Hypergeometric Solutions of Linear Differential Equations with Rational Function Coefficients

Vijay Jung Kunwar

Department of Mathematics
Florida State University

June 11, 2014

Introduction

- We consider linear differential operators $L \in \mathbb{C}(x)[\partial]$ of order 2 such that:

Introduction

- We consider linear differential operators $L \in \mathbb{C}(x)[\partial]$ of order 2 such that:
  1. L is irreducible,
  2. L has no Liouvillian solutions
  3. L has only regular singularities, i.e, L is Fuchsian.
  4. $\partial = \frac{d}{dx}$.

Introduction

- We consider linear differential operators $L \in \mathbb{C}(x)[\partial]$ of order 2 such that:
  1. L is irreducible,
  2. L has no Liouvillian solutions
  3. L has only regular singularities, i.e, L is Fuchsian.
  4. $\partial = \frac{d}{dx}$.
- We want to find ${}_2F_1$-type solutions (if they exist), i.e, solutions of the form:
  $y = \exp(\int r\, dx) \left(r_0 S(f) + r_1 S(f)' \right) \neq 0$ such that $L(y) = 0$, where $S(x) = {}_2F_1(a, b; c \,|\, x)$, $r, r_0, r_1, f \in \mathbb{C}(x)$.

Introduction

- We consider linear differential operators $L \in \mathbb{C}(x)[\partial]$ of order 2 such that:
  1. L is irreducible,
  2. L has no Liouvillian solutions
  3. L has only regular singularities, i.e, L is Fuchsian.
  4. $\partial = \frac{d}{dx}$.
- We want to find ${}_2F_1$-type solutions (if they exist), i.e, solutions of the form:
  $y = \exp(\int r\,dx)\left(r_0 S(f) + r_1 S(f)'\right) \neq 0$ such that $L(y) = 0$,
  where $S(x) = {}_2F_1(a, b; c \,|\, x)$, $r, r_0, r_1, f \in \mathbb{C}(x)$.
- Why this format?
  Conjecture: if L has a convergent solution in $\mathbb{Z}[[x]]$ then it has a solution of this format.

An Example

- Consider the differential operator:

$$L = \partial^2 + \frac{\left(16\,x^3+16\,x^2+23\,x-5\right)}{3\,x(2\,x-1)(x^2+2\,x+5)}\,\partial - \frac{875\,x^2}{9\,(2\,x-1)^2(x^2+2\,x+5)}$$

An Example

- Consider the differential operator:

$$L = \partial^2 + \frac{\left(16\,x^3+16\,x^2+23\,x-5\right)}{3\,x(2\,x-1)(x^2+2\,x+5)}\,\partial - \frac{875\,x^2}{9\,(2\,x-1)^2(x^2+2\,x+5)}$$

- L has singularities at the roots of $x, 2x-1, x^2+2x+5$ and at ∞ (singularities of L come from roots of the leading coefficient or poles of other coefficients).

An Example

- Consider the differential operator:

$$L = \partial^2 + \frac{\left(16\,x^3+16\,x^2+23\,x-5\right)}{3\,x(2\,x-1)(x^2+2\,x+5)}\,\partial - \frac{875\,x^2}{9\,(2\,x-1)^2(x^2+2\,x+5)}$$

- L has singularities at the roots of $x, 2x-1, x^2+2x+5$ and at ∞ (singularities of L come from roots of the leading coefficient or poles of other coefficients).

- Our algorithm on 'five singularities' solves L:

$$Sol_1(L) = \frac{20\,(2x-1)^{\frac{1}{6}}}{9\,(x^2+2x+5)^{\frac{7}{6}}\,x^{\frac{14}{3}}} \cdot \left[(x^2+2\,x+5)x^4\ {}_2F_1\left(\frac{1}{6}, \frac{1}{3};\ 1 \mid \frac{4\,(2\,x-1)}{x^4\,(x^2+2\,x+5)}\right) - (x^3+x^2+2\,x-2)^2\ {}_2F_1\left(\frac{7}{6}, \frac{4}{3};\ 2 \mid \frac{4\,(2\,x-1)}{x^4\,(x^2+2\,x+5)}\right)\right]$$

Why Second Order?

- First order differential operators are easy to solve.

Why Second Order?

- First order differential operators are easy to solve.
- For higher order, the most natural way is to find if the differential operator can be reduced to lower order using factors, symmetric products, symmetric powers etc.

Why Second Order?

- First order differential operators are easy to solve.
- For higher order, the most natural way is to find if the differential operator can be reduced to lower order using factors, symmetric products, symmetric powers etc.
- Mark van Hoeij and Michael F. Singer have developed algorithms to solve higher order differential operators (up to order 4) using order reduction.

Why Second Order?

- First order differential operators are easy to solve.
- For higher order, the most natural way is to find if the differential operator can be reduced to lower order using factors, symmetric products, symmetric powers etc.
- Mark van Hoeij and Michael F. Singer have developed algorithms to solve higher order differential operators (up to order 4) using order reduction.
- Complete algorithms for second order differential operators are very useful to solve higher order differential operators.

Related Works

- J. Kovacic developed algorithm to find Liouvillian solutions.

Related Works

- J. Kovacic developed algorithm to find Liouvillian solutions.
- Q. Yuan, R. Debeerst, M. van Hoeij and W. Koepf developed algorithms to solve differential operators with irregular singularities.

Related Works

- J. Kovacic developed algorithm to find Liouvillian solutions.
- Q. Yuan, R. Debeerst, M. van Hoeij and W. Koepf developed algorithms to solve differential operators with irregular singularities.
- Fuchsian differential operators correspond to hypergeometric solutions.

Related Works

- J. Kovacic developed algorithm to find Liouvillian solutions.
- Q. Yuan, R. Debeerst, M. van Hoeij and W. Koepf developed algorithms to solve differential operators with irregular singularities.
- Fuchsian differential operators correspond to hypergeometric solutions.
- M. van Hoeij and R. Vidunas developed the tables of rational functions for 4 singularities (Heun equation).

Related Works

- J. Kovacic developed algorithm to find Liouvillian solutions.
- Q. Yuan, R. Debeerst, M. van Hoeij and W. Koepf developed algorithms to solve differential operators with irregular singularities.
- Fuchsian differential operators correspond to hypergeometric solutions.
- M. van Hoeij and R. Vidunas developed the tables of rational functions for 4 singularities (Heun equation).
- T. Fang and M. van Hoeij developed algorithm for 2-descent, which finds ${}_2F_1$-type solutions whenever f has degree 2, and also reduces a differential operator to another with fewer singularities.

Our Contribution

Let $L_{inp} \in \mathbb{C}(x)[\partial]$ be a second order linear differential operator with rational function coefficients. Let L_{inp} be irreducible and has no Liouvillian solutions.

Our Contribution

Let $L_{inp} \in \mathbb{C}(x)[\partial]$ be a second order linear differential operator with rational function coefficients. Let L_{inp} be irreducible and has no Liouvillian solutions.

We have developed algorithms to find ${}_2F_1$-type solutions of L_{inp} in the following cases:

Our Contribution

Let $L_{inp} \in \mathbb{C}(x)[\partial]$ be a second order linear differential operator with rational function coefficients. Let L_{inp} be irreducible and has no Liouvillian solutions.

We have developed algorithms to find ${}_2F_1$-type solutions of L_{inp} in the following cases:

1. L_{inp} has five regular singularities where at least one of them is *logarithmic*. This is the topic of today!

Our Contribution

Let $L_{inp} \in \mathbb{C}(x)[\partial]$ be a second order linear differential operator with rational function coefficients. Let L_{inp} be irreducible and has no Liouvillian solutions.

