Hypergeometric Solutions of Second Order Linear Differential Equations with Five Singularities

Abstract

Let $L \in \mathbb{C}(x)[\partial]$ be a second order, irreducible, linear differential operator with rational function coefficients and $\partial = \frac{d}{dx}$. Suppose L has five regular singularities and at least one of them is logarithmic. The goal in this paper is to solve such L in terms of ${}_2F_1$-hypergeometric functions, i.e, to find ${}_2F_1$-type solution:

$$y = \exp\Big(\int r\,dx\Big) \cdot \Big(r_0 S(f) + r_1 S(f)'\Big) \neq 0 \tag{1}$$

such that $L(y) = 0$, where $S(x) = {}_2F_1(a, b; c \,|\, x)$ and $f, r, r_0, r_1 \in \mathbb{C}(x)$.

1 Introduction

Differential equations have a huge impact in human society as they occur significantly in every branch of science. Linear homogeneous differential equations with rational function coefficients (i.e, with singularities) are very common in mathematics, combinatorics, physics and engineering. Finding closed form solutions (solutions expressible in terms of well studied special functions, for example; Bessel, Kummer, Liouvillian, Hypergeometric etc.) of such differential equations is a fascinating area of research in computer algebra [5, 22, 11, 23, 1, 8, 9].

Although there is no complete algorithm which can find closed form solution of every second order differential equation, there are algorithms to treat some classes of differential equations. For example, Kovacic's algorithm [24] finds Liouvillian solutions and the algorithm in [10] finds solutions of the differential equations with so-called irregular singularities in terms of Bessel, Kummer functions. The hypergeometric case, which corresponds to Fuchsian differential equations (equations with only regular singularities), is interesting as it incorporates a broader area (dessin d'enfants, Belyi and near Belyi maps, constellations, ...) of mathematics. This motivates us to work on hypergeometric solutions of differential equations.

Gauss Hypergeometric Equation (GHE) has 3 regular singularities at $\{0, 1, \infty\}$ and has ${}_2F_1(a, b; c \,|\, x)$ as a solution where the parameters a, b, c are determined by exponent differences (e_0, e_1, e_∞) of GHE at $0, 1, \infty$ by $(e_0, e_1, e_\infty) = (1-c, c-a-b, b-a)$ (up to $\pm$). The corresponding differential operator, also called Gauss Hypergeometric Differential Operator (GHDO) is denoted as $H^{a,b}_{c,x}$ (*see Section 2 for details*). If we choose $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3})$ then we have $(a, b, c) = (\frac{1}{12}, \frac{5}{12}, 1)$. Then the differential operator $H^{\frac{1}{12}, \frac{5}{12}}_{1,x}$ has a basis of two solutions at $x = 0$, consisting of an analytic solution: ${}_2F_1(\frac{1}{12}, \frac{5}{12}; 1 \,|\, x)$ and a logarithmic solution.

Finding ${}_2F_1$-type solution of a second order differential operator L corresponds to finding transformations that send $H^{a,b}_{c,x}$ to L:

(i) Change of variables: $y(x) \mapsto y(f)$
(ii) Gauge transformation: $y \mapsto r_0 y + r_1 y'$
(iii) Exponential product: $y \mapsto \exp(\int r\,dx) \cdot y$

The function f in (i) above is called the *pullback* function. These transformations are denoted as $\xrightarrow{f}_C$, $\xrightarrow{r_0, r_1}_G$ and $\xrightarrow{r}_E$ respectively. Let $S(x)$ be a special function that satisfies a second order differential operator. All of these transformations send expressions in terms of S to expressions in terms of S. So any solver for finding solutions in terms of S, if it is complete, then it must be able to deal with transformations $\xrightarrow{f}_C$, $\xrightarrow{r_0, r_1}_G$ and $\xrightarrow{r}_E$. So it must be able to find any solution of the form (1).

Our task in this project is, given an input differential operator L_{inp}, find

$$H^{a,b}_{c,x} \xrightarrow{f}_C H^{a,b}_{c,f} \xrightarrow{r_0,r_1}_G \xrightarrow{r}_E L_{inp}.$$

Once we find such transformations, we compute a ${}_2F_1$-type solution of L_{inp} as:

$$S(x) \xrightarrow{f}_C S(f) \xrightarrow{r_0,r_1}_G \xrightarrow{r}_E \exp(\int r dx)\big(r_0 S(f) + r_1 S(f)'\big)$$

where $S(x) ={}_2 F_1(a, b; c \,|\, x)$.
There are algorithms [3] to find the transformations $\xrightarrow{r_0,r_1}_G$ and $\xrightarrow{r}_E$ but to apply them we first need $H^{a,b}_{c,f}$ (or equivalently, f and $H^{a,b}_{c,x}$). Thus the crucial part is to compute f. We compute f from the singularities of $H^{a,b}_{c,f}$. Since we do not yet know $H^{a,b}_{c,f}$, the only singularities of $H^{a,b}_{c,f}$ that we know are the singularities of L_{inp} that can not *disappear* (turn to regular points) under the transformations $\xrightarrow{r_0,r_1}_G$ and $\xrightarrow{r}_E$.

Definition 1.1. *A singularity of a differential operator is called a non-removable singularity if it stays singular under any combination of transformations* $\xrightarrow{r_0,r_1}_G$ *and* $\xrightarrow{r}_E$.

A singularity p of L_{inp} that can become a regular point under $\xrightarrow{r_0,r_1}_G$ and/or $\xrightarrow{r}_E$ need not be a singularity of $H^{a,b}_{c,f}$. Such (*removable*) singularities provide no information about f. They include *apparent* singularities (singularities p where all solutions are analytic at $x = p$, such singularities can disappear under $\xrightarrow{r_0,r_1}_G$). More generally, if there exist functions u, y_1, y_2 with y_1, y_2 analytic at $x = p$ such that uy_1, uy_2 is a basis of local solutions of a second order differential operator L at $x = p$, then $x = p$ is removable (such p can be sent to an apparant singularity with $\xrightarrow{r}_E$).

1.1 Motivation

Any second order linear differential equation with 3 regular singularities is ${}_2F_1$-solvable (essentially by definition). In this case we need to find a Möbius transformation from these 3 singularities to $\{0, 1, \infty\}$ and applying that to ${}_2F_1(a, b; c \,|\, x)$ gives a solution. This case is done in [11]. For differential equations with $n > 3$ singularities, the main task is to develop a complete table consisting of all rational maps f which produce n singularities from $0, 1, \infty$. The case where a differential equation has 4 regular singularities (*Heun equation*) is done in [5]. That motivates our focus on the differential equations with 5 regular singularities.
Differential equations with logarithmic singularities are very common. Section 3 in [15] mentions 92 integer sequences coming from counting paths in a 2D lattice, of which 36 appear to be holonomic (their generating function satisfies a linear homogeneous differential equation with polynomial coefficients). Of these 36 differential equations, there are 19 with algebraic solutions. All remaining 17 equations are ${}_2F_1$-solvable and have logarithmic singularities.
We examined many integer sequences $u(0), u(1), u(2), \ldots$ in [25] whose generating functions $y = \sum_n u(n)x^n \in \mathbb{Z}[x]$ are (a) convergent, and (b) holonomic, i.e; y satisfies a linear differential equation with rational function coefficients. Such differential equations are also known as globally nilpotent differential equations [19] or CIS (convergent integer power series)-equations [18]. All such *second order* differential equations tested so far turned out to have hypergeometric (${}_2F_1$ in this case) solutions or algebraic solutions. We are interested in hypergeometric solutions, the algebraic solutions can be found using [24]. In fact we observed the same for differential equations of order three from [25], in this case the differential equation reduces to a second order differential equation with ${}_2F_1$-type solution. More surprisingly, all differential equations discussed above lie in the same class, namely $\mathit{Class}\Big(H^{\frac{1}{12},\frac{5}{12}}_{1,x}\Big)$:

Definition 1.2. *The class of a differential operator L, denoted* $\mathrm{Class}(L)$*, is a minimal set of operators with the following properties:*

1. $L \in \mathrm{Class}(L)$,
2. *If L_1 can be solved in terms of L_2 (this means solutions of L_1 are expressible in terms of solutions of L_2 using the transformations* $\xrightarrow{f}_C$, $\xrightarrow{r_0,r_1}_G$, $\xrightarrow{r}_E$*) and* $\mathrm{Class}(L) \cap \{L_1, L_2\} \neq \emptyset$ *then* $\{L_1, L_2\} \subseteq \mathrm{Class}(L)$.

Definition 1.3. *If transformations in property 2 above involve algebraic functions, the class is denoted as* $\mathit{Class}^{alg}(L)$.

Remark 1.4. *$Class(L) \subseteq Class^{alg}(L)$.*

It turns out that f in logarithmic case has degree bound 18 and at most 2 branch points outside $\{0, 1, \infty\}$ (*see Section 3 for more details*). For arbitrary a, b, c, the degree bound for such f would be 60 for 4 singularities, and 96 for 5 singularities. If $L_1 \in Class^{alg}(L_2)$, then the monodromy groups of L_1 and L_2 are commensurable. Kisao Takeuchi classified [10, Section 2, Table (1)] commensurable classes of arithmetic triangle groups. The first class (Section 4, Diagram (I)) in Takeuchi's table corresponds to the reciprocals of exponent differences of the GHDO's in $Class\left(H_{1,x}^{\frac{1}{12},\frac{5}{12}}\right)$. We show the diagram here:

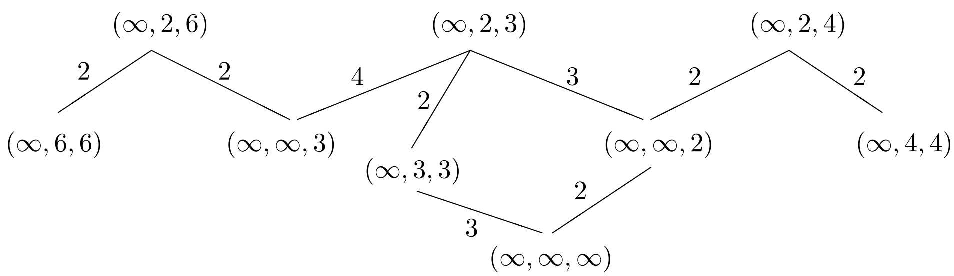


Fig. 1: [10, Section 4, Diagram (I)], which gives the reciprocals of exponent differences of GHDO's in $Class\left(H_{1,x}^{\frac{1}{12},\frac{5}{12}}\right)$

Each triangle group in Figure 1 corresponds to the denominators of exponent differences of GHDO whereas ∞ corresponds to exponent difference 0 (logarithmic singularity, *see Section 2*). This diagram includes all logarithmic cases in Takeuchi's classification. From the classification [10, Section 2, Table (1)], we observe the following:
If a differential operator L has (i) logarithmic singularities and (ii) arithmetic monodromy group, then $L \in Class^{alg}\left(H_{1,x}^{\frac{1}{12},\frac{5}{12}}\right)$.
$(\infty, 2, 3)$ in Figure 1 corresponds to the GHDO with exponent differences $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3})$ (up to $\pm$ and mod $\mathbb{Z}$). This choice of the exponent differences gives $(a, b, c) = (\frac{1}{12}, \frac{5}{12}, 1)$. The correspondence can also be given as:

$$3 \longleftrightarrow \pm\frac{1}{3} + \mathbb{Z}, \;\; 2 \longleftrightarrow \pm\frac{1}{2} + \mathbb{Z} \text{ and } \infty \longleftrightarrow 0 + \mathbb{Z}.$$

The numbers along the lines in Figure 1 represent the degree of the pullback function f in $\xrightarrow{f}_C$ which produces one triple of exponent differences from another. For example, a degree 2 pullback produces the exponent differences $(0, 0, \frac{1}{3})$ from $(0, \frac{1}{2}, \frac{1}{6})$.
Taking $(e_0, e_1, e_\infty) = (0, 0, \frac{1}{3})$ gives $(a, b, c) = (\frac{1}{3}, \frac{2}{3}, 1)$, and taking $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{6})$ gives $(a, b, c) = (\frac{1}{6}, \frac{1}{3}, 1)$. That means $H_{1,x}^{\frac{1}{3},\frac{2}{3}}$ can be solved in terms of solutions of $H_{1,x}^{\frac{1}{6},\frac{1}{3}}$ using the pullback f of degree 2 (Moreover if a differential operator L can be solved in terms of solutions of $H_{1,x}^{\frac{1}{3},\frac{2}{3}}$, then L can also be solved in terms of solutions of $H_{1,x}^{\frac{1}{6},\frac{1}{3}}$). Such f has the branching pattern $[1, 1], [2], [2]$ above $0, 1, \infty$ respectively, i.e, f ramifies of order 2 above 1 and ∞. A quick computation gives $f = -4x(x - 1)$.
Our ultimate goal is to solve all logarithmic cases, yet we want to deal with the differential equations associated with Figure 1 first because that covers nearly all cases with logarithmic singularities. The other cases, for example, differential equations solvable in terms of the GHDO with $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{5})$, which have lower degree bound for f and hence smaller table than $H_{1,x}^{\frac{1}{12},\frac{5}{12}}$, can be done in the similar way. Any choice of exponent differences corresponding to Figure 1 is solvable in terms of solutions of $H_{1,x}^{\frac{1}{12},\frac{5}{12}}$, $H_{1,x}^{\frac{1}{8},\frac{3}{8}}$ or $H_{1,x}^{\frac{1}{6},\frac{1}{3}}$ which correspond to $(\infty, 2, 3), (\infty, 2, 4)$ and $(\infty, 2, 6)$ respectively. Hence, treating these 3 cases covers everything for this project.

2 Preliminaries and Notations

This section gives a brief summary of the prior results and notations that are necessary for this paper.

2.1 Differential Operators

Definition 2.1. *Let K be a field with characteristic zero. A derivation ∂ in K is a map $\partial : K \longrightarrow K$ with the following properties:*

$$\partial(a+b) = \partial(a) + \partial(b),$$
$$\partial(ab) = \partial(a)b + a\partial(b)$$

where $a, b \in K$.

Remark 2.2.

1. *A field K equipped with such derivation is called a differential field.*
2. *$C_K := \{k \in K \mid \partial(k) = 0\}$ is also a field, called the constant field of K.*

The associated ring, also called a ring of differential operators, is denoted by $K[\partial]$.

Definition 2.3. *Given a differential field K with derivation ∂, a differential operator L is an element of $K[\partial]$ given as: $L = \sum_{i=0}^{n} a_i \partial^i$ where $a_i \in K$.*

Remark 2.4. *If $a_n \neq 0$ then we say that L has order n and write* $\deg(L) = n$.

Note: $K[\partial]$ is non commutative in general. For example, $\partial x = x\partial + 1$ when $K = \mathbb{C}(x)$ and $\partial = \frac{d}{dx}$.
The solutions y of differential equation $L(y) = 0$ lie in a *universal extension* Ω of K, where Ω is a minimal differential ring in which every operator $L \in K[\partial]$ has precisely $\deg(L)$ linearly independent solutions, more details can be found in [22].

Definition 2.5. *The set of all solutions of a differential operator L is called its solution space. It is denoted by $V(L)$ and defined as:*

$$V(L) := \{y \in \Omega \mid L(y) = 0\}$$

2.2 Singularities

Consider a differential operator $L = \sum_{i=0}^{n} a_i \partial^i$ where $a_i \in K$. After clearing denominators, we may assume that the a_i's are polynomials.

Definition 2.6.

1. *A point $p \in \overline{C_K}$ is called a* regular *(or* non-singular*) point when $a_n(p) \neq 0$. Otherwise it is called a* singular *point (or a* singularity*).*
2. *The point $p = \infty$ is called regular if the change of variable $x \mapsto 1/x$ produces an operator $L_{1/x}$ with a regular point at $x = 0$.*

Remark 2.7. *Let y be a solution of a differential operator L. Singularities of y are also singularities of L but the converse is not true, see apparent singularities in Section 2.4.*

Definition 2.8. *Given $p \in \overline{C_K} \cup \{\infty\}$, we define the local parameter t_p as*

$$t_p = \begin{cases} x - p & \text{if } p \neq \infty \\ \frac{1}{x} & \text{if } p = \infty. \end{cases}$$

Below we discuss the types of singularities.

