Rigorous High-Precision Solution of Fuchsian ODEs and Applications

Marc Mezzarobba

CNRS, École polytechnique

Dynaverse 2025: Algebraic methods in dynamics and particle physics

Saarbrücken, Oct. 16, 2025

Rigorous High-Precision Solution of Fuchsian ODEs

$$a_r(x)\,y^{(r)}(x)+\cdots+a_1(x)\,y'(x)+a_0(x)\,y(x)=0$$

```
SageMath version 10.7, Release Date: 2025-08-09
sage: from ore_algebra import OreAlgebra
sage: POL.<x> = QQ[]; DOP.<Dx> = OreAlgebra(POL)
sage: (x^2 + 1)*Dx^2 + Dx + 1
```

Rigorous High-Precision Solution of Fuchsian ODEs

$$\overset{\overset{a_i \in \mathbb{C}[x]}{\downarrow}}{a_r}(x)\, y^{(r)}(x) + \cdots + \overset{\overset{a_i \in \mathbb{C}[x]}{\downarrow}}{a_1}(x)\, y'(x) + \overset{\overset{a_i \in \mathbb{C}[x]}{\downarrow}}{a_0}(x)\, y(x) = 0$$

```
SageMath version 10.7, Release Date: 2025-08-09
sage: from ore_algebra import OreAlgebra
sage: POL.<x> = QQ[]; DOP.<Dx> = OreAlgebra(POL)
sage: (x^2 + 1)*Dx^2 + Dx + 1
```

Rigorous High-Precision Solution of Fuchsian ODEs

$$\overset{\overset{a_i\in\mathbb{C}[x]}{\downarrow}}{a_r(x)}\,y^{(r)}(x)+\cdots+\overset{\overset{a_i\in\mathbb{C}[x]}{\downarrow}}{a_1(x)}\,y'(x)+\overset{\overset{a_i\in\mathbb{C}[x]}{\downarrow}}{a_0(x)}\,\underset{\underset{y\colon\mathbb{C}\to\mathbb{C}}{\uparrow}}{y(x)}=0$$

```
SageMath version 10.7, Release Date: 2025-08-09
sage: from ore_algebra import OreAlgebra
sage: POL.<x> = QQ[]; DOP.<Dx> = OreAlgebra(POL)
sage: (x^2 + 1)*Dx^2 + Dx + 1
```

Rigorous High-Precision Solution of Fuchsian ODEs

$$\overset{a_i \in \mathbb{C}[x]}{\overset{\downarrow}{a_r}}(x)\, y^{(r)}(x) + \cdots + \overset{a_i \in \mathbb{C}[x]}{\overset{\downarrow}{a_1}}(x)\, y'(x) + \overset{a_i \in \mathbb{C}[x]}{\overset{\downarrow}{a_0}}(x)\, \underset{y\colon \mathbb{C} \to \mathbb{C}}{\underset{\uparrow}{y}}(x) = 0$$

Operator notation: $L(y) = 0$ where $L = a_r(x)\, D_x^r + \cdots + a_1(x)\, D + a_0(x)$

```
SageMath version 10.7, Release Date: 2025-08-09
sage: from ore_algebra import OreAlgebra
sage: POL.<x> = QQ[]; DOP.<Dx> = OreAlgebra(POL)
sage: (x^2 + 1)*Dx^2 + Dx + 1
```

Rigorous High-Precision Solution of Fuchsian ODEs

$$\overset{\substack{a_i \in \mathbb{C}[x] \\ \downarrow}}{a_r}(x)\, y^{(r)}(x) + \cdots + \overset{\substack{a_i \in \mathbb{C}[x] \\ \downarrow}}{a_1}(x)\, y'(x) + \overset{\substack{a_i \in \mathbb{C}[x] \\ \downarrow}}{a_0}(x)\, \underset{\substack{\uparrow \\ y\colon \mathbb{C} \to \mathbb{C}}}{y}(x) = 0$$

Operator notation: $L(y) = 0$ where $L = a_r(x)\, D_x^r + \cdots + a_1(x)\, D + a_0(x)$

```
SageMath version 10.7, Release Date: 2025-08-09
sage: from ore_algebra import OreAlgebra
sage: POL.<x> = QQ[]; DOP.<Dx> = OreAlgebra(POL)
sage: (x^2 + 1)*Dx^2 + Dx + 1
```

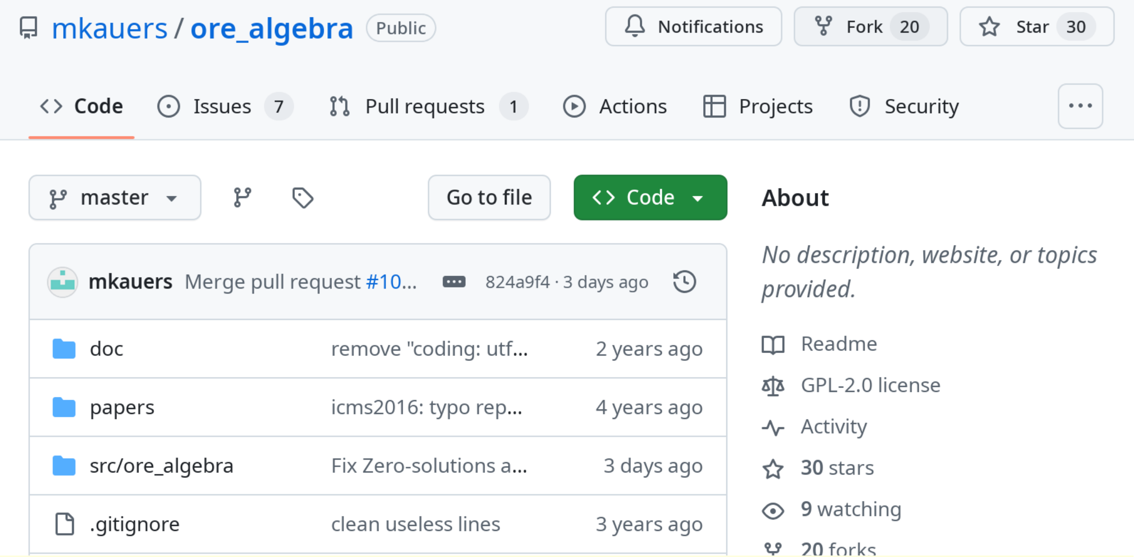
Rigorous High-Precision Solution of Fuchsian ODEs

$$\overset{\substack{a_i \in \mathbb{C}[x] \\ \downarrow}}{a_r(x)}\, y^{(r)}(x) + \cdots + \overset{\substack{a_i \in \mathbb{C}[x] \\ \downarrow}}{a_1(x)}\, y'(x) + \overset{\substack{a_i \in \mathbb{C}[x] \\ \downarrow}}{a_0(x)}\, \underset{\substack{\uparrow \\ y\colon \mathbb{C} \to \mathbb{C}}}{y(x)} = 0$$

Operator notation: $L(y) = 0$ where $L = a_r(x)\, D_x^r + \cdots + a_1(x)\, D + a_0(x)$

```
SageMath version 10.7, Release Date: 2025-08-09
sage: from ore_algebra import OreAlgebra
sage: POL.<x> = QQ[]; DOP.<Dx> = OreAlgebra(POL)
sage: (x^2 + 1)*Dx^2 + Dx + 1
(x^2 + 1)*Dx^2 + Dx + 1
```

Rigorous High-Precision Solution of Fuchsian ODEs



mkauers / ore_algebra Public

About

No description, website, or topics provided.

mkauers Merge pull request #10... 824a9f4 · 3 days ago

doc	remove "coding: utf...	2 years ago
papers	icms2016: typo rep...	4 years ago
src/ore_algebra	Fix Zero-solutions a...	3 days ago
.gitignore	clean useless lines	3 years ago

Readme
GPL-2.0 license
Activity
30 stars
9 watching
20 forks

```
$ sage -pip install --no-build-isolation git+https://github.com/mkauers/ore_algebra.git
```

Rigorous High-Precision **Solution** of Fuchsian ODEs

Theorem. [Cauchy, ~1839]

Assume $a_r(x) \neq 0$ for $|x| < \rho$.

For any choice of $y(0), \ldots, y^{(r-1)}(0) \in \mathbb{C}$, the differential equation

$$a_r(x)\, y^{(r)}(x) + a_{r-1}(x)\, y^{(r-1)}(x) + \cdots + a_0(x)\, y(x) = 0$$

has a unique analytic solution $y\colon \{|x| < \rho\} \to \mathbb{C}$.

Rigorous High-Precision **Solution** of Fuchsian ODEs

Theorem. [Cauchy, ~1839]

Assume $a_r(x) \neq 0$ for $|x| < \rho$.

For any choice of $y(0), \ldots, y^{(r-1)}(0) \in \mathbb{C}$, the differential equation

$$a_r(x)\, y^{(r)}(x) + a_{r-1}(x)\, y^{(r-1)}(x) + \cdots + a_0(x)\, y(x) = 0$$

has a unique analytic solution $y \colon \{|x| < \rho\} \to \mathbb{C}$.

“Local” basis of solutions:

$$\left\{ \begin{array}{rcl} f_0(x) & = & 1 + 0\,x + \cdots + 0\,x^{r-1} + \blacksquare\, x^r + \cdots \\ f_1(x) & = & 0 + 1\,x + \cdots + 0\,x^{r-1} + \blacksquare\, x^r + \cdots \\ & \vdots & \\ f_{r-1}(x) & = & 0 + 0\,x + \cdots + 1\,x^{r-1} + \blacksquare\, x^r + \cdots \end{array} \right.$$

Rigorous High-Precision Solution of Fuchsian ODEs

Theorem. [Cauchy, ~1839]

Assume $a_r(x) \neq 0$ for $|x| < \rho$.

For any choice of $y(0), \ldots, y^{(r-1)}(0) \in \mathbb{C}$, the differential equation

$$a_r(x)\,y^{(r)}(x) + a_{r-1}(x)\,y^{(r-1)}(x) + \cdots + a_0(x)\,y(x) = 0$$

has a unique analytic solution $y\colon \{|x| < \rho\} \to \mathbb{C}$.

“Local” basis of solutions:

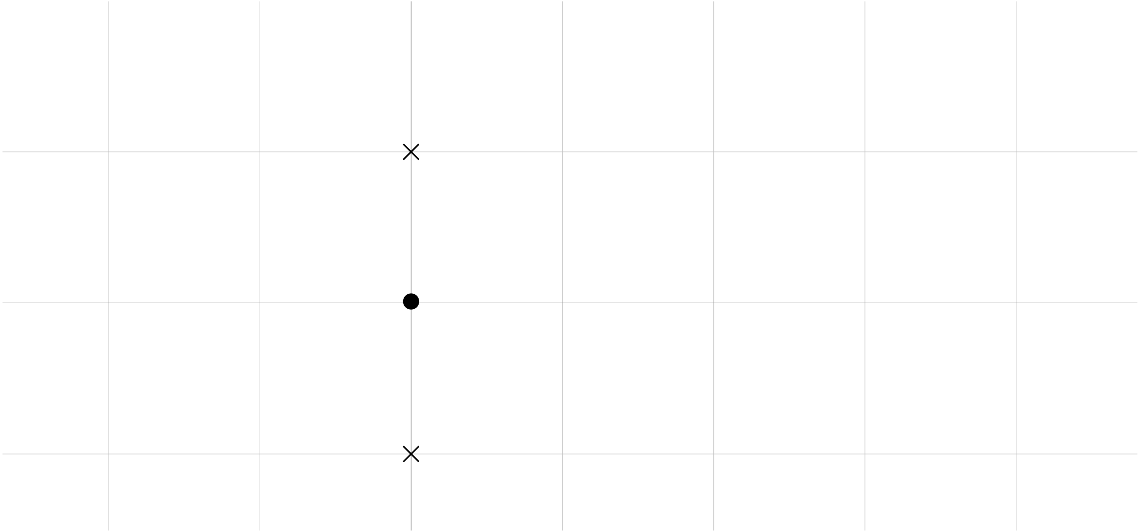
$$\begin{cases} f_0(x) = 1 + 0\,x + \cdots + 0\,x^{r-1} + \blacksquare\,x^r + \cdots \\ f_1(x) = 0 + 1\,x + \cdots + 0\,x^{r-1} + \blacksquare\,x^r + \cdots \\ \quad\vdots \\ f_{r-1}(x) = 0 + 0\,x + \cdots + 1\,x^{r-1} + \blacksquare\,x^r + \cdots \end{cases}$$

Fundamental matrix:

$$F(x) = \begin{pmatrix} f_0(x) & \ldots & f_{r-1}(x) \\ f_0'(x) & & f_{r-1}'(x) \\ \vdots & & \vdots \\ \dfrac{f_0^{(r-1)}(x)}{(r-1)!} & \ldots & \dfrac{f_{r-1}^{(r-1)}(x)}{(r-1)!} \end{pmatrix}$$

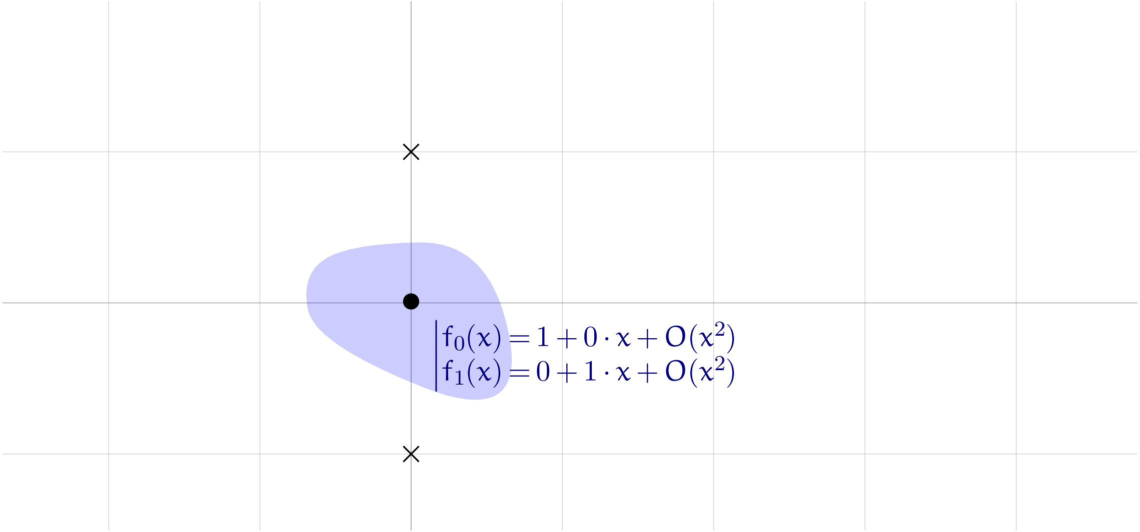
Rigorous High-Precision **Solution** of Fuchsian ODEs

$$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$$



Rigorous High-Precision **Solution** of Fuchsian ODEs

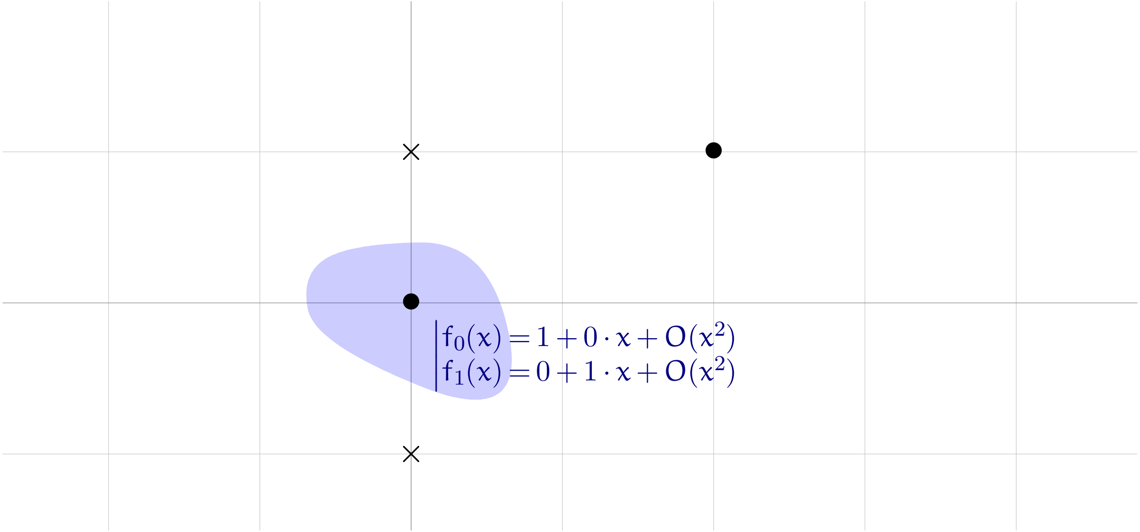
$$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$$



$$\left|\begin{array}{l} f_0(x)=1+0\cdot x+O(x^2) \\ f_1(x)=0+1\cdot x+O(x^2) \end{array}\right.$$