We have developed algorithms to find ${}_2F_1$-type solutions of L_{inp} in the following cases:

1. L_{inp} has five regular singularities where at least one of them is *logarithmic*. This is the topic of today!
2. L_{inp} has hypergeometric solution of degree three, i.e, L_{inp} is solvable in terms of ${}_2F_1(a, b; c \mid f)$ where f is a rational function of degree three.

Formal Solutions, Example

- $L(y) = 144x(x-1)y'' + (216x-72)y' + 5y = 0$

Formal Solutions, Example

- $L(y) = 144x(x-1)y'' + (216x-72)y' + 5y = 0$
- Formal solutions at $x = 0$ (dots = higher powers of x);
  $y_1 = x^0 + ...$
  $y_2 = x^{\frac{1}{2}} + ...$
  Only the dominant term is listed.

Formal Solutions, Example

- $L(y) = 144x(x-1)y'' + (216x-72)y' + 5y = 0$
- Formal solutions at $x = 0$ (dots = higher powers of x);
  $y_1 = x^0 + ...$
  $y_2 = x^{\frac{1}{2}} + ...$
  Only the dominant term is listed.
- Exponents at $x = 0$ are: 0 and $\frac{1}{2}$.

Formal Solutions, Example

- $L(y) = 144x(x-1)y'' + (216x-72)y' + 5y = 0$
- Formal solutions at $x=0$ (dots = higher powers of x);
  $y_1 = x^0 + ...$
  $y_2 = x^{\frac{1}{2}} + ...$
  Only the dominant term is listed.
- Exponents at $x=0$ are: 0 and $\frac{1}{2}$.
- Formal solutions at $x=1$ (dots = higher powers of $x-1$);
  $y_1 = (x-1)^0 + ...$
  $y_2 = \log(x-1)y_1 + ...$

Formal Solutions, Example

- $L(y) = 144x(x-1)y'' + (216x - 72)y' + 5y = 0$
- Formal solutions at $x = 0$ (dots = higher powers of x);
  $y_1 = x^0 + ...$
  $y_2 = x^{\frac{1}{2}} + ...$
  Only the dominant term is listed.
- Exponents at $x = 0$ are: 0 and $\frac{1}{2}$.
- Formal solutions at $x = 1$ (dots = higher powers of $x - 1$);
  $y_1 = (x-1)^0 + ...$
  $y_2 = \log(x-1)y_1 + ...$
- Exponents at $x = 1$ are: 0, 0. The point $x = 1$ is a logarithmic singularity.

Formal Solutions, Example

- $L(y) = 144x(x-1)y'' + (216x-72)y' + 5y = 0$
- Formal solutions at $x=0$ (dots = higher powers of x);
  $y_1 = x^0 + ...$
  $y_2 = x^{\frac{1}{2}} + ...$
  Only the dominant term is listed.
- Exponents at $x=0$ are: 0 and $\frac{1}{2}$.
- Formal solutions at $x=1$ (dots = higher powers of $x-1$);
  $y_1 = (x-1)^0 + ...$
  $y_2 = \log(x-1)y_1 + ...$
- Exponents at $x=1$ are: 0, 0. The point $x=1$ is a logarithmic singularity.
- Regular points have exponents 0, 1.

Formal Solutions, Example

- $L(y) = 144x(x-1)y'' + (216x-72)y' + 5y = 0$
- Formal solutions at $x = 0$ (dots = higher powers of x);
  $y_1 = x^0 + ...$
  $y_2 = x^{\frac{1}{2}} + ...$
  Only the dominant term is listed.
- Exponents at $x = 0$ are: 0 and $\frac{1}{2}$.
- Formal solutions at $x = 1$ (dots = higher powers of $x - 1$);
  $y_1 = (x-1)^0 + ...$
  $y_2 = \log(x-1)y_1 + ...$
- Exponents at $x = 1$ are: 0, 0. The point $x = 1$ is a logarithmic singularity.
- Regular points have exponents 0, 1.
- A change of variables $x \mapsto x^2$ turns $x = 0$ into a regular point. It turns $x = 1$ into two logarithmic singularities $x = \pm 1$.

Gauss Hypergeometric Differential Operator

- Gauss hypergeometric differential operator has the following form;

  $H_{c,x}^{a,b} = x(1-x)\partial^2 + (c-(a+b+1)x)\partial - ab$

Gauss Hypergeometric Differential Operator

- Gauss hypergeometric differential operator has the following form;

  $H_{c,x}^{a,b} = x(1-x)\partial^2 + (c-(a+b+1)x)\partial - ab$

- $H_{c,x}^{a,b}$ has 3 regular singularities at $0, 1, \infty$ with exponent differences $(e_0, e_1, e_\infty) = (1-c, c-a-b, b-a)$ up to sign.

Gauss Hypergeometric Differential Operator

- Gauss hypergeometric differential operator has the following form;

  $$H_{c,x}^{a,b} = x(1-x)\partial^2 + (c-(a+b+1)x)\partial - ab$$

- $H_{c,x}^{a,b}$ has 3 regular singularities at $0, 1, \infty$ with exponent differences $(e_0, e_1, e_\infty) = (1-c, c-a-b, b-a)$ up to sign.
- The Gauss hypergeometric function ${}_2F_1(a, b; c \,|\, x)$ is a solution of $H_{c,x}^{a,b}$ where:

  $${}_2F_1(a, b; c \,|\, x) = \sum_{n=0}^{\infty} \frac{(a)_n (b)_n}{(c)_n} \cdot \frac{x^n}{n!}$$

Gauss Hypergeometric Differential Operator

- Gauss hypergeometric differential operator has the following form;

  $$H_{c,x}^{a,b} = x(1-x)\partial^2 + (c - (a+b+1)x)\partial - ab$$

- $H_{c,x}^{a,b}$ has 3 regular singularities at $0, 1, \infty$ with exponent differences $(e_0, e_1, e_\infty) = (1-c, c-a-b, b-a)$ up to sign.
- The Gauss hypergeometric function ${}_2F_1(a,b;c\,|\,x)$ is a solution of $H_{c,x}^{a,b}$ where:

  $${}_2F_1(a,b;c\,|\,x) = \sum_{n=0}^{\infty} \frac{(a)_n(b)_n}{(c)_n} \cdot \frac{x^n}{n!}$$

- The Pochhammer symbol $(a)_n$ is defined as:

  $$(a)_n = \begin{cases} 1 & \text{if } n = 0 \\ a(a+1)\dots(a+n-1) & \text{otherwise} \end{cases}$$

Transformations

- We define the following transformations on a second order differential operator:
  - (i) Change of variables: $y(x) \mapsto y(f)$
  - (ii) Gauge transformation: $y \mapsto r_0 y + r_1 y'$
  - (iii) Exponential product: $y \mapsto \exp(\int r\, dx)\, y$

Transformations

- We define the following transformations on a second order differential operator:

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

  The function f in (i) above is called the *pullback* function.

Transformations

- We define the following transformations on a second order differential operator:
  - (i) Change of variables: $y(x) \mapsto y(f)$
  - (ii) Gauge transformation: $y \mapsto r_0 y + r_1 y'$
  - (iii) Exponential product: $y \mapsto \exp(\int r\, dx)\, y$

  The function f in (i) above is called the *pullback* function.

- These transformations are denoted $\xrightarrow{f}_C$, $\xrightarrow{r_0, r_1}_G$ and $\xrightarrow{r}_E$.

Transformations

- We define the following transformations on a second order differential operator:
  (i) Change of variables: $y(x) \mapsto y(f)$
  (ii) Gauge transformation: $y \mapsto r_0 y + r_1 y'$
  (iii) Exponential product: $y \mapsto \exp(\int r\, dx)\, y$

  The function f in (i) above is called the *pullback* function.

- These transformations are denoted $\xrightarrow{f}_C$, $\xrightarrow{r_0, r_1}_G$ and $\xrightarrow{r}_E$.

