Summary of the PhD Defense Presentation

Vijay Jung Kunwar

Notation:

\[ \partial \quad \frac{d}{dx} \]

\[ L_{inp} \] Input differential operator; a second order linear differential operator

\[ H_{c,x}^{a,b} \] Gauss hypergeometric differential operator

\[ S(x) = 2F_1(a, b; c | x) \] Gauss hypergeometric function; a solution of \( H_{c,x}^{a,b} \)

\( (e_0, e_1, e_\infty) \) Exponent differences of \( H_{c,x}^{a,b} \) at \((0, 1, \infty)\)

\[ \frac{f}{c} \] Change of variables

\[ \frac{r_0, r_1}{c} \] Gauge transformation

\[ \rightarrow \] Exponential product

\[ [g_0, g_1, \ldots, g_k] \] \( k \)-constellation; \( g_i \in S_n \) (details in the slides)

Given:

A second order linear differential operator \( L_{inp} \) with rational function coefficients (see slides for motivation);

1. irreducible and has no Liouvillian solutions,
2. has five regular singularities where at least one of the singularities is logarithmic, and
3. has arithmetic monodromy group.

Goal:

Solve \( L_{inp} \) in terms of \( 2F_1(a, b; c | f) \), i.e, find a solution \( y \) of the following form (also called the closed form solution):

\[ y = \exp\left( \int dx \left( r_0 S(f) + r_1 S(f)^{'} \right) \right) \neq 0 \]

where \( S(x) = 2F_1(a, b; c | x) \), \( r, r_0, r_1, f \in \mathbb{C}(x) \).

Problem Discussion:

For the solver program to be complete we need the complete tables of Belyi, Belyi-1 and Belyi-2 maps. We use the following correspondence to prove the completeness:

Belyi maps \( \leftrightarrow \) dessins, i.e, equivalence classes of 3-constellations \([g_0, g_1, g_\infty]\) mod conjugation

Belyi-1 maps \( \leftrightarrow \) near dessins, i.e, equivalence classes of 4-constellations \([g_0, g_1, g_t, g_\infty]\) mod conjugation

Belyi-2 maps \( \leftrightarrow \) we have algorithms to compute such maps

- We compute Belyi, Belyi-1 maps solving polynomial equations and using other techniques.
- We compute dessins, near dessins using combinatorial search plus various techniques to prevent computational explosion.

Proof: Compare!
> $f := -(x^2 - 3) / (3 * x^2 - 1) * x^4$;

\[
f := \frac{(x^2 - 3) \cdot x^4}{3 \cdot x^2 - 1}
\]  
\hspace{1cm} (1)

> factor(1 - f);

\[
\frac{(x - 1)^3 \cdot (x + 1)^3}{3 \cdot x^2 - 1}
\]  
\hspace{1cm} (2)

> read PlotDessin: plots[pointplot](pts); # plots $f^{-1}( [0, 1])$.

\[
> \text{algcurves[monodromy]}(\text{numer}(f - y), y, x)[-1];
\]

\[
[[0, [[2, 3, 5, 4]]], [1, [[1, 3, 2], [4, 5, 6]]], [\infty, [[1, 4, 6, 3]]]]
\]  
\hspace{1cm} (3)

0: cycles of length 4, 1, 1
1: cycles of length 3, 3
inf: cycles of length 4, 1, 1 (1-cycles are not printed)
Dessin:

```
[\begin{tikzpicture}
    \begin{scope}
        \draw (0,0) circle (1cm);
        \filldraw (0,0) circle (0.1cm);
        \draw (-1,0) -- (1,0);
        \filldraw (-1,0) circle (0.1cm);
        \filldraw (1,0) circle (0.1cm);
    \end{scope}
\end{tikzpicture}
```

“Labelled dessin”, i.e, 3-constellation:

```
[\begin{tikzpicture}
    \begin{scope}
        \draw (0,0) circle (1cm);
        \filldraw (0,0) circle (0.1cm);
        \draw (-1,0) -- (1,0);
        \filldraw (-1,0) circle (0.1cm);
        \filldraw (1,0) circle (0.1cm);
        \node at (0,1) {1};
        \node at (0,-1) {2};
        \node at (1,1) {3};
        \node at (-1,1) {4};
        \node at (1,-1) {5};
        \node at (-1,-1) {6};
    \end{scope}
\end{tikzpicture}
```

Read off permutations:

- black vertices: \( x = -\sqrt{3}, x = 0, x = \sqrt{3} \) \( g_0 = (1) (2 4 6 3) (5) \)
- white vertices: \( x = -1, x = 1 \) \( g_1 = (1 2 3) (4 5 6) \)
- faces: \( x = \frac{1}{\sqrt{3}}, x = \frac{1}{\sqrt{3}}, x = \infty \) \( g_{\infty} = (2) (6) (1 3 5 4) \)
  (follow outside face, \( x = \infty \), clockwise)

Note:

Since the labelling was done arbitrarily, we should expect the result from “algcurves[monodromy]” to match only up to conjugation (same dessin, not the same 3-constellation). We have a program that can find the conjugation.