Definition 2.9. *Let $L = \sum_{i=0}^{n} a_i \partial^i$ where a_i are polynomials. A singularity p of L is:*
(1) regular singularity ($p \neq \infty$) if $t_p^i \cdot \frac{a_{n-i}}{a_n}$ is analytic at $x = p$ for $1 \leq i \leq n$.
(2) regular singularity ($p = \infty$) if $L_{1/x}$ has a regular singularity at $x = 0$.
(3) irregular singularity otherwise.

Definition 2.10. *A differential operator is called Fuchsian (or regular singular) if all of its singularities are regular singularities.*

This paper considers only Fuchsian operators of order 2. The non-Fuchsian case (L having at least one irregular singularity) was treated in [22]. The following classical theorem gives the structure of local solutions of a second order differential operator at a regular singularity or a non-singular point:

Theorem 2.11. *Let $L \in K[\partial]$ be an operator of order 2 and $p \in \overline{C_K}$. If $x = p$ is a regular singularity or a non-singular point of L, then there exists the following basis of $V(L)$ in the neighborhood of $x = p$;*

$$y_1 = t_p^{e_1} \sum_{i=0}^{\infty} a_i t_p^i, \ a_0 \neq 0 \text{ and}$$

$$y_2 = t_p^{e_2} \sum_{i=0}^{\infty} b_i t_p^i + c y_1 \log(t_p), \ b_0 \neq 0 \text{ where } e_1, e_2, a_i, b_i, c \in \overline{C_K}$$

such that:
(i) If $e_1 = e_2$ then $c \neq 0$.
(ii) Conversely, if $c \neq 0$ then $e_1 - e_2 \in \mathbb{Z}$.

More details can be found in [20, 22].

Remark 2.12. *In Theorem 2.11:*

1. *If $c \neq 0$ then $x = p$ is called a* logarithmic *singularity.*
2. *The constants e_1, e_2 are called* local exponents *or* exponents *of L at $x = p$.*

For a second order differential operator $L = \partial^2 + a_0\partial + a_1 \in K[\partial]$, these exponents e_1, e_2 of a regular singular point p can be obtained as the roots of the *indicial equation*:
1. $X(X-1) + q_0 X + q_1 = 0$, where $q_i = \lim_{x \mapsto p}(x-p)^{i+1} a_i, \quad i \in \{0, 1\} \quad (\text{ if } p \in \overline{C_K} \text{ })$.
2. If $p = \infty$ then take the indicial equation of $L_{1/x}$ at $x = 0$.

Remark 2.13.

1. *Logarithmic singularities are non-removable. They stay logarithmic under the transformations $\xrightarrow{f}_C$, $\xrightarrow{r_0, r_1}_G$ and $\xrightarrow{r}_E$.*
2. *If $e_1 - e_2 \in \mathbb{Z}$ and $x = p$ is non logarithmic then the point $x = p$ is either a regular point or a removable singularity.*
3. *$x = p$ is non-singular $\iff \{e_1, e_2\} = \{0, 1\}$ and $c = 0$.*
4. *$x = p$ is a non-removable singularity $\iff c \neq 0$ or $e_1 - e_2 \notin \mathbb{Z}$.*

Proofs and more details can be found in [21].

Definition 2.14. *Let e_1, e_2 be the exponents of L at $x = p$. The exponent difference of L at $x = p$ is denoted $\Delta_p(L)$ (or Δ_p) and is defined as $\Delta_p(L) = \pm(e_1 - e_2)$.*

Let $\Delta_{p_1}, \Delta_{p_2}$ be the exponent differences of L at p_1, p_2 respectively. We say that Δ_{p_1} and Δ_{p_2} *match* if $\Delta_{p_1} \equiv \Delta_{p_2} \mod \mathbb{Z}$.

Definition 2.15. *The* singularity structure *of L is:*

$$\mathit{Sing}(L) = \{(p, \Delta_p(L) \mod \mathbb{Z}) : p \text{ is a non-removable singularity}\}.$$

It is often more convenient to express singularities in terms of monic irreducible polynomials.

Definition 2.16. *Let F be a field of constants with characteristic 0.*

$$\mathit{places}(F) := \{f \in F[x] \mid f \text{ is monic and irreducible}\} \bigcup \{\infty\}.$$

The degree *of a place p is 1 if $p = \infty$ and $\deg(p)$ otherwise.*

Example 2.17. *Consider the following differential operator:*

$$L = 2\left(2x^2-1\right)\left(8x^2-1\right)\partial^2 + 4x\left(24x^2-7\right)\partial + 24x^2 - 3.$$

We obtain the singularity structure of L as:

$$Sing(L) = \Big\{\Big(\frac{1}{\sqrt{2}}, \frac{1}{6}\Big), \Big(-\frac{1}{\sqrt{2}}, \frac{1}{6}\Big), \Big(\frac{1}{2\sqrt{2}}, \frac{1}{3}\Big), \Big(-\frac{1}{2\sqrt{2}}, \frac{1}{3}\Big)\Big\}.$$

In terms of places($\mathbb{Q}$) it is written as:

$$Sing(L) = \Big\{\Big(x^2 - \frac{1}{2}, \frac{1}{6}\Big), \Big(x^2 - \frac{1}{8}, \frac{1}{3}\Big)\Big\}.$$

For the rest of the paper, we will consider $K = \mathbb{C}(x)$.

2.3 Gauss Hypergeometric Differential Equation

The Gauss hypergeometric differential equation (GHE) has the following form:

$$x(1-x)y'' + (c-(a+b+1)x)y' - aby = 0. \tag{2}$$

It has three regular singularities at $0, 1$, and ∞. It has exponents $\{0, 1-c\}$ at $x=0$, $\{0, c-a-b\}$ at $x=1$ and $\{a, b\}$ at $x=\infty$. The corresponding differential operator is denoted by:

$$H^{a,b}_{c,x} = x(1-x)\partial^2 + (c-(a+b+1)x)\partial - ab. \tag{3}$$

One of the solutions of the GHE at $x=0$ is ${}_2F_1(a,b;c\,|\,x)$. Computing a ${}_2F_1$-type solution of a second order L_{inp} (inp = input) corresponds to computing transformations from $H^{a,b}_{c,x}$ to L_{inp}.

Remark 2.18. *The exponent differences of $H^{a,b}_{c,x}$ can be obtained from the parameters a, b, c and vice versa: $(e_0, e_1, e_\infty) = (1-c, c-a-b, b-a)$.*

Remark 2.19. *We assume that $H^{a,b}_{c,x}$ has no Liouvillian solutions. For such $H^{a,b}_{c,x}$, the points $0, 1, \infty$ are never non-singular or removable singularities. So if $H^{a,b}_{c,x}$ has $e_p \in \mathbb{Z}$ (with $p \in \{0, 1, \infty\}$) then p is a logarithmic singularity.*

2.4 Properties of Transformations

For second order operators, we use the notation $L_1 \longrightarrow L_2$ if L_1 can be transformed to L_2 with any combination of the three transformations from *Section 1*. If $L_1 \longrightarrow L_2$ then $L_1 \xrightarrow{f}_C \xrightarrow{r_0,r_1}_G \xrightarrow{r}_E L_2$. More details can be found in [1].

Remark 2.20.

1. $\xrightarrow{r_0,r_1}_G$ *and* $\xrightarrow{r}_E$ *are equivalence relations.*

2. Δ_p *remains same under* $\xrightarrow{r}_E$ *but may change by an integer under* $\xrightarrow{r_0,r_1}_G$.
   So if $H^{a,b}_{c,x} \xrightarrow{f}_C H^{a,b}_{c,f} \xrightarrow{r_0,r_1}_G \xrightarrow{r}_E L_{inp}$ *for some input* L_{inp} *with* a, b, c, f *unknown, then* $\Delta_p(H^{a,b}_{c,f})$ *can be ($\bmod\ \mathbb{Z}$ and up to $\pm$) read from* $\Delta_p(L_{inp})$,

   $$Sing(L_{inp}) = Sing(H^{a,b}_{c,f}).$$

   Hence a, b, c, f should be reconstructed from Sing(L_{inp}).

3. *If one of e_0, e_1, e_∞ is in $\frac{1}{2} + \mathbb{Z}$ then $H^{a,b}_{c,x}$ is determined, up to the equivalence relation $\xrightarrow{r_0,r_1}_G \xrightarrow{r}_E$, by the triple (e_0, e_1, e_∞) up to $\pm$ and $\bmod\ \mathbb{Z}$.*
   If $\{e_0, e_1, e_\infty\} \bigcap (\frac{1}{2} + \mathbb{Z}) = \emptyset$ then the triple leaves two separate cases for $H^{a,b}_{c,x}$ up to $\xrightarrow{r_0,r_1}_G \xrightarrow{r}_E$; we need to consider (e_0, e_1, e_∞) up to $\pm$ and $\bmod\ \mathbb{Z}$, and (e_0+1, e_1, e_∞) up to $\pm$. See Theorem 8, Section 5.3 in [23] for details.

Because of the transformation $H^{a,b}_{c,f} \xrightarrow{r_0,r_1}_G \xrightarrow{r}_E L_{inp}$ in *Remark 2.20* only non-removable singularities of L_{inp} provide usable data for f and $H^{a,b}_{c,f}$.

Definition 2.21. *Two operators L_1, L_2 are called projectively equivalent (notation: $L_1 \sim_p L_2$) if $L_1 \xrightarrow{r_0,r_1}_G \xrightarrow{r}_E L_2$.*

Definition 2.22. *Let $f : \mathbb{P}^1 \to \mathbb{P}^1$ be a rational function of degree n, where the degree of a rational function is defined as the maximum of the degrees of its numerator and denominator. A point $b \in \mathbb{P}^1$ is called a branch point if $\#(f^{-1}(b)) < n$, i.e; f has multiple roots above b. The multiple root (if any) $a \in \mathbb{P}^1$ is called a ramification point. Set of all branch points is called the branched set. The branching pattern of a rational function f above a point q is given as a list of multiplicities of all points $p \in f^{-1}(q)$.*

Example 2.23. *Consider the following function:*

$$f = -\frac{1}{4}\,\frac{(3\,x-1)^2}{(x-3)\,(x-1)^3\,x^2} \quad \textit{where} \quad 1-f = \frac{1}{4}\,\frac{\left(-1+3\,x-6\,x^2+2\,x^3\right)^2}{(x-3)\,(x-1)^3\,x^2}$$

Then branching pattern of f above $0, 1, \infty$ is $[2,4],[2,2,2],[1,2,3]$.

Singularity structure of a differential operator is preserved under the transformations $\xrightarrow{r_0,r_1}_G$ and $\xrightarrow{r}_E$. However, the change of variables $\xrightarrow{f}_C$ can change everything. The following lemma gives the effect of $\xrightarrow{f}_C$ on the singularities and their exponent differences (see [12] for more details):

Lemma 2.24. *Let e_0, e_1, e_∞ be the exponent differences of $H^{a,b}_{c,x}$ at $0, 1, \infty$. Let $H^{a,b}_{c,f}$ be the operator obtained from $H^{a,b}_{c,x}$ by applying $x \mapsto f$. Let $d = \Delta_p$ be the exponent difference of $H^{a,b}_{c,f}$ at $x = p$. Then:*

1. *If p is a root of f with multiplicity m, then $d = me_0$.*
2. *If p is a root of $1-f$ with multiplicity m, then $d = me_1$.*
3. *If p is a pole of f of order m, then $d = me_\infty$.*

Example 2.25. *Let L be the Gauss hypergeometric differential operator with $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{4})$, i.e; $L := H^{\frac{1}{8},\frac{3}{8}}_{1,x}$:*

$$L := 64x(x-1)\partial^2 + 32(3x-2)\partial + 3$$

Singularity structure of L is the following:

```
> Sing(L);
```

$$\left\{[x,0],[\infty,-\frac{1}{4}],[x-1,\frac{1}{2}]\right\}$$

Exponent difference is defined up to $\pm$. Let M be the differential operator obtained after applying the change of variables with $f = \frac{(1-x)(4x+1)}{(x+1)^3}$, i.e; $M := H^{\frac{1}{8},\frac{3}{8}}_{1,f}$;

$$M := 16(x+1)^2(x-1)(4x+1)(x+7)(2x-7)\partial^2 + 16(x+1)(x+4)(8x^3-48x^2-75x+35)\partial + 3(2x-7)^3$$

We find the following singularity structure of M;

```
> Sing(M);
```

$$\{[\infty,0],[x+7,\frac{1}{2}],[x-1,0],[x+1,-\frac{3}{4}],[x+\frac{1}{4},0]\}$$

Following diagram illustrates the result:

$H_{1,f}^{\frac{1}{8},\frac{3}{8}}$:

p	∞	1	$-\frac{1}{4}$	-7	-1
Δ_p	0	0	0	$\frac{1}{2}$	$\frac{3}{4}$

$$f = \frac{(1-x)(4x+1)}{(x+1)^3} \quad \Big\uparrow \quad 1-f = \frac{x^2(x+7)}{(x+1)^3}$$

$H_{1,x}^{\frac{1}{8},\frac{3}{8}}$:

p	0	1	∞
Δ_p	0	$\frac{1}{2}$	$\frac{1}{4}$

p: singularity, Δ_p: exponent difference

Fig. 2: Effect of $\xrightarrow{f}_C$ on the singularity structure

The branching pattern of f above $0, 1, \infty$ is $[1, 1, 1], [1, 2], [3]$. Exponent differences of the base operator $H_{1,x}^{\frac{1}{8},\frac{3}{8}}$ get multiplied by the corresponding multiplicities of f to produce the exponent differences of the resulting operator $H_{1,f}^{\frac{1}{8},\frac{3}{8}}$. The point 0 above 1 becomes a regular point (exponent difference is $2 \cdot \frac{1}{2} = 1$) and thus does not show up in $Sing\left(H_{1,f}^{\frac{1}{8},\frac{3}{8}}\right)$.

Remark 2.26. *Let $H_{c,x}^{a,b}$ be the Gauss hypergeometric differential operator. Suppose $[a_1, \ldots, a_i]$, $[b_1, \ldots, b_j], [c_1, \ldots, c_k]$ be the branching pattern of f above $0, 1, \infty$ respectively. Using* Lemma 2.24 *and* Remark 2.13*, the singularities of $H_{c,f}^{a,b}$ are as follows:*

$P_0 = \{x : f(x) = 0 \textit{ and } (e_0 \in \mathbb{Z} \textit{ or } a_l\, e_0 \notin \mathbb{Z}) \textit{ for } 1 \le l \le i\}$
$P_1 = \{x : 1 - f(x) = 0 \textit{ and } (e_1 \in \mathbb{Z} \textit{ or } b_l\, e_1 \notin \mathbb{Z}) \textit{ for } 1 \le l \le j\}$
$P_\infty = \{x : \frac{1}{f(x)} = 0 \textit{ and } (e_\infty \in \mathbb{Z} \textit{ or } c_l\, e_\infty \notin \mathbb{Z}) \textit{ for } 1 \le l \le k\}$

where (e_0, e_1, e_∞) are the exponent differences of $H_{c,x}^{a,b}$ at $(0, 1, \infty)$ respectively. The union of P_0, P_1 and P_∞ are the non-removable singularities of $H_{c,f}^{a,b}$, or L_{inp} by Remark 2.20.