- $\xrightarrow{r_0, r_1}_G$ and $\xrightarrow{r}_E$ are equivalence relations. They do not affect the true singularities of a differential operator. $\xrightarrow{f}_C$ can change everything.

Effect of $\xrightarrow{f}_C$

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

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

p: singularity, Δ_p: exponent difference

Effect of $\xrightarrow{f}_C$

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

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

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

p: singularity, Δ_p: exponent difference

Effect of $\xrightarrow{f}_C$

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

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

↑

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

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

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

p: singularity, Δ_p: exponent difference

Computing ${}_2F_1$-type Solutions

Let L_{inp} be the input differential operator of order 2, and $S(x) = {}_2F_1(a, b; c \,|\, x)$.

Computing ${}_2F_1$-type Solutions

Let L_{inp} be the input differential operator of order 2, and $S(x) = {}_2F_1(a, b; c \mid x)$.

- If we find the transformations such that :

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

Computing ${}_2F_1$-type Solutions

Let L_{inp} be the input differential operator of order 2, and $S(x) = {}_2F_1(a, b; c \mid x)$.

- If we find the transformations such that :

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

then we get a solution of L_{inp} in the same fashion as:

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

Computing ${}_2F_1$-type Solutions

Let L_{inp} be the input differential operator of order 2, and $S(x) = {}_2F_1(a,b;c\,|\,x)$.

- If we find the transformations such that :

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

  then we get a solution of L_{inp} in the same fashion as:

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

- There are algorithms to compute $\xrightarrow{r_0,r_1}_G \xrightarrow{r}_E$. The crucial part is to compute f and a, b, c.

Motivation

- Differential equations with ${}_2F_1$-type solutions are very common in Combinatorics, Physics and Engineering.

Motivation

- Differential equations with ${}_2F_1$-type solutions are very common in Combinatorics, Physics and Engineering.
- To find 'closed form solutions' (solutions in terms of very well studied special functions; Airy, Bessel, Kummer, Whittaker, Liouvillian, Hypergeometric) we need a complete algorithm that treats the hypergeometric case.

Motivation

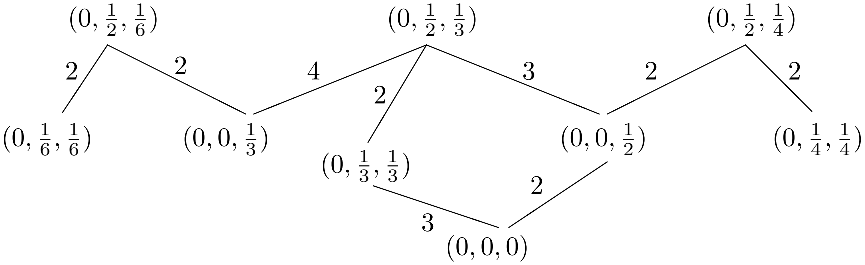
- Differential equations with ${}_2F_1$-type solutions are very common in Combinatorics, Physics and Engineering.
- To find 'closed form solutions' (solutions in terms of very well studied special functions; Airy, Bessel, Kummer, Whittaker, Liouvillian, Hypergeometric) we need a complete algorithm that treats the hypergeometric case.
- There are many integer sequences in `oeis.org` whose generating functions are convergent and holonomic. Such generating functions satisfy linear differential operators. Such differential operators of order 2 and 3 tested so far have logarithmic singularities and have ${}_2F_1$- type solutions.

Motivation Contd.

- Moreover, such differential operators lie in the same class (minimal network of differential operators in terms of solvability), namely, $Class\left(H_{1,x}^{\frac{1}{12},\frac{5}{12}}\right)$;
  $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3}) \Leftrightarrow (a, b, c) = (\frac{1}{12}, \frac{5}{12}, 1)$

Motivation Contd.

- Moreover, such differential operators lie in the same class (minimal network of differential operators in terms of solvability), namely, $Class\left(H_{1,x}^{\frac{1}{12},\frac{5}{12}}\right)$;
  $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{3}) \Leftrightarrow (a, b, c) = (\frac{1}{12}, \frac{5}{12}, 1)$
- K. Takeuchi classified commensurable classes of arithmetic triangle groups. The first class gives (e_0, e_1, e_∞) of Gauss hypergeometric differential operators that lie in $Class\left(H_{1,x}^{\frac{1}{12},\frac{5}{12}}\right)$:



Degree Bounds and Types of f

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

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

where e_p is the ramification order of f at p.

Degree Bounds and Types of f

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

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

where e_p is the ramification order of f at p.

- Riemann-Hurwitz's formula gives the following for our project:

1. Belyi maps: zero-dimensional families $f(x)$, ramify only above $\{0, 1, \infty\}$, degree bound 18.
2. Belyi-1 maps: one-dimensional families $f(x, t)$, ramify above one point outside $\{0, 1, \infty\}$, degree bound 12.
3. Belyi-2 maps: two-dimensional families $f(x, s, t)$, ramify above two points outside $\{0, 1, \infty\}$, degree bound 6.

Computing f

- We can compute Belyi and near Belyi (Belyi-1, Belyi-2) maps in Maple using polynomial equations and other techniques.

Computing f

- We can compute Belyi and near Belyi (Belyi-1, Belyi-2) maps in Maple using polynomial equations and other techniques.
- Smaller cases are easy to find. For larger cases we use Elimination, Resultants, Parametrization etc. There are no maps of degree 17 for our project. We use special techniques given by F. Beukers and H. Montanus to compute degree 18 Belyi maps.

Computing f

- We can compute Belyi and near Belyi (Belyi-1, Belyi-2) maps in Maple using polynomial equations and other techniques.
- Smaller cases are easy to find. For larger cases we use Elimination, Resultants, Parametrization etc. There are no maps of degree 17 for our project. We use special techniques given by F. Beukers and H. Montanus to compute degree 18 Belyi maps.
- The major task is to prove that we have computed ALL Belyi and near Belyi maps relevant to our project.

The Major Task

Let L_{inp} be a second order linear differential operator with five regular singularities where at least one singularity is logarithmic. Suppose L_{inp} has ${}_2F_1$-type solution with the choice of exponent differences given in Takeuchi's diagram.

The Major Task

Let L_{inp} be a second order linear differential operator with five regular singularities where at least one singularity is logarithmic. Suppose L_{inp} has ${}_2F_1$-type solution with the choice of exponent differences given in Takeuchi's diagram.

- **We have to develop a complete table T of relevant rational functions $f(x), f(x,t)$ and $f(x,s,t)$ such that there exists at least one $f \in T$ and a suitable Möbius transformation m for which**

$$H_{c,x}^{a,b} \xrightarrow{f(m)}_C H_{c,f(m)}^{a,b} \xrightarrow{r_0,r_1}_G \xrightarrow{r}_E L_{inp}$$

Completeness of the Table

- **Question: How do we prove that the table is complete?**

Completeness of the Table

- **Question: How do we prove that the table is complete?**
- **Proof:**
  We use the following correspondence (more details later);
  Belyi maps $\longleftrightarrow$ dessins
  Belyi-1 maps $\longleftrightarrow$ near dessins
  Belyi-2 maps $\longleftrightarrow$ algorithms

Completeness of the Table

- **Question: How do we prove that the table is complete?**
- **Proof:**
  We use the following correspondence (more details later);
  Belyi maps $\longleftrightarrow$ dessins
  Belyi-1 maps $\longleftrightarrow$ near dessins
  Belyi-2 maps $\longleftrightarrow$ algorithms
- Once we have a complete table, we can develop a differential solver from it.