Rigorous High-Precision **Solution** of Fuchsian ODEs

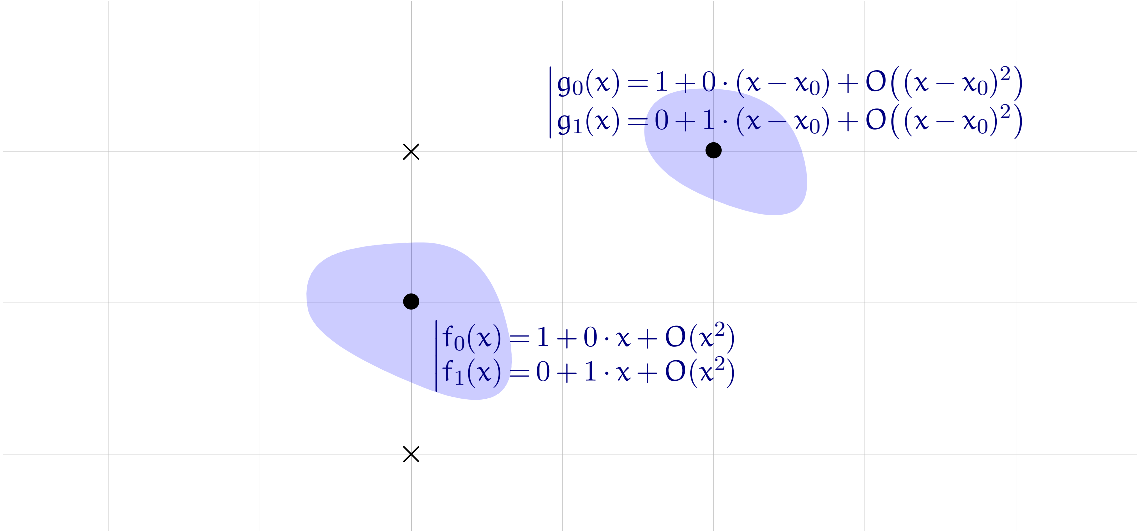
$$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$$



$$\left|\begin{array}{l} f_0(x)=1+0\cdot x+O(x^2) \\ f_1(x)=0+1\cdot x+O(x^2) \end{array}\right.$$

Rigorous High-Precision Solution of Fuchsian ODEs

$$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$$

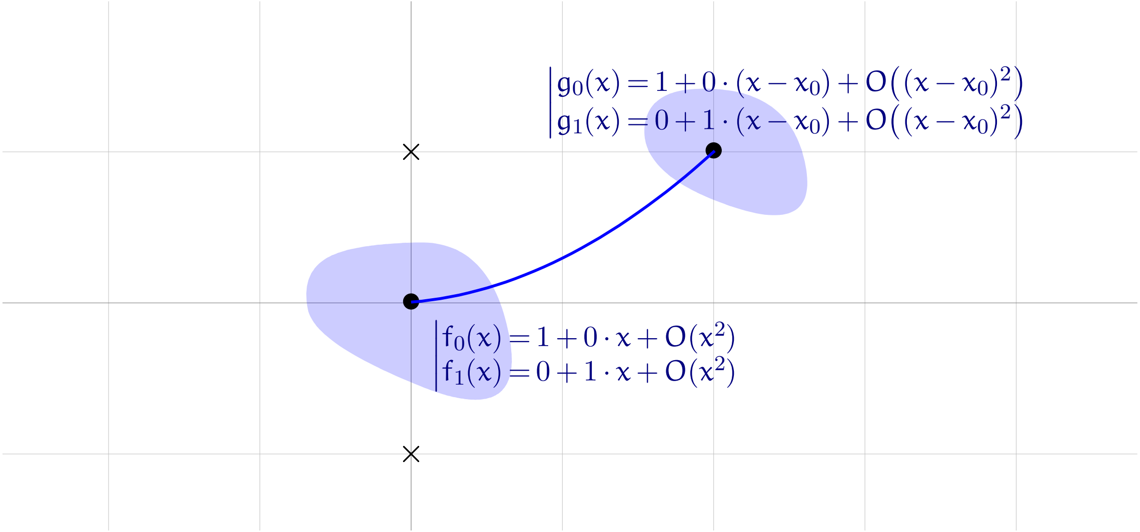


$$\left|\begin{aligned} g_0(x) &= 1 + 0\cdot(x-x_0) + O\big((x-x_0)^2\big) \\ g_1(x) &= 0 + 1\cdot(x-x_0) + O\big((x-x_0)^2\big) \end{aligned}\right.$$

$$\left|\begin{aligned} f_0(x) &= 1 + 0\cdot x + O(x^2) \\ f_1(x) &= 0 + 1\cdot x + O(x^2) \end{aligned}\right.$$

Rigorous High-Precision **Solution** of Fuchsian ODEs

$$(x^2+1)\,y''(x) + 2\,x\,y'(x) = 0$$

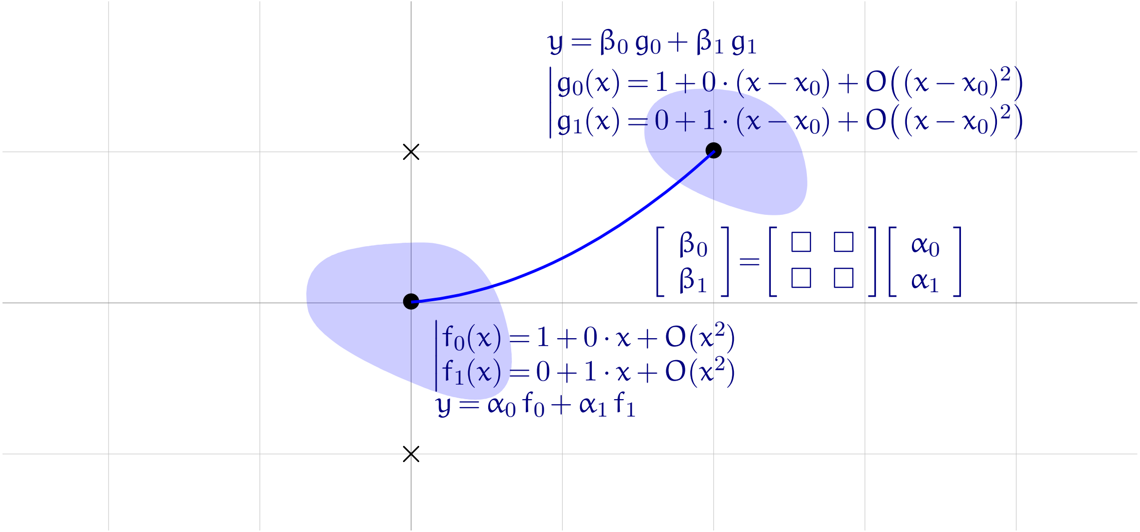


$$\left|\begin{array}{l} g_0(x) = 1 + 0 \cdot (x - x_0) + O\big((x-x_0)^2\big) \\ g_1(x) = 0 + 1 \cdot (x - x_0) + O\big((x-x_0)^2\big) \end{array}\right.$$

$$\left|\begin{array}{l} f_0(x) = 1 + 0 \cdot x + O(x^2) \\ f_1(x) = 0 + 1 \cdot x + O(x^2) \end{array}\right.$$

Rigorous High-Precision Solution of Fuchsian ODEs

$$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$$



$$y=\beta_0\,g_0+\beta_1\,g_1$$

$$\left|\begin{aligned}g_0(x)&=1+0\cdot(x-x_0)+O\big((x-x_0)^2\big)\\ g_1(x)&=0+1\cdot(x-x_0)+O\big((x-x_0)^2\big)\end{aligned}\right.$$

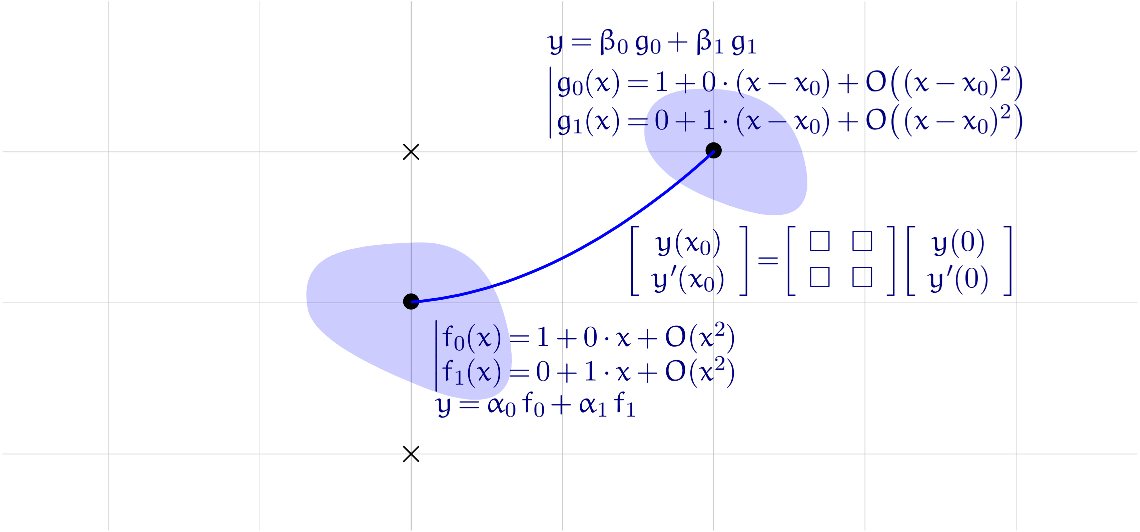
$$\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix}=\begin{bmatrix}\square & \square\\ \square & \square\end{bmatrix}\begin{bmatrix}\alpha_0\\ \alpha_1\end{bmatrix}$$

$$\left|\begin{aligned}f_0(x)&=1+0\cdot x+O(x^2)\\ f_1(x)&=0+1\cdot x+O(x^2)\end{aligned}\right.$$

$$y=\alpha_0\,f_0+\alpha_1\,f_1$$

Rigorous High-Precision **Solution** of Fuchsian ODEs

$$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$$



$$y=\beta_0\,g_0+\beta_1\,g_1$$

$$\left|\begin{array}{l} g_0(x)=1+0\cdot(x-x_0)+O\big((x-x_0)^2\big)\\ g_1(x)=0+1\cdot(x-x_0)+O\big((x-x_0)^2\big)\end{array}\right.$$

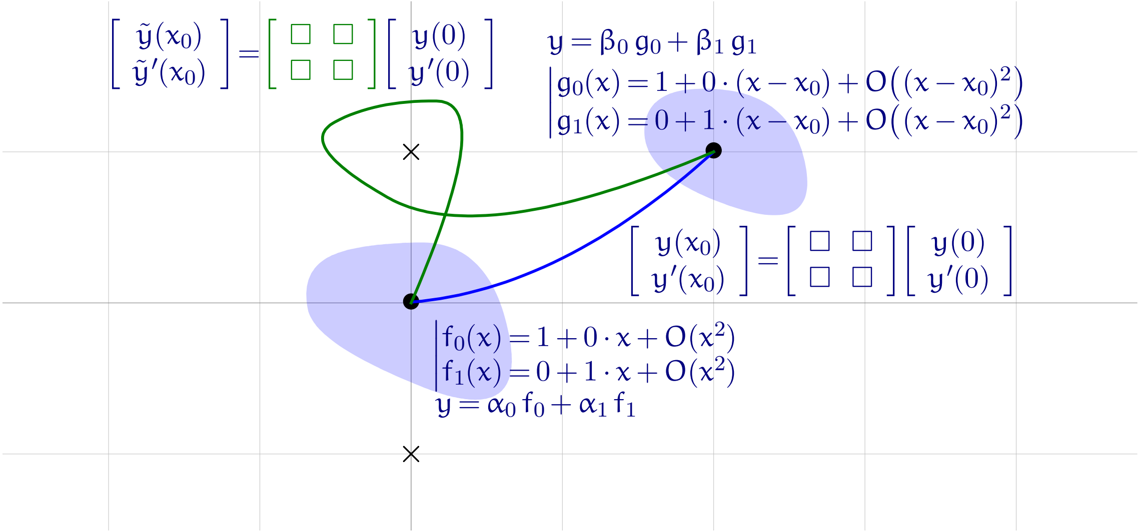
$$\begin{bmatrix} y(x_0)\\ y'(x_0)\end{bmatrix}=\begin{bmatrix}\Box & \Box\\ \Box & \Box\end{bmatrix}\begin{bmatrix} y(0)\\ y'(0)\end{bmatrix}$$

$$\left|\begin{array}{l} f_0(x)=1+0\cdot x+O(x^2)\\ f_1(x)=0+1\cdot x+O(x^2)\end{array}\right.$$

$$y=\alpha_0\,f_0+\alpha_1\,f_1$$

Rigorous High-Precision Solution of Fuchsian ODEs

$$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$$



$$\begin{bmatrix} \tilde{y}(x_0) \\ \tilde{y}'(x_0) \end{bmatrix} = \begin{bmatrix} \square & \square \\ \square & \square \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

$$y = \beta_0\, g_0 + \beta_1\, g_1$$

$$\left| \begin{aligned} g_0(x) &= 1 + 0 \cdot (x - x_0) + O\big((x-x_0)^2\big) \\ g_1(x) &= 0 + 1 \cdot (x - x_0) + O\big((x-x_0)^2\big) \end{aligned} \right.$$

$$\begin{bmatrix} y(x_0) \\ y'(x_0) \end{bmatrix} = \begin{bmatrix} \square & \square \\ \square & \square \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

$$\left| \begin{aligned} f_0(x) &= 1 + 0 \cdot x + O(x^2) \\ f_1(x) &= 0 + 1 \cdot x + O(x^2) \end{aligned} \right.$$

$$y = \alpha_0\, f_0 + \alpha_1\, f_1$$

Rigorous High-Precision Solution of Fuchsian ODEs

```
sage: (Dx - 1).numerical_solution(ini=[1], path=[0,1])
sage: dop = (x^2 + 1)*Dx^2 + 2*x*Dx
sage: dop.numerical_transition_matrix([0, 1])
sage: dop.numerical_solution(ini=[0,1], path=[0,1])
sage: dop.numerical_solution(ini=[0,1], path=[0,2+I])
sage: dop.numerical_solution(ini=[0,1], path=[0,1+I,2*I,I-1,0,2+I])
```