2.5 An Example with Hypergeometric Solution

Consider the following differential operator:

$$L = (x-16)(x^2+18x-15)\partial^2 + (x+7)(x-39)\partial - \frac{1}{36}\frac{(25x^3 - 1006x^2 - 5523x - 894)}{(x^2-3)}$$

L has the following singularity structure:

```
> Sing(L);
```

$$\left\{[\infty, -\frac{5}{3}], [x^2-3, 1], [x^2+18\,x-15, 0]\right\}$$

L has five regular singularities; the roots of $x^2 + 18\,x - 15$ and $x^2 - 3$ are logarithmic singularities. Our algorithm solves L, see `www.math.fsu.edu/~vkunwar/FiveSings/` for more details. One of the solutions is:

$$Sol(L) = h_1(x)S(f) + h_2(x)S(f)'$$

where $h_1(x) = \frac{1}{3}\,\frac{\left(x^3-36\,x^2+69\,x-54\right)\left(x^2-3\right)}{(4\,x^3-29\,x^2+42\,x-21)^{5/4}}$, $h_2(x) = \frac{\left(x^2-3\right)\left(x^2+18\,x-15\right)}{(4\,x^3-29\,x^2+42\,x-21)^{1/4}(x+7)}$ and

$S(f) = {}_2F_1\left(\frac{1}{12}, \frac{5}{12}; 1 \mid \frac{4}{9}\,\frac{\left(x^2+18\,x-15\right)^2\left(x^2-3\right)^3}{\left(4\,x^3-29\,x^2+42\,x-21\right)^3}\right)$.
The crucial task is to find the parameters of hypergeometric function; i.e, to find the constants a, b, c and the rational function f. For this project, the parameters a, b, c are computed from the finite choices of exponent differences corresponding to Figure 1. Hence the major task is to compute f. The remaining part of the paper will be focused on the theoretical and computational aspects of our method on finding f.

3 Types and Bounds for f

For a rational function $f : \mathbb{P}^1 \to \mathbb{P}^1$ of degree n, total amount of ramification is given by:

$$\sum_{p\in\mathbb{P}^1}(e_p - 1) = 2n - 2 \qquad \text{(Riemann-Hurwitz)} \tag{4}$$

where e_p is the ramification order of f at p. Let the amount of ramification of f be $R_{01\infty}$ (above $\{0, 1, \infty\}$) and R_{out} (above $\mathbb{P}^1 \setminus \{0, 1, \infty\}$). As in [2], using (4), we find the largest bounds for the degree of f and ramification outside $\{0, 1, \infty\}$ for our project as:

$$\deg(f) \leq 18 \quad \text{and} \quad R_{out} \leq 2$$

when we choose $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3})$. We have to compute all rational functions (up to Möbius transformation) that can occur as f in the solution (1) of L_{inp} in this project. The bound on ramification outside $\{0, 1, \infty\}$ further classifies such f's as:

1. *Belyi maps*: $R_{out} = 0$
2. *Belyi-1 maps*: $R_{out} = 1$
3. *Belyi-2 maps*: $R_{out} = 2$

Belyi maps are zero-dimensional families. But Belyi-1 (resp. Belyi-2) maps are one (resp two)-dimensional families as they ramify above 1 (resp. 2) arbitrary points outside $\{0, 1, \infty\}$. We use the term *near Belyi* maps for such maps. We summarize the bounds in the following table:

(e_0, e_1, e_∞)	GHDO	R_{out}	Type	max. degree
$(0, \frac{1}{2}, \frac{1}{3})$	$H_{1,x}^{\frac{1}{12},\frac{5}{12}}$	0	Belyi	18
		1	Belyi-1	12
		2	Belyi-2	6
$(0, \frac{1}{2}, \frac{1}{4})$	$H_{1,x}^{\frac{1}{8},\frac{3}{8}}$	0	Belyi	12
		1	Belyi-1	8
		2	Belyi-2	4
$(0, \frac{1}{2}, \frac{1}{6})$	$H_{1,x}^{\frac{1}{6},\frac{1}{3}}$	0	Belyi	9
		1	Belyi-1	6
		2	Belyi-2	3

Tab. 1: Bounds and types

The data in Figure 1 and Table 1 indicate that the case $H_{1,x}^{\frac{1}{12},\frac{5}{12}}$ alone requires more work than the other two cases $H_{1,x}^{\frac{1}{8},\frac{3}{8}}$ and $H_{1,x}^{\frac{1}{6},\frac{1}{3}}$ combined together. Additionally, $H_{1,x}^{\frac{1}{12},\frac{5}{12}}$ shares some part from both $H_{1,x}^{\frac{1}{8},\frac{3}{8}}$ and $H_{1,x}^{\frac{1}{6},\frac{1}{3}}$ in terms of solvability (*see Figure 1*).

Our solver will be complete if the tables for $H_{1,x}^{\frac{1}{12},\frac{5}{12}}$, $H_{1,x}^{\frac{1}{8},\frac{3}{8}}$ and $H_{1,x}^{\frac{1}{6},\frac{1}{3}}$ are complete. The major task in this project is to prove that our tables are complete, i.e; How do we know that our tables contain all such maps up to $\mathrm{Aut}(\mathbb{P}^1)$? The next section addresses the completeness for Belyi maps. Near Belyi maps will be discussed later.

4 Belyi Maps

Definition 4.1. *A rational map $f : \mathbb{P}^1 \to \mathbb{P}^1$ is called a Belyi map if its branched set lies inside $\{0, 1, \infty\}$. That means f is unramified outside $\{0, 1, \infty\}$.*

Definition 4.2. *Let f be a Belyi map. The $(0, \frac{1}{2}, \frac{1}{3})$-singularity-count of f is the sum of*

1. *the number of roots of f (not counting with multiplicity)*
2. *the number of roots of $1 - f$ that do not have multiplicity 2*
3. *the number of poles of f that do not have multiplicity 3.*

The motivation for Definition 4.2 is that this counts the number of singular points (including removable singularities) after a change of variables $x \mapsto f$ applied to the hypergeometric equation with exponent differences $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3})$. In general, we can define the same for any (e_0, e_1, e_∞). Usually we want to count only non-removable singularities, then replace 'do not have multiplicity k' by 'whose multiplicity is not divisible by k' in the above definition. We usually count only non removable singularities, there are some Belyi maps which produce removable singularities.

Remark 4.3. *Consider $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3})$ and take the branching pattern $[1, 2, 3, 4], [2, 2, 2, 4]$, $[1, 3, 3, 3]$ above $0, 1, \infty$ respectively. Such branching pattern produces a Belyi map f with singularity-count 6. But the only singularity above 1 is a removable singularity (its exponent difference is $4 \cdot \frac{1}{2} = 2$). So f produces 5 non-removable singularities and 1 removable singularity, we denote this as $5 + 1$ singularities.* ***Algorithm 5.8*** *in Section 4.4 will skip this branching pattern (and its dessin) when singularity-count $d = 5$, and will find it when $d = 6$. We omit such $5 + 1$ singularities (and their Belyi maps) from our Belyi table for $d = 5$ because the corresponding differential operator will be solved by our Belyi-1 solver. Some Belyi-1 maps $g(x, s)$ (see section 5 for more details) from our table for some $s \in \mathbb{P}^1$ will cover such f (additional ramified point in Belyi-1 maps produces a removable singularity). So we don't compute such Belyi maps. Likewise, Belyi maps with 4+1 singularities are found in $d = 5$, but we also skip them.*[1] *They are covered by Belyi-1 maps in $d = 4$.*

The crucial part on finding ${}_2F_1$-type solution of a differential operator L_{inp} is to compute f and a, b, c such that:

$$H^{a,b}_{c,x} \xrightarrow{f}_C H^{a,b}_{c,f} \xrightarrow{r_0, r_1}_G \xrightarrow{r}_E L_{inp}$$

In particular, we want to have $Sing(H^{a,b}_{c,f}) = Sing(L_{inp})$. The Gauss hypergeometric differential operator $H^{a,b}_{c,x}$ has singularities at $0, 1$ and ∞. So the singularity structure $Sing(H^{a,b}_{c,f})$ depends solely on the branching pattern of f above $0, 1, \infty$ and the choice of a, b, c. Belyi maps are very special as their branching occurs only above $0, 1$ and ∞.
The main task is to compute all Belyi maps and near Belyi maps (up to Möbius transformation) whose singularity-count is 5. The goal in this section is to find all Belyi maps f (up to Möbius transformation) with $(0, \frac{1}{2}, \frac{1}{3})$-singularity-count 5 (Note: the cases < 5 are done previously, see [23, 5] for details). We have also done the cases $(0, \frac{1}{2}, \frac{1}{4})$ and $(0, \frac{1}{2}, \frac{1}{6})$ but we explain only $(0, \frac{1}{2}, \frac{1}{3})$ here for convenience of writing. We prove completeness by computing dessins.

Definition 4.4. *[14] A sequence $[g_1, g_2, \ldots, g_k]$ of permutations in S_n is called a k-constellation if the following properties hold:*

1. *the group $\langle g_1, g_2, \ldots, g_k \rangle$ acts transitively on the set of n points;*
2. *$g_1 g_2 \cdots g_k = 1$.*

Here k is called length and n is called degree of the constellation. The group $\langle g_1, g_2, \ldots, g_k \rangle$ is called the *cartographic group* or the *monodromy group* of the constellation $[g_1, g_2, \ldots, g_k]$.

Definition 4.5. *Any two k-constellations $[g_1, g_2, \cdots, g_k]$ and $[h_1, h_2, \cdots, h_k]$ are said to be equivalent or conjugated (notation; $[g_1, g_2, \cdots, g_k] \sim [h_1, h_2, \cdots, h_k]$) if there exists $\sigma \in S_n$ such that $h_i = \sigma g_i \sigma^{-1}$ for all $i \in \{1, 2, \cdots, k\}$.*

[1] 5 out of 416 Belyi maps in [4] produce 4+1 singularities. We don't include them in our Belyi table.

We will work with 3, 4 and 5-constellations in this paper. The braid group B_k generated by the braids $\sigma_1, \dots, \sigma_{k-1}$ acts on a k-constellation in the following way:

$$\begin{aligned} \sigma_i : \; & g_i \mapsto g_{i+1}, \\ & g_{i+1} \mapsto g_{i+1}^{-1} g_i g_{i+1} \quad \text{and} \\ & g_j \mapsto g_j, \; j \neq i, i+1. \end{aligned}$$

i.e, $\sigma_i : [g_1, \dots, g_{i-1}, g_i, g_{i+1}, \dots, g_k] \mapsto [g_1, \dots, g_{i-1}, g_{i+1}, g_{i+1}^{-1} g_i g_{i+1}, \dots, g_k]$

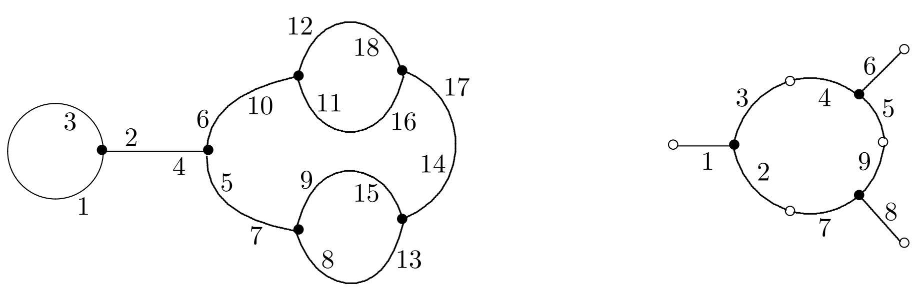
Definition 4.6. *Any two Belyi maps f and g are said to be Möbius equivalent if there exists a Möbius transformation m such that $f = g(m)$. A Belyi map f up to Möbius equivalence corresponds to a 3-constellation $[g_0, g_1, g_\infty]$ up to equivalence (i.e, conjugation). We use the notation g_0, g_1, g_∞ as these are the monodromy permutations around $0, 1, \infty$ respectively.*

Definition 4.7. *A dessin is a connected and oriented graph whose vertices are bi-colored (say, black and white) in such a way that any edge joins a black and a white vertex.*

Remark 4.8. *Given a Belyi map f, the corresponding dessin is the graph of $f^{-1}([0,1])$ where*

1. *$f^{-1}(\{0\})$ is the set of black vertices,*
2. *$f^{-1}(\{1\})$ is the set of white vertices,*
3. *$f^{-1}((0,1))$ are the edges and*
4. *$f^{-1}(\{\infty\})$ corresponds to the set of faces.*

Here are two examples of dessins which correspond to the Belyi maps with $(\frac{1}{3}, \frac{1}{2}, 0)$-singularity-count 5:



I. A clean planar dessin of degree 18

II. A planar dessin of degree 9

Fig. 3: Planar dessins

Definition 4.9. *A dessin in which each white vertex has valence (total number of edges coming out of the vertex) 2 is called a* clean *dessin. It is customary to omit the white vertices of a* clean *dessin. In such a case, any curve joining black vertices corresponds to an element of $f^{-1}(\{1\})$.*

In Figure 3, black vertex represents a point in $f^{-1}(\{0\})$, i.e; a point above 0 and white vertex represents a point in $f^{-1}(\{1\})$, i.e; a point above 1. The curves joining any two neighbouring black and white vertices are called the *edges*. The corresponding Belyi map projects each edge homeomorphically to $(0, 1)$. The number of edges of a dessin is called its *degree*.

There is a correspondence [7] between dessins, Belyi maps up to Möbius equivalence and 3-constellations $[g_0, g_1, g_\infty]$ up to conjugation. The ordering around black (resp. white) vertices in the dessin correspond to the cycles in g_0 (resp. g_1) and their valences correspond to the length of cycles. Faces on the dessin correspond to the points above ∞. So they produce the cycles in g_∞; labels on the faces build the cycles.

We placed labels in the dessins above to obtain permutations from the diagram but dessins are the graphs without any labelling. Labels are also useful as they help us to understand the procedure of inserting edges into existing dessins (see Figure 5 for details). These 'labelled dessins' are 3-constellations. A dessin is basically a '3-constellation without labels', more precisely, an equivalence class of 3-constellations mod conjugation. Any two conjugated 3-constellations represent the same dessin (with different labelling). The genus of a dessin can be computed from the Riemann-Hurwitz formula as:

$$\#\text{ black vertices} + \#\text{ white vertices} + \#\text{ faces} - \#\text{ edges} = 2 - 2 \cdot \text{genus}$$

We consider the Belyi maps $f : \mathbb{P}^1 \to \mathbb{P}^1$. So our dessins are planar, i.e; their genus is zero. The dessin in Figure 3.I has 6 black vertices, 9 double edges (i.e, 18 edges plus 9 white vertices) and 5 faces. This is a clean and planar dessin. Corresponding 3-constellation $[g_0, g_1, g_\infty]$ of order 18 can be read from Figure 3.I as:

$$g_0 = (1\ 2\ 3)\ (4\ 5\ 6)\ (7\ 8\ 9)\ (10\ 11\ 12)\ (13\ 14\ 15)\ (16\ 17\ 18).$$

$$g_1 = (1\ 3)\ (2\ 4)\ (5\ 7)\ (6\ 10)\ (8\ 13)\ (9\ 15)\ (11\ 16)\ (12\ 18)\ (14\ 17).$$

We often omit g_∞ because $g_\infty = (g_0 \cdot g_1)^{-1}$.

Each planar dessin determines a Belyi map $f : \mathbb{P}^1 \to \mathbb{P}^1$ up to Möbius equivalence. The dessin in Figure 3.I corresponds to the following degree 18 Belyi map (up to Möbius equivalence) with $(\frac{1}{3}, \frac{1}{2}, 0)$-singularity-count 5:

$$f = \frac{4}{27} \frac{\left(x^6 - 4\,x^5 + 5\,x^2 + 4\,x + 4\right)^3}{(x-4)\left(5\,x^2 + 4\,x + 4\right)^2 x^5}. \tag{5}$$

Swapping 0 and ∞ results in replacing f by $\frac{1}{f}$, ${}_2F_1(\frac{1}{12}, \frac{5}{12}; 1 \,|\, \frac{1}{f})$ satisfies a differential operator $L \in Class\left(H_{1,x}^{\frac{1}{12}, \frac{5}{12}}\right)$ which has five non removable singularities (with at least one logarithmic singularity). The main task is to tabulate all such f's.

