Solving Second Order Linear Differential Equations with Klein's Theorem

Mark van Hoeij (Florida State University (USA))
Jacques-Arthur Weil (LACO−XLIM, Limoges (France))
hoeij@math.fsu.edu jacques-arthur.weil@unilim.fr

Talk presented by Thomas Cluzeau
LACO-XLIM, Limoges (France)

This talk is dedicated to the memory of Manuel Bronstein

Liouvillian Solutions of Second Order Linear ODE's: The Problem

$$y'' + a_1(x)y'(x) + a_0(x)y(x) = 0, \qquad a_i(x) \in k$$

k is a differential field, e.g $C(x)$, $C(x, \exp(x))$

A solution y is called

1. Rational: if $y \in k$
2. Exponential: if $y'/y \in k$
3. Liouvillian: if y can be presented by any combination of: algebraic extensions, arithmetic operations, exp( ), and $\int$

The problem is to compute Liouvillian solutions.

Liouvillian Solutions of Second Order Linear ODE's: Algorithms

1. Kovacic, 1977, 1986
   - If a Liouvillian solution exists then $\exists$ solution of the form $y = \exp\left(\int \omega\right)$ with ω algebraic. Minimal polynomial of ω is computed using *semi-invariants* and a recursive formula.
2. Ulmer & Weil, 1996
   - Compute minpoly ω from *invariants* (easier to implement).
3. Fakler, 1997, computes algebraic solutions y in nicer form:
   - Gives the minpoly of y instead of minpoly of ω.
4. Klein (1877) ... Berkenbosch, van Hoeij, and Weil (2002)
   - Write Liouvillian solutions as hypergeometric functions composed with a function (called the *pullback*) in k.
   - Formulas for pullback given in B.H.W. using *semi-invariants*.

Liouvillian Solutions of Second Order Linear ODE's: Algorithms

1. Kovacic, 1977, 1986
   - If a Liouvillian solution exists then $\exists$ solution of the form $y = \exp\left(\int \omega\right)$ with ω algebraic. Minimal polynomial of ω is computed using *semi-invariants* and a recursive formula.
2. Ulmer & Weil, 1996
   - Compute minpoly ω from *invariants* (easier to implement).
3. Fakler, 1997, computes algebraic solutions y in nicer form:
   - Gives the minpoly of y instead of minpoly of ω.
4. Klein (1877) ... Berkenbosch, van Hoeij, and Weil (2002)
   - Write Liouvillian solutions as hypergeometric functions composed with a function (called the *pullback*) in k.
   - Formulas for pullback given in B.H.W. using *semi-invariants*.

Liouvillian Solutions of Second Order Linear ODE's: Algorithms

1. Kovacic, 1977, 1986
   - If a Liouvillian solution exists then $\exists$ solution of the form $y = \exp\left(\int \omega\right)$ with ω algebraic. Minimal polynomial of ω is computed using *semi-invariants* and a recursive formula.
2. Ulmer & Weil, 1996
   - Compute minpoly ω from *invariants* (easier to implement).
3. Fakler, 1997, computes algebraic solutions y in nicer form:
   - Gives the minpoly of y instead of minpoly of ω.
4. Klein (1877) ... Berkenbosch, van Hoeij, and Weil (2002)
   - Write Liouvillian solutions as hypergeometric functions composed with a function (called the *pullback*) in k.
   - Formulas for pullback given in B.H.W. using *semi-invariants*.

Liouvillian Solutions of Second Order Linear ODE's: Algorithms

1. Kovacic, 1977, 1986
   - If a Liouvillian solution exists then $\exists$ solution of the form $y = \exp\left(\int \omega\right)$ with ω algebraic. Minimal polynomial of ω is computed using *semi-invariants* and a recursive formula.
2. Ulmer & Weil, 1996
   - Compute minpoly ω from *invariants* (easier to implement).
3. Fakler, 1997, computes algebraic solutions y in nicer form:
   - Gives the minpoly of y instead of minpoly of ω.
4. Klein (1877) ... Berkenbosch, van Hoeij, and Weil (2002)
   - Write Liouvillian solutions as hypergeometric functions composed with a function (called the *pullback*) in k.
   - Formulas for pullback given in B.H.W. using *semi-invariants*.

$$L(y) := y'' + a_1(x)y'(x) + a_0(x)y(x) = 0, \qquad a_i(x) \in k$$

Write Liouvillian Solutions as $H(f) \cdot \exp\left(\int v\right)$ where H is a Hypergeometric function from Klein's table and $f, v \in k$.