Automatic bounds on truncation errors in series expansions

Interval arithmetic for error propagation

Rigorous High-Precision Solution of Fuchsian ODEs

```
sage: (Dx - 1).numerical_solution(ini=[1], path=[0,1])
[2.718281828459045 +/- 3.57e-17]
sage: dop = (x^2 + 1)*Dx^2 + 2*x*Dx

sage: dop.numerical_transition_matrix([0, 1])

sage: dop.numerical_solution(ini=[0,1], path=[0,1])

sage: dop.numerical_solution(ini=[0,1], path=[0,2+I])

sage: dop.numerical_solution(ini=[0,1], path=[0,1+I,2*I,I-1,0,2+I])
```

Automatic bounds on truncation errors in series expansions

Interval arithmetic for error propagation

Rigorous High-Precision Solution of Fuchsian ODEs

```
sage: (Dx - 1).numerical_solution(ini=[1], path=[0,1])
[2.7182818284590452 +/- 3.57e-17]
sage: dop = (x^2 + 1)*Dx^2 + 2*x*Dx
sage: dop.numerical_transition_matrix([0, 1])
[    [1.0000000000000000 +/- 2e-21] [0.78539816339744831 +/- 3.89e-19]]
[                   [+/- 1.02e-21]  [0.50000000000000000 +/- 2.9e-21]]
sage: dop.numerical_solution(ini=[0,1], path=[0,1])
sage: dop.numerical_solution(ini=[0,1], path=[0,2+I])
sage: dop.numerical_solution(ini=[0,1], path=[0,1+I,2*I,I-1,0,2+I])
```

Automatic bounds on truncation errors in series expansions

Interval arithmetic for error propagation

Rigorous High-Precision Solution of Fuchsian ODEs

```
sage: (Dx - 1).numerical_solution(ini=[1], path=[0,1])
[2.7182818284590452 +/- 3.57e-17]
sage: dop = (x^2 + 1)*Dx^2 + 2*x*Dx
sage: dop.numerical_transition_matrix([0, 1])
[    [1.0000000000000000 +/- 2e-21] [0.78539816339744831 +/- 3.89e-19]]
[                    [+/- 1.02e-21]   [0.5000000000000000 +/- 2.9e-21]]
sage: dop.numerical_solution(ini=[0,1], path=[0,1])
[0.78539816339744831 +/- 5.40e-19]
sage: dop.numerical_solution(ini=[0,1], path=[0,2+I])
sage: dop.numerical_solution(ini=[0,1], path=[0,1+I,2*I,I-1,0,2+I])
```

Automatic bounds on truncation errors in series expansions

Interval arithmetic for error propagation

Rigorous High-Precision Solution of Fuchsian ODEs

```
sage: (Dx - 1).numerical_solution(ini=[1], path=[0,1])
[2.718281828459045 +/- 3.57e-17]
sage: dop = (x^2 + 1)*Dx^2 + 2*x*Dx
sage: dop.numerical_transition_matrix([0, 1])
[    [1.0000000000000000 +/- 2e-21] [0.7853981633974483 +/- 3.89e-19]]
[                    [+/- 1.02e-21]  [0.5000000000000000 +/- 2.9e-21]]
sage: dop.numerical_solution(ini=[0,1], path=[0,1])
[0.7853981633974483 +/- 5.40e-19]
sage: dop.numerical_solution(ini=[0,1], path=[0,2+I])
[1.178097245096172 +/- 3.58e-17] + [0.1732867951399863 +/- 2.68e-18]*I
sage: dop.numerical_solution(ini=[0,1], path=[0,1+I,2*I,I-1,0,2+I])
```

Automatic bounds on truncation errors in series expansions

Interval arithmetic for error propagation

Rigorous High-Precision Solution of Fuchsian ODEs

```
sage: (Dx - 1).numerical_solution(ini=[1], path=[0,1])
[2.718281828459045 +/- 3.57e-17]
sage: dop = (x^2 + 1)*Dx^2 + 2*x*Dx
sage: dop.numerical_transition_matrix([0, 1])
[    [1.0000000000000000 +/- 2e-21] [0.7853981633974483 +/- 3.89e-19]]
[                    [+/- 1.02e-21]  [0.5000000000000000 +/- 2.9e-21]]
sage: dop.numerical_solution(ini=[0,1], path=[0,1])
[0.7853981633974483 +/- 5.40e-19]
sage: dop.numerical_solution(ini=[0,1], path=[0,2+I])
[1.178097245096172 +/- 3.58e-17] + [0.1732867951399863 +/- 2.68e-18]*I
sage: dop.numerical_solution(ini=[0,1], path=[0,1+I,2*I,I-1,0,2+I])
[4.319689898685965 +/- 3.33e-18] + [0.1732867951399863 +/- 3.00e-18]*I
```

Automatic bounds on truncation errors in series expansions

Interval arithmetic for error propagation

Rigorous High-Precision Solution of Fuchsian ODEs

```
sage: (Dx - 1).numerical_solution(ini=[1], path=[0,1], eps=1e-30)
sage: (Dx - 1).numerical_solution(ini=[1], path=[0,1], eps=1e-1000)
```

’Standard’ numerical methods like RK4 need $\Omega(2^n)$ steps for n digits.

High precision requires order-adaptive methods, large steps.

Rigorous **High-Precision** Solution of Fuchsian ODEs

```
sage: (Dx - 1).numerical_solution(ini=[1], path=[0,1], eps=1e-30)
[2.7182818284590452353602874713527 +/- 4.78e-32]
sage: (Dx - 1).numerical_solution(ini=[1], path=[0,1], eps=1e-1000)
```

'Standard' numerical methods like RK4 need $\Omega(2^n)$ steps for n digits.

High precision requires order-adaptive methods, large steps.

Rigorous High-Precision Solution of Fuchsian ODEs

```
sage: (Dx - 1).numerical_solution(ini=[1], path=[0,1], eps=1e-30)
[2.71828182845904523536028747135 +/- 4.78e-32]
sage: (Dx - 1).numerical_solution(ini=[1], path=[0,1], eps=1e-1000)
[2.7182818284590452353602874713526624977572470936999595749669676277240766303535475
9457138217852516642742746639193200305992181741359662904357290033429526059563073813
2328627943490763233829880753195251019011573834187930702154089149934884167509244761
4606680822648001684774118537423454424371075390777449920695517027618386062613313845
8300075204493382656029760673711320070932870912744374704723069697720931014169283681
9025515108657463772111252389784425056953696770785449969967946864454905987931636889
2300987931277361782154249992295763514822082698951936680331825288693984964651058209
3923982948879332036250944311730123819706841614039701983767932068328237646480429531
1802328782509819455815301756717361332069811250996181881593041690351598888519345807
2738667385894228792284998920868058257492796104841984443634632449684875602336248270
4197862320900216099023530436994184914631409343173814364054625315209618369088870701
6768396424378140592714563549061303107208510383750510115747704171898610687396965521
26715468895703503540 +/- 5.53e-1002]
```

'Standard' numerical methods like RK4 need $\Omega(2^n)$ steps for n digits.

High precision requires order-adaptive methods, large steps.

Rigorous High-Precision Solution of **Fuchsian** ODEs

$$L = a_r(x)\, D^r + \cdots + a_0(x)$$

Recall: if $a_r(\xi) \neq 0$, full basis of **analytic** solutions around ξ

Rigorous High-Precision Solution of Fuchsian ODEs

$$L = a_r(x)\, D^r + \cdots + a_0(x)$$

Recall: if $a_r(\xi) \neq 0$, full basis of **analytic** solutions around ξ

Definition. A **regular singular point** of L is a point $\xi \in \mathbb{C}$ where $a_r(\xi) = 0$ but there is a full basis of solutions of the form

$$\overset{\overset{\in\mathbb{C}}{\downarrow}}{(x-\xi)^{\lambda}} \Big(\underset{\uparrow}{f_0}(x) + \underset{\uparrow}{f_1}(x) \log(x-\xi) + \cdots + f_K(x) \log(x-\xi)^{\overset{\overset{\in\mathbb{N}}{\downarrow}}{K}} \Big).$$

convergent power series

Rigorous High-Precision Solution of Fuchsian ODEs

$$L = a_r(x)\,D^r + \cdots + a_0(x)$$

Recall: if $a_r(\xi) \neq 0$, full basis of **analytic** solutions around ξ

Definition. A **regular singular point** of L is a point $\xi \in \mathbb{C}$ where $a_r(\xi) = 0$ but there is a full basis of solutions of the form

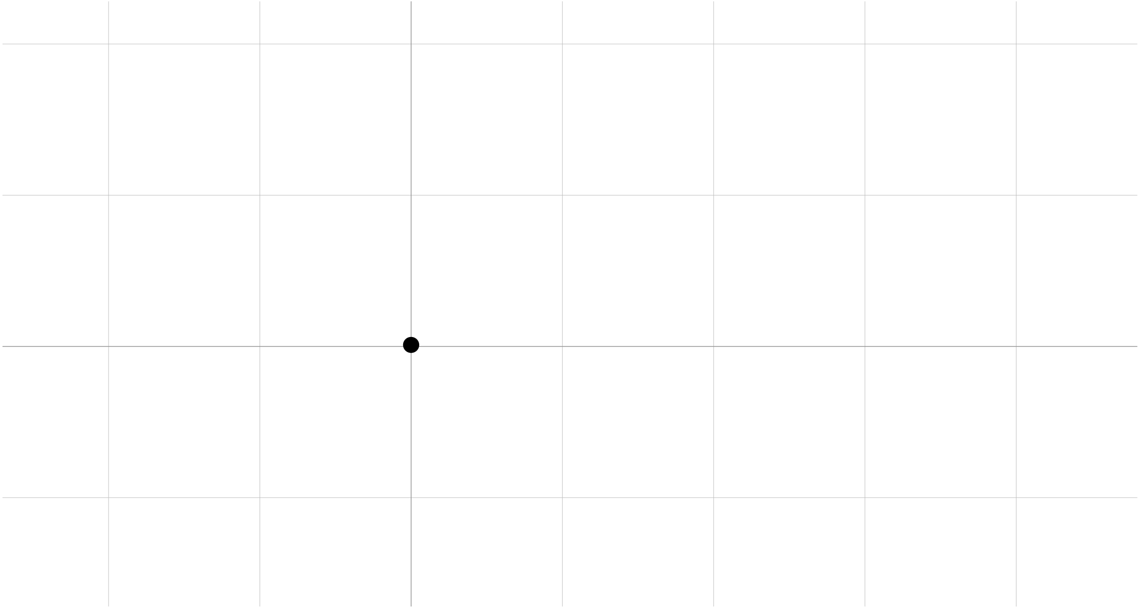
$$(x-\xi)^{\overset{\in\mathbb{C}}{\overset{\downarrow}{\lambda}}}\left(\underset{\uparrow}{f_0(x)} + \underset{\uparrow}{f_1(x)}\log(x-\xi) + \cdots + f_K(x)\log(x-\xi)^{\overset{\in\mathbb{N}}{\overset{\downarrow}{K}}}\right).$$

convergent power series

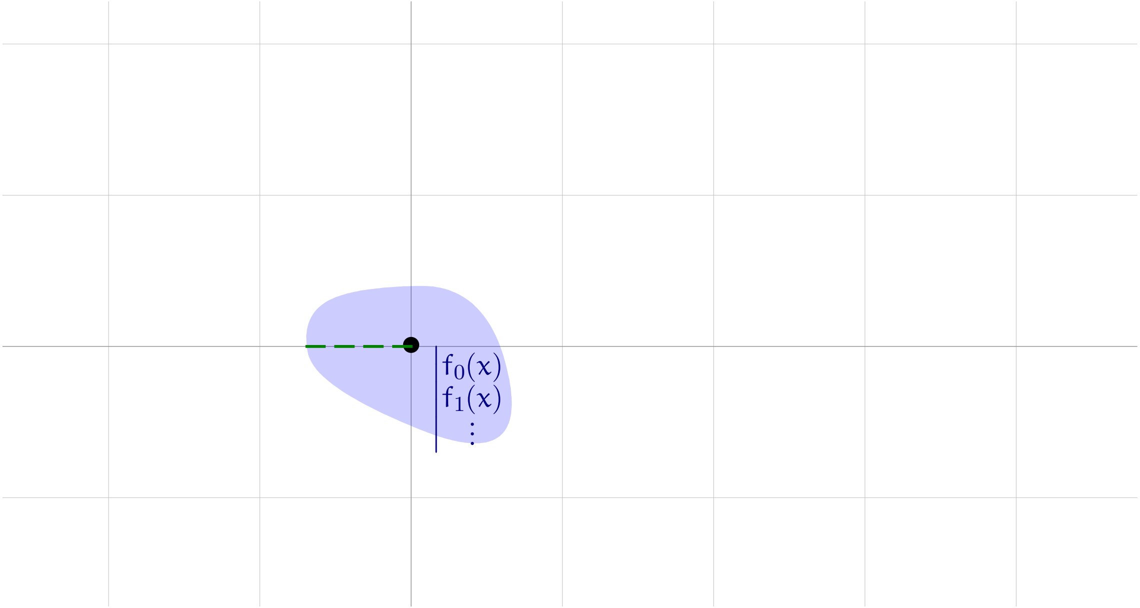
An equation is **Fuchsian** if all its singular points (in $\mathbb{P}^1(\mathbb{C})$) are regular.