Now we explain the procedure to compute all dessins of degree ≤ 18 (equivalently, all 3-constellations $[g_0, g_1, g_\infty]$ of degree ≤ 18 up to conjugacy) that are relevant to our project, i.e, that are planar and have singularity-count 5.

4.1 Computing 3-constellations

We begin with the 3-constellation of degree 1. We can draw it as the 'labelled dessin' (Recall that a dessin means the equivalence class of 3-constellations mod conjugacy). Then we compute 3-constellations of higher degree recursively, i.e; given a 'labelled dessin' of degree $n-1$, insert one more edge to get a 'labelled dessin' of degree n for $n = 2, 3, \ldots$. Inserting an edge means the following modifications on g_0, g_1: (i) inserting a new number n into an existing cycle, or (ii) introducing a new 1-cycle with that number n. Lets draw the 'labelled dessin' of degree 1:

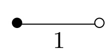


Fig. 4: 'Labelled dessin' of degree 1

The corresponding permutations are: $g_0 = (1)$, $g_1 = (1)$. Now we want to insert an edge to produce 'labelled dessins' of degree 2. The asterisks indicate the possible places to insert edge # 2:

$$g_0 = (1\,*)(*),\ \ g_1 = (1\,*)(*)$$

This procedure gives the following 4 candidates:
$(i)\, g_0 = (1\,2),\ g_1 = (1\,2)$ $(ii)\, g_0 = (1\,2),\ g_1 = (1)(2)$ $(iii)\, g_0 = (1)(2),\ g_1 = (1\,2)$
$(iv)\, g_0 = (1)(2),\ g_1 = (1)(2)$
Candidate (iv) is not acceptable as that gives a disconnected graph which is not a dessin. Given a 'labelled dessin' D of degree $n-1$, there are n^2 choices to insert the new edge (labelled 'n') into g_0, g_1. After discarding the one choice yielding a disconnected graph, we get $n^2 - 1$ 'labelled dessins' of degree n from D.

Now we explain with an example, how the algorithm $\mathbf{Insert}(g_i, j, n)$, $i \in \{0, 1\}$, $1 \leq j \leq n$ inserts an edge n at j^{th} position in g_i:

Example 4.10. *Let $g_0 = (1\,2)(4\,5)(6\,8) \in S_8$ be given. We want to insert edge #9 at 6^{th} position in g_0; i.e, we want to compute* $\mathbf{Insert}(g_0, 6, 9)$.
Step 1: *Rewrite g_0 in complete form (including 1-cycles) so that all edges 1 – 8 appear:*

$$g_0 = (3)(7)(1\,2)(4\,5)(6\,8)$$

Step 2: *Placeholders (asterisks) indicate all possible positions in g_0 where we can insert 9:*

$$g_0 = (3\,*)(7\,*)(1\,*\,2\,*)(4\,*\,5\,*)(6\,*\,8\,*)(*)$$

Note that there are 9 possibilities in total.
Step 3: *Locate 6^{th} placeholder and insert 9 there:*

$$\mathbf{Insert}(g_0, 6, 9) := (3)(7)(1\,2)(4\,5\,9)(6\,8) = (1\,2)(4\,5\,9)(6\,8)$$

The following algorithm computes 3-constellations of degree $\leq n$.
Note: we will not write g_∞ in algorithms unless required. Given g_0 and g_1, we can compute $g_\infty = (g_0 \cdot g_1)^{-1}$. So a 3-constellation will be denoted as $[g_0, g_1]$.

Algorithm 4.1: Compute all 3-constellations of degree $\leq n$
Input: n
Output: A table with all 3-constellations of degrees $1, 2, \ldots, n$.
Step 1: Table[1] $:= \{[(1), (1)]\}$ (the 3-constellation in Figure 4).
Step 2: Table[n] $:= \{\, [\,\mathbf{Insert}(g_0, i, n), \mathbf{Insert}(g_1, j, n)\,] \mid [\,g_0, g_1\,] \in$ Table[$n-1$], $1 \leq i, j \leq n,\ \{i, j\} \neq \{n\}\,\}$

The following diagram illustrates the procedure of computing 3-constellations:

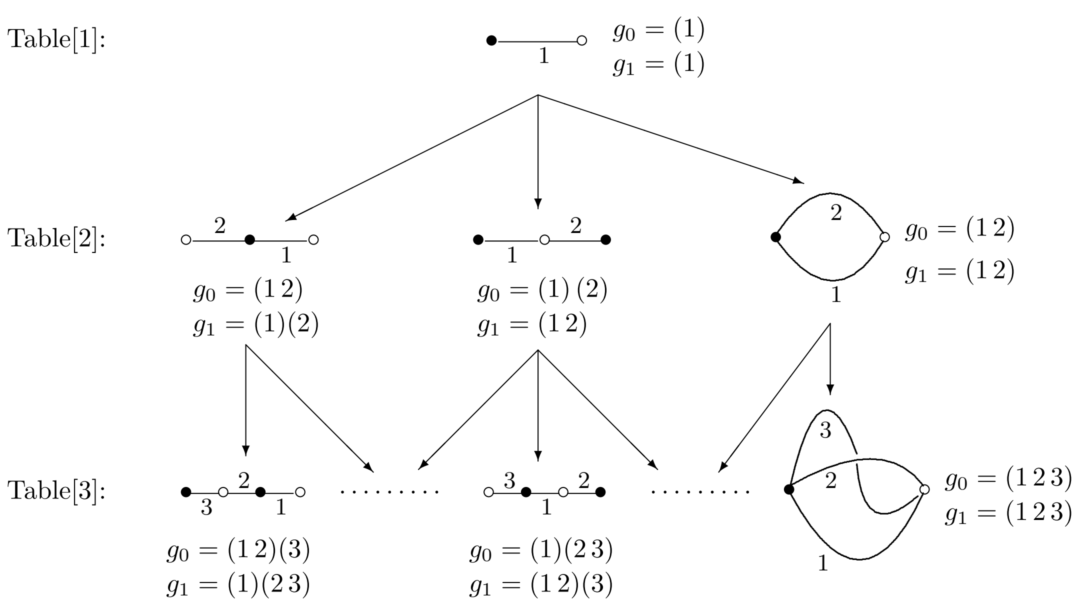


Fig. 5: Computing 3-constellations

Let T_n denote the number of 3-constellations of degree n; i.e, the number of elements of Table[n]. Then we have the following recurrence relation:

$$T_1 = 1,\ \ T_n = (n^2 - 1) \cdot T_{n-1} \ \text{ which gives}$$

$$T_n = \frac{(n-1)!(n+1)!}{2} = 1, 3, 24, 360, 8640, 302400, 14515200, 914457600, 73156608000, \ldots$$

This sequence has a huge growth. An efficient C-implementation of **Algorithm 4.1** could compute 3-constellations up to $n = 8$, but Maple will run out of memory at that point. To prevent computational explosion, we implement some special measures in **Algorithm 4.1** that are compatible with our project. We will discuss in brief about these features in the next sections, details can be found in [9].

4.2 Computing Dessins

Algorithm 4.1 has a huge growth because it returns the same dessin many times. Our target $n = 18$ is unreachable unless we identify conjugated 3-constellations and discard all but one of them (not discarding conjugated 3-constellations means computing the same dessin many times). Table[2] in Figure 5 has three non-conjugated 3-constellations. So these are three distinct dessins. Table[3] has twenty-four 3-constellations. After discarding 17 conjugates we get only 7 distinct dessins on that level.
Let $g = (m_1, m_2, \ldots)(n_1, n_2, \ldots)\ldots \in S_n$. Suppose $\sigma \in S_n$.
Denote $g^\sigma := (\sigma(m_1), \sigma(m_2), \ldots)(\sigma(n_1), \sigma(n_2), \ldots)\ldots = \sigma g \sigma^{-1}$
Given $D = [g_0, g_1, g_\infty]$ denote $D^\sigma = [g_0^\sigma, g_1^\sigma, g_\infty^\sigma]$.

Definition 4.11. *If D_1, D_2 are 3-constellations of degree n, they represent the same dessin if and only if $\exists\, \sigma \in S_n$ such that $D_1 = D_2^\sigma$.*

Conjugation is a reordering of the numbers in g_0, g_1, g_∞. We represent the reordering with a permutation $\pi \in S_n$. We represent π as a list $[\,\pi(1), \pi(2), \ldots, \pi(n)\,]$ with $\pi(\,i\,) \in \{1, 2, \ldots, n\}$.
The permutation $\pi \in S_n$ is computed as follows:
Step 1: Choose a base point $b \in \{1, 2, \ldots, n\}$. Take $\pi := [\,b\,]$.
Step 2: Let l be the last element of π, compute $g_0^k(l)$, $k = 1, 2, \ldots$ and append them to the list π until $g_0^k(l) \in \pi$. If π has n elements, then stop.
Step 3: Consider $g_1(c)$ for each $c \in \pi$ and append the first $g_1(c)$ that is not in π to the list π. Then return to **Step 2**.
Let $\pi = [\,a_1, a_2, \ldots, a_n\,]$ be the ordering obtained from $[g_0, g_1]$ with the base point b. Then $[g_0^\sigma, g_1^\sigma]$ with base point $\sigma(b)$ will return the permutation $\sigma\pi = [\,\sigma(a_1), \sigma(a_2), \ldots, \sigma(a_n)\,]$. Moreover

$$(\sigma\pi)^{-1} g_i^\sigma (\sigma\pi) = \pi^{-1}\sigma^{-1}\sigma g_i \sigma^{-1}\sigma\pi = \pi^{-1} g_i \pi,\ i \in \{0, 1\}$$

That means conjugating g_i by π is the same as conjugating g_i^σ by $\sigma\pi$. The remaining issue is how to match the base points. Any two conjugated 3-constellations will produce the same set of 3-constellations if we repeat this procedure over all $b \in \{1, 2, \ldots, n\}$ and compute the conjugation for each. We sort this set with the help of suitable ordering (for example, the lexicographic ordering) and use the first element which will be unique for this dessin.

4.3 Discarding Non-planar Dessins

Given a rational Belyi map $f : X \longrightarrow \mathbb{P}^1$ of degree n, the genus of X is given by the following formula [7]:

$$2g(X) - 2 = n - n_0 - n_1 - n_\infty$$

where n_i = number of distinct elements in $f^{-1}(\{i\})$, which is the number of cycles in g_i. We discard all 3-constellations which are non-planar,i.e; those with positive genus. For example; the last 3-constellation of Table[3] in Figure 5 is non-planar (that has genus 1).
Finally we want to consider only those dessins which are relevant to our project, i.e; the dessins with singularity-count 5. The following section explains this procedure for the exponent differences $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3})$. The cases $(0, \frac{1}{2}, \frac{1}{4})$ and $(0, \frac{1}{2}, \frac{1}{6})$ are done similarly.

4.4 Choosing Relevant Dessins

Implementation of the following measure helps us discard many irrelevant dessins. The following algorithm gives the 'weighted' singularity-count of a dessin:

Definition 4.12. *Let $p \in \{0, 1, \infty\}$. Given a list $L_p = [l_1, l_2, \dots, l_n]$, $l_i \in \mathbb{N}$ (branching pattern above p) and the exponent difference e_p of $H^{a,b}_{c,x}$, suppose d be the denominator of e_p (take $d = \infty$ if p is a logarithmic singularity). The following formula gives the weight w_i assigned to each l_i:*

$$w_i = \begin{cases} 1 & \text{if } d = \infty \text{ or } l_i > d \text{ or } l_i \le d-2 \ ; \\ 0 & \text{if } l_i = d \ ; \\ \frac{1}{2} & \text{if } l_i = d-1 \ . \end{cases}$$

*The case $l_i = d$ corresponds to a regular point, while the case $l_i = d-1$ is counted half to ensure that the total weighted singularity-count does not decrease when the **Insert** program inserts an edge.*

The sum $W_p := \sum_{i=1}^{n} w_i$ gives the weighted singularity-count above p.

Weighted-singularity-count of a 3-constellation is similar to actual singularity count with a small modification that some of the points here count as half-singularities, which ensures that the total weighted count does not decrease no matter how many edges we attach to a 3-constellation.

Remark 4.13. *Let D be a planar dessin of degree n. Given the exponent differences (e_0, e_1, e_∞) of $H^{a,b}_{c,x}$, let w be the weighted singularity-count and d be the singularity-count of D. Then;*

1. $w \le d$
2. *Let $\tilde{D}$ be a planar dessin of degree $n+1$ obtained after inserting an edge in D and $\tilde{w}$ be the weighted-singularity-count of $\tilde{D}$, then:*

   $w \le \tilde{w}$.

Property #2 follows from the fact that if $n+1 \in \{i, j\}$ then the number of vertices increases by 1, and if $n+1 \notin \{i, j\}$ then the number of faces increases by 1. Using remark 4.13, we can discard a dessin as soon as its weighted singularity-count exceeds 5.

Remark 4.14. *These special features of discarding 3-constellations on the basis of conjugation, genus and weighted-singularity-count are crucial in this procedure as each of them reduces the number of cases by a very large factor. The growth of 3-constellations is so high that if we do not implement any one of these measures, the computer runs out of memory long before we reach $n = 18$.*

Now we put all algorithms together to give the main algorithm which computes all dessins with $(0, \frac{1}{2}, \frac{1}{3})$-singularity-count d. The other cases $(0, \frac{1}{2}, \frac{1}{4})$ and $(0, \frac{1}{2}, \frac{1}{6})$ are done similarly. We ran this algorithm for $d \le 6$ and $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{k})$, $k \in \{3, 4, 6\}$; the results can be found in [4].

Algorithm 4.2: Compute All Dessins with a Specific $(0, \frac{1}{2}, \frac{1}{3})$-singularity-count
Input: d
Output: all planar dessins $[\, g_0, g_1 \,]$ with $(0, \frac{1}{2}, \frac{1}{3})$-singularity-count $= d$.

```
Table[1] := { [ (1), (1) ] };
  for n from 2 to 6(d − 2) do
    Table[n] := { };
      for [g̃0, g̃1] in Table[n − 1] do
          for i from 1 to n do
            g0 := Insert(g̃0, i, n);
              for j from 1 to n while {i, j} ≠ {n} do
                g1 := Insert(g̃1, j, n);
                g∞ := (g0 · g1)^−1;
                 'if [g0, g1, g∞] is non-planar (genus > 0) or has weighted-singularity-count > d
                   then discard it'
                          [ ĝ0, ĝ1 ] := dessin of [g0, g1]
                            (Recall that it is same for conjugated 3-constellations);
```

```
                    Table[n] := Table[n] ⋃{ [ ĝ_0, ĝ_1 ] };
                end do;
            end do;
        end do;
    end do;
    ANS := { };
        for n from 1 to 6(d − 2) do
            for D in Table[n] do
                if ( (0, 1/2, 1/3)-singularity-count of D = d then
                    ANS := ANS ⋃{D};
                end if;
            end do;
        end do;
    Return ANS;
```

Implementation of these special features discards many dessins. So the number of elements of Table[n] grows much slower, and the computation no longer runs out of memory. We computed all dessins with $(0, \frac{1}{2}, \frac{1}{k})$-singularity-count $d \leq 6$ where $k \in \{3, 4, 6\}$ (degree $n \leq 24$). Although we are interested in $d = 5$, we ran **Algorithm 4.2** for $d = 3, 4, 5, 6$. The outputs contain the following number of dessins of degree $n = 1, 2, \ldots, 6(d-2)$:

d	n	dessin count for $(0, \frac{1}{2}, \frac{1}{3})$, degree $= 1, \ldots, n$
3	≤ 6	1, 2, 1, 1, 0, 2
4	≤ 12	0, 1, 3, 4, 3, 6, 4, 6, 4, 4, 0, 6
5	$\mathbf{\leq 18}$	**0, 0, 2, 6, 12, 19, 22, 26, 32, 39, 36, 50, 40, 42, 32, 32, 0, 26**
6	≤ 24	0,0,0,9,23,59,112,176,240,315,332,429,437,470,518,579,536,620,512,444,336,336,0,191

Tab. 2: Dessin count for $d = 3, 4, 5, 6$

Dessins for $d = 6$, $n = 24$, $(0, \frac{1}{2}, \frac{1}{3})$ were previously found by Beukers and Montanus [7]. They used a combination of computer computation and hand computation and found 190 dessins (we emailed them their missing dessin and they have used it to correct their website). This incident shows why it is important to use only machine computations to find the dessins, if any human interaction is needed then the chance of a gap is too high.