Advantage: A more compact representation of the solutions.

Sketch of the Approach:

- Invariants $\Longrightarrow$ The differential Galois group $G(L)$ and v.
- $G(L)$ and Klein's table $\Longrightarrow$ H.
- a_0, a_1, v and a pre-computed formula $\Longrightarrow$ f.

Our contribution: Formulas to compute v and f using invariants.

- Makes it easy to implement: need only rational sols of linear ODE's (and exponential sols of L itself, if L is reducible).
- Available in Maple 10 and in Bernina.

$$L(y) := y'' + a_1(x)y'(x) + a_0(x)y(x) = 0, \qquad a_i(x) \in k$$

Write Liouvillian Solutions as $H(f) \cdot \exp\left(\int v\right)$ where H is a Hypergeometric function from Klein's table and $f, v \in k$.

Advantage: A more compact representation of the solutions.

Sketch of the Approach:

- Invariants $\Longrightarrow$ The differential Galois group $G(L)$ and v.
- $G(L)$ and Klein's table $\Longrightarrow$ H.
- a_0, a_1, v and a pre-computed formula $\Longrightarrow$ f.

Our contribution: Formulas to compute v and f using invariants.

- Makes it easy to implement: need only rational sols of linear ODE's (and exponential sols of L itself, if L is reducible).
- Available in Maple 10 and in Bernina.

$$L(y) := y'' + a_1(x)y'(x) + a_0(x)y(x) = 0, \qquad a_i(x) \in k$$

Write Liouvillian Solutions as $H(f) \cdot \exp\left(\int v\right)$ where H is a Hypergeometric function from Klein's table and $f, v \in k$.

Advantage: A more compact representation of the solutions.

Sketch of the Approach:

- Invariants $\Longrightarrow$ The differential Galois group $G(L)$ and v.
- $G(L)$ and Klein's table $\Longrightarrow H$.
- a_0, a_1, v and a pre-computed formula $\Longrightarrow f$.

Our contribution: Formulas to compute v and f using invariants.

- Makes it easy to implement: need only rational sols of linear ODE's (and exponential sols of L itself, if L is reducible).
- Available in Maple 10 and in Bernina.

Solutions and Differential Galois Groups

$$L(y) := y'' + a_1(x)y'(x) + a_0(x)y(x) = 0, \qquad a_i(x) \in k$$

May assume $a_1 = 0$.

To L is associated a diff. Galois group $G(L)$.

$G(L)$ is a group of 2×2 matrices, acts on sol. space.

- Discriminate between groups via computing semi-invariants (Kovacic, Singer-Ulmer) or invariants (Ulmer-Weil)
- Invariants are found by finding *rational solutions* (in k) of an auxilliary operator: the *symmetric power* $Sym^m(L)$.

Solutions and Differential Galois Groups

$$L(y) := y'' + a_1(x)y'(x) + a_0(x)y(x) = 0, \qquad a_i(x) \in k$$

May assume $a_1 = 0$.

To L is associated a diff. Galois group $G(L)$.

$G(L)$ is a group of 2×2 matrices, acts on sol. space.

- Discriminate between groups via computing semi-invariants (Kovacic, Singer-Ulmer) or invariants (Ulmer-Weil)
- Invariants are found by finding *rational solutions* (in k) of an auxilliary operator: the *symmetric power* $Sym^m(L)$.

Invariants and the UW-Kovacic algorithm

$$L(y) := y'' + a_1(x)y'(x) + a_0(x)y(x) = 0, \qquad a_i(x) \in k$$

Assume L is irreducible (no exponential sols).

Projective group $PG(L)$:= $G(L)$ mod center.

We compute $PG(L)$ (and later also v and f) from invariants:

1. If $\exists$ invariant(s) of degree 4: group is D_n or D_∞.
   ($n = 2$ is a special case, there we will compute v and f from the invariant of degree 6).
2. else, if $\exists$ invariant of degree 6: group is A_4
3. else, if $\exists$ invariant of degree 8: group is S_4
4. else, if $\exists$ invariant of degree 12: group is A_5
5. else: group is PSL_2 (no Liouvillian solutions).

Invariants and the UW-Kovacic algorithm

$$L(y) := y'' + a_1(x)y'(x) + a_0(x)y(x) = 0, \qquad a_i(x) \in k$$

Assume L is irreducible (no exponential sols).