Examples: differential equations satisfied by algebraic functions
periods of algebraic varieties

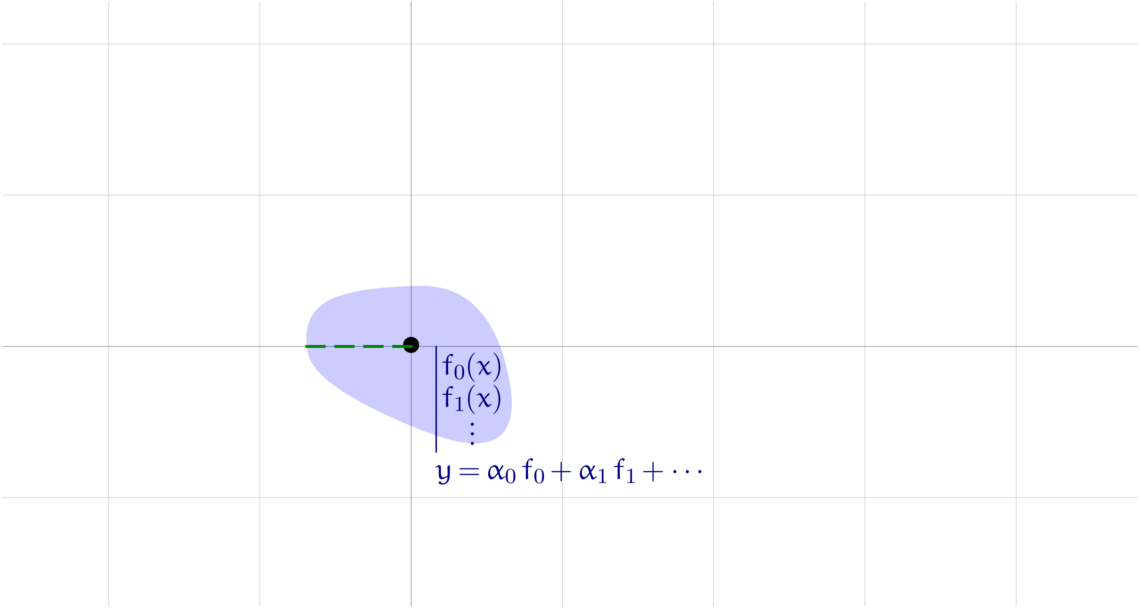
Rigorous High-Precision Solution of **Fuchsian** ODEs



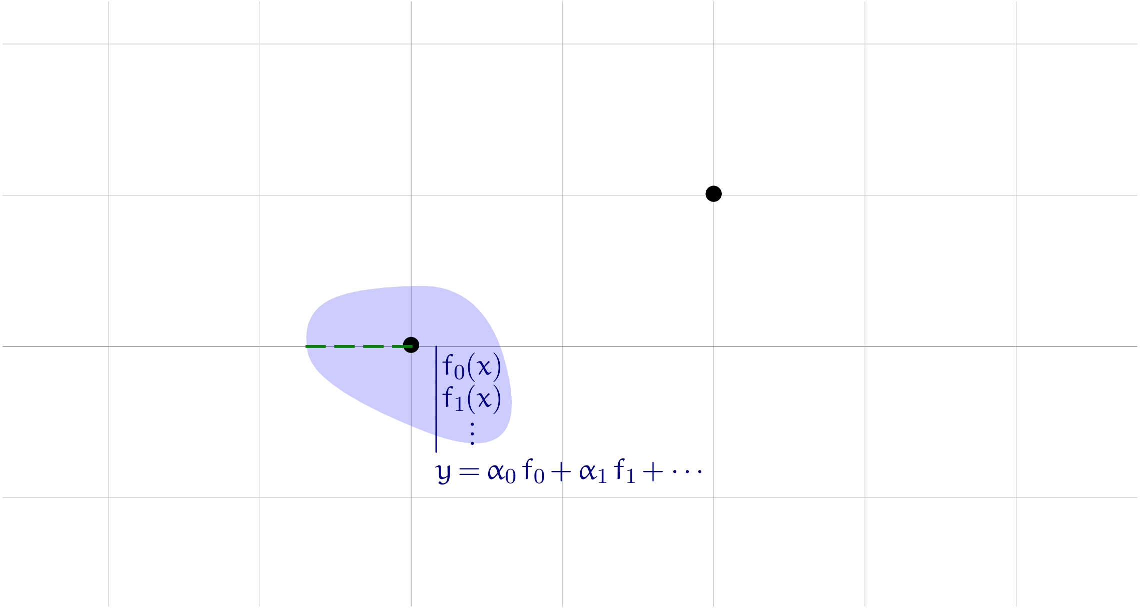
Rigorous High-Precision Solution of **Fuchsian** ODEs



Rigorous High-Precision Solution of **Fuchsian** ODEs

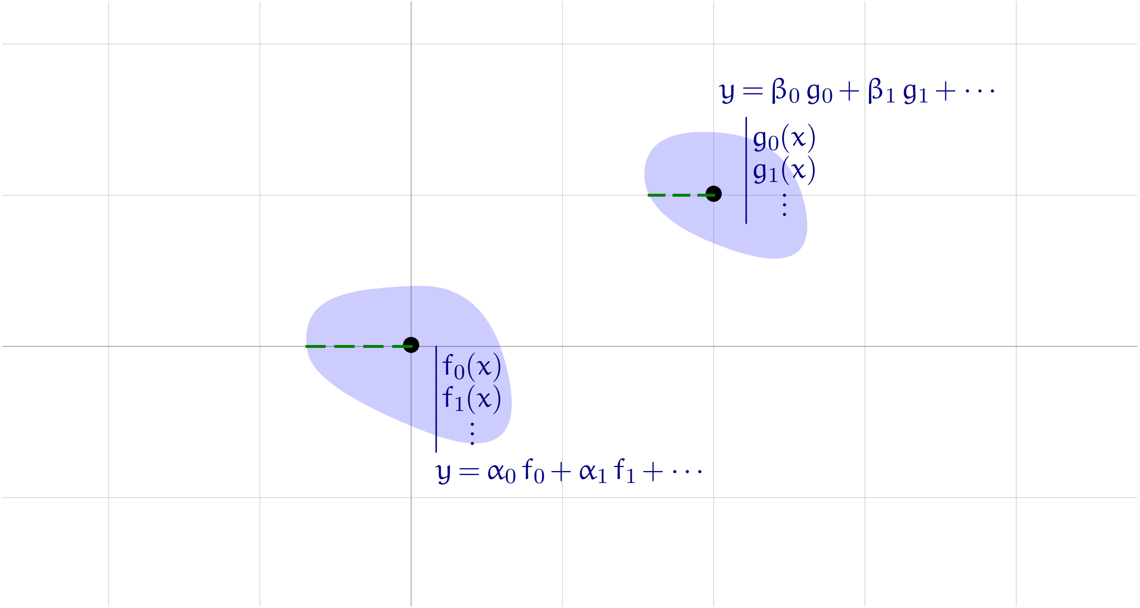


Rigorous High-Precision Solution of **Fuchsian** ODEs



$$y = \alpha_0 \, f_0 + \alpha_1 \, f_1 + \cdots$$

Rigorous High-Precision Solution of **Fuchsian** ODEs



$y = \beta_0\, g_0 + \beta_1\, g_1 + \cdots$

$g_0(x)$

$g_1(x)$

$\vdots$

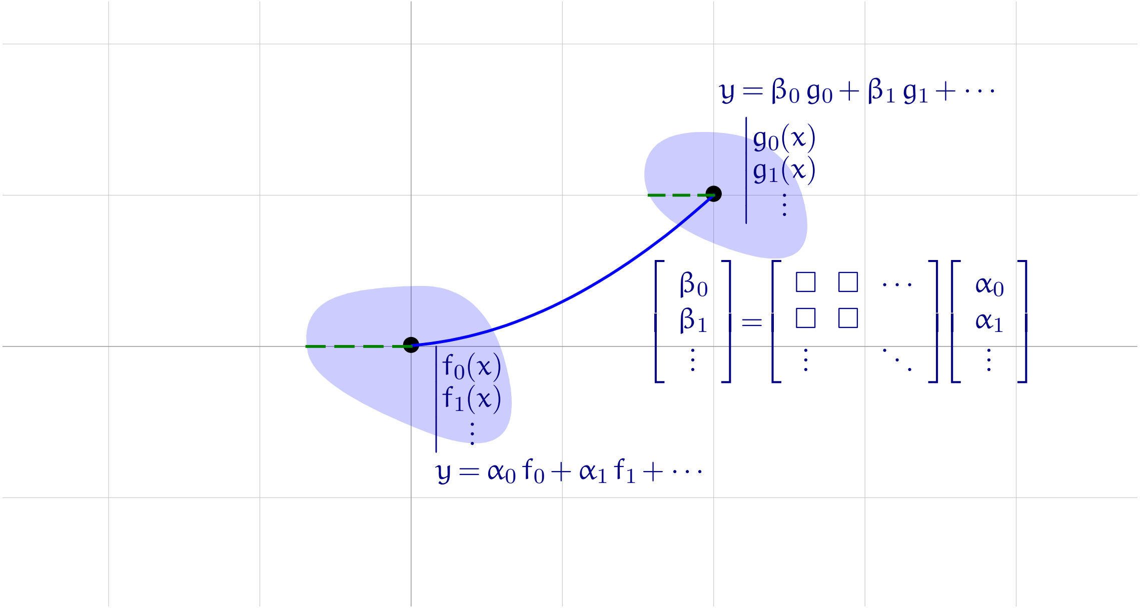
$f_0(x)$

$f_1(x)$

$\vdots$

$y = \alpha_0\, f_0 + \alpha_1\, f_1 + \cdots$

Rigorous High-Precision Solution of **Fuchsian** ODEs



$$y = \beta_0\, g_0 + \beta_1\, g_1 + \cdots$$

$$\begin{array}{|l} g_0(x) \\ g_1(x) \\ \quad\vdots \end{array}$$

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \end{bmatrix} = \begin{bmatrix} \square & \square & \cdots \\ \square & \square & \\ \vdots & & \ddots \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \end{bmatrix}$$

$$\begin{array}{|l} f_0(x) \\ f_1(x) \\ \quad\vdots \end{array}$$

$$y = \alpha_0\, f_0 + \alpha_1\, f_1 + \cdots$$

Rigorous High-Precision Solution of **Fuchsian** ODEs

```
sage: dop = 2*x^2*(x-1)*Dx^2 + 3*x*(x-1)*Dx + x + 1

sage: dop.local_basis_expansions(0)

sage: dop.local_basis_expansions(1)

sage: mat = dop.numerical_transition_matrix([0,1])

sage: [list(row) for row in mat.rows()]
```

Rigorous High-Precision Solution of **Fuchsian** ODEs

```
sage: dop = 2*x^2*(x-1)*Dx^2 + 3*x*(x-1)*Dx + x + 1

sage: dop.local_basis_expansions(0)

[x^(-1) - 2*1 - x - 4/9*x^2 - 11/45*x^3,
 x^(1/2) + 2/5*x^(3/2) + 1/5*x^(5/2) + 16/135*x^(7/2) + 116/1485*x^(9/2)]

sage: dop.local_basis_expansions(1)

sage: mat = dop.numerical_transition_matrix([0,1])

sage: [list(row) for row in mat.rows()]
```

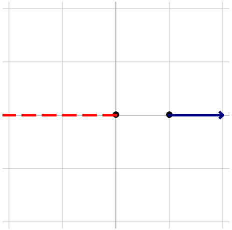
Rigorous High-Precision Solution of **Fuchsian** ODEs

```
sage: dop = 2*x^2*(x-1)*Dx^2 + 3*x*(x-1)*Dx + x + 1

sage: dop.local_basis_expansions(0)

[x^(-1) - 2*1 - x - 4/9*x^2 - 11/45*x^3,
 x^(1/2) + 2/5*x^(3/2) + 1/5*x^(5/2) + 16/135*x^(7/2) + 116/1485*x^(9/2)]

sage: dop.local_basis_expansions(1)

[1 - (x - 1)*log(x - 1) + 5/4*(x - 1)^2*log(x - 1) - 3/8*(x - 1)^2 - 4/3*(x - 1)^3
*log(x - 1) + 67/144*(x - 1)^3 + 395/288*(x - 1)^4*log(x - 1) - 215/432*(x - 1)^4,
 (x - 1) - 5/4*(x - 1)^2 + 4/3*(x - 1)^3 - 395/288*(x - 1)^4]

sage: mat = dop.numerical_transition_matrix([0,1])

sage: [list(row) for row in mat.rows()]
```

Rigorous High-Precision Solution of **Fuchsian** ODEs

```
sage: dop = 2*x^2*(x-1)*Dx^2 + 3*x*(x-1)*Dx + x + 1

sage: dop.local_basis_expansions(0)

[x^(-1) - 2*1 - x - 4/9*x^2 - 11/45*x^3,
 x^(1/2) + 2/5*x^(3/2) + 1/5*x^(5/2) + 16/135*x^(7/2) + 116/1485*x^(9/2)]

sage: dop.local_basis_expansions(1)

[1 - (x - 1)*log(x - 1) + 5/4*(x - 1)^2*log(x - 1) - 3/8*(x - 1)^2 - 4/3*(x - 1)^3
*log(x - 1) + 67/144*(x - 1)^3 + 395/288*(x - 1)^4*log(x - 1) - 215/432*(x - 1)^4,
 (x - 1) - 5/4*(x - 1)^2 + 4/3*(x - 1)^3 - 395/288*(x - 1)^4]

sage: mat = dop.numerical_transition_matrix([0,1])

sage: [list(row) for row in mat.rows()]

[[[-3.506629926405794‍7 +/- 3.63e-17] + [+/- 2.25e-20]*I,
  [2.1726930052195791 +/- 4.40e-17] + [+/- 1.32e-20]*I],
 [[0.6130903679245153‍2 +/- 2.18e-18] + [-11.01640281565456‍2 +/- 6.06e-17]*I,
  [-0.8076293231375796‍1 +/- 1.47e-19] + [6.825716383703759‍9 +/- 4.22e-17]*I]]
```

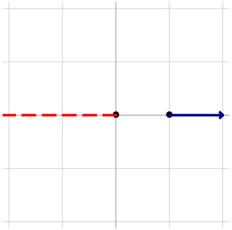
A Toy Example



$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$

```
sage: dop = x*Dx^2 + Dx
sage: dop.local_basis_expansions(1)
sage: dop.numerical_transition_matrix([1, 2])
sage: log(2.)
sage: dop.numerical_transition_matrix([1, -1])
```

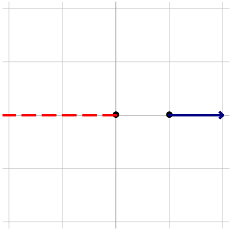
A Toy Example



$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$

```
sage: dop = x*Dx^2 + Dx
sage: dop.local_basis_expansions(1)
[1, (x - 1) - 1/2*(x - 1)^2 + 1/3*(x - 1)^3 - 1/4*(x - 1)^4]
sage: dop.numerical_transition_matrix([1, 2])
sage: log(2.)
sage: dop.numerical_transition_matrix([1, -1])
```