After computing the dessins, the next task is to compute the corresponding Belyi maps. If we have a Belyi map (up to Möbius equivalence) for each dessin, then our table of Belyi maps is complete. Dessins give the branching pattern of corresponding Belyi maps which give a way to compute the maps. Small cases are easy to compute, cases up to degree 16 can be computed using Gröbner basis. There are no dessins for degree 17 and we use the special techniques given in [7] to compute Belyi maps of degree 18. An example is given in the next section.

5 Belyi-1 Maps

Belyi-1 maps have one more branch point t outside $\{0, 1, \infty\}$, which has only one ramification point $\tilde{t}$, with multiplicity 2. Such point $\tilde{t}$ is called a simple ramified point. These maps correspond to 4-constellations $[g_0, g_1, g_t, g_\infty]$ where g_t is a 2-cycle. The point $t \notin \{0, 1, \infty\}$ can vary, which produces these maps as one dimensional families.

Hence, up to equivalence there is a correspondence:

$$[g_0, g_1, g_t, g_\infty] \longleftrightarrow \text{an element of } K(x)$$

where K is an algebraic extension[2] of $\mathbb{Q}(t)$.

[2] in all case for $d = 5$, the field K turned out to be isomorphic to $\mathbb{Q}(s)$. We use `parametrization` in Maple to find such isomorphism.

Definition 5.1.

1. *A near-dessin of a Belyi-1 map is an equivalence class of 4-constellations* $[g_0, g_1, g_t, g_\infty]$ *mod conjugation where* g_t *is a 2-cycle.*

2. *Belyi-1 maps (up to Möbius transformation) correspond to 4-constellations* $[g_0, g_1, g_t, g_\infty]$ *(up to conjugation and braid group action).*

Example 5.2. *Consider the following one-dimensional family of functions:*

$$f_1(x,s) = \frac{4}{27}\,\frac{\left(sx^3 - 2\,sx^2 + sx - 3\right)^3}{sx^3 - 2\,sx^2 + sx - 4}$$

The branching pattern of f_1 *above* $0, 1, \infty$ *is* $[3,3,3], [1,2,2,2,2], [1,1,1,6]$. *Using the Riemann-Hurwitz formula, we find that there is one more* branch point $t \notin \{0,1,\infty\}$ *and the ramification pattern of* f_1 *above* t *is* $[1,1,1,1,1,1,1,2]$. *So,* f_1 *is a Belyi-1 map. We compute* t *using its corresponding* ramification point *(Note that the derivative of* f_1 *vanishes at ramification points). For* f_1*, we get* $t = \frac{1}{19683}\frac{(4\,s-81)^3}{s-27}$. *For each fixed* $t \notin \{0,1,\infty\}$*, we get* 3 *distinct values of* s *which produce* 3 *distinct Belyi-1 maps up to Möbius equivalence. These three Belyi-1 maps have the same branching pattern, but their near-dessins differ. However, analytic continuation of* t *around* $0, 1, \infty$ *permutes these three near-dessins. Such near-dessins lie in the same orbit under the action of braid group.*
Now consider another one-dimensional family of functions:

$$f_2(x,s) = \frac{\left(sx^3 - 2\,sx^2 - 9\,x^2 + 18\,x + sx - 3\right)^3}{27\left(sx^3 - 2\,sx^2 - 9\,x^2 + 18\,x + sx - 1\right)}$$

f_2 *is also a Belyi-1 map with the same branching pattern as* f_1. *The fourth* branch point *for* f_2 *is* $t = \frac{2}{19683}\frac{\left(2\,s^3+27\,s^2+486\,s-1458\right)^3}{s^4(s^3+27\,s^2+243\,s-729)}$. *For each fixed* t *in this case, we get* 9 *values of* s *which correspond to* 9 *distinct near-dessins, again, in one orbit.*

f_1 *and* f_2 *are two distinct families of Belyi-1 maps as their monodromy groups are different. For* f_1*, the monodromy group* $\langle g_0, g_1, g_t, g_\infty\rangle$ *is a group of order* 1296*, and for* f_2 *it equals* S_9. *Our combinatorial search shows that near-dessins with branching type* $[3,3,3], [1,2,2,2,2], [1,1,1,1,1,1,1,2]$, $[1,1,1,6]$ *belong to* 2 *distinct braid orbits. This result implies that* $\{f_1(x,s), f_2(x,s)\}$ *completely cover this branching pattern. Galois theory further tells us if* $\mathbb{C}(x)/\mathbb{C}(f)$ *has subfields. We use these monodromy groups to find decompositions (if any) of Belyi-1 maps. Our computation shows that* f_1 *has a decomposition* $g(h)$ *where each* g, h *has degree* 3 *in* x. *Both* f_1, f_2 *are Belyi-1 maps with* $(\frac{1}{3}, \frac{1}{2}, 0)$*-singularity-count* 5. *Our task is to compute all such Belyi-1 maps and to prove completeness.*

The degree bound for Belyi-1 maps in our project is 12 (Table 1). We use the following steps to compute such maps:

1. Compute all possible branching patterns for degree $n \le 12$. Note that the candidate branching patterns must (i) satisfy Riemann-Hurwitz formula (6), (ii) produce a Belyi-1 map, and (iii) have singularity-count 5 .

2. Compute all *near-dessins* (if any) for each branching pattern

3. Group them together by braid orbit

4. Compute a Belyi-1 map for each orbit

For example, near-dessins of degree 10 for the choice $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3})$ are computed as follows. *Let's switch the roots and poles of* f*, so we assume* $(e_0, e_1, e_\infty) = (\frac{1}{3}, \frac{1}{2}, 0)$.
Step 1: *Finding the list of candidate branching patterns*:
Our program produces the following list of possible branching patterns for Belyi-1 maps of degree 10:

$$\begin{aligned}B_{10} = \{&[\,[1,3,3,3],[2,2,2,2,2],[1,1,1,7]\,],[\,[1,3,3,3],[2,2,2,2,2],[1,1,2,6]\,],\\&[\,[1,3,3,3],[2,2,2,2,2],[1,1,3,5]\,],[\,[1,3,3,3],[2,2,2,2,2],[1,1,4,4]\,],\\&[\,[1,3,3,3],[2,2,2,2,2],[1,2,2,5]\,],[\,[1,3,3,3],[2,2,2,2,2],[1,2,3,4]\,],\\&[\,[1,3,3,3],[2,2,2,2,2],[1,3,3,3]\,],[\,[1,3,3,3],[2,2,2,2,2],[2,2,2,4]\,],\\&[\,[1,3,3,3],[2,2,2,2,2],[2,2,3,3]\,]\}.\end{aligned}$$

where branching patterns are above $0, 1$ and ∞ respectively. The branching pattern above the fourth point t outside $\{0,1,\infty\}$ is [1,1,1,1,1,1,1,1,2].

Step 2: *Computing near-dessins, i.e; equivalence classes of 4-constellations mod conjugation*:

1. For g_0, we have a 1-cycle and three 3-cycles. As we are computing these permutations up to equivalence, we can take $g_0 = (1\,2\,3)\,(4\,5\,6)\,(7\,8\,9)\,(10)$.
2. g_1 has five 2-cycles. Total number of $g_1 \in S_{10}$ that are a product of 5 disjoint 2-cycles is $9{\cdot}7{\cdot}5{\cdot}3{\cdot}1 = 945$. We loop over all such g_1's.
3. g_t has a 2-cycle (and eight 1-cycles). Hence we have $\binom{10}{2} = 45$ choices for g_t. We loop over all such g_t's.
4. For each of the $945 \cdot 45 = 42525$ triples (g_0, g_1, g_t), we check the following two properties:
   i. Is the group $\langle g_0, g_1, g_t \rangle$ transitive?
   ii. Does the product $g_0 g_1 g_t$ have 4 disjoint cycles? ($g_0 g_1 g_t = g_\infty^{-1}$ and $|f^{-1}(\{\infty\})| = 4$)

   After computing 4-constellations we found that only the following branching patterns actually occur above ∞ (here we omit the branching at $0, 1, t$ because for degree 10 they all happened to be the same):

   $$[1,1,1,7], [1,1,2,6], [1,1,3,5], [1,1,4,4], [1,2,2,5], [1,2,3,4], [2,2,3,3].$$

5. Item 4 produced a list of 4-constellations. Next we compute the near-dessins, i.e. the equivalence classes mod conjugation, similar to the procedure in Section 4. We also group together those near-dessins that fall into the same orbit under the action of braid group. One Belyi-1 map $f(x, s) \in K(x)$, computed below, covers precisely one braid orbit. To check that the f's we computed (*see below*) are complete, we need to compute their near-dessins, and then check that every braid orbit occurs among our f's. For all such f's, we further checked that the degree of $[K : \mathbb{Q}(t)]$ equals the number of *near-dessins* in that orbit. This means for a fixed t, each near-dessin corresponds to precisely one value of s.

Remark 5.3. *Out of* 9 *candidates in* B_{10} *from Step 1, only* 7 *of them allowed a near-dessin, and hence, a family of Belyi-1 maps. For degree* 10*, there are no* 4*-constellations corresponding to the branching patterns* $[1,3,3,3]$ *and* $[2,2,2,4]$ *above infinity, which means there are no Belyi-1 maps for those patterns. Some branching patterns may produce more than one family of Belyi-1 maps, see Example 5.2.*

Step 3: *Grouping near-dessins by braid orbit:*
Applying braid action we find that each branching pattern given in Step 2 above has only one braid orbit, i.e, for degree 10, we do not have the situation like Example 5.2.
Step 4: *Computing Belyi-1 maps:*
Let's compute the Belyi-1 map with branching pattern
$[\,[1,1,3,5], [2,2,2,2,2], [1,1,1,1,1,1,1,1,2], [1,3,3,3]\,]$ above $0, 1, t$ and ∞ respectively. Note: to compute the map(s), we only need the branching pattern. But to prove that we found all of them, we need to compare them with the orbit(s) of the near-dessins.
Step (i): General structure of f:
To make the computation easier, let's take the branching pattern as [[1,3,3,3],[1,1,3,5],[2,2,2,2,2]] above $0, 1, \infty$ respectively. Let's place the unramified root of f at $x = 1$, and the roots of $(1 - f)$ with multiplicity $3, 5$ at $x = 0, x = \infty$ respectively. This fixes our f up to Möbius transformation and the map has now the following form:

$$f := \frac{c\,(x-1)\left(x^3 + a_2\,x^2 + a_1\,x + a_0\right)^3}{\left(x^5 + b_4\,x^4 + b_3\,x^3 + b_2\,x^2 + b_1\,x + b_0\right)^2}.$$

Step (ii): Generating equations:
The numerator of $(1-f)$ must have the form: $x^3(Ax^2+Bx+C)$ where A and C are non zero. The coefficients of x^n for $n = 0, .., 2, 6, .., 10$ from the numerator of $(1 - f)$ produce the following equations:
$eqns := [\,1 - c, b_0^2 + c\,a_0^3, 2\,b_4 - 3\,c\,a_2 + c, 2\,b_1\,b_0 + 3\,c\,a_1\,a_0^2 - c\,a_0^3, 2\,b_3 + b_4^2 + 3\,c\,a_2 - 3\,c\,a_1 - 3\,c\,a_2^2, 2\,b_2\,b_0 + b_1^2 - 3\,c\,a_1\,a_0^2 + 3\,c\,a_2\,a_0^2 + 3\,c\,a_1^2 a_0, 2\,b_4\,b_3 + 2\,b_2 - 6\,c\,a_2\,a_1 + 3\,c\,a_1 + 3\,c\,a_2^2 - 3\,c\,a_0 - c\,a_2^3, 2\,b_4\,b_2 + 2\,b_1 + b_3^2 - 3\,c\,a_2^2 a_1 - 6\,c\,a_2\,a_0 + 6\,c\,a_2\,a_1 + 3\,c\,a_0 + c\,a_2^3 - 3\,c\,a_1^2\,]$.
Step (iii): Elimination and Resultants:
We have 8 equations with 9 unknowns, which produces a one dimensional family. We can recursively eliminate the unknowns c, b_4, b_3, b_1, b_2 and b_0 from their corresponding linear equations. Then we have three unknowns $\{a_0, a_1, a_2\}$ and two non trivial equations left. The equations are rather big, but we can compute their resultant with respect to a_2 and then factor. This produces a polynomial relation between a_0 and a_1, i.e. an

algebraic curve which turned out to have genus 0, which means that $\mathbb{C}(a_0, a_1) \cong \mathbb{C}(s)$ for some s. We can find such isomorphism using Maple's `parametrization` and we obtain $a_0 = -s^4$ and $a_1 = \frac{1}{9}s(-16+42s+s^3)$.
Step (iv): The result:
We update f each time when we eliminate an unknown. After re-arranging $\{0,1,\infty\}$ back to the original ramification pattern, we get g as:

$$g = 1 - \frac{1}{f} = \frac{64\,x^3(s-1)^8(9\,x^2+16\,x+6\,s^2x-40\,sx+s^4+8\,s^3)}{(1-x)(9\,x^3+15\,x^2-48\,sx^2+6\,s^2x^2-16\,sx+42\,s^2x+s^4x-9\,s^4)^3}.$$

Remark 5.4. *There are some Belyi-1 maps which produce 5+2 singularities, i.e. 5 non removable and 2 removable singularities. We will skip such maps because the corresponding differential operator will be solved by Belyi-2 maps, see Section 6 for more details.*

Remark 5.5. *For each Belyi-1 map, we compute the size of its braid orbit. In the case where $[\mathbb{Q}(s):\mathbb{Q}(t)]$ is larger than the orbit size, we compute a subfield $\mathbb{Q}(t) \subseteq \mathbb{Q}(\tilde{s}) \subset \mathbb{Q}(s)$ such that $f \in \mathbb{Q}(\tilde{s}, x)$ and then rewrite f in terms of $\tilde{s}$.*

Remark 5.6. ***Completeness:*** *For $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3})$, there are 68 Belyi-1 maps $f \in \mathbb{Q}(s)(x)$. For each f in our table, we compute 4-constellation $[g_0, g_1, g_t, g_\infty]$ for some value of s (for example, with Maple's* `monodromy`*). Then we check if for every braid orbit (see Steps 2 and 3 above) our table has a Belyi-1 map with a 4-constellation in that orbit.*

6 Belyi-2 Maps

Our Belyi-2 maps have degree ≤ 6 and appear only for the case $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3})$. The branching patterns for these maps are $[1,1,1,1], [2,2], [1,3]$ for degree 4 and $[1,1,1,1,2], [2,2,2], [3,3]$ for degree 6.
Belyi-2 maps have two branch points outside $\{0,1,\infty\}$ that are free to move. Hence these maps are two dimensional families. We compute these maps using the data from $Sing(L_{inp})$; the singularity structure of input differential operator L_{inp}. Since 5 singularities, up to Möbius equivalence, have two degrees of freedom, this carries just enough information to extract the parameters in a 2-dimensional family. In this section we will explain the algorithm to compute Belyi-2 maps of degree four and will illustrate the procedure with an example. Degree six case can be done similarly. The implementation and more details can be found at `www.math.fsu.edu/~vkunwar/FiveSings/`.
We can write the generic map for the branching pattern $[1,1,1,1], [2,2], [1,3]$ as:

$$f = k_1 \frac{(x^4 + c_3x^3 + c_2x^2 + c_1x + c_0)}{(x-a_1)(x-a_2)^3} \quad \text{where} \quad 1 - f = k_2 \frac{(x^2 + b_1x + b_0)^2}{(x-a_1)(x-a_2)^3}.$$