The Differential Solver	Table
$L = \partial^2 + \frac{(x+7)(x-39)}{(x-16)(x^2+18x-15)}\partial - \frac{25x^3-1006x^2-5523x-894}{36(x^2+18x-15)(x-16)(x^2-3)}$ $Sol = e^{\int r dx}\left(r_0 S(f) + r_1 S(f)'\right)$, $r, r_0, r_1 \in \mathbb{C}(x)$ and $S(f) = {}_2F_1\left(\frac{1}{12}, \frac{5}{12}; 1 \mid \frac{4(x^2+18x-15)^2(x^2-3)^3}{9(4x^3-29x^2+42x-21)^3}\right)$	**Belyi maps:** $F_1(x) = \frac{4(2x-5)(7x+20)^4}{x^5(5x+28)^2(5x+12)}$ $\vdots$ $F_{383}(x) = \ldots$ **Belyi-1 maps:** $G_1(x,s) = \frac{-64(x^2+sx-s)^3 x^2}{s^3(x-1)^3(8x^2+9sx-9s)}$ $\vdots$ $G_{100}(x,s) = \ldots$ **Belyi-2 maps:**
$\exists\, m = \frac{ax+b}{cx+d}$ such that $f = \begin{cases} F_i(x) \circ m & \text{or} \\ G_j(x,s)\|_{s=?} \circ m & \text{or} \\ \text{Belyi-2 map} \end{cases}$	

The Correspondence

- A dessin is a connected, oriented and bi-colored graph where any two vertices of different color are joined by an edge.

The Correspondence

- A dessin is a connected, oriented and bi-colored graph where any two vertices of different color are joined by an edge.
- Dessin of a Belyi map f is the graph of $f^{-1}([0,1])$.

The Correspondence

- A dessin is a connected, oriented and bi-colored graph where any two vertices of different color are joined by an edge.
- Dessin of a Belyi map f is the graph of $f^{-1}([0,1])$.
- A sequence $[g_1, g_2, \cdots, g_k]$ of permutations in S_n is called a constellation (or a k-constellation) of degree n if:
    1. the group $\langle g_1, g_2, \cdots, g_k \rangle$ is transitive,
    2. $g_1 g_2 \cdots g_k = 1$.

The Correspondence

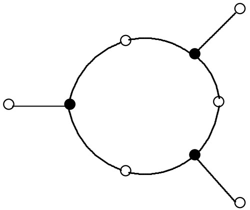
- A dessin is a connected, oriented and bi-colored graph where any two vertices of different color are joined by an edge.
- Dessin of a Belyi map f is the graph of $f^{-1}([0,1])$.
- A sequence $[g_1, g_2, \cdots, g_k]$ of permutations in S_n is called a constellation (or a k-constellation) of degree n if:
  1. the group $\langle g_1, g_2, \cdots, g_k \rangle$ is transitive,
  2. $g_1 g_2 \cdots g_k = 1$.
- There is a correspondence between dessins, Belyi maps (up to Möbius transformation) and 3-constellations (up to conjugation).

The Correspondence

- A dessin is a connected, oriented and bi-colored graph where any two vertices of different color are joined by an edge.
- Dessin of a Belyi map f is the graph of $f^{-1}([0,1])$.
- A sequence $[g_1, g_2, \cdots, g_k]$ of permutations in S_n is called a constellation (or a k-constellation) of degree n if:
  1. the group $\langle g_1, g_2, \cdots, g_k \rangle$ is transitive,
  2. $g_1 g_2 \cdots g_k = 1$.
- There is a correspondence between dessins, Belyi maps (up to Möbius transformation) and 3-constellations (up to conjugation).
- The braid group B_k generated by the braids $\sigma_1, \ldots, \sigma_{k-1}$ acts on a k-constellation in the following way:
  $\sigma_i : [g_1, \ldots, g_i, g_{i+1}, \ldots, g_k] \mapsto [g_1, \ldots, g_{i+1}, g_{i+1}^{-1} g_i g_{i+1}, \ldots, g_k]$

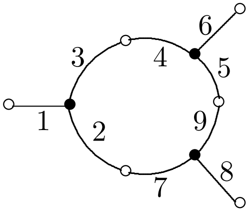
The Correspondence Contd.

- Here is an example of a dessin of degree 9:



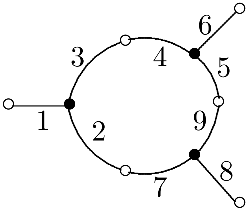
The Correspondence Contd.

- Here is an example of a dessin of degree 9:



The Correspondence Contd.

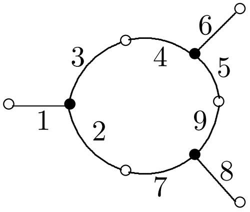
- Here is an example of a dessin of degree 9:



- This dessin has 3 black vertices (points above 0), 6 white vertices (points above 1) and 2 faces (correspond to poles).

The Correspondence Contd.

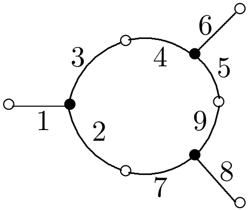
- Here is an example of a dessin of degree 9:



- Dessins do not have labels. The above 'labelled dessin' is useful to read the correspondence.

The Correspondence Contd.

- Here is an example of a dessin of degree 9:

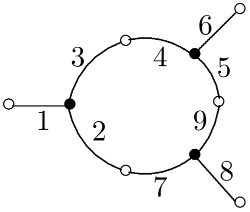


- This dessin corresponds to the following 3-constellation of degree 9 (unique up to conjugation):

$$g_0 = (1\ 2\ 3)\ (4\ 5\ 6)\ (7\ 8\ 9)$$
$$g_1 = (1)\ (6)\ (8)\ (2\ 7)\ (3\ 4)\ (5\ 9)$$
$$g_\infty = (g_0 g_1)^{-1} = (1\ 3\ 6\ 5\ 8\ 7)\ (2\ 9\ 4).$$

The Correspondence Contd.

- Here is an example of a dessin of degree 9:



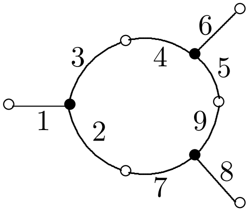
- and the following Belyi map (up to Möbius transformation):

$$f = \frac{4}{27}\,\frac{\left(x^3+1\right)^3}{x^3}$$

$$1 - f = -\frac{1}{27}\,\frac{\left(x^3+4\right)\left(2\,x^3-1\right)^2}{x^3}$$

The Correspondence Contd.

- Here is an example of a dessin of degree 9:



- A dessin is the equivalence class of 3-constellations mod conjugation. Conjugated 3-constellations give the same dessin (with different labelling).

Computing Relevant Dessins

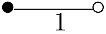
We have developed the table of Belyi maps. To prove the completeness we first enumerate all '5 singularity' dessins using combinatorial search including various techniques to prevent computational explosion. Then we compare the table of dessins with our table of Belyi maps. Steps:

Computing Relevant Dessins

We have developed the table of Belyi maps. To prove the completeness we first enumerate all ‘5 singularity’ dessins using combinatorial search including various techniques to prevent computational explosion. Then we compare the table of dessins with our table of Belyi maps. Steps:

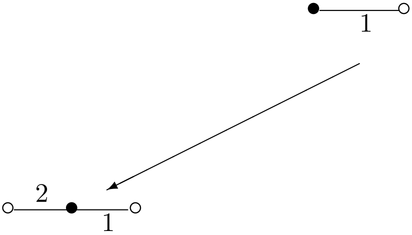
- Computing 3-constellations
- Computing dessins, i.e, discarding conjugates
- Discarding non-planar dessins, as well as dessins whose Weighted Singularity Count is too high
- Choosing only relevant dessins