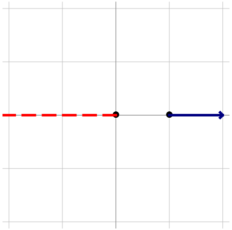
A Toy Example



$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$

```
sage: dop = x*Dx^2 + Dx
sage: dop.local_basis_expansions(1)
[1, (x - 1) - 1/2*(x - 1)^2 + 1/3*(x - 1)^3 - 1/4*(x - 1)^4]
sage: dop.numerical_transition_matrix([1, 2])
[    [1.0000000000000000 +/- 1e-21] [0.6931471805599453 +/- 5.83e-19]]
[                   [+/- 5.28e-29]    [0.5000000000000000 +/- 1e-22]]
sage: log(2.)
sage: dop.numerical_transition_matrix([1, -1])
```

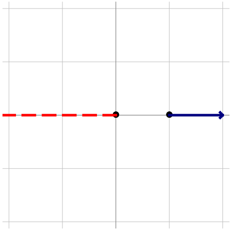
A Toy Example



$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$

```
sage: dop = x*Dx^2 + Dx
sage: dop.local_basis_expansions(1)
[1, (x - 1) - 1/2*(x - 1)^2 + 1/3*(x - 1)^3 - 1/4*(x - 1)^4]
sage: dop.numerical_transition_matrix([1, 2])
[    [1.0000000000000000 +/- 1e-21] [0.6931471805599453 +/- 5.83e-19]]
[                    [+/- 5.28e-29]    [0.5000000000000000 +/- 1e-22]]
sage: log(2.)
0.693147180559945
sage: dop.numerical_transition_matrix([1, -1])
```

A Toy Example



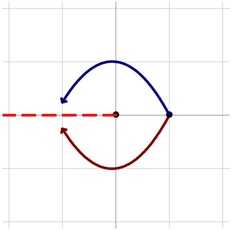
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$

```
sage: dop = x*Dx^2 + Dx
sage: dop.local_basis_expansions(1)
[1, (x - 1) - 1/2*(x - 1)^2 + 1/3*(x - 1)^3 - 1/4*(x - 1)^4]
sage: dop.numerical_transition_matrix([1, 2])
[    [1.0000000000000000 +/- 1e-21] [0.6931471805599453 +/- 5.83e-19]]
[                    [+/- 5.28e-29]     [0.5000000000000000 +/- 1e-22]]
sage: log(2.)
0.693147180559945
sage: dop.numerical_transition_matrix([1, -1])
```

```
ValueError: Step 1 --> -1 passes through or too close to singular point 0 (to
compute the connection to a singular point, make it a vertex of the path)
```

Branches of Log

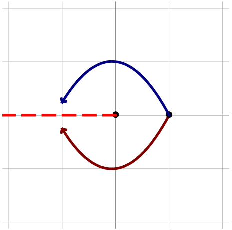
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b \log x$$



```
sage: dop.numerical_transition_matrix([1, I, -1], 1e-5)
sage: dop.numerical_transition_matrix([1, -I, -1], 1e-5)
sage: dop.numerical_transition_matrix([1, I, -1, -I, 1], 1e-5)
```

Branches of Log

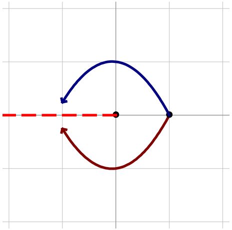
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
sage: dop.numerical_transition_matrix([1, I, -1], 1e-5)
[[1.00000 +/- 2e-10] + [+/- 1.09e-10]*I  [+/- 3.58e-10] + [3.14159 +/- 2.66e-6]*I]
[      [+/- 6.04e-11] + [+/- 6.04e-11]*I [-1.00000 +/- 2.3e-10] + [+/- 2.03e-10]*I]
sage: dop.numerical_transition_matrix([1, -I, -1], 1e-5)

sage: dop.numerical_transition_matrix([1, I, -1, -I, 1], 1e-5)
```

Branches of Log

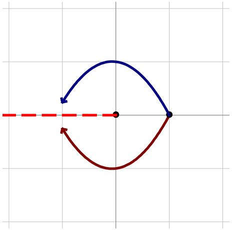
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
sage: dop.numerical_transition_matrix([1, I, -1], 1e-5)
[[1.00000 +/- 2e-10] + [+/- 1.09e-10]*I  [+/- 3.58e-10] + [3.14159 +/- 2.66e-6]*I]
[     [+/- 6.04e-11] + [+/- 6.04e-11]*I [-1.00000 +/- 2.3e-10] + [+/- 2.03e-10]*I]
sage: dop.numerical_transition_matrix([1, -I, -1], 1e-5)
[[1.00000 +/- 2e-10] + [+/- 1.09e-10]*I [+/- 3.58e-10] + [-3.14159 +/- 2.66e-6]*I]
[     [+/- 6.04e-11] + [+/- 6.04e-11]*I [-1.00000 +/- 2.3e-10] + [+/- 2.03e-10]*I]
sage: dop.numerical_transition_matrix([1, I, -1, -I, 1], 1e-5)
```

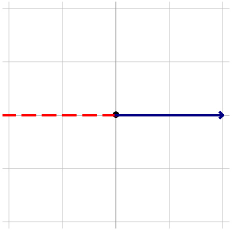
Branches of Log

$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
sage: dop.numerical_transition_matrix([1, I, -1], 1e-5)
[[1.00000 +/- 2e-10] + [+/- 1.09e-10]*I  [+/- 3.58e-10] + [3.14159 +/- 2.66e-6]*I]
[     [+/- 6.04e-11] + [+/- 6.04e-11]*I [-1.00000 +/- 2.3e-10] + [+/- 2.03e-10]*I]
sage: dop.numerical_transition_matrix([1, -I, -1], 1e-5)
[[1.00000 +/- 2e-10] + [+/- 1.09e-10]*I [+/- 3.58e-10] + [-3.14159 +/- 2.66e-6]*I]
[     [+/- 6.04e-11] + [+/- 6.04e-11]*I [-1.00000 +/- 2.3e-10] + [+/- 2.03e-10]*I]
sage: dop.numerical_transition_matrix([1, I, -1, -I, 1], 1e-5)
[[1.00000 +/- 1.1e-9] + [+/- 1.10e-9]*I [+/- 4.00e-9] + [6.28319 +/- 4.70e-6]*I]
[     [+/- 5.61e-10] + [+/- 5.61e-10]*I [1.00000 +/- 2.09e-9] + [+/- 2.07e-9]*I]
```

Connection to Singular Points

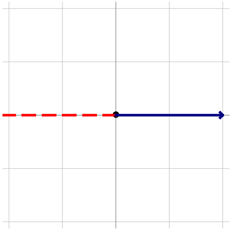


$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$

```
sage: dop.local_basis_expansions(0)
sage: dop.numerical_transition_matrix([0, 2])
sage: dop.numerical_transition_matrix([1, 0])
sage: dop.local_basis_expansions(1)
```

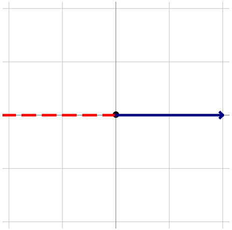
Connection to Singular Points

$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
sage: dop.local_basis_expansions(0)
[log(x), 1]
sage: dop.numerical_transition_matrix([0, 2])
sage: dop.numerical_transition_matrix([1, 0])
sage: dop.local_basis_expansions(1)
```

Connection to Singular Points

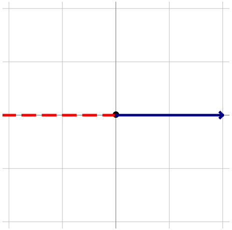


$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$

```
sage: dop.local_basis_expansions(0)
[log(x), 1]
sage: dop.numerical_transition_matrix([0, 2])
[[0.6931471805599453 +/- 5.83e-19]                  1.0000000000000000]
[                0.5000000000000000                                  0]
sage: dop.numerical_transition_matrix([1, 0])

sage: dop.local_basis_expansions(1)
```

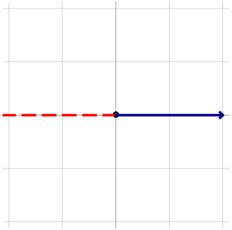
Connection to Singular Points

$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
sage: dop.local_basis_expansions(0)
[log(x), 1]
sage: dop.numerical_transition_matrix([0, 2])
[[0.69314718055994531 +/- 5.83e-19]                     1.0000000000000000]
[                0.50000000000000000                                     0]
sage: dop.numerical_transition_matrix([1, 0])
[                  0 1.0000000000000000]
[1.0000000000000000                   0]
sage: dop.local_basis_expansions(1)
```

Connection to Singular Points



$$x\,y''(x)+y'(x)=0 \qquad y(x)=a+b\log x$$

```
sage: dop.local_basis_expansions(0)
[log(x), 1]
sage: dop.numerical_transition_matrix([0, 2])
[[0.6931471805599453 +/- 5.83e-19]                  1.000000000000000]
[                 0.5000000000000000                                 0]
sage: dop.numerical_transition_matrix([1, 0])
[                 0 1.000000000000000]
[1.000000000000000                  0]
sage: dop.local_basis_expansions(1)
[1, (x - 1) - 1/2*(x - 1)^2 + 1/3*(x - 1)^3 - 1/4*(x - 1)^4]
```

Summary So Far

Input.

- a differential operator $L = a_r(x)\,D^r + \cdots + a_0(x)$ with $a_i \in \mathbb{C}[x]$,
- a path in $\mathbb{C} \setminus \{\text{singular points}\}$ but possibly joining regular singular points,
- a working precision ε,

Output.

connection matrix

entries represented as intervals containing the mathematical value

intervals → points as $\varepsilon \to 0$

SageMath code available (GNU GPL)

Iterated Integrals

[Ablinger, Blümlein, Raab, Schneider, 2014]

$$\int_0^1 \frac{x_5\,dx_5}{x_5-1} \int_{x_5}^1 \frac{dx_4}{x_4\sqrt{x_4-\frac{1}{4}}} \int_{x_4}^1 \frac{dx_3}{x_3\sqrt{x_3-\frac{1}{4}}} \int_{x_3}^1 \frac{dx_2}{1-x_2} \int_{x_2}^1 \frac{dx_1}{1-x_1} = ?$$

(with suitable branch choices)

```
sage: from ore_algebra.examples.iint import diffop, h, w, f, iint_value
sage: dop = diffop([h[0], w[8], w[8], f[1], f[1]])
sage: ini = [0, 0, 0, 1/3, -2/3-i*pi/3, 11/12+2*i*pi/3-pi^2/6]
sage: iint_value(dop, ini, 1e-450)
```

Iterated Integrals

[Ablinger, Blümlein, Raab, Schneider, 2014]

$$I(x) = \int_x^1 \frac{x_5\,dx_5}{x_5 - 1} \int_{x_5}^1 \frac{dx_4}{x_4\sqrt{x_4 - \frac{1}{4}}} \int_{x_4}^1 \frac{dx_3}{x_3\sqrt{x_3 - \frac{1}{4}}} \int_{x_3}^1 \frac{dx_2}{1 - x_2} \int_{x_2}^1 \frac{dx_1}{1 - x_1} = ?$$

(with suitable branch choices)

$$\begin{aligned}
&(4\,x^9 - 13\,x^8 + 15\,x^7 - 7\,x^6 + x^5)\,I^{(6)}(x) + (54\,x^8 - 140\,x^7 + 120\,x^6 - 36\,x^5 + 2\,x^4)\,I^{(5)}(x)\\
+\,&(54\,x^8 - 140\,x^7 + 120\,x^6 - 36\,x^5 + 2\,x^4)\,I^{(5)}(x) + (222\,x^6 - 303\,x^5 + 90\,x^4 + 3\,x^3 - 3\,x^2)\,I^{(3)}(x)\\
+\,&(48\,x^5 - 37\,x^4 + x^3 - 6\,x^2 + 6\,x)\,I''(x) + (-x^2 + 6\,x - 6)\,I'(x) = 0
\end{aligned}$$

```
sage: from ore_algebra.examples.iint import diffop, h, w, f, iint_value
sage: dop = diffop([h[0], w[8], w[8], f[1], f[1]])
sage: ini = [0, 0, 0, 1/3, -2/3-i*pi/3, 11/12+2*i*pi/3-pi^2/6]
sage: iint_value(dop, ini, 1e-450)
```

Iterated Integrals

[Ablinger, Blümlein, Raab, Schneider, 2014]

$$I(x) = \int_x^1 \frac{x_5\,dx_5}{x_5-1} \int_{x_5}^1 \frac{dx_4}{x_4\sqrt{x_4-\frac{1}{4}}} \int_{x_4}^1 \frac{dx_3}{x_3\sqrt{x_3-\frac{1}{4}}} \int_{x_3}^1 \frac{dx_2}{1-x_2} \int_{x_2}^1 \frac{dx_1}{1-x_1} = ?$$

(with suitable branch choices)

$$\begin{aligned} &(4\,x^9-13\,x^8+15\,x^7-7\,x^6+x^5)\,I^{(6)}(x)+(54\,x^8-140\,x^7+120\,x^6-36\,x^5+2\,x^4)\,I^{(5)}(x)\\ +&(54\,x^8-140\,x^7+120\,x^6-36\,x^5+2\,x^4)\,I^{(5)}(x)+(222\,x^6-303\,x^5+90\,x^4+3\,x^3-3\,x^2)\,I^{(3)}(x)\\ +&(48\,x^5-37\,x^4+x^3-6\,x^2+6\,x)\,I''(x)+(-x^2+6\,x-6)\,I'(x)=0 \end{aligned}$$

```
sage: from ore_algebra.examples.iint import diffop, h, w, f, iint_value
sage: dop = diffop([h[0], w[8], w[8], f[1], f[1]])
sage: ini = [0, 0, 0, 1/3, -2/3-i*pi/3, 11/12+2*i*pi/3-pi^2/6]
sage: iint_value(dop, ini, 1e-450)
```

Iterated Integrals

[Ablinger, Blümlein, Raab, Schneider, 2014]

$$I(x) = \int_x^1 \frac{x_5\,dx_5}{x_5 - 1} \int_{x_5}^1 \frac{dx_4}{x_4\sqrt{x_4 - \frac{1}{4}}} \int_{x_4}^1 \frac{dx_3}{x_3\sqrt{x_3 - \frac{1}{4}}} \int_{x_3}^1 \frac{dx_2}{1 - x_2} \int_{x_2}^1 \frac{dx_1}{1 - x_1} = ?$$

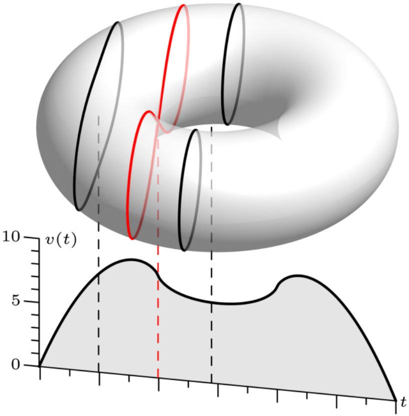
(with suitable branch choices)

$$\begin{aligned}
&(4\,x^9 - 13\,x^8 + 15\,x^7 - 7\,x^6 + x^5)\,I^{(6)}(x) + (54\,x^8 - 140\,x^7 + 120\,x^6 - 36\,x^5 + 2\,x^4)\,I^{(5)}(x) \\
&+ (54\,x^8 - 140\,x^7 + 120\,x^6 - 36\,x^5 + 2\,x^4)\,I^{(5)}(x) + (222\,x^6 - 303\,x^5 + 90\,x^4 + 3\,x^3 - 3\,x^2)\,I^{(3)}(x) \\
&+ (48\,x^5 - 37\,x^4 + x^3 - 6\,x^2 + 6\,x)\,I''(x) + (-x^2 + 6\,x - 6)\,I'(x) = 0
\end{aligned}$$