We are in the case $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3})$. So roots of $x - a_1$ and $(x^4 + c_3x^3 + c_2x^2 + c_1x + c_0)$ are the non removable singularities of $H_{1,f}^{\frac{1}{12},\frac{5}{12}}$; we extract them from $Sing(L_{inp})$. We find the remaining part of f by solving equations.
We developed algorithms to compute such maps. They use the data from $Sing(L_{inp})$ and return the Belyi-2 maps f such that $Sing(H_{1,f}^{\frac{1}{12},\frac{5}{12}}) = Sing(L_{inp})$. Before giving the algorithm, let's observe, with an example, what it need to do:

Example 6.1. *Consider the following differential operator:*

$$L = \partial^2 + \frac{1}{3}\,\frac{\left(5\,x^5 - 56\,x^3 + 90\,x^2 - 48\,x - 18\right)}{x\,(x^2+x-3)\,(x^3-4\,x^2+3\,x+3)}\,\partial + \frac{1}{144}\,\frac{(16\,x^4 + 99\,x^3 - 370\,x^2 + 414\,x - 45)}{x\,(x^2+x-3)\,(x^3-4\,x^2+3\,x+3)}$$

Singularity structure of L in terms of places($\mathbb{Q}$) is:
$Sing(L) = \left\{[\infty, 0], [x, \frac{1}{3}], [x^3 - 4\,x^2 + 3\,x + 3, 0]\right\}$
Our main task is to compute $f = -3\,\frac{(x^3-4\,x^2+3\,x+3)}{x\,(x-3)^3}$ from $Sing(L)$ such that $1 - f = \frac{\left(x^2-3\,x+3\right)^2}{x\,(x-3)^3}$. Once we find such f then we can show that $Sing(H_{1,f}^{\frac{1}{12},\frac{5}{12}}) = Sing(L)$ and $\exp(\int r\,dx)\ {}_2F_1(\frac{1}{12}, \frac{5}{12}; 1\,|\,f)$ for some $r \in \mathbb{Q}(x)$ is a solution of L. Notice that f has the branching pattern $[1,1,1,1], [2,2], [1,3]$ above $0, 1, \infty$ respectively. It is easy to check that f is a Belyi-2 map and thus L is an example of a differential operator solvable in terms of Belyi-2 maps. $Sing(L)$ gives the numerator of f and a part of its denominator. However we need to know the constant factor -3 and the factor $(x-3)$ with multiplicity 3. We need algorithms which produce such Belyi-2 maps (if they exist) from given singularity structure.

In Example 6.1, the fact that the numerator of $(1-f)$ is a square will be used to generate equations. The implementation only considers solutions defined over the base field (i.e, field of definition). Let $C \subseteq \mathbb{C}$ be the base field of L_{inp} (the smallest field C such that $L_{inp} \in C(x)[\partial]$).
Note: The equations $EQa, EQb_1, EQc, EQd, EQns$ appearing in these algorithms are the results of the computation performed on the generic case of f and $1-f$ as explained above.
The following algorithm explains the procedure to compute Belyi-2 maps of degree 4.

Algorithm 6.1: Find Belyi-2 maps of degree 4 with $(0, \frac{1}{2}, \frac{1}{3})$-singularity-count 5.
Input: The base field $C \subseteq \mathbb{C}$ of input differential operator L_{inp}, variable x and $Sing(L_{inp})$ in terms of places(C)
Output: $\{f \in C(x) : f$ is a Belyi-2 map of degree 4 with the branching pattern $[1,1,1,1],[2,2],[1,3]$ such that $Sing(H_{1,f}^{\frac{1}{12},\frac{5}{12}}) = Sing(L_{inp})\}$.
Note: We are in the case $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3})$ and f has the branching pattern $[1,1,1,1],[2,2],[1,3]$. That means the roots of f and the pole of f with order 1 can be extracted from $Sing(L_{inp})$. Roots of $1-f$ and the pole of order 3 produce removable singularities, so they do not appear in $Sing(L_{inp})$ (see Figure 2 and Remark 2.26). To make the computation easier, let's make some changes which we will revert at the end. Let's take the branching pattern of f as $[1,3],[1,1,1,1],[2,2]$. Let's assume the following with this new branching pattern:

1. The root of f with multiplicity 1 is at infinity and
2. The sum of the roots of $1-f$ is zero

The assumptions 1 and 2 above correspond to non removable singularities, i.e, $Sing(L_{inp})$. If $Sing(L_{inp})$ is not compatible with these assumptions then we will make appropriate adjustments (transformations) in $Sing(L_{inp})$, see **Step 2** and **Step 3** below, which we will revert at the end. These changes will have the following effects on f: (i) numerator of f has degree 3 in x and (ii) the coefficient of x^3 in the numerator of $1-f$ vanishes. Then we get a Belyi-2 map, say F, in the following form:

$$F = \frac{2\, b_1\, (x-a)^3}{(x^2 + b_1\, x + b_0)^2} \tag{6}$$

such that the numerator of $1-F$ does not contain any duplicated roots and does not have any term with degree 3 in x.
Step 1: $Candidates := \{\,\}$;
Check the following three conditions in $Sing(L_{inp})$;

1. L_{inp} must have 5 non removable singularities; Compute the degree $deg(a(x))$ of $a(x)$ for each $[a(x), b] \in Sing(L_{inp})$. The sum $\sum deg(a(x))$ must be 5.
   Note: $a(x) = x - \infty$, which is denoted ∞ and replaced by 1 in our implementation, should also count as degree 1.
2. We need exactly one $[a(x), b] \in Sing(L_{inp})$ where $b \in \{\pm\frac{1}{3}, \pm\frac{2}{3}\} \bmod \mathbb{Z}$ and $deg(a(x)) = 1$.
3. b must be 0 mod $\mathbb{Z}$ for the remaining $[a(x), b]$.

If $Sing(L_{inp})$ does not satisfy these three conditions then stop.
Step 2: Let $P = \prod a(x)$ where $[a(x), b] \in Sing(L_{inp})$ and ∞ is replaced by 1. If the singularity with exponent difference $b \in \{\pm\frac{1}{3}, \pm\frac{2}{3}\} \bmod \mathbb{Z}$ (*second condition in Step 1 above*) is not at ∞ then find an appropriate Möbius transformation $m : x \mapsto \frac{a_1 x + a_2}{a_3 x + a_4}$ and compose that with P such that $P(m)$ will have that singularity at ∞.
Step 3: Let $\tilde{P}$ be the numerator of $P(m)$. $\tilde{P}$ should be a degree 4 polynomial in $C[x]$ whose roots are the singularities of L_{inp}, one of them is at ∞ now. If $\tilde{P}$ has degree 3 then that means one singularity of L_{inp} with $b = 0$ was already at ∞, and after applying m that should go to 0. In such a case, multiply $\tilde{P}$ by x to adjust the singularity at 0, and to get a degree 4 polynomial in $C[x]$. Let P_1 be the degree 4 polynomial; i.e,

$$P_1 = \begin{cases} \tilde{P} & \text{if } \tilde{P} \text{ has degree 4} \\ \tilde{P} \cdot x & \text{if } \tilde{P} \text{ has degree 3.} \end{cases}$$

Find a suitable translation $\tau : x \mapsto x - t$ and compose it with P_1 to eliminate the third degree term. Then make the result monic to obtain $P_2 = x^4 + p_2\, x^2 + p_1\, x + p_0$.
Note: EQb_1, EQa and $EQns$ in the following steps are the results of computations on F and $1 - F$.
Step 4: Solve the following equation for b_1:
$EQb_1 := b_1^9 + 24\, p_2 b_1^7 - 168\, p_1 b_1^6 - 78\, p_2^2 b_1^5 + 1080\, p_0 b_1^5 + 336\, p_1 p_2 b_1^4 + 80\, p_2^3 b_1^3 + 1728\, p_0 p_2 b_1^3 - 636\, p_1^2 b_1^3 - 168\, p_1 p_2^2 b_1^2 - 864\, p_0 p_1 b_1^2 - 27\, p_2^4 b_1 - 432\, p_0^2 b_1 + 216\, p_2^2 p_0 b_1 - 120\, p_2 p_1^2 b_1 - 8\, p_1^3$.
Step 4.1: For each $b_1 \in C$, substitute the value of b_1 in the following equation and solve that for a :
$EQa := b_1\, p_2 - p_1 - b_1^3 - 6\, b_1^2\, a - 6\, b_1\, a^2$.
Step 4.1.1: For each $a \in C$, substitute the values of b_1 and a in the following equations and solve their *gcd* for b_0:
$EQns := \{2\, b_0\, b_1 - p_1 - 6\, b_1\, a^2,\ b_0^2 - p_0 + 2\, b_1\, a^3,\ 2\, b_0 - p_2 + b_1^2 + 6\, b_1\, a\}$.
Step 4.1.1a: Substitute the values of a, b_1 and b_0 in F. Skip those F which do not have degree 4.
Step 4.1.1b: If F has degree 4, then $F_1 = 1 - \frac{1}{F}$ gives the map with right branching pattern $[1,1,1,1], [2,2], [1,3]$ (we had set the branching pattern as $[1,3], [1,1,1,1], [2,2]$ for F).
Step 4.1.1c: $f := F_1(\tilde{\tau}(\tilde{m}))$, where $\tilde{\tau} : x \mapsto x + t$ (inverse of *Step 3*) and $\tilde{m}$ is the inverse of m (*Step 2*) gives a candidate Belyi-2 map. $Candidates := Candidates \bigcup \{f\}$;
Step 5: Return $Candidates$.

Example 6.2. *Let's compute the Belyi-2 map of degree 4 for the differential operator considered in **Example 6.1**. Take $C = \mathbb{Q} \subset \mathbb{C}$. The input to **Algorithm 6.1** is the base field $C = \mathbb{Q}$ and the singularity structure:*
$Sing(L) = \big\{[\infty, 0], [x, \frac{1}{3}], [x^3 - 4\, x^2 + 3\, x + 3, 0]\big\}$ *(We replace ∞ by 1 in our implementation).*
Step 1: *It is easy to check that $Sing(L)$ satisfies all three conditions.*
Step 2: $P = x\ \big(x^3 - 4\, x^2 + 3\, x + 3\big)$, $m : x \mapsto \frac{1}{x}$.
Step 3: $\tilde{P} = 3\, x^3 + 3\, x^2 - 4\, x + 1$ *is a degree 3 polynomial. So* $P_1 = \tilde{P} \cdot x = 3x^4 + 3x^3 - 4x^2 + x$.
$\tau : x \mapsto x - \frac{1}{4}$, $P_2 = x^4 - \frac{41}{24}\, x^2 + \frac{9}{8}\, x - \frac{137}{768}$. *Hence* $[p_0, p_1, p_2] = [-\frac{137}{768}, \frac{9}{8}, -\frac{41}{24}]$.
Step 4: *Substituting p_0, p_1 and p_2 in EQb_1 (*Algorithm 5.9, Step 4*) we get:*
$EQb_1 = b_1^9 - 41\, b_1^7 - 189\, b_1^6 - \frac{10087}{24}\, b_1^5 - \frac{2583}{4}\, b_1^4 - \frac{292547}{432}\, b_1^3 - \frac{6051}{16}\, b_1^2 - \frac{74269}{768}\, b_1 - \frac{729}{64}$.
The only solution of $EQb_1 = 0$ in $C = \mathbb{Q}$ is $b_1 = -\frac{3}{2}$.
Step 4.1: *Substituting the values of b_1, p_1 and p_2 in EQa (*Algorithm 5.9, Step 4.1*) we get*
$EQa = \frac{77}{16} - \frac{27}{2}\, a + 9\, a^2$ *which gives* $a = \frac{7}{12}, \frac{11}{12}$.
Step 4.1.1: *Substituting the values of p_0, p_1, p_2, b_1 and $a = \frac{7}{12}$ in $EQns$ (*Algorithm 5.9, Step 4.1.1*) we get*
$EQns = \big\{\frac{31}{16} - 3\, b_0, 2\, b_0 - \frac{31}{24}, b_0^2 - \frac{961}{2304}\big\}$ *which has the solution $b_0 = \frac{31}{48}$. Repeating the same procedure with $a = \frac{11}{12}$ gives $EQns = \big\{\frac{103}{16} - 3\, b_0, 2\, b_0 - \frac{103}{24}, b_0^2 - \frac{4913}{2304}\big\}$ which has no solution.*
Step 4.1.1a: $F = \frac{2\, b_1\, (x-a)^3}{(x^2 + b_1\, x + b_0)^2}$.
Substituting $a = \frac{7}{12}$, $b_1 = -\frac{3}{2}$, $b_0 = \frac{31}{48}$ we get $F = -4\, \frac{(12\, x - 7)^3}{(48\, x^2 - 72\, x + 31)^2}$.
Step 4.1.1b: $F_1 = 1 - \frac{1}{F} = \frac{3}{4}\, \frac{(4\, x - 1)\big(192\, x^3 + 48\, x^2 - 316\, x + 137\big)}{(12\, x - 7)^3}$ *gives the right branching pattern.*
Step 4.1.1c: $\tilde{\tau} : x \mapsto x + \frac{1}{4}$ *and* $\tilde{m} : x \mapsto \frac{1}{x}$.
Hence $f := F_1(\tilde{\tau}(\tilde{m})) = -3\, \frac{(x^3 - 4\, x^2 + 3\, x + 3)}{x\, (x - 3)^3}$ *is the required Belyi-2 map.*

7 Additional Features

Once the table is complete, our algorithm is mainly the '*table look up*', where we choose candidate f's from the table. Since the tables are big, it is important that we discard the non-candidate entries (the entries which do not lead to the solution) as quickly as we can. In this section, we will discuss some features which help us to detect non-candidates so that we can readily discard them. These features make our algorithm faster and more efficient. So we add these features along with the maps in our table. We will also discuss about the decompositions, which give smaller (usually better) solutions.

7.1 Five Point Invariants

Definition 7.1. *Let $P_5 = \{S \subseteq \mathbb{P}^1(\mathbb{C})\,;\, |S| = 5\}$. A function $I : P_5 \to \mathbb{C}$ is called a* five point invariant *if it is invariant under Möbius transformation.*

Since Möbius transformations have three degrees of freedom, and $S \in P_5$ has five degrees of freedom; there are $5 - 3 = 2$ algebraically independent five point invariants.

Definition 7.2. *Let $[p_1, p_2, p_3, p_4]$ be a quadruple of distinct points in the Riemann sphere $\mathbb{P}^1(\mathbb{C}) = \mathbb{C} \cup \{\infty\}$. Their cross-ratio is denoted $(p_1, p_2; p_3, p_4)$ and defined as:*

$$(p_1, p_2; p_3, p_4) = \frac{(p_1 - p_3)\,(p_2 - p_4)}{(p_2 - p_3)\,(p_1 - p_4)}$$

Remark 7.3.

1. *If a point $p_i = \infty$, then the cross-ratio is computed by removing any factor containing p_i.*
2. *The cross-ratio depends on the ordering of the points $p_1, \ldots, p_4$, but it is invariant under Möbius transformation.*

Definition 7.4. *The* j-invariant *of an elliptic curve $y^2 = x^3 + px + q$ is defined as:*

$$j = 1728 \cdot \frac{4p^3}{4p^3 + 27q^2}$$

Remark 7.5. *Let $p_1, p_2, p_3, p_4 \in \mathbb{P}^1(\mathbb{C})$ be any four points.*

1. *The* j-invariant *of $y^2 = \prod(x - p_i)$ can be obtained by moving (with a Möbius transformation) one point to ∞, the sum of other 3 points to 0, and then applying definition 7.4.*
2. *Alternatively, the* j-invariant *can also be computed as $j = 256 \cdot \frac{(\lambda^2-\lambda+1)^2}{\lambda^2(\lambda-1)^2}$ where λ is the cross-ratio of $p_1, \ldots, p_4$.*
3. *The* j-invariant *is invariant under Möbius transformations as well as reordering of the points $p_1, \ldots, p_4$.*

Definition 7.6. *Let $P_5 = \{S \subseteq \mathbb{P}^1(\mathbb{C})\,;\, |S| = 5\}$. Define $I_5 : P_5 \to \mathbb{C}$ as*

$$I_5(S) = \sum_{\substack{T \subseteq S \\ |T|=4}} j(T).$$

Remark 7.7. *I_5 is a* five point invariant*. Another five point invariant is*

$$\tilde{I}_5(S) = \prod_{\substack{T \subseteq S \\ |T|=4}} j(T).$$

(actually $\tilde{I}_5$ is a cube of a five point invariant)

Remark 7.8.