Computing ‘Labelled Dessins’ or 3-constellations



$$g_0 = (1)$$
$$g_1 = (1)$$

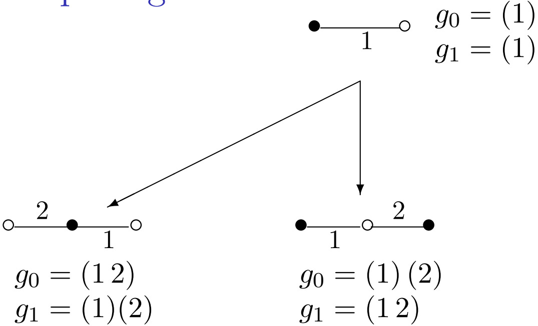
Computing ‘Labelled Dessins’ or 3-constellations



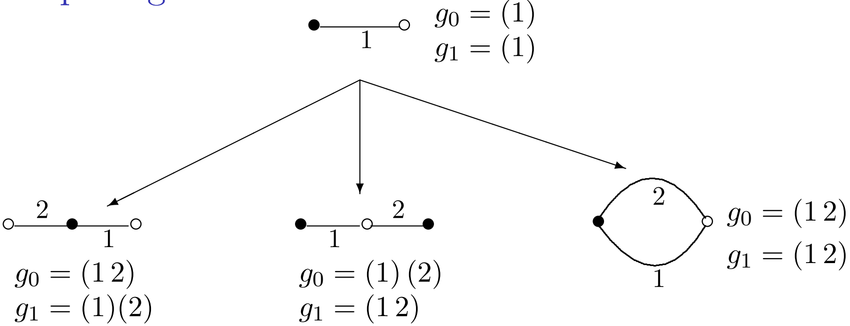
$g_0 = (1)$
$g_1 = (1)$

$g_0 = (1\,2)$
$g_1 = (1)(2)$

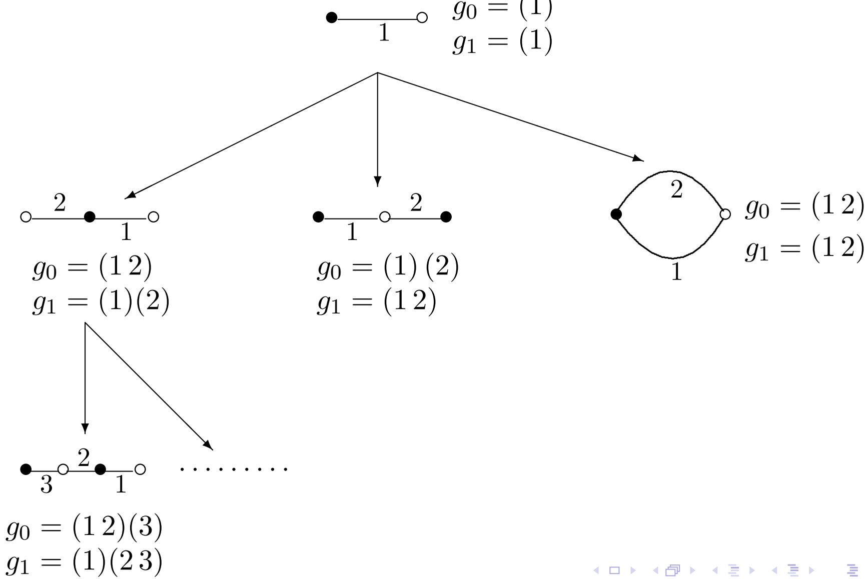
Computing ‘Labelled Dessins’ or 3-constellations



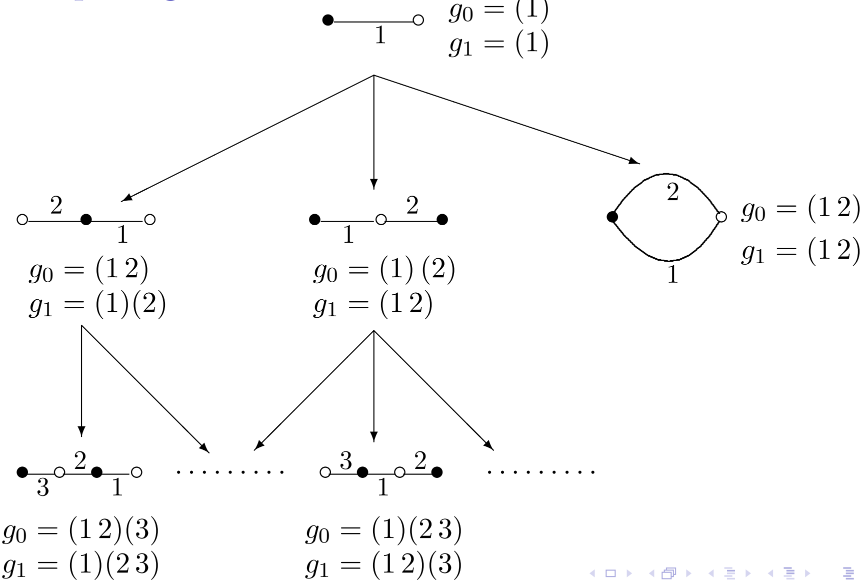
Computing 'Labelled Dessins' or 3-constellations



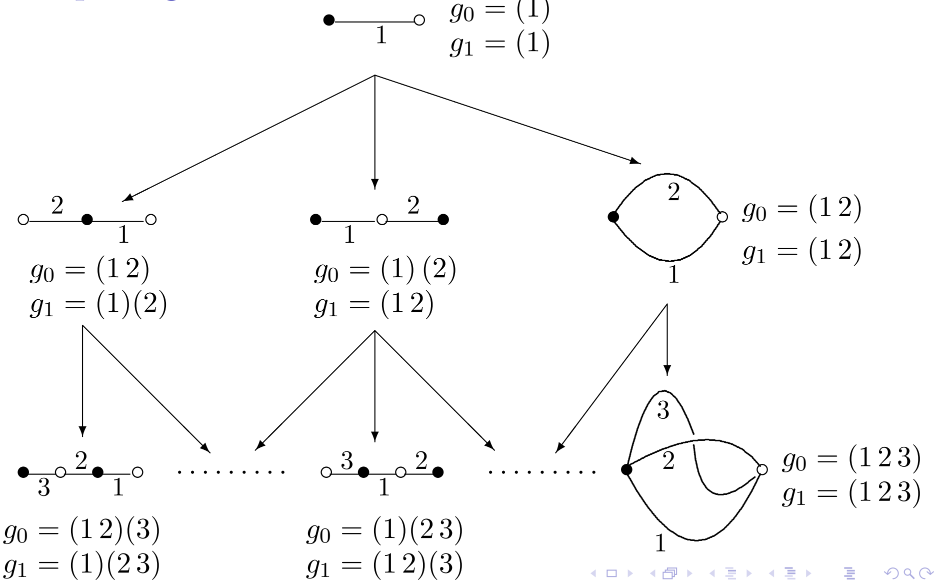
Computing ‘Labelled Dessins’ or 3-constellations



Computing ‘Labelled Dessins’ or 3-constellations



Computing 'Labelled Dessins' or 3-constellations



Computing ‘Labelled Dessins’ or 3-constellations Contd.

- A ‘labelled dessin’ of degree $n-1$ produces n^2-1 ‘labelled dessins’ of degree n.

Computing 'Labelled Dessins' or 3-constellations Contd.

- A 'labelled dessin' of degree $n-1$ produces n^2-1 'labelled dessins' of degree n.
- The number of 'labelled dessins' grows very rapidly:

$$T_n = \frac{(n-1)!(n+1)!}{2}$$

$$i.e,\ T_n = 1, 3, 24, 360, 8640, 302400, 14515200, 914457600, \ldots$$

Computing 'Labelled Dessins' or 3-constellations Contd.