```
sage: from ore_algebra.examples.iint import diffop, h, w, f, iint_value
sage: dop = diffop([h[0], w[8], w[8], f[1], f[1]])
sage: ini = [0, 0, 0, 1/3, -2/3-i*pi/3, 11/12+2*i*pi/3-pi^2/6]
sage: iint_value(dop, ini, 1e-450)
[0.9708046956249312405601198753795434484693323352038280840782977209392206854324710
4239693854364491332569490848002229625708401731986330961511387598035782070940314675
5465965635142182477075724871974708848482417024966620766917318410552708526685622051
3768662261407821735674025074740233664153471278995228943701841129875878281962011936
9482802194121377017892861103337273067129383060974585592538401278094630392014977458
05231094292565735252931676116881079814401169 +/- 9.45e-452]
```

Volumes of Semi-Algebraic Sets

[Lairez, M., Safey El Din, 2019]



For a compact s.a. set V:

- The **“slice volume”** function satisfies a Picard-Fuchs equation
- Except at critical values of the projection, it is analytic

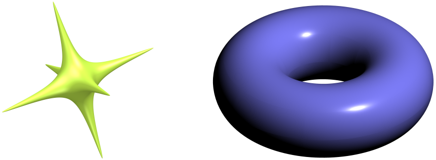
💡 Compute initial values by recursive calls, integrate the equation

If V defined by $O(1)$ polynomials of degree $O(1)$:

- Grid: $\tilde{O}(2^{p\,d})$
- Monte-Carlo: $2^{\Omega(p)}$ (uncertified)
- Iterated quadrature: $\tilde{O}(p^{d})$ (plausibly)
- Differential equation: $\tilde{O}(2^{O(d^2)}\,p)$

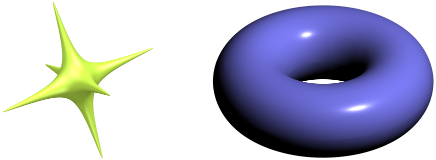
$d =$ ambient dimension, $p =$ precision

Volumes of Semi-Algebraic Sets



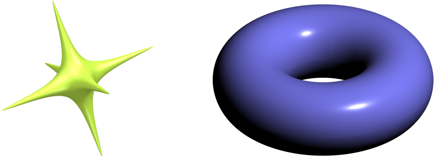
```
sage: import volume  # partial implementation with many limitations

sage: P.<X,Y,Z> = QQ[]

sage: volume.volume1(X^2+Y^2+Z^2-(1 - 2^10*(X^2*Y^2+Y^2*Z^2+Z^2*X^2)), 1000)

sage: volume.volume1((X^2+Y^2+Z^2+3)^2 - 16 * (X^2+Y^2), 1000)

sage: n((2*pi)^2)
```

Volumes of Semi-Algebraic Sets



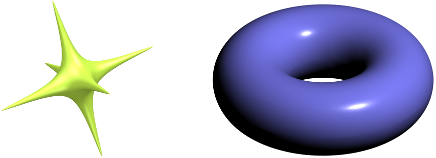
```
sage: import volume  # partial implementation with many limitations

sage: P.<X,Y,Z> = QQ[]

sage: volume.volume1(X^2+Y^2+Z^2-(1 - 2^10*(X^2*Y^2+Y^2*Z^2+Z^2*X^2)), 1000)
[0.10857542146036093773950339599420761981091787444660747544447582299328536067303292
819494347441406406613662423462795980877810349323467815682039679678371920510986067
27381212801675342463658744569084990812694753049172281973755844219253394024113105541
4354736564257531342253838423061375866757194540902 +/- 5.36e-295]

sage: volume.volume1((X^2+Y^2+Z^2+3)^2 - 16 * (X^2+Y^2), 1000)

sage: n((2*pi)^2)
```

Volumes of Semi-Algebraic Sets



```
sage: import volume  # partial implementation with many limitations

sage: P.<X,Y,Z> = QQ[]

sage: volume.volume1(X^2+Y^2+Z^2-(1 - 2^10*(X^2*Y^2+Y^2*Z^2+Z^2*X^2)), 1000)

[0.10857542146036093773950339599420761981091787444660747544447582299328536067303292
2819494347441406406613662423462795980877810349323467815682039679678371920510986067
2738121280167534246365874456908499081269475304917228197375584421925339402411310554
143547365642575313422538384230613758667571945409402 +/- 5.36e-295]

sage: volume.volume1((X^2+Y^2+Z^2+3)^2 - 16 * (X^2+Y^2), 1000)

[39.47841760435743447533796399950460454125479762896316250565339750488017928967682
0972007093614874208927296103655096092576311089392488120186910442470711940790643961
5994248263025222860241649313613151234774110773686597586662860761781549410471176552
810330423267736500623681932392754930801323050666844 +/- 2.86e-293]

sage: n((2*pi)^2)
```

Volumes of Semi-Algebraic Sets

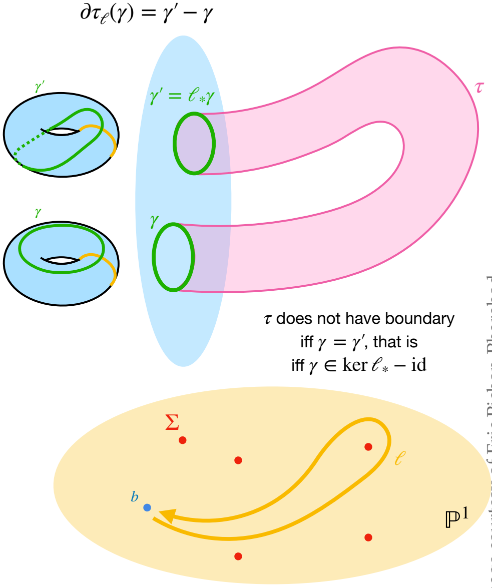


```
sage: import volume  # partial implementation with many limitations

sage: P.<X,Y,Z> = QQ[]

sage: volume.volume1(X^2+Y^2+Z^2-(1 - 2^10*(X^2*Y^2+Y^2*Z^2+Z^2*X^2)), 1000)

[0.1085754214603609377395033959942076198109178744466074754444758229932853606730329
2819494347441406406613662423462795980877810349323467815682039679678371920510986067
2738121280167534246365874456908499081269475304917228197375584421925339402411310554
143547365642575313422538384230613758667571945409402 +/- 5.36e-295]

sage: volume.volume1((X^2+Y^2+Z^2+3)^2 - 16 * (X^2+Y^2), 1000)

[39.478417604357434475337963999504604541254797628963162505653397504880179289676820
9720070936148742089272961036550960925763110893924881201869104424707119407906439615
9942482630252228602416493136131512347741107736865975866628607617815494104711765528
10330423267736500623681932392754930801323050666844 +/- 2.86e-293]

sage: n((2*pi)^2)