1. *I_5 and $\tilde{I}_5$ are algebraically independent.*
2. *We use I_5 for Belyi maps, and both I_5 and $\tilde{I}_5$ for Belyi-1 maps. We do not use these invariants for Belyi-2 maps.*

Algorithm and details to compute I_5 and $\tilde{I}_5$ can be found in `www.math.fsu.edu/~vkunwar/FiveSings/FivePointInvariants/`. For a chosen $H^{a,b}_{c,x}$, each f in the table produces $H^{a,b}_{c,f}$ with five non removable singularities. Such f can only lead to a solution of a differential operator L_{inp} if $Sing(H^{a,b}_{c,f})$ matches $Sing(L_{inp})$ up to Möbius equivalence (our tables are complete up to Möbius equivalence). I_5 is a function on a set of five points which is invariant under Möbius transformation. It assigns a specific number to each set of five points. If there is a Möbius transformation between any two such sets, then they must have same I_5.

With each Belyi map f in the table, we attach the I_5 of non removable singularities of $H^{a,b}_{c,f}$ and the minimal polynomial of I_5. We compute the I_5 of the non removable singularities of L_{inp} and its minimal polynomial. We compare the minimal polynomial of I_5 from L_{inp} with the minimal polynomials attached to each Belyi map in the table. We discard those entries on the table whose minimal polynomials do not match the minimal polynomial from L_{inp}. This way, a large portion of the Belyi table is skipped. In case of Belyi-1 maps $f(x,s)$ the values of I_5 and $\tilde{I}_5$ are elements of $\mathbb{Q}(s)$. We compare I_5 and $\tilde{I}_5$ of Belyi-1 maps and $Sing(L_{inp})$. This gives two polynomial equations for s. We compute their gcd to find an equation for s. If the gcd is 1 then we can discard $f(x,s)$, otherwise we solve the gcd to find the value(s) of s. We do not use invariants for Belyi-2 maps because we have algorithms to compute such maps explicitly.

7.2 Exponent Differences

A necessary condition for f in the table to be a *candidate* is that the sorted lists of exponent differences (counted with multiplicity) in $Sing(L_{inp})$ and $Sing(H^{a,b}_{c,f})$ match mod $\mathbb{Z}$. This property is used to discard non-candidate Belyi-1 maps instantly before comparing the five point invariants. We attach the list of exponent differences of $H^{a,b}_{c,f}$ to each Belyi-1 map $f(x,s)$. We consider only those Belyi-1 maps whose list of exponent differences matches with the list from L_{inp} mod $\mathbb{Z}$.

7.3 Decompositions

Our group theoretic computations show that many f's in our tables are decomposable (see Figure 1). Solutions in terms of decompositions (if they exist) involve smaller degree pullbacks f. Such solutions are smaller and more preferable. For instance, if a map f of degree 12 from the table of $H^{\frac{1}{12},\frac{5}{12}}_{1,x}$ has a decomposition: $f = g(h)$ where $g = -4x(x-1)$ is a degree 2 pull-back which produces the exponent differences $(0,0,\frac{1}{3})$ from $(0,\frac{1}{2},\frac{1}{3})$ and h is a degree 6 rational function, then a differential operator which is solvable in terms of ${}_2F_1(\frac{1}{12},\frac{5}{12};1\,|\,f)$ is also solvable in terms of ${}_2F_1(\frac{1}{3},\frac{2}{3};1\,|\,h)$ $\quad((e_0,e_1,e_\infty)=(0,0,\frac{1}{3}) \Leftrightarrow (a,b,c)=(\frac{1}{3},\frac{2}{3},1))$. The later solution is smaller and more preferable (*see the example in Section 8.1 for details*). Our algorithms use all necessary[3] pull-backs in Figure 1 and find the decompositions (if any).

8 Main Algorithm

Once we have complete tables for all cases; $H^{\frac{1}{12},\frac{5}{12}}_{1,x}$, $H^{\frac{1}{8},\frac{3}{8}}_{1,x}$ and $H^{\frac{1}{6},\frac{1}{3}}_{1,x}$, the final task is to build the solver program. Let $C \subseteq \mathbb{C}$ be the base field, i.e. the field of constants of input differential operator L_{inp}. We give the algorithms to solve L_{inp} in terms of ${}_2F_1$-hypergeometric functions with the choice $(e_0,e_1,e_\infty) \in \{(0,\frac{1}{2},\frac{1}{k}), k \in \{3,4,6\}\}$. The algorithms not only find solutions in terms of ${}_2F_1(a,b;c\,|\,f)$ but also compute a decomposition $f = g(h)$ if that exists and leads to a smaller solution in terms of ${}_2F_1(\tilde{a},\tilde{b};\tilde{c}\,|\,h)$ (*see Figure 1 and Example 8.1 for more details*). The following algorithm computes candidate Belyi and near Belyi maps:

Algorithm 8.1: ComputeCandidates_02k

Compute candidate Belyi and near Belyi maps f such that $Sing\left(H^{\frac{k-2}{4k},\frac{k+2}{4k}}_{1,f}\right) = Sing(L_{inp})$, where $k \in \{3,4,6\}$

Note: This program uses the tables `Belyi_k20` and `Belyi_one_k20` which are the tables for Belyi and Belyi-1 maps for $(e_0,e_1,e_\infty) = (\frac{1}{k},\frac{1}{2},0)$, $\;k \in \{3,4,6\}$. These tables use $(\frac{1}{k},\frac{1}{2},0)$. But we use $(0,\frac{1}{2},\frac{1}{k})$, so the maps f from these tables are replaced by $\frac{1}{f}$. When $k=3$, this program also uses algorithms (see **Algorithm 6.1**) to compute Belyi-2 maps.

Input: A second order linear differential operator $L_{inp} \in C(x)[\partial]$, variable x, Tables of Belyi and Belyi-1 maps, exponent differences $(0,\frac{1}{2},\frac{1}{k})$ and the base field $C \subseteq \mathbb{C}$

(For example, if $k=3$ then the tables in the input are `Belyi_320` and `Belyi_one_320`)

Output: $\{[f,(0,\frac{1}{2},\frac{1}{k})]\,|\,f$ is a Belyi or near Belyi map s.t. $Sing\left(H^{\frac{k-2}{4k},\frac{k+2}{4k}}_{1,f}\right) = Sing(L_{inp})\}$

[3] Not all pullbacks in Figure 1 are necessary. For example, degree 4 pullback from $(0,\frac{1}{2},\frac{1}{3})$ to $(0,0,\frac{1}{3})$ is not needed. We use degree 2 pullback which produces exponent differences $(0,0,\frac{1}{3})$ from $(0,\frac{1}{2},\frac{1}{6})$ to cover that case.

Step 1: Compute the singularity structure of L_{inp}, i.e. $Sing(L_{inp})$. If L_{inp} does not have 5 non removable regular singularities or none of the exponent differences is zero mod $\mathbb{Z}$ then stop (L_{inp} must have at least one logarithmic singularity).
Step 2: Compute five point invariants of $Sing(L_{inp})$, denote them as $I_5(L_{inp})$ and $\tilde{I}_5(L_{inp})$. Let $MinPolyI_5(L_{inp})$ be the minimal polynomial of $I_5(L_{inp})$ over $\mathbb{Q}$. Let E be the list of exponent differences (counted with multiplicity) of $Sing(L_{inp})$.
Step 3: *Now we compute candidate Belyi and Belyi-1 maps:* Let $Candidates := \{\}$.
Step 3.1: *Compute candidate Belyi maps:* For each entry $i = [F, a, g]$ in `Belyi_k20` (where F is a Belyi map, a is its I_5 and g is the minimal polynomial of a) check if $g = MinPolyI_5(L_{inp})$. If they are equal then $Candidates := Candidates \bigcup \{F\}$.
Step 3.2: *Compute candidate Belyi-1 maps:* For each entry $[f_1(x,s), e]$ in the table `Belyi_one_k20` (where $f_1(x,s)$ is a family of Belyi-1 maps and e is the list of exponent differences (counted with multiplicity)) check if $e \equiv E \bmod \mathbb{Z}$. If they *match* then compute the singularity structure that $f_1(x,s)$ produces from $(e_0, e_1, e_\infty) = (\frac{1}{k}, \frac{1}{2}, 0)$ and its five point invariants (these are functions in s). Equate I_5 and $\tilde{I}_5$ of L_{inp} and f_1. This produces two equations in $C[s]$. Take their gcd and solve for s. For each s (if any), let $F_1 = f_1$ evaluated at such s. Then $Candidates := Candidates \bigcup \{F_1\}$.
Step 4: *Compute final candidates, i.e. f such that $Sing\left(H_{1,f}^{\frac{k-2}{4k},\frac{k+2}{4k}}\right) = Sing(L_{inp})$:*
Let $FinalCandidates := \{\}$. This loop runs through all entries in $Candidates$.
For each map $\tilde{f}$ in $Candidates$ compute the singularity structure which the pullback $\tilde{f}$ produces from $(e_0, e_1, e_\infty) = (\frac{1}{k}, \frac{1}{2}, 0)$. Then compute Möbius transformations from these singularities to the singularities of L_{inp}. For each Möbius transformation m, $FinalCandidates := FinalCandidates \bigcup \{[\frac{1}{\tilde{f}(m)}, (0, \frac{1}{2}, \frac{1}{k})]\}$.
Step 5: *Compute Belyi-2 maps*: If $k = 3$ then run algorithm **Algorithm 6.1** and another algorithm with the input C, x and $Sing(L_{inp})$ in terms of places(C). For each Belyi-2 map f_2 in the output, append $[f_2, (0, \frac{1}{2}, \frac{1}{3})]$ in *FinalCandidates*.
Step 6: Return *FinalCandidates*.

For $k \in \{3, 4, 6\}$, the following algorithm solves a second order linear differential operator L_{inp} with 5 regular singularities in terms of ${}_2F_1(\frac{k-2}{4k}, \frac{k+2}{4k}; 1 \,|\, f)$ or a decomposition, where $f \in C(x) \setminus C$:

Algorithm 8.2: Solver5_02k
Input: A second order linear differential operator $L_{inp} \in C(x)[\partial]$, variable x, $k \in \{3, 4, 6\}$ and the base field $C \subseteq \mathbb{C}$.
Output: $y = \exp\left(\int r\, dx\right) \cdot \left(r_0 S(f) + r_1 S(f)'\right) \neq 0$ such that $L_{inp}(y) = 0$, where $S(f) = {}_2F_1(\frac{k-2}{4k}, \frac{k+2}{4k}; 1 \,|\, f)$ or a decomposition, and $f \in C(x) \setminus C$.
Step 1: Run **Algorithm 8.1** with L_{inp}, x, the tables `Belyi_k20, Belyi_one_k20`, exponent differences $(0, \frac{1}{2}, \frac{1}{k})$ and the base field C as inputs. The output is *FinalCandidates*, i.e, the set of lists $[f, (0, \frac{1}{2}, \frac{1}{k})]$ such that $Sing\left(H_{1,f}^{\frac{k-2}{4k},\frac{k+2}{4k}}\right) = Sing(L_{inp})$.
Step 2: *Compute the decompositions of FinalCandidates: $RefinedCandidates := \{\}$*. This loop runs through the entries in $FinalCandidates$. For each element $[f, (0, \frac{1}{2}, \frac{1}{k})]$ in $FinalCandidates$ compute all possible decompositions of f (Figure 1 and Section 7.3). Include the outputs in *RefinedCandidates*.
Step 3: This loop runs through *RefinedCandidates*.
For each element $[F, (e_0, e_1, e_\infty)]$ in *RefinedCandidates* ((e_0, e_1, e_∞) must be the reciprocals of one of the triples in Figure 1), take the base GHDO $H_{c,x}^{a,b}$ with exponent differences (e_0, e_1, e_∞). For instance if $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3})$ then take:
$H_{1,x}^{\frac{1}{12},\frac{5}{12}} := x(1-x)\partial^2 + (c - (a+b+1)x)\partial - ab$ with $a = \frac{1}{12}, b = \frac{5}{12}$ and $c = 1$ (these correspond to $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3})$). Apply change of variables $x \mapsto F$ on $H_{c,x}^{a,b}$, which produces $H_{c,F}^{a,b}$ such that $Sing(H_{c,F}^{a,b}) = Sing(L_{inp})$.
Step 3.1: For each $H_{c,F}^{a,b}$ in **Step 3**, compute the projective equivalence [3] between $H_{c,F}^{a,b}$ and L_{inp}. The output could be zero (meaning they are not equivalent) in which case we take the next $H_{c,F}^{a,b}$, or we get

a non zero map G of the form:
$G = \exp(\int r dx)(r_0 + r_1\partial)$, where $r, r_0, r_1 \in C(x)$.
Step 3.2: $S(F) = {}_2F_1(a, b; c \,|\, F)$ is a solution of $H^{a,b}_{c,F}$. Apply the operator G obtained in **Step 3.1** to $S(F)$. That gives a solution of L_{inp}. Repeat this procedure for all $RefinedCandidates$ to obtain a list of solutions of L_{inp}.
Step 4: From the list of solutions of L_{inp}, choose the best solution with the shortest length.

Now we give the main algorithm:

Algorithm 8.3: Solver5
Solve a second order linear differential operator with five regular singularities in terms of ${}_2F_1(\frac{k-2}{4k}, \frac{k+2}{4k}; 1 \,|\, f)$ or a decomposition, where $f \in C(x)$ and $k \in \{3, 4, 6\}$.
Input: A second order linear differential operator $L_{inp} \in C(x)[\partial]$ with five regular singularities where at least one singularity is logarithmic, variable x, and the base field $C \subset \mathbb{C}$.
Output: $y = \exp\Big(\int r\, dx\Big) \cdot \Big(r_0 S(f) + r_1 S(f)'\Big) \neq 0$ such that $L_{inp}(y) = 0$, where $S(f) = {}_2F_1(\frac{k-2}{4k}, \frac{k+2}{4k}; 1 \,|\, f)$, $k \in \{3, 4, 6\}$ or a decomposition, and $f \in C(x) \setminus C$.
Let's first run **Algorithm 8.2** with $k = 6$. This case has the smallest degree bound:
Step 1: Call **Algorithm 8.2** with L_{inp}, x, $k = 6$ and C.
*If **Step 1** can't solve L_{inp} then we run **Algorithm 8.2** with $k = 4$:*
Step 2: Call **Algorithm 8.2** with L_{inp}, x, $k = 4$ and C.
*If **Step 2** can't solve L_{inp} then we finally run **Algorithm 8.2** with $k = 3$:*
Step 3: Call **Algorithm 8.2** with L_{inp}, x, $k = 3$ and C.