- A 'labelled dessin' of degree $n-1$ produces n^2-1 'labelled dessins' of degree n.
- The number of 'labelled dessins' grows very rapidly:

$$T_n = \frac{(n-1)!(n+1)!}{2}$$

$i.e,\ T_n = 1, 3, 24, 360, 8640, 302400, 14515200, 914457600, \ldots$

- **Problem:** The program computes every 3-constellation. A dessin is a conjugacy class of 3-constellations. To prevent computing the same dessin many times, we should compute only one element from each conjugacy class (otherwise the output will be"error, out of memory" long before we reach $n = 18$). The next step is to identify conjugated 3-constellations and discard all but one of them.

Computing Dessins

- Any two 3-constellations $[g_0, g_1, g_\infty]$ and $[\tilde{g}_0, \tilde{g}_1, \tilde{g}_\infty]$ represent the same dessin iff $\exists\, \sigma \in S_n$ such that $\tilde{g}_i = \sigma g_i \sigma^{-1}$, $i \in \{0, 1, \infty\}$.

Computing Dessins

- Any two 3-constellations $[g_0, g_1, g_\infty]$ and $[\tilde{g}_0, \tilde{g}_1, \tilde{g}_\infty]$ represent the same dessin iff $\exists\, \sigma \in S_n$ such that $\tilde{g}_i = \sigma g_i \sigma^{-1}$, $i \in \{0, 1, \infty\}$.
- Conjugation is a reordering of the numbers in g_0, g_1, g_∞. We detect that reordering using the action of g_0 and g_1.

Computing Dessins

- Any two 3-constellations $[g_0, g_1, g_\infty]$ and $[\tilde{g}_0, \tilde{g}_1, \tilde{g}_\infty]$ represent the same dessin iff $\exists\, \sigma \in S_n$ such that $\tilde{g}_i = \sigma g_i \sigma^{-1}$, $i \in \{0, 1, \infty\}$.
- Conjugation is a reordering of the numbers in g_0, g_1, g_∞. We detect that reordering using the action of g_0 and g_1.
- Take a base point $b \in \{1, \ldots, n\}$ and apply the repeated action of g_0 and g_1 on b. That produces an ordering $\pi = [a_1, a_2, \ldots, a_n]$.

Computing Dessins

- Any two 3-constellations $[g_0, g_1, g_\infty]$ and $[\tilde{g}_0, \tilde{g}_1, \tilde{g}_\infty]$ represent the same dessin iff $\exists\, \sigma \in S_n$ such that $\tilde{g}_i = \sigma g_i \sigma^{-1}$, $i \in \{0, 1, \infty\}$.
- Conjugation is a reordering of the numbers in g_0, g_1, g_∞. We detect that reordering using the action of g_0 and g_1.
- Take a base point $b \in \{1, \ldots, n\}$ and apply the repeated action of g_0 and g_1 on b. That produces an ordering $\pi = [a_1, a_2, \ldots, a_n]$.
- We will obtain $\sigma\pi = [\sigma(a_1), \sigma(a_2), \ldots, \sigma(a_n)]$ after applying the repeated action of $\tilde{g}_0$ and $\tilde{g}_1$ on $\sigma(b)$. Moreover,

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

Computing Dessins Contd.

- Conjugation in g_i by π is the same as conjugation in $\tilde{g}_i$ by $\sigma\pi$.

Computing Dessins Contd.

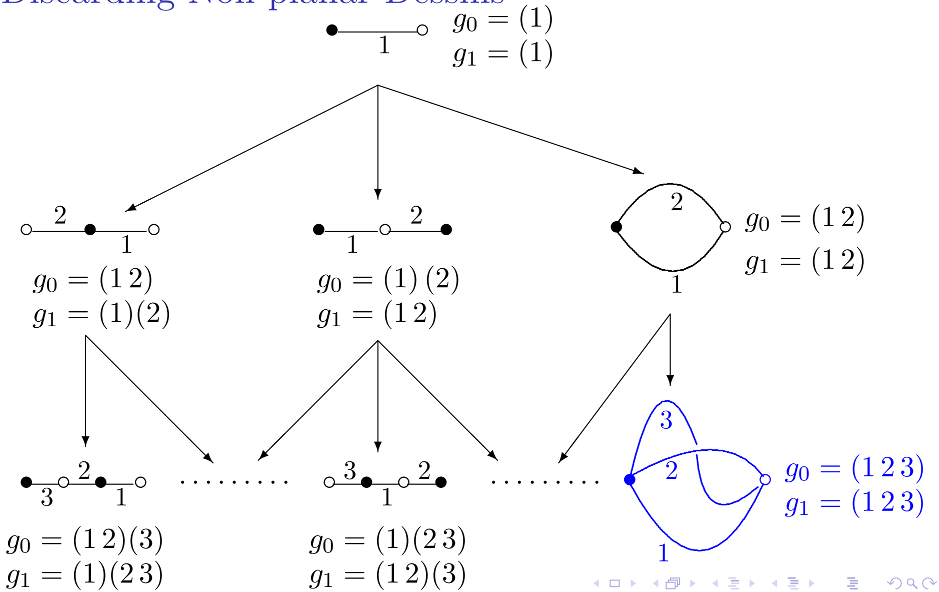
- Conjugation in g_i by π is the same as conjugation in $\tilde{g}_i$ by $\sigma\pi$.
- Computing the permutations from all $b \in \{1, 2, \ldots, n\}$ and conjugating gives two equal sets. We sort these sets with suitable ordering and check the first elements to detect conjugated 3-constellations.

Computing Dessins Contd.

- Conjugation in g_i by π is the same as conjugation in $\tilde{g}_i$ by $\sigma\pi$.
- Computing the permutations from all $b \in \{1, 2, \ldots, n\}$ and conjugating gives two equal sets. We sort these sets with suitable ordering and check the first elements to detect conjugated 3-constellations.
- Including this procedure discards conjugated 3-constellations and gives the following growth:

$$T_n = 1, 3, 7, 26, 97, 624, 4163, 34470, 314493, 3202839, \ldots$$

Discarding Non-planar Dessins



Discarding Non-planar Dessins

- The genus of a dessin of degree n is given by:

$$2g - 2 = n - \#\text{ black vertices - }\#\text{ white vertices - }\#\text{ faces.}$$

Discarding Non-planar Dessins

- The genus of a dessin of degree n is given by:

$$2g - 2 = n - \# \text{ black vertices - } \# \text{ white vertices - } \# \text{ faces.}$$

- Our dessins are planar (drawn in $\mathbb{P}^1$). So their genus must be zero.

Discarding Non-planar Dessins

- The genus of a dessin of degree n is given by:

  $$2g - 2 = n - \#\text{ black vertices - }\#\text{ white vertices - }\#\text{ faces.}$$

- Our dessins are planar (drawn in $\mathbb{P}^1$). So their genus must be zero.
- We compute the genus of each dessin and discard the non-planar dessins (genus > 0).

Discarding Non-planar Dessins

- The genus of a dessin of degree n is given by:

$$2g - 2 = n - \# \text{ black vertices - } \# \text{ white vertices - } \# \text{ faces.}$$

- Our dessins are planar (drawn in $\mathbb{P}^1$). So their genus must be zero.
- We compute the genus of each dessin and discard the non-planar dessins (genus > 0).
- Including this feature produces the following growth:

$$T_n = 1, 3, 6, 20, 60, 291, 1310, 6975, 37746, 215602, 1262874, \ldots$$

Discarding Non-planar Dessins

- The genus of a dessin of degree n is given by:

$$2g - 2 = n - \# \text{ black vertices - } \# \text{ white vertices - } \# \text{ faces.}$$

- Our dessins are planar (drawn in $\mathbb{P}^1$). So their genus must be zero.
- We compute the genus of each dessin and discard the non-planar dessins (genus > 0).
- Including this feature produces the following growth:

$$T_n = 1, 3, 6, 20, 60, 291, 1310, 6975, 37746, 215602, 1262874, \ldots$$

- The growth is much smaller, but still too large to reach $n = 18$ (we still get "error! out of memory").