39.4784176043574
```

Periods of Algebraic Varieties

- Computation based on **Picard-Fuchs differential equations** with polynomial coefficients
- E.g., follow a basis of solutions along loops in $\mathbb{C}$
- Used to reconstruct **exact geometric data** (homology…)

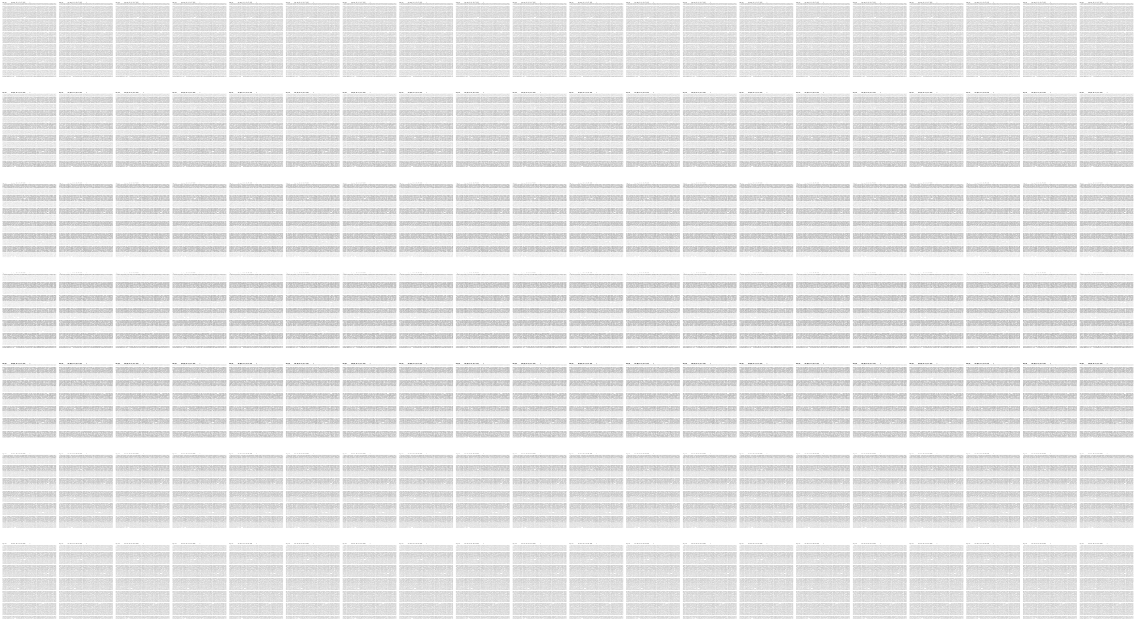
[Chudnovsky, Chudnovsky, 1990;
Sertöz, 2018;
Lairez, Sertöz, 2019, 2020;
Lairez, Pichon-Pharabod, Vanhove, 2024; …]



τ does not have boundary
iff $\gamma = \gamma'$, that is
iff $\gamma \in \ker \ell_* - \mathrm{id}$

image courtesy of Eric Pichon-Pharabod

Equations can get large



Sequence Positivity via Asymptotics

- **Yu–Chen** (2022): uniqueness problem related to minimal surfaces

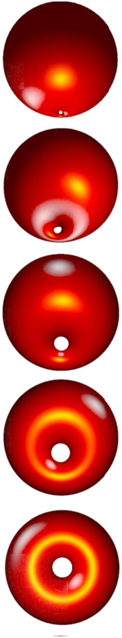
  ↓

  complete positivity of a sequence

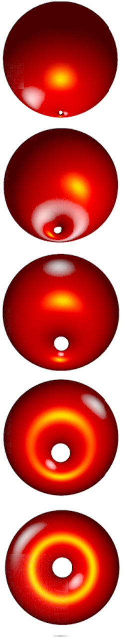
$$(d_n) = \left(72, 1932, 31248, \frac{790101}{2}, \frac{17208645}{4}, \frac{338898609}{8}, \dots\right)$$

given by an **explicit recurrence relation** (equivalently, ODE)

$$\boxed{\cdots}\, d_{n+6} + \cdots + \boxed{\cdots}\, d_{n+1} + \boxed{\cdots}\, d_n = 0$$



Sequence Positivity via Asymptotics



- **Yu–Chen** (2022): uniqueness problem related to minimal surfaces

  ↓

  complete positivity of a sequence

$$(d_n) = \left(72, 1932, 31248, \frac{790101}{2}, \frac{17208645}{4}, \frac{338898609}{8}, \dots\right)$$

  given by an **explicit recurrence relation** (equivalently, ODE)

$$\boxed{\cdots}\; d_{n+6} + \cdots + \boxed{\cdots}\; d_{n+1} + \boxed{\cdots}\; d_n = 0$$

- Routine calculation: $d_n \sim (\mathbf{C} + o(1)) \cdot (\sqrt{2}+1)^{2n} n^3 \ln n$

Sequence Positivity via Asymptotics

- **Yu–Chen** (2022): uniqueness problem related to minimal surfaces

  $\downarrow$

  complete positivity of a sequence

  $$(d_n) = \left(72, 1932, 31248, \frac{790101}{2}, \frac{17208645}{4}, \frac{338898609}{8}, \ldots\right)$$

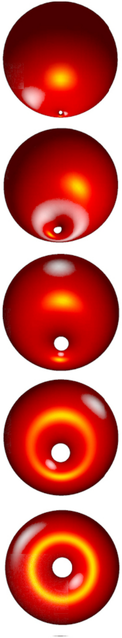
  given by an **explicit recurrence relation** (equivalently, ODE)

  $$\boxed{\cdots}\; d_{n+6} + \cdots + \boxed{\cdots}\; d_{n+1} + \boxed{\cdots}\; d_n = 0$$

- Routine calculation: $d_n \sim (\mathbf{C} + o(1)) \cdot (\sqrt{2}+1)^{2n} n^3 \ln n$
- **Melczer–M.** (2022):

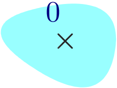
  $$d_n \geqslant (\sqrt{2}+1)^{2n} (8.07\, n^3 \ln n + 1.37\, n^3 - 1196\, n^2 \ln^2 n), \qquad n \geqslant 1000$$

  Corollary: $d_n > 0$ for $n \geqslant 1000$.



Effective Singularity Analysis

$$\mathrm{d}(z) = 72 + 1932\, z + 31248\, z^2 + \cdots$$



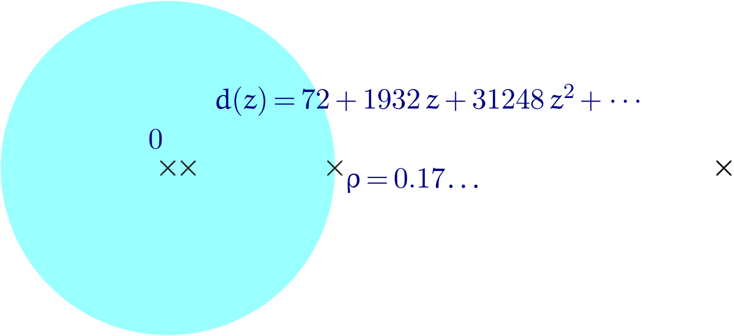
Effective Singularity Analysis

$$\mathrm{d}(z) = 72 + 1932\,z + 31248\,z^2 + \cdots$$

Effective Singularity Analysis

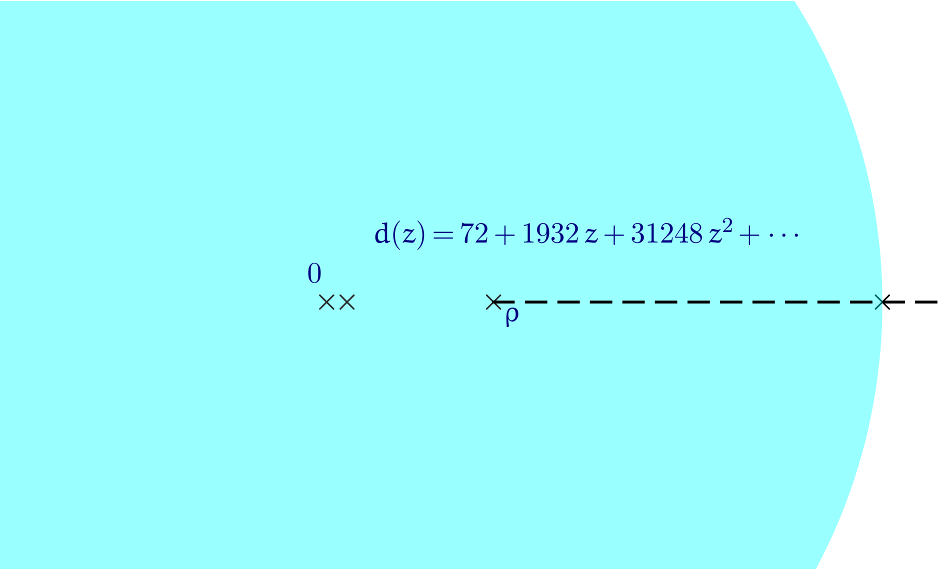
$$\mathrm{d}(z) = 72 + 1932\, z + 31248\, z^2 + \cdots$$

Effective Singularity Analysis



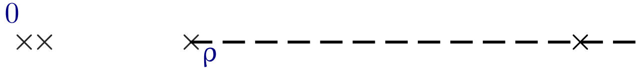
$$\mathrm{d}(z) = 72 + 1932\,z + 31248\,z^2 + \cdots$$

Effective Singularity Analysis



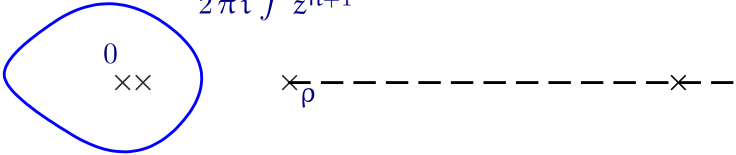
$$\mathrm{d}(z) = 72 + 1932\, z + 31248\, z^2 + \cdots$$

Effective Singularity Analysis

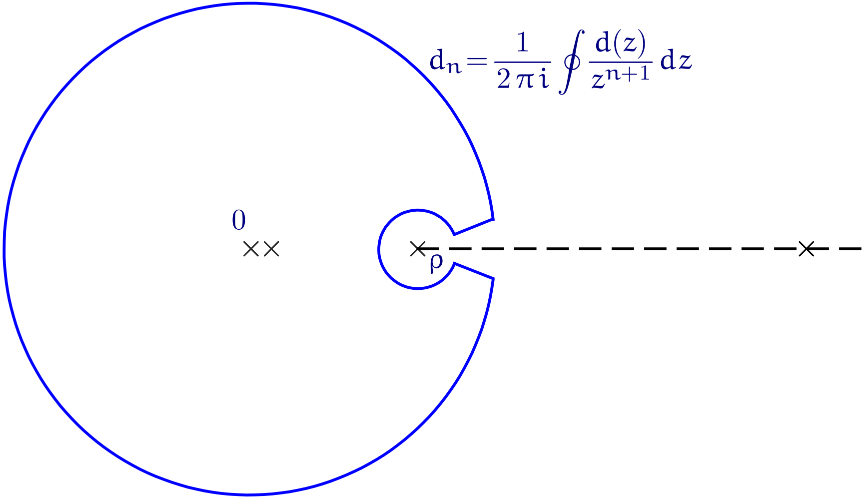


Effective Singularity Analysis

$$d_n = \frac{1}{2\pi i} \oint \frac{d(z)}{z^{n+1}}\, dz$$

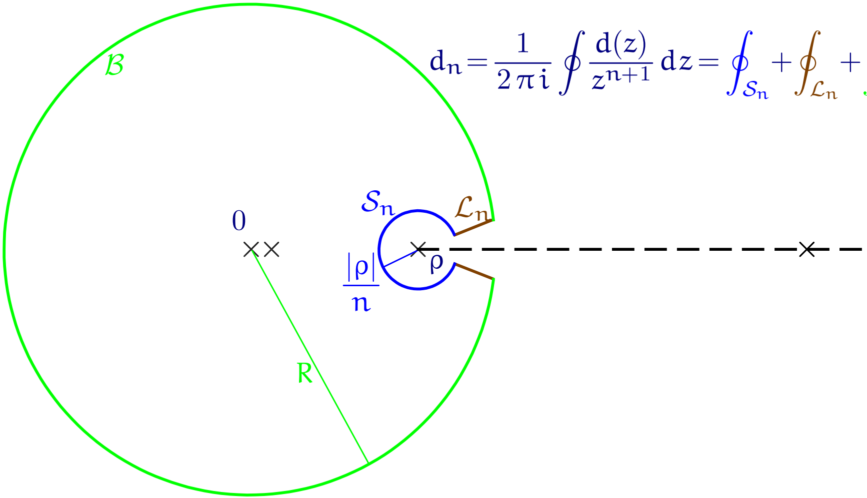


Effective Singularity Analysis



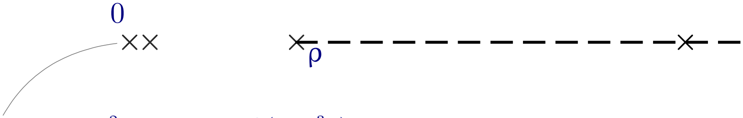
$$d_n = \frac{1}{2\pi i}\oint \frac{d(z)}{z^{n+1}}\,dz$$

Effective Singularity Analysis



$$d_n = \frac{1}{2\pi i}\oint \frac{d(z)}{z^{n+1}}\,dz = \oint_{\mathcal{S}_n} + \oint_{\mathcal{L}_n} + \oint_{\mathcal{B}}$$

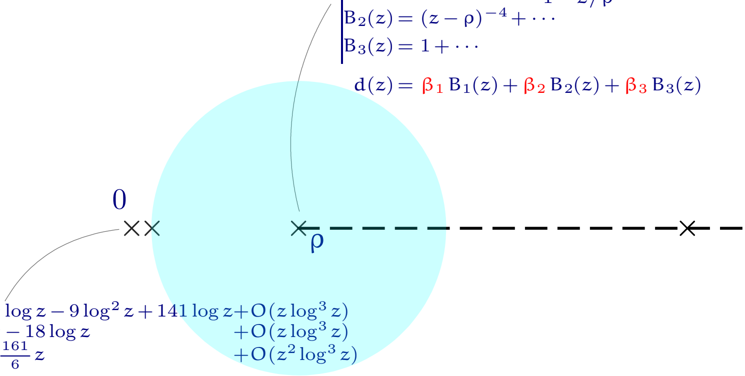
Effective Singularity Analysis



$$\left|\begin{array}{ll} A_1(z)=z^{-1}\log z-9\log^2 z+141\log z & +O(z\log^3 z)\\ A_2(z)=z^{-1}-18\log z & +O(z\log^3 z)\\ A_3(z)=1+\frac{161}{6}z & +O(z^2\log^3 z) \end{array}\right.$$

$$d(z)=0\cdot A_1(z)+0\cdot A_2(z)+72\,A_3(z)$$

Effective Singularity Analysis



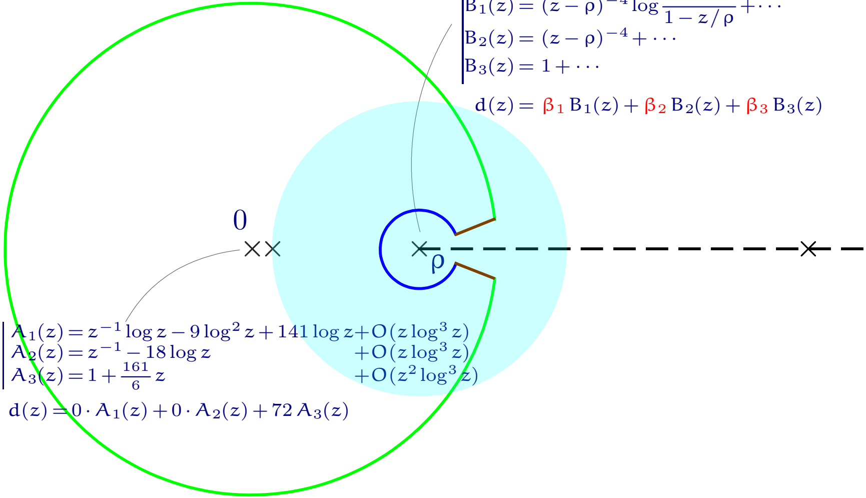
$$\left|\begin{aligned} B_1(z) &= (z-\rho)^{-4}\log\frac{1}{1-z/\rho}+\cdots \\ B_2(z) &= (z-\rho)^{-4}+\cdots \\ B_3(z) &= 1+\cdots \end{aligned}\right.$$

$$d(z) = \beta_1\,B_1(z)+\beta_2\,B_2(z)+\beta_3\,B_3(z)$$

$$\left|\begin{aligned} A_1(z) &= z^{-1}\log z - 9\log^2 z + 141\log z + O(z\log^3 z) \\ A_2(z) &= z^{-1} - 18\log z + O(z\log^3 z) \\ A_3(z) &= 1+\tfrac{161}{6}z + O(z^2\log^3 z) \end{aligned}\right.$$

$$d(z) = 0\cdot A_1(z) + 0\cdot A_2(z) + 72\,A_3(z)$$

Effective Singularity Analysis



$$\begin{aligned} B_1(z) &= (z-\rho)^{-4}\log\frac{1}{1-z/\rho}+\cdots \\ B_2(z) &= (z-\rho)^{-4}+\cdots \\ B_3(z) &= 1+\cdots \end{aligned}$$

$$d(z) = \beta_1\,B_1(z) + \beta_2\,B_2(z) + \beta_3\,B_3(z)$$

$$\begin{aligned} A_1(z) &= z^{-1}\log z - 9\log^2 z + 141\log z + O(z\log^3 z) \\ A_2(z) &= z^{-1} - 18\log z + O(z\log^3 z) \\ A_3(z) &= 1 + \tfrac{161}{6}z + O(z^2\log^3 z) \end{aligned}$$

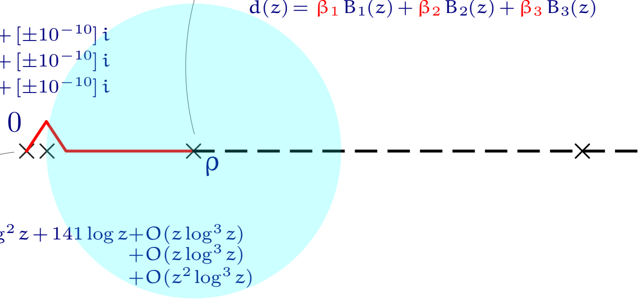
$$d(z) = 0\cdot A_1(z) + 0\cdot A_2(z) + 72\,A_3(z)$$

Effective Singularity Analysis

$$\begin{aligned} B_1(z) &= (z-\rho)^{-4}\log\frac{1}{1-z/\rho}+\cdots \\ B_2(z) &= (z-\rho)^{-4}+\cdots \\ B_3(z) &= 1+\cdots \end{aligned}$$

$$d(z) = \beta_1\,B_1(z) + \beta_2\,B_2(z) + \beta_3\,B_3(z)$$

$$\begin{aligned} \beta_1 &= [0.0420 \pm 10^{-4}] + [\pm 10^{-10}]\,i \\ \beta_2 &= [0.0598 \pm 10^{-4}] + [\pm 10^{-10}]\,i \\ \beta_3 &= [0.7302 \pm 10^{-4}] + [\pm 10^{-10}]\,i \end{aligned}$$



$$\begin{aligned} A_1(z) &= z^{-1}\log z - 9\log^2 z + 141\log z + O(z\log^3 z) \\ A_2(z) &= z^{-1} - 18\log z + O(z\log^3 z) \\ A_3(z) &= 1 + \tfrac{161}{6}\,z + O(z^2\log^3 z) \end{aligned}$$

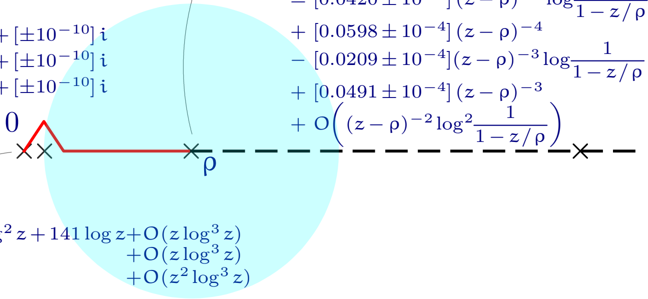
$$d(z) = 0\cdot A_1(z) + 0\cdot A_2(z) + 72\,A_3(z)$$

Effective Singularity Analysis

$$\begin{aligned} B_1(z) &= (z-\rho)^{-4}\log\frac{1}{1-z/\rho}+\cdots \\ B_2(z) &= (z-\rho)^{-4}+\cdots \\ B_3(z) &= 1+\cdots \end{aligned}$$

$$\begin{aligned} d(z) = {} & \beta_1 B_1(z)+\beta_2 B_2(z)+\beta_3 B_3(z) \\ = {} & [0.0420\pm 10^{-4}]\,(z-\rho)^{-4}\log\frac{1}{1-z/\rho} \\ & + [0.0598\pm 10^{-4}]\,(z-\rho)^{-4} \\ & - [0.0209\pm 10^{-4}](z-\rho)^{-3}\log\frac{1}{1-z/\rho} \\ & + [0.0491\pm 10^{-4}]\,(z-\rho)^{-3} \\ & + O\left((z-\rho)^{-2}\log^2\frac{1}{1-z/\rho}\right) \end{aligned}$$

$$\begin{aligned} \beta_1 &= [0.0420\pm 10^{-4}]+[\pm 10^{-10}]\,i \\ \beta_2 &= [0.0598\pm 10^{-4}]+[\pm 10^{-10}]\,i \\ \beta_3 &= [0.7302\pm 10^{-4}]+[\pm 10^{-10}]\,i \end{aligned}$$



$$\begin{aligned} A_1(z) &= z^{-1}\log z - 9\log^2 z + 141\log z + O(z\log^3 z) \\ A_2(z) &= z^{-1} - 18\log z + O(z\log^3 z) \\ A_3(z) &= 1+\frac{161}{6}z + O(z^2\log^3 z) \end{aligned}$$

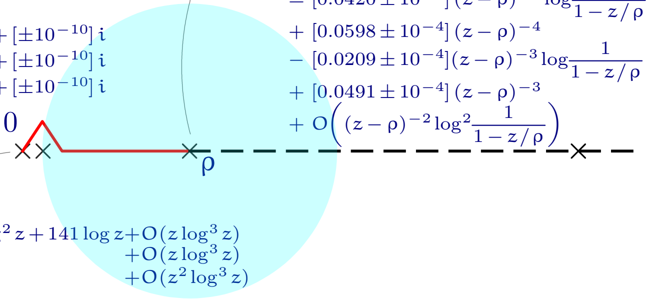
$$d(z) = 0\cdot A_1(z) + 0\cdot A_2(z) + 72\,A_3(z)$$

Effective Singularity Analysis

$$\begin{aligned} B_1(z) &= (z-\rho)^{-4}\log\frac{1}{1-z/\rho}+\cdots \\ B_2(z) &= (z-\rho)^{-4}+\cdots \\ B_3(z) &= 1+\cdots \end{aligned}$$

$$\begin{aligned} d(z) = {} & \beta_1 B_1(z)+\beta_2 B_2(z)+\beta_3 B_3(z) \\ = {} & [0.0420\pm 10^{-4}]\,(z-\rho)^{-4}\log\frac{1}{1-z/\rho} \\ + {} & [0.0598\pm 10^{-4}]\,(z-\rho)^{-4} \\ - {} & [0.0209\pm 10^{-4}](z-\rho)^{-3}\log\frac{1}{1-z/\rho} \\ + {} & [0.0491\pm 10^{-4}]\,(z-\rho)^{-3} \\ + {} & O\left((z-\rho)^{-2}\log^2\frac{1}{1-z/\rho}\right) \end{aligned}$$

$$\begin{aligned} \beta_1 &= [0.0420\pm 10^{-4}]+[\pm 10^{-10}]\,i \\ \beta_2 &= [0.0598\pm 10^{-4}]+[\pm 10^{-10}]\,i \\ \beta_3 &= [0.7302\pm 10^{-4}]+[\pm 10^{-10}]\,i \end{aligned}$$



$$\begin{aligned} A_1(z) &= z^{-1}\log z - 9\log^2 z + 141\log z + O(z\log^3 z) \\ A_2(z) &= z^{-1} - 18\log z + O(z\log^3 z) \\ A_3(z) &= 1+\frac{161}{6}z + O(z^2\log^3 z) \end{aligned}$$

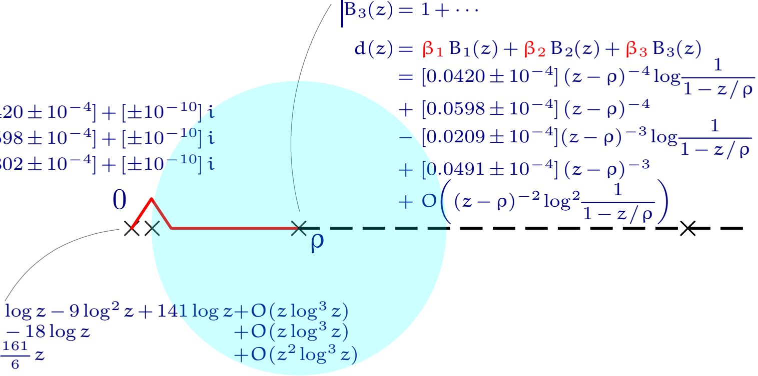
$$d(z) = 0\cdot A_1(z) + 0\cdot A_2(z) + 72\,A_3(z)$$

Effective Singularity Analysis

$$\begin{aligned} B_1(z) &= (z-\rho)^{-4}\log\frac{1}{1-z/\rho}+\cdots\\ B_2(z) &= (z-\rho)^{-4}+\cdots\\ B_3(z) &= 1+\cdots \end{aligned}$$

$$\begin{aligned} d(z) &= \beta_1\,B_1(z)+\beta_2\,B_2(z)+\beta_3\,B_3(z)\\ &= [0.0420\pm10^{-4}]\,(z-\rho)^{-4}\log\frac{1}{1-z/\rho}\\ &+ [0.0598\pm10^{-4}]\,(z-\rho)^{-4}\\ &- [0.0209\pm10^{-4}](z-\rho)^{-3}\log\frac{1}{1-z/\rho}\\ &+ [0.0491\pm10^{-4}]\,(z-\rho)^{-3}\\ &+ O\left((z-\rho)^{-2}\log^2\frac{1}{1-z/\rho}\right) \end{aligned}$$

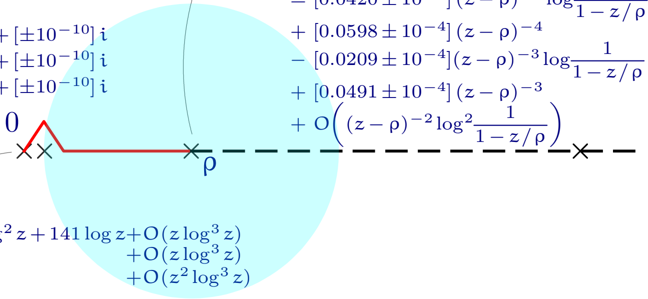
$$\begin{aligned} \beta_1 &= [0.0420\pm10^{-4}]+[\pm10^{-10}]\,i\\ \beta_2 &= [0.0598\pm10^{-4}]+[\pm10^{-10}]\,i\\ \beta_3 &= [0.7302\pm10^{-4}]+[\pm10^{-10}]\,i \end{aligned}$$



$$\begin{aligned} A_1(z) &= z^{-1}\log z-9\log^2 z+141\log z+O(z\log^3 z)\\ A_2(z) &= z^{-1}-18\log z+O(z\log^3 z)\\ A_3(z) &= 1+\tfrac{161}{6}z+O(z^2\log^3 z) \end{aligned}$$

$$d(z)=0\cdot A_1(z)+0\cdot A_2(z)+72\,A_3(z)$$

Effective Singularity Analysis



$$\begin{aligned} B_1(z) &= (z-\rho)^{-4}\log\frac{1}{1-z/\rho} + \cdots \\ B_2(z) &= (z-\rho)^{-4} + \cdots \\ B_3(z) &= 1 + \cdots \end{aligned}$$

$$\begin{aligned} d(z) = {} & \beta_1 B_1(z) + \beta_2 B_2(z) + \beta_3 B_3(z) \\ = {} & [0.0420 \pm 10^{-4}]\,(z-\rho)^{-4}\log\frac{1}{1-z/\rho} \\ & + [0.0598 \pm 10^{-4}]\,(z-\rho)^{-4} \\ & - [0.0209 \pm 10^{-4}](z-\rho)^{-3}\log\frac{1}{1-z/\rho} \\ & + [0.0491 \pm 10^{-4}]\,(z-\rho)^{-3} \\ & + O\left((z-\rho)^{-2}\log^2\frac{1}{1-z/\rho}\right) \end{aligned}$$

$$\begin{aligned} \beta_1 &= [0.0420 \pm 10^{-4}] + [\pm 10^{-10}]\,i \\ \beta_2 &= [0.0598 \pm 10^{-4}] + [\pm 10^{-10}]\,i \\ \beta_3 &= [0.7302 \pm 10^{-4}] + [\pm 10^{-10}]\,i \end{aligned}$$

$$\begin{aligned} A_1(z) &= z^{-1}\log z - 9\log^2 z + 141\log z + O(z\log^3 z) \\ A_2(z) &= z^{-1} - 18\log z + O(z\log^3 z) \\ A_3(z) &= 1 + \frac{161}{6}z + O(z^2\log^3 z) \end{aligned}$$

$$d(z) = 0 \cdot A_1(z) + 0 \cdot A_2(z) + 72\,A_3(z)$$

Automatic Asymptotics with Error Bounds

[Dong, Melczer, M., 2024]