Projective group $PG(L)$:= $G(L)$ mod center.

We compute $PG(L)$ (and later also v and f) from invariants:

1. If $\exists$ invariant(s) of degree 4: group is D_n or D_∞.
   ($n = 2$ is a special case, there we will compute v and f from the invariant of degree 6).
2. else, if $\exists$ invariant of degree 6: group is A_4
3. else, if $\exists$ invariant of degree 8: group is S_4
4. else, if $\exists$ invariant of degree 12: group is A_5
5. else: group is PSL_2 (no Liouvillian solutions).

Pullbacks

Definition

Let $L \in C(z)\left[\frac{d}{dz}\right]$ and $\mathcal{L} \in k\left[\partial\right]$ be differential operators.

1. $\mathcal{L}$ is a *proper pullback* of L by $f \in k$ if the change of variable $z \mapsto f$ changes L into $\mathcal{L}$. Then:

   Solutions $y(z)$ of $L \iff$ Solutions $y(f)$ of $\mathcal{L}$.

2. $\mathcal{L}$ is a *(weak) pullback* of L by $f \in k$ if $\exists v \in k$ such that we can transform L into $\mathcal{L}$ by doing a
   - change of variable: $z \mapsto f$, followed by
   - scaling: multiplying all solutions by $\exp(\int v)$.

   Then:

   Solutions $y(z)$ of $L \iff$ Solutions $y(f) \cdot \exp(\int v)$ of $\mathcal{L}$.

Pullbacks

Definition

Let $L \in C(z)\left[\frac{d}{dz}\right]$ and $\mathcal{L} \in k\left[\partial\right]$ be differential operators.

1. $\mathcal{L}$ is a *proper pullback* of L by $f \in k$ if the change of variable $z \mapsto f$ changes L into $\mathcal{L}$. Then:

   Solutions $y(z)$ of $L \iff$ Solutions $y(f)$ of $\mathcal{L}$.

2. $\mathcal{L}$ is a *(weak) pullback* of L by $f \in k$ if $\exists v \in k$ such that we can transform L into $\mathcal{L}$ by doing a
   - change of variable: $z \mapsto f$, followed by
   - scaling: multiplying all solutions by $\exp(\int v)$.

   Then:

   Solutions $y(z)$ of $L \iff$ Solutions $y(f) \cdot \exp(\int v)$ of $\mathcal{L}$.

Klein's pullback theorem

To each $G \in \{D_n, A_4, S_4, A_5\}$, one associates a Standard Equation (we scaled them in such a way that the invariant has value 1)

$$St_{D_2} = \partial^2 + \frac{4}{3}\frac{z}{(z^2-1)}\partial - \frac{5}{144}\frac{z^2+3}{(z^2-1)^2} \quad (1)$$

$$St_{A_4} = \partial^2 + \frac{2\left(3z^2-1\right)}{3z\left(z^2-1\right)}\partial + \frac{5}{144z^2\left(z^2-1\right)} \quad (2)$$

$$\ldots \quad (3)$$

Theorem (Klein)

Let L be a second order irreducible linear differential operator over k with projective differential Galois group $PG(L)$. If $PG(L)$ is finite then L is a (weak) pullback of $St_{PG(L)}$.

This means: can write solutions of L as $H_{PG(L)}(f)\exp\left(\int v\right)$ where $H_G(z)$ = Hypergeometric sols of St_G.

Klein's pullback theorem

To each $G \in \{D_n, A_4, S_4, A_5\}$, one associates a Standard Equation (we scaled them in such a way that the invariant has value 1)

$$St_{D_2} = \partial^2 + \frac{4}{3}\frac{z}{(z^2-1)}\partial - \frac{5}{144}\frac{z^2+3}{(z^2-1)^2} \quad (1)$$

$$St_{A_4} = \partial^2 + \frac{2\left(3\,z^2-1\right)}{3z\left(z^2-1\right)}\partial + \frac{5}{144z^2\left(z^2-1\right)} \quad (2)$$

$$\ldots \quad (3)$$

Theorem (Klein)

Let L be a second order irreducible linear differential operator over k with projective differential Galois group $PG(L)$. If $PG(L)$ is finite then L is a (weak) pullback of $St_{PG(L)}$.

This means: can write solutions of L as $H_{PG(L)}(f)\exp\left(\int v\right)$ where $H_G(z) =$ Hypergeometric sols of St_G.