Weighted Singularity Count

- This tool plays a significant role on controlling the growth.

Weighted Singularity Count

- This tool plays a significant role on controlling the growth.
- It is a real valued function, say W, on 3-constellations with the following properties:
  1. W never decreases when we add an edge,
  2. $W(D) \leq \text{Singularity-Count}(D)$ for every dessin D.

Weighted Singularity Count

- This tool plays a significant role on controlling the growth.
- It is a real valued function, say W, on 3-constellations with the following properties:
  1. W never decreases when we add an edge,
  2. $W(D) \leq \text{Singularity-Count}(D)$ for every dessin D.
- We can discard a 3-constellation D as soon as $W(D)$ exceeds the desired number of singularities. This tool is very useful as each 3-constellation contributes $n^2 - 1$ new 3-constellations in the next level.

Putting it All Together

By discarding

1. all but one member from each conjugacy class
2. non-planar dessins
3. dessins whose Weighted Singularity Count is too high

the table grows much more slowly. Not only are we able to compute all relevant dessins for $d = 5\,(n \leq 18)$ we can also do the same for $d = 6\,(n \leq 24)$.

Choosing Relevant Dessins

- Finally, we consider only those dessins which produce 5 non removable singularities from $0, 1, \infty$ with $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{k})$ where $k \in \{3, 4, 6\}$.

Choosing Relevant Dessins

- Finally, we consider only those dessins which produce 5 non removable singularities from $0, 1, \infty$ with $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{k})$ where $k \in \{3, 4, 6\}$.
- We computed all such dessins which produce up to 6 singularities (degree ≤ 24). The details for $(0, \frac{1}{2}, \frac{1}{3})$ up to 5 singularities are as follows:

d	n	dessin count for $(0, \frac{1}{2}, \frac{1}{3})$
3	≤ 6	1, 2, 1, 1, 0, 2
4	≤ 12	0, 1, 3, 4, 3, 6, 4, 6, 4, 4, 0, 6
5	≤ 18	0,0,2,6,12,19,22,26,32,39,36,50,40,42,32,32,0,26

Choosing Relevant Dessins

- Finally, we consider only those dessins which produce 5 non removable singularities from $0, 1, \infty$ with $(e_0, e_1, e_\infty) = (0, \frac{1}{2}, \frac{1}{k})$ where $k \in \{3, 4, 6\}$.
- We computed all such dessins which produce up to 6 singularities (degree ≤ 24). The details for $(0, \frac{1}{2}, \frac{1}{3})$ up to 5 singularities are as follows:

d	n	dessin count for $(0, \frac{1}{2}, \frac{1}{3})$
3	≤ 6	1, 2, 1, 1, 0, 2
4	≤ 12	0, 1, 3, 4, 3, 6, 4, 6, 4, 4, 0, 6
5	≤ 18	0,0,2,6,12,19,22,26,32,39,36,50,40,42,32,32,0,26

- COMPLETENESS: Once each member from our table of Belyi maps corresponds to a member from the table of dessins and vice versa, the table of Belyi maps is complete.

Computing Relevant Near Dessins

- There is a correspondence between Belyi-1 maps (up to Möbius transformation) and 4-constellations $[g_0, g_1, g_t, g_\infty]$ (up to conjugation and braid action) where g_t is a 2-cycle.

Computing Relevant Near Dessins

- There is a correspondence between Belyi-1 maps (up to Möbius transformation) and 4-constellations $[g_0, g_1, g_t, g_\infty]$ (up to conjugation and braid action) where g_t is a 2-cycle.
- Computing relevant near dessins involves the following steps:
  1. Listing all branching patterns (up to degree 12) which produce 5 non removable singularities from $\{0, 1, \infty\}$.
  2. Computing near dessins (4-constellations mod conjugation) for each branching pattern.
  3. Grouping near dessins together by braid orbits.

Computing Relevant Near Dessins

- There is a correspondence between Belyi-1 maps (up to Möbius transformation) and 4-constellations $[g_0, g_1, g_t, g_\infty]$ (up to conjugation and braid action) where g_t is a 2-cycle.
- Computing relevant near dessins involves the following steps:
  1. Listing all branching patterns (up to degree 12) which produce 5 non removable singularities from $\{0, 1, \infty\}$.
  2. Computing near dessins (4-constellations mod conjugation) for each branching pattern.
  3. Grouping near dessins together by braid orbits.
- The next slides will explain the procedure of computing relevant near dessins of degree 9 for $(e_0, e_1, e_\infty) = (\frac{1}{3}, \frac{1}{2}, 0)$.

Listing Branching Patterns

- Branching patterns above $0, 1$ are $[3, 3, 3], [1, 2, 2, 2, 2]$ respectively. Following is the list of branching patterns above ∞:
  $[1, 1, 1, 6], [1, 1, 2, 5], [1, 1, 3, 4], [1, 2, 2, 4], [1, 2, 3, 3], [2, 2, 2, 3]$

- 4 poles and a root above 1 produce 5 non removable singularities.

Computing Near Dessins

- g_0 has three 3-cycles. We can fix $g_0 = (1\,2\,3)\,(4\,5\,6)\,(7\,8\,9)$. (We are working mod conjugation)

Computing Near Dessins

- g_0 has three 3-cycles. We can fix $g_0 = (1\,2\,3)\,(4\,5\,6)\,(7\,8\,9)$. (We are working mod conjugation)
- g_1 has one 1-cycle and four 2-cycles. For g_1 we have $9 \cdot 7 \cdot 5 \cdot 3 \cdot 1 = 945$ choices.

Computing Near Dessins

- g_0 has three 3-cycles. We can fix $g_0 = (1\,2\,3)\,(4\,5\,6)\,(7\,8\,9)$. (We are working mod conjugation)
- g_1 has one 1-cycle and four 2-cycles. For g_1 we have $9 \cdot 7 \cdot 5 \cdot 3 \cdot 1 = 945$ choices.
- g_t is a 2-cycle. For g_t we have $\binom{9}{2} = 36$ choices.

Computing Near Dessins

- g_0 has three 3-cycles. We can fix $g_0 = (1\,2\,3)\,(4\,5\,6)\,(7\,8\,9)$. (We are working mod conjugation)
- g_1 has one 1-cycle and four 2-cycles. For g_1 we have $9 \cdot 7 \cdot 5 \cdot 3 \cdot 1 = 945$ choices.
- g_t is a 2-cycle. For g_t we have $\binom{9}{2} = 36$ choices.
- For each of the $945 \cdot 36 = 34020$ triples (g_0, g_1, g_t) we check the following:
  1. Is $\langle g_0, g_1, g_t \rangle$ transitive?
  2. Does the product $g_0 g_1 g_t$ have 4 disjoint cycles? $(g_0 g_1 g_t = g_\infty^{-1})$

Computing Near Dessins

- g_0 has three 3-cycles. We can fix $g_0 = (1\,2\,3)\,(4\,5\,6)\,(7\,8\,9)$. (We are working mod conjugation)
- g_1 has one 1-cycle and four 2-cycles. For g_1 we have $9 \cdot 7 \cdot 5 \cdot 3 \cdot 1 = 945$ choices.
- g_t is a 2-cycle. For g_t we have $\binom{9}{2} = 36$ choices.
- For each of the $945 \cdot 36 = 34020$ triples (g_0, g_1, g_t) we check the following:
  1. Is $\langle g_0, g_1, g_t \rangle$ transitive?
  2. Does the product $g_0 g_1 g_t$ have 4 disjoint cycles? $(g_0 g_1 g_t = g_\infty^{-1})$
- Computing near dessins (4-constellations mod conjugation) is similar to the procedure of computing dessins. Here we use the action of g_0, g_1 and g_t.