8.1 An Example

Consider the following differential operator:

$$L := \partial^2 + \frac{\left(8\,x^4 - x^2 + 2\,x - 3\right)}{x\,(x+1)\,(4\,x+3)\,(x^2 - 2\,x + 3)}\partial - \frac{4\,x^2}{\left(x^2 - 2\,x + 3\right)^2 (x+1)^2\,(4\,x+3)}$$

Following is the procedure to solve this operator using our algorithm in Maple:
Step 1: Read the program $Solver5$ from `http://www.math.fsu.edu/~vkunwar/FiveSings/`.
Step 2: L has the following singularity structure:

```
> Sing(L);
```

$$\left\{[x, 4/3], [x+1, 0], [x+3/4, 1/3], [x^2 - 2\,x + 3, 0]\right\}$$

L has five regular singularities (exponent differences are constant) and three of them are logarithmic (exponent differences are 0). So L is a differential operator we want to solve. It is easy to see that L can't be solved with the choice $k = 4$.
Let's compute five point invariants I_5 and $\tilde{I}_5$ of L, and minimal polynomial of $I_5(L)$:

```
> I5(L);
```

$$-259058528/59049$$

```
> I5tilde(L);
```

$$-11874715/472392$$

```
> MinPoly_I5(L);
```

$$x + 259058528/59049$$

E is the sorted list of exponent differences of L:

```
> E;
```

$$[0, 0, 0, 1/3, 4/3]$$

Step 3: First we try to solve L with the choice $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{6})$, i.e; using **Solver5_02k** with $k = 6$;

```
>  Solver5_02k(L, x, 6, {});
```

$$\{\ \}$$

Solver5_02k with $k = 6$ does not solve L. It finds some *RefinedCandidates*, but fails at projective equivalence.

Step 4: Now we try to solve L with the choice $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{4})$, i.e; using **Solver5_02k** with $k = 4$;

```
>  Solver5_02k(L, x, 4, {});
```

$$\{\ \}$$

Solver5_02k with $k = 4$ does not solve L. It does not find any *Candidates*.

Step 5: We finally try to solve L with the choice $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3})$, i.e; using **Solver5_02k** with $k = 3$;

```
>  Solver5_02k(L, x, 3, {});
```

$$\left\{ \frac{(x+1)^{1/3}(x^2-2\,x+3)^{1/6}}{x^{2/3}}\,{}_2F_1\Big(1/6, 1/2;\ 1 \,|\, \frac{x^4+4x+3}{x^4}\Big) \right\}$$

The details of this procedure are the following:

Step 5.1: *Run **Algorithm 8.1** with* $L, x, \texttt{Belyi\_320}, \texttt{Belyi\_one\_320}, (\frac{1}{3}, \frac{1}{2}, 0)$ *and* $\{\}$:

1. The program first searches the entries on the table `Belyi_320` to find Belyi maps whose minimal polynomial of five point invariant I_5 matches with that of L. Here are such Belyi maps:
$$F_1 = \left\{ \frac{4\,(4\,x-3)\big(x^2+2\,x+3\big)(x-1)^2}{x^8}, \frac{4\,(4\,x+3)x^4}{(x+1)^4(x^2-2\,x+3)^2}, \frac{128\,(2\,x-3)\big(x^4-36\,x+54\big)^3}{(x-2)^2(x^2+4\,x+12)x^{12}}, \frac{128\,(2\,x+3)\big(x^4-4\,x-6\big)^3}{(x+2)^6(x^2-4\,x+12)^3x^4} \right\}$$

2. The program then searches the table `Belyi_one_320` for those Belyi-1 maps whose sorted list of exponent differences match with E. It compares five point invariants I_5 and $\tilde{I}_5$ of matching entries to obtain two polynomials and solves their gcd for 's'(parameter of Belyi-1 families). The procedure finds the following map:
$$F_2 = \left\{ \frac{-(x-1)^4(x+1)^3(x-7)}{16\,(3\,x^2+2\,x+1)^2} \right\}$$
Note that this is also a Belyi map, we can check that from its branching above $0, 1, \infty$. Candidate Belyi-1 map $f(x, s)$ from the table reduced to this Belyi map because the fourth branch point t happened to be in $\{0, 1, \infty\}$ for this particular value of s.

3. Let $F := F_1 \bigcup F_2$. For each map g in F, we compute Möbius transformations from the singularities of $H_{1,g}^{\frac{1}{12},\frac{5}{12}}$ to $Sing(L)$. We compose g with these Möbius transformations. Reciprocals of the results (we use $(0, \frac{1}{2}, \frac{1}{k})$) give the following maps:
$$Fs = \left\{ \frac{(x+1)^4\big(x^2-2\,x+3\big)^2}{4\,(4\,x+3)x^4}, \frac{-x^8}{4\,(x^2-2\,x+3)(4\,x+3)(x+1)^2}, \frac{64(x+1)^6\big(x^2-2\,x+3\big)^3x^4}{(4\,x+3)(8\,x^4-4\,x-3)^3}, \frac{-64(x+1)^2\big(x^2-2\,x+3\big)x^{12}}{(4\,x+3)(8\,x^4+36\,x+27)^3} \right\}$$
(Two maps in F are Möbius equivalent)

4. The program calls the algorithms to find Belyi-2 maps. There are no such maps.

Hence, **Algorithm 8.1** returns the following:

$$FinalCandidates := \{[\frac{(x+1)^4\big(x^2-2\,x+3\big)^2}{4\,(4\,x+3)x^4}, (0, \tfrac{1}{2}, \tfrac{1}{3})], [\frac{-x^8}{4\,(x^2-2\,x+3)(4\,x+3)(x+1)^2}, (0, \tfrac{1}{2}, \tfrac{1}{3})],$$
$$[\frac{64(x+1)^6\big(x^2-2\,x+3\big)^3x^4}{(4\,x+3)(8\,x^4-4\,x-3)^3}, (0, \tfrac{1}{2}, \tfrac{1}{3})], [\frac{-64(x+1)^2\big(x^2-2\,x+3\big)x^{12}}{(4\,x+3)(8\,x^4+36\,x+27)^3}, (0, \tfrac{1}{2}, \tfrac{1}{3})]\}$$

Step 5.2: *Run **Algorithm 8.2** with* $L, x, 3$ *and* C:

We compute decompositions of *FinalCandidates*. For $(0, \frac{1}{2}, \frac{1}{3})$ it is enough to consider the only decomposition $f = g(h)$ where $g = \frac{x^2}{4\,(x-1)}$ produces exponent differences $(0, \frac{1}{3}, \frac{1}{3})$ from $(0, \frac{1}{2}, \frac{1}{3})$. $RefinedCandidates := \{\}$.

The first entry $i = \frac{(x+1)^4\big(x^2-2\,x+3\big)^2}{4\,(4\,x+3)x^4}$ has the decomposition $i = g(h)$ where $h \in \{\frac{x^4+4x+3}{4x+3}, \frac{x^4+4x+3}{x^4}\}$.

Second entry $i = \frac{-x^8}{4\,(x^2-2\,x+3)(4\,x+3)(x+1)^2}$ has the decomposition $i = g(h)$ with $h \in \{\frac{x^4}{x^4+4x+3}, -\frac{x^4}{4x+3}\}$.
The other two maps don't have any decompositions. This procedure gives the following *RefinedCandidates*:
$RefinedCandidates := \{[\frac{(x^4+4\,x+3)}{x^4}, (0, \frac{1}{3}, \frac{1}{3})], [\frac{(x^4+4\,x+3)}{4\,x+3}, (0, \frac{1}{3}, \frac{1}{3})], [\frac{x^4}{x^4+4\,x+3}, (0, \frac{1}{3}, \frac{1}{3})],$
$[\frac{-x^4}{4x+3}, (0, \frac{1}{3}, \frac{1}{3})], [\frac{64(x+1)^6(x^2-2\,x+3)^3x^4}{(4\,x+3)(8\,x^4-4\,x-3)^3}, (0, \frac{1}{2}, \frac{1}{3})], [\frac{-64(x+1)^2(x^2-2\,x+3)x^{12}}{(4\,x+3)(8\,x^4+36\,x+27)^3}, (0, \frac{1}{2}, \frac{1}{3})]\}$
Step 5.2a: *Now we apply projective equivalence [3]:*
For the candidate $f = \frac{x^4+4x+3}{x^4}$ we take GHDO with $(e_0, e_1, e_\infty) = (0, \frac{1}{3}, \frac{1}{3})$ and apply change of variable $x \mapsto f$. That produces the following operator:
$L_1 := \partial^2 + \frac{(-x^2-15+8\,x^4-14\,x)}{x(x+1)(4\,x+3)(x^2-2\,x+3)}\partial + \frac{12}{(x^2-2\,x+3)(4\,x+3)x^2}$

```
>  equiv(L1, L);
```

$$\frac{(x+1)^{1/3}\,(x^2-2\,x+3)^{1/6}}{x^{2/3}}$$

${}_2F_1\left(\frac{1}{6}, \frac{1}{2}; 1 \mid \frac{x^4+4x+3}{x^4}\right)$ is a solution of L_1. Hence
$\frac{(x+1)^{1/3}\,(x^2-2\,x+3)^{1/6}}{x^{2/3}} \cdot {}_2F_1\left(\frac{1}{6}, \frac{1}{2}; 1 \mid \frac{x^4+4x+3}{x^4}\right)$ is a solution of L.
Repeating the procedure with candidate $f = \frac{x^4+4x+3}{4\,x+3}$ and $(e_0, e_1, e_\infty) = (0, \frac{1}{3}, \frac{1}{3})$ produces the following operator:
$L_2 := \partial^2 + \frac{(12\,x^4-x^2+2\,x-3)}{x(x+1)(4\,x+3)(x^2-2\,x+3)}\partial + \frac{12\,x^2}{(x^2-2\,x+3)(4\,x+3)^2}$

```
>  equiv(L2, L);
```

$$(x+1)^{1/3}\left(\frac{x^2-2x+3}{4x+3}\right)^{1/6}$$

${}_2F_1\left(\frac{1}{6}, \frac{1}{2}; 1 \mid \frac{x^4+4x+3}{4x+3}\right)$ is a solution of L_2. Hence
$(x+1)^{1/3}\left(\frac{x^2-2x+3}{4x+3}\right)^{1/6} \cdot {}_2F_1\left(\frac{1}{6}, \frac{1}{2}; 1 \mid \frac{x^4+4x+3}{4x+3}\right)$ is a solution of L.
Repeating the procedure with candidate $f = \frac{x^4}{x^4+4x+3}$ and $(e_0, e_1, e_\infty) = (0, \frac{1}{3}, \frac{1}{3})$ produces the following operator:
$L_3 := \partial^2 + \frac{(8\,x^4-x^2+18\,x+9)}{x(x+1)(4\,x+3)(x^2-2\,x+3)}\partial - \frac{12\,x^2}{(x^2-2\,x+3)^2(x+1)^2(4\,x+3)}$

```
>  equiv(L3, L);
```

$$0$$

This choice does not solve L.
Other candidates do not solve L; they stop at projective equivalence, returning 0.
Step 7: Of these two solutions
$\left\{\frac{(x+1)^{\frac{1}{3}}\,(x^2-2\,x+3)^{\frac{1}{6}}}{x^{\frac{2}{3}}}\;{}_2F_1\left(\frac{1}{6}, \frac{1}{2}; 1 \mid \frac{x^4+4x+3}{x^4}\right), (x+1)^{\frac{1}{3}}\left(\frac{x^2-2x+3}{4x+3}\right)^{\frac{1}{6}}\;{}_2F_1\left(\frac{1}{6}, \frac{1}{2}; 1 \mid \frac{x^4+4x+3}{4x+3}\right)\right\}$, our program returns the following (best) solution:

```
>  Solver5(L, x, {});
```

$$\left\{\frac{(x+1)^{1/3}(x^2-2\,x+3)^{1/6}}{x^{2/3}} \cdot {}_2F_1\left(1/6, 1/2;\, 1 \mid \frac{x^4+4x+3}{x^4}\right)\right\}$$

which is the solution obtained in Step 5.

References

[1] R. Debeerst, M. van Hoeij, W. Koepf: *Solving Differential Equations in Terms of Bessel Functions*, ISSAC'08 Proceedings,39-46, (2008).

[2] R. S. Maier, R. Vidunas, G. Filipuk: *A Classification of Covering yielding Heun to Hypergeometric Reductions* (paper in progress).

[3] M. van Hoeij: *An implementation for finding equivalence map*, www.math.fsu.edu/∼hoeij/files/equiv.

[4] M. van Hoeij: *A database of dessins with* $(\frac{1}{k}, \frac{1}{2}, 0)$*-singularity-count* $\leq$ *6, where* $k \in \{3, 4, 6\}$, http://www.math.fsu.edu/∼hoeij/files/dessins/.

[5] M. van Hoeij, R. Vidunas: *Belyi functions for hyperbolic hypergeometric-to-Heun transformations*, arXiv:1212.3803.

[6] M. Assis, S. Boukraa, S. Hassani, M. van Hoeij, J-M. Maillard, B. M. McCoy: *Diagonal Ising susceptibility: elliptic integrals, modular forms and Calabi-Yau equations.* http://arxiv.org/abs/1110.1705, 2011.

[7] F. Beukers, H. Montanus: *Explicit calculation of elliptic fibrations of K3-surfaces and their Belyi-maps*, Cambridge Univ. Press, Cambridge, 2008.

[8] V. J. Kunwar, M. van Hoeij: *Second Order Differential Equations with Hypergeometric Solutions of Degree Three*, ISSAC 2013 Proceedings, 235-242, 2013.
Implementation: www.math.fsu.edu/∼vkunwar/hypergeomdeg3/.

[9] V. J. Kunwar: *Hypergeometric Solutions of Linear Differential Equations with Rational Function Coefficients*, Ph.D thesis and implementation, available at www.math.fsu.edu/∼vkunwar (2014).

[10] Q. Yuan, M. van Hoeij: *Finding all Bessel type solutions for Linear Differential Equations with Rational Function Coefficients*, ISSAC'2010 Proceedings, 37-44, (2010).

[11] T. Fang, M. van Hoeij: *2-Descent for Second Order Linear Differential Equations*, ISSAC'2011 Proceedings, 107-114 (2011).

[12] A. Bostan, F. Chyzak, M. van Hoeij, L. Pech: *Explicit formula for the generating series of diagonal 3D rook paths*, Seminaire Lotharingien de Combinatoire, B66a (2011).

[13] T. Cluzeau, M. van Hoeij: *A Modular Algorithm to Compute the Exponential Solutions of a Linear Differential Operator*, J. Symb. Comput., 38, 1043-1076 (2004).

[14] S. K. Lando, A. K. Zvonkin: *Graphs on Surfaces and Their Applications*, ISBN 3-540-00203-0 Springer-Verlag Berlin Heidelberg 2004

[15] A. Bostan and M. Kauers: *Automatic Classification of Restricted Lattice Walks*, Proceedings of FPSAC'09, pp. 201–215

[16] K. Takeuchi: *Commensurability classes of arithmetic triangle groups*, J. Fac. Sci, Univ. Tokyo, Sect.1 A, pp 201–272, 1977.

[17] M. van Hoeij: *Solving Third Order Linear Differential Equation in Terms of Second Order Equations*, ISSAC'07 Proceedings, 355-360, 2007. Implementation: www.math.fsu.edu/∼hoeij/files/ReduceOrder.

[18] M. van Hoeij: *Linear Differential Equations with a Convergent Integer Series Solution*, Project Description for NSF grant 1319547, 2013.

[19] A. Bostan, S. Boukraa, S. Hassani, J.-M. Maillard, J.-A. Weil, N. Zenine: *Globally nilpotent differential operators and the square Ising model*, arXiv:0812.4931.

[20] Z. X. Wang, D. R. Guo: *Special Functions* World Scientific Publishing Co. Pte. Ltd, Singapore, 1989.

[21] R. Vidunas: *Algebraic transformations of Gauss hypergeometric functions.* Funkcialaj Ekvacioj, Vol 52 (Aug 2009), 139-180.

[22] Q. Yuan: *Finding all Bessel type solutions for Linear Differential Equations with Rational Function Coefficients*, Ph.D thesis and implementation, available at www.math.fsu.edu/∼qyuan (2012).

[23] T. Fang: *Solving Linear Differential Equations in Terms of Hypergeometric Functions by 2-Descent*, Ph.D thesis and implementation, available at www.math.fsu.edu/∼tfang (2012).

[24] J. Kovacic: *An algorithm for solving second order linear homogeneous equations*, J. Symbolic Computations, 2, 3-43 (1986).

[25] oeis.org: *The Online Encyclopedia of Integer Sequences.*