The Algorithm: Example of the A_4 Group

Suppose for example the input of our algorithm is a differential operator L with group $PG(L) = A_4$. How would the algorithm determine $PG(L)$, the pullback f, and the solutions of L?

Group..

1. L irreducible. No invariants of degree $1, 2, 4$ and an invariant of degree 6 with value I_6. So the projective group is A_4.

Scaling..v

2. Divide solutions of L by $I_6^{1/6} \Longrightarrow$ new operator L_S that must be a *proper pullback* of St_{A_4} (because both operators have invariant value 1, and $y(z) \mapsto y(f)$ sends 1 to 1).

Pullback..f

3. Write $L_S = \partial^2 + a_1\partial + a_0$. Compute $g := 2a_1 + \frac{a_0'}{a_0}$, and the pullback mapping is $f = \pm\sqrt{1 + \frac{64}{5}\frac{a_0}{g^2}}$ it is rational!
Solutions: $H_{A_4}(f) \cdot I_6^{1/6}$ for any solution H_{A_4} of St_{A_4}

The Algorithm: Example of the A_4 Group

Suppose for example the input of our algorithm is a differential operator L with group $PG(L) = A_4$. How would the algorithm determine $PG(L)$, the pullback f, and the solutions of L?

Group..

1. L irreducible. No invariants of degree $1, 2, 4$ and an invariant of degree 6 with value I_6. So the projective group is A_4.

Scaling..v

2. Divide solutions of L by $I_6^{1/6} \Longrightarrow$ new operator L_S that must be a *proper pullback* of St_{A_4} (because both operators have invariant value 1, and $y(z) \mapsto y(f)$ sends 1 to 1).

Pullback..f

3. Write $L_S = \partial^2 + a_1\partial + a_0$. Compute $g := 2a_1 + \frac{a_0'}{a_0}$, and the pullback mapping is $f = \pm\sqrt{1 + \frac{64}{5}\frac{a_0}{g^2}}$ it is rational!
Solutions: $H_{A_4}(f) \cdot I_6^{1/6}$ for any solution H_{A_4} of St_{A_4}

The Algorithm: Example of the A_4 Group

Suppose for example the input of our algorithm is a differential operator L with group $PG(L) = A_4$. How would the algorithm determine $PG(L)$, the pullback f, and the solutions of L?

Group..

1. L irreducible. No invariants of degree $1, 2, 4$ and an invariant of degree 6 with value I_6. So the projective group is A_4.

Scaling..v

2. Divide solutions of L by $I_6^{1/6} \Longrightarrow$ new operator L_S that must be a *proper pullback* of St_{A_4} (because both operators have invariant value 1, and $y(z) \mapsto y(f)$ sends 1 to 1).

Pullback..f

3. Write $L_S = \partial^2 + a_1\partial + a_0$. Compute $g := 2a_1 + \frac{a_0'}{a_0}$, and the pullback mapping is $f = \pm\sqrt{1 + \frac{64}{5}\frac{a_0}{g^2}}$ it is rational!

Solutions: $H_{A_4}(f) \cdot I_6^{1/6}$ for any solution H_{A_4} of St_{A_4}

The Algorithm: Example of the A_4 Group

Suppose for example the input of our algorithm is a differential operator L with group $PG(L) = A_4$. How would the algorithm determine $PG(L)$, the pullback f, and the solutions of L?

Group..

1. L irreducible. No invariants of degree $1, 2, 4$ and an invariant of degree 6 with value I_6. So the projective group is A_4.

Scaling..v

2. Divide solutions of L by $I_6^{1/6} \implies$ new operator L_S that must be a *proper pullback* of St_{A_4} (because both operators have invariant value 1, and $y(z) \mapsto y(f)$ sends 1 to 1).

Pullback..f

3. Write $L_S = \partial^2 + a_1\partial + a_0$. Compute $g := 2a_1 + \frac{a_0'}{a_0}$, and the pullback mapping is $f = \pm\sqrt{1 + \frac{64}{5}\frac{a_0}{g^2}}$ it is rational!

   Solutions: $H_{A_4}(f) \cdot I_6^{1/6}$ for any solution H_{A_4} of St_{A_4}

How the pullback formula was found

For $G = A_4$ the *pullback formula* on the previous page was $f = \pm\sqrt{1 + \frac{64}{5}\frac{a_0}{g^2}}$ where $g = 2a_1 + \frac{a_0'}{a_0}$.