Braid Orbits

- Applying braid group action on each near dessin produces braid orbits.

Braid Orbits

- Applying braid group action on each near dessin produces braid orbits.
- Analytic continuation of the fourth branch point t around $0, 1, \infty$ permutes the near dessins on the same braid orbits.

Braid Orbits

- Applying braid group action on each near dessin produces braid orbits.
- Analytic continuation of the fourth branch point t around $0, 1, \infty$ permutes the near dessins on the same braid orbits.
- Our computation produces braid orbits with the following branching patterns above ∞:
  $[1, 1, 1, 6], [1, 1, 1, 6], [1, 1, 2, 5], [1, 1, 3, 4], [1, 2, 2, 4], [1, 2, 3, 3], [2, 2, 2, 3]$

Braid Orbits

- Applying braid group action on each near dessin produces braid orbits.
- Analytic continuation of the fourth branch point t around $0, 1, \infty$ permutes the near dessins on the same braid orbits.
- Our computation produces braid orbits with the following branching patterns above ∞: $[1,1,1,6], [1,1,1,6], [1,1,2,5], [1,1,3,4], [1,2,2,4], [1,2,3,3], [2,2,2,3]$
- Following are the Belyi-1 maps with branching pattern $[1,1,1,6]$:

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

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

Braid Orbits Contd.

- For f_1 the fourth branch point $t = \frac{1}{19683} \frac{(4\,s-81)^3}{s-27}$.

Braid Orbits Contd.

- For f_1 the fourth branch point $t = \frac{1}{19683}\frac{(4\,s-81)^3}{s-27}$.
- For each fixed value of $t \notin \{0, 1, \infty\}$, we get 3 distinct values of s which produce 3 distinct near dessins.

Braid Orbits Contd.

- For f_1 the fourth branch point $t = \frac{1}{19683}\frac{(4\,s-81)^3}{s-27}$.
- For each fixed value of $t \notin \{0, 1, \infty\}$, we get 3 distinct values of s which produce 3 distinct near dessins.
- Analytic continuation of t around $0, 1, \infty$ permutes the values of s, i.e, the corresponding near dessins.

Braid Orbits Contd.

- For f_1 the fourth branch point $t = \frac{1}{19683}\frac{(4\,s-81)^3}{s-27}$.
- For each fixed value of $t \notin \{0, 1, \infty\}$, we get 3 distinct values of s which produce 3 distinct near dessins.
- Analytic continuation of t around $0, 1, \infty$ permutes the values of s, i.e, the corresponding near dessins.
- For f_2 we get $t = \frac{2}{19683}\frac{\left(2\,s^3+27\,s^2+486\,s-1458\right)^3}{s^4(s^3+27\,s^2+243\,s-729)}$.

Braid Orbits Contd.

- For f_1 the fourth branch point $t = \frac{1}{19683}\frac{(4s-81)^3}{s-27}$.
- For each fixed value of $t \notin \{0,1,\infty\}$, we get 3 distinct values of s which produce 3 distinct near dessins.
- Analytic continuation of t around $0,1,\infty$ permutes the values of s, i.e, the corresponding near dessins.
- For f_2 we get $t = \frac{2}{19683}\frac{\left(2s^3+27s^2+486s-1458\right)^3}{s^4(s^3+27s^2+243s-729)}$.
- COMPLETENESS: We choose a value of s with $t \notin \{0,1,\infty\}$ for each Belyi-1 map $f(x,s)$. Then we compute monodromy g_0, g_1, g_t, g_∞ using Maple. The table of Belyi-1 maps is complete if $\forall$ braid orbit $\exists$ a Belyi-1 map f in our table with $[g_0, g_1, g_t, g_\infty]$ on that orbit.

Braid Orbits Contd.

- For f_1 the fourth branch point $t = \frac{1}{19683}\frac{(4s-81)^3}{s-27}$.
- For each fixed value of $t \notin \{0, 1, \infty\}$, we get 3 distinct values of s which produce 3 distinct near dessins.
- Analytic continuation of t around $0, 1, \infty$ permutes the values of s, i.e, the corresponding near dessins.
- For f_2 we get $t = \frac{2}{19683}\frac{\left(2s^3+27s^2+486s-1458\right)^3}{s^4(s^3+27s^2+243s-729)}$.
- COMPLETENESS: We choose a value of s with $t \notin \{0, 1, \infty\}$ for each Belyi-1 map $f(x, s)$. Then we compute monodromy g_0, g_1, g_t, g_∞ using Maple. The table of Belyi-1 maps is complete if $\forall$ braid orbit $\exists$ a Belyi-1 map f in our table with $[g_0, g_1, g_t, g_\infty]$ on that orbit.
- Monodromy groups of f_1 and f_2 have different order. Hence $\{f_1, f_2\}$ completely cover the branching pattern $[1, 1, 1, 6]$.

Completeness of Belyi-2 maps

- Our program gives two branching patterns for Belyi-2 maps which occur only for $(0, \frac{1}{2}, \frac{1}{3})$;
  $[1, 1, 1, 1], [2, 2], [1, 3]$ and $[1, 1, 1, 1, 2], [2, 2, 2], [3, 3]$

Completeness of Belyi-2 maps

- Our program gives two branching patterns for Belyi-2 maps which occur only for $(0, \frac{1}{2}, \frac{1}{3})$; $[1, 1, 1, 1], [2, 2], [1, 3]$ and $[1, 1, 1, 1, 2], [2, 2, 2], [3, 3]$
- Five singularities up to Möbius transformation have two degrees of freedom, which is just enough to extract the parameters of a 2-dimensional family.

Completeness of Belyi-2 maps

- Our program gives two branching patterns for Belyi-2 maps which occur only for $(0, \frac{1}{2}, \frac{1}{3})$;
$[1, 1, 1, 1], [2, 2], [1, 3]$ and $[1, 1, 1, 1, 2], [2, 2, 2], [3, 3]$
- Five singularities up to Möbius transformation have two degrees of freedom, which is just enough to extract the parameters of a 2-dimensional family.
- The generic Belyi-2 map with branching pattern $[1, 1, 1, 1], [2, 2], [1, 3]$ is the following:

$$f = k_1 \cdot \frac{(x^4 + a_3x^3 + a_2x^2 + a_1x + a_0)}{(x - b_1)(x - b_2)^3}, \; 1-f = k_2 \cdot \frac{(x^2 + c_1x + c_0)^2}{(x - b_1)(x - b_2)^3}$$

Completeness of Belyi-2 maps

- Our program gives two branching patterns for Belyi-2 maps which occur only for $(0, \frac{1}{2}, \frac{1}{3})$;
  $[1,1,1,1], [2,2], [1,3]$ and $[1,1,1,1,2], [2,2,2], [3,3]$
- Five singularities up to Möbius transformation have two degrees of freedom, which is just enough to extract the parameters of a 2-dimensional family.
- The generic Belyi-2 map with branching pattern $[1,1,1,1], [2,2], [1,3]$ is the following:

$$f = k_1 \cdot \frac{(x^4 + a_3x^3 + a_2x^2 + a_1x + a_0)}{(x-b_1)(x-b_2)^3}, \ 1-f = k_2 \cdot \frac{(x^2 + c_1x + c_0)^2}{(x-b_1)(x-b_2)^3}$$

- $(x^4 + a_3x^3 + a_2x^2 + a_1x + a_0)$ and $(x - b_1)$ are obtained from the singularities of L_{inp}. Then we have 5 equations with 5 unknowns.

THANK YOU!

- Thank you Dr. van Hoeij for your invaluable guide, support and friendship. You are a great advisor!
- Thank you my committee members for your help, support and cooperation.
- I dedicate this achievement to my sweet family. It would not be possible without their love, care and company.