Our algorithm contains a pullback formula for each group G. These formulas were found as follows:

- Take a standard equation for G from Klein's table.
- Key idea: Scale it so that the invariant has value 1. Doing this to all operators reduces weak pullpacks to proper pullbacks!
- Change of variable $z \mapsto F$. One obtains a differential operator $\partial^2 + a_1\partial + a_0$ where $a_1, a_0 \in C(F, F', F'')$.
- Use differential elimination to express F in terms of a_1, a_0.
- For A_4 we got $F = \pm\sqrt{1 + \frac{64}{5}\frac{a_0}{g^2}}$ where $g = 2a_1 + \frac{a_0'}{a_0}$.
- For S_4 we got $F = \frac{-7}{144}\frac{g^2}{a_0}$.
- For other groups: see paper.

Example: group A_4 concretely

$$L(y) :=$$
$$y'' - \frac{1}{144}\frac{404\,(e^x)^2x - 27\,x^2 + 108\,x^3 + 54\,x^4 + 9\,x^6 - 36\,x^5 + 216\,(e^x)^4 + \cdots}{(x-e^x)^2(x+e^x)^2(x-1)^2}y = 0$$

Group

1. No invariants of degree 1 or 2 or 4
2. Invariant of degree 6, value $I_6 = \frac{\left(x^2 - e^{2x}\right)^2}{e^x(x-1)^3}$: $Sym^6(L)(I_6) = 0$
   $\Longrightarrow PG(L) = A_4$

Normalize

3. Rescale operator L: Get L_S such that $Sym^6(L_S)(1) = 0$.
   L_S is a proper pullback of St_{A_4} because $Sym^6(St_{A_4})(1) = 0$.

Pullback

4. Apply *pullback formula* to coeffs of L_S gives pullback $f = \frac{e^x}{x}$
5. Solutions are

$$\frac{\left(x^2-e^{2x}\right)^{2/3}}{\sqrt{x-1}}\left(C_1\frac{{}_2F_1\left([\frac{7}{24},\frac{19}{24}],[\frac{3}{4}],\frac{e^{2x}}{x^2}\right)}{e^{\frac{x}{4}}x^{\frac{7}{12}}} + C_2\frac{e^{\frac{x}{4}}\,{}_2F_1\left([\frac{13}{24},\frac{25}{24}],[\frac{5}{4}],\frac{e^{2x}}{x^2}\right)}{x^{\frac{13}{12}}}\right)$$

Example: group A_4 concretely

$$L(y) :=$$
$$y'' - \frac{1}{144}\,\frac{404\,(e^x)^2 x - 27\,x^2 + 108\,x^3 + 54\,x^4 + 9\,x^6 - 36\,x^5 + 216\,(e^x)^4 + \cdots}{(x-e^x)^2(x+e^x)^2(x-1)^2}\,y = 0$$

Group

1. No invariants of degree 1 or 2 or 4
2. Invariant of degree 6, value $I_6 = \frac{\left(x^2-e^{2x}\right)^2}{e^x(x-1)^3}$: $Sym^6(L)(I_6) = 0$ $\Longrightarrow PG(L) = A_4$

Normalize

3. Rescale operator L: Get L_S such that $Sym^6(L_S)(1) = 0$. L_S is a proper pullback of St_{A_4} because $Sym^6(St_{A_4})(1) = 0$.

Pullback

4. Apply *pullback formula* to coeffs of L_S gives pullback $f = \frac{e^x}{x}$
5. Solutions are

$$\frac{\left(x^2-e^{2x}\right)^{2/3}}{\sqrt{x-1}}\left(C_1\,\frac{{}_2F_1\left([\frac{7}{24},\frac{19}{24}],[\frac{3}{4}],\frac{e^{2x}}{x^2}\right)}{e^{\frac{x}{4}}\,x^{\frac{7}{12}}} + C_2\,\frac{e^{\frac{x}{4}}\,{}_2F_1\left([\frac{13}{24},\frac{25}{24}],[\frac{5}{4}],\frac{e^{2x}}{x^2}\right)}{x^{\frac{13}{12}}}\right)$$

Example: group A_4 concretely

$$L(y) :=$$
$$y'' - \frac{1}{144}\,\frac{404\,(e^x)^2 x - 27\,x^2 + 108\,x^3 + 54\,x^4 + 9\,x^6 - 36\,x^5 + 216\,(e^x)^4 + \cdots}{(x-e^x)^2(x+e^x)^2(x-1)^2}\,y = 0$$

Group

1. No invariants of degree 1 or 2 or 4
2. Invariant of degree 6, value $I_6 = \frac{\left(x^2-e^{2x}\right)^2}{e^x(x-1)^3}$: $Sym^6(L)(I_6) = 0$
   $\Longrightarrow PG(L) = A_4$

Normalize

3. Rescale operator L: Get L_S such that $Sym^6(L_S)(1) = 0$.
   L_S is a proper pullback of St_{A_4} because $Sym^6(St_{A_4})(1) = 0$.

Pullback

4. Apply *pullback formula* to coeffs of L_S gives pullback $f = \frac{e^x}{x}$
5. Solutions are

$$\frac{\left(x^2-e^{2x}\right)^{2/3}}{\sqrt{x-1}}\left(C_1\,\frac{{}_2F_1\left([\frac{7}{24},\frac{19}{24}],[\frac{3}{4}],\frac{e^{2x}}{x^2}\right)}{e^{\frac{x}{4}}\,x^{\frac{7}{12}}} + C_2\,\frac{e^{\frac{x}{4}}\,{}_2F_1\left([\frac{13}{24},\frac{25}{24}],[\frac{5}{4}],\frac{e^{2x}}{x^2}\right)}{x^{\frac{13}{12}}}\right)$$

Example: group A_5

$L(y) := 48x(x-1)(75x-139)y'' + (2520x^2 - 47712x/5 + 3336)y' + (36001/75 - 19x)y = 0.$

1. $PG(L)$ equals A_5 in this example.
2. Both the standard Kovacic algorithm and our pullback method need to compute the invariant of degree 12.
3. However, the pullback method produces much smaller solutions:
    1. Solutions in Maple 9.5 (standard Kovacic): 236789 bytes.
    2. Solutions in Maple 10 (using pullback): 1360 bytes.
4. The old output is very large is because it contains an algebraic function represented by its minimal polynomial, and every coefficient of this polynomial is a large rational function.
5. In contrast, the output from the pullback method contains only one large rational function, namely f (which has degree 31 in this example).

Example: group A_5

$L(y) := 48x(x-1)(75x-139)y'' + (2520x^2 - 47712x/5 + 3336)y' + (36001/75 - 19x)y = 0.$

1. $PG(L)$ equals A_5 in this example.
2. Both the standard Kovacic algorithm and our pullback method need to compute the invariant of degree 12.
3. However, the pullback method produces much smaller solutions:
   1. Solutions in Maple 9.5 (standard Kovacic): 236789 bytes.
   2. Solutions in Maple 10 (using pullback): 1360 bytes.
4. The old output is very large is because it contains an algebraic function represented by its minimal polynomial, and every coefficient of this polynomial is a large rational function.
5. In contrast, the output from the pullback method contains only one large rational function, namely f (which has degree 31 in this example).

Example: group A_5

$L(y) := 48x(x-1)(75x-139)y'' + (2520x^2 - 47712x/5 + 3336)y' + (36001/75 - 19x)y = 0.$

1. $PG(L)$ equals A_5 in this example.
2. Both the standard Kovacic algorithm and our pullback method need to compute the invariant of degree 12.
3. However, the pullback method produces much smaller solutions:
    1. Solutions in Maple 9.5 (standard Kovacic): 236789 bytes.
    2. Solutions in Maple 10 (using pullback): 1360 bytes.
4. The old output is very large is because it contains an algebraic function represented by its minimal polynomial, and every coefficient of this polynomial is a large rational function.
5. In contrast, the output from the pullback method contains only one large rational function, namely f (which has degree 31 in this example).

Conclusion

- Keys to the algorithm are:
  1. We choose standard equations with invariant value 1.
  2. Given an equation we want to solve, we compute its invariant, and then scale it so that it too has value 1.
  3. This reduces a weak pullback to a proper pullback,
  4. which allows us to find a formula for the pullback.
- Easy to implement (one can simply add the pullback formulas to existing Kovacic implementations).
- Slightly faster than Kovacic due to smaller output size.
- $\exists$ extensions to order 3 by Berkenbosch (no algo but good)
- Other works on special functions using special forms (e.g Cheb-Terrab 2004) or essential singularities (e.g Bronstein and Lafaille 2002): get non-Liouvillian functions.

Thank you for your attention.