```
sage: from ore_algebra.analytic.singularity_analysis import bound_coefficients

sage: seqini = [72, 1932, 31248, 790101/2, 17208645/4, 338898609/8, 1551478257/4]

sage: deq = (8388593*x^2*(3*x^4 - 164*x^3 + 370*x^2 - 164*x + 3)*(x + 1)^2*(x^2
      - 6*x + 1)^2*(x - 1)^3*Dx^3 + 8388593*x*(x + 1)*(x^2 - 6*x + 1)*(66*x^8 -
      3943*x^7 + 18981*x^6 - 16759*x^5 - 30383*x^4 + 47123*x^3 - 17577*x^2 + 971*x
      - 15)*(x - 1)^2*Dx^2 + 16777186*(x - 1)*(210*x^12 - 13761*x^11 + 101088*x^10
      - 178437*x^9 - 248334*x^8 + 930590*x^7 - 446064*x^6 - 694834*x^5 + 794998*x^4
      - 267421*x^3 + 24144*x^2 - 649*x + 6)*Dx + 6341776308*x^12 - 427012938072*x^11
      + 2435594423178*x^10 - 2400915979716*x^9 - 10724094731502*x^8
      + 26272536406048*x^7 - 8496738740956*x^6 - 30570113263064*x^5 +
      39394376229112*x^4 - 19173572139496*x^3 + 3825886272626*x^2 - 170758199108*x
      + 2701126946)

sage: bound_coefficients(deq, seqini, order=2)
```

based on code by Ruiwen Dong

Automatic Asymptotics with Error Bounds

[Dong, Melczer, M., 2024]

```
sage: from ore_algebra.analytic.singularity_analysis import bound_coefficients

sage: seqini = [72, 1932, 31248, 790101/2, 17208645/4, 338898609/8, 1551478257/4]

sage: deq = (8388593*x^2*(3*x^4 - 164*x^3 + 370*x^2 - 164*x + 3)*(x + 1)^2*(x^2
      - 6*x + 1)^2*(x - 1)^3*Dx^3 + 8388593*x*(x + 1)*(x^2 - 6*x + 1)*(66*x^8 -
      3943*x^7 + 18981*x^6 - 16759*x^5 - 30383*x^4 + 47123*x^3 - 17577*x^2 + 971*x
      - 15)*(x - 1)^2*Dx^2 + 16777186*(x - 1)*(210*x^12 - 13761*x^11 + 101088*x^10
      - 178437*x^9 - 248334*x^8 + 930590*x^7 - 446064*x^6 - 694834*x^5 + 794998*x^4
      - 267421*x^3 + 24144*x^2 - 649*x + 6)*Dx + 6341776308*x^12 - 427012938072*x^11
      + 2435594423178*x^10 - 2400915979716*x^9 - 10724094731502*x^8
      + 26272536406048*x^7 - 8496738740956*x^6 - 30570113263064*x^5 +
      39394376229112*x^4 - 19173572139496*x^3 + 3825886272626*x^2 - 170758199108*x
      + 2701126946)

sage: bound_coefficients(deq, seqini, order=2)

1.00000000000000*5.828427124746190?^n*(([8.0719562915189 +/- 5.34e-14] + [+/- 3.5
7e-20]*I)*n^3*log(n) + ([1.3714048996380 +/- 2.19e-14] + [+/- 1.25e-14]*I)*n^3 + (
[50.5091308739016 +/- 8.57e-14] + [+/- 2.23e-19]*I)*n^2*log(n) + ([29.698551451828
 +/- 2.38e-13] + [+/- 7.81e-14]*I)*n^2 + B([6308.749020581030 +/- 1.88e-14]*n*log(
n)^2, n >= 15))
```

based on code by Ruiwen Dong

Symbolic-Numeric Factoring

[van der Hoeven, 2007; Chyzak, Goyer, M., 2022]

```
sage: ((4*x^2 + 6*x + 2)*Dx^2 + (4*x + 3)*Dx - 1).factor()
sage: from ore_algebra.examples import polya
sage: polya.dop[9]  # cf. Beukers & Vlasenko 2021
sage: polya.dop[9].is_provably_irreducible()  # Maple DFactor: > 1 min 40 s
```

code by Alexandre Goyer

Symbolic-Numeric Factoring

[van der Hoeven, 2007; Chyzak, Goyer, M., 2022]

```
sage: ((4*x^2 + 6*x + 2)*Dx^2 + (4*x + 3)*Dx - 1).factor()
[(-2*x - 2)*Dx - 1, (-2*x - 1)*Dx + 1]
sage: from ore_algebra.examples import polya
sage: polya.dop[9]  # cf. Beukers & Vlasenko 2021
sage: polya.dop[9].is_provably_irreducible()  # Maple DFactor: > 1 min 40 s
```

code by Alexandre Goyer

Symbolic-Numeric Factoring

[van der Hoeven, 2007; Chyzak, Goyer, M., 2022]

```
sage: ((4*x^2 + 6*x + 2)*Dx^2 + (4*x + 3)*Dx - 1).factor()
[(-2*x - 2)*Dx - 1, (-2*x - 1)*Dx + 1]
sage: from ore_algebra.examples import polya
sage: polya.dop[9]  # cf. Beukers & Vlasenko 2021
(-914457600*z^18 + 270648576*z^16 - 11059840*z^14 + 140448*z^12 - 660*z^10 + z^8)*
Dz^9 + (-74071065600*z^17 + 19486697472*z^15 - 696769920*z^13 + 7584192*z^11 - 297
00*z^9 + 36*z^7)*Dz^8 + (-2370274099200*z^16 + 550176012288*z^14 - 17038986624*z^1
2 + 156601632*z^10 - 498696*z^8 + 462*z^6)*Dz^7 + (-38714476953600*z^15 + 78616610
79552*z^13 - 208388936448*z^11 + 1587838560*z^9 - 3984288*z^7 + 2646*z^5)*Dz^6 + (
-348430292582400*z^14 + 61302652637184*z^12 - 1371240707328*z^10 + 8466726048*z^8
- 16052916*z^6 + 6951*z^4)*Dz^5 + (-1742151462912000*z^13 + 262594912849920*z^11 -
 487272756800*z^9 + 23679448320*z^7 - 32006700*z^5 + 7770*z^3)*Dz^4 + (-464573723
4432000*z^12 + 592052526858240*z^10 - 8923673318400*z^8 + 32838948864*z^6 - 290544
76*z^4 + 3025*z^2)*Dz^3 + (-5973090729984000*z^11 + 633557545205760*z^9 - 75524219
59680*z^7 + 19955195136*z^5 - 10089816*z^3 + 255*z)*Dz^2 + (-2986545364992000*z^10
 + 258683438039040*z^8 - 2355318466560*z^6 + 4133152512*z^4 - 935592*z^2 + 1)*Dz -
 331838373888000*z^9 + 22924354682880*z^7 - 152023080960*z^5 + 156432384*z^3 - 921
6*z
sage: polya.dop[9].is_provably_irreducible()  # Maple DFactor: > 1 min 40 s
```

code by Alexandre Goyer

Symbolic-Numeric Factoring

[van der Hoeven, 2007; Chyzak, Goyer, M., 2022]

```
sage: ((4*x^2 + 6*x + 2)*Dx^2 + (4*x + 3)*Dx - 1).factor()
[(-2*x - 2)*Dx - 1, (-2*x - 1)*Dx + 1]
sage: from ore_algebra.examples import polya
sage: polya.dop[9]  # cf. Beukers & Vlasenko 2021
(-914457600*z^18 + 270648576*z^16 - 11059840*z^14 + 140448*z^12 - 660*z^10 + z^8)*
Dz^9 + (-74071065600*z^17 + 19486697472*z^15 - 696769920*z^13 + 7584192*z^11 - 297
00*z^9 + 36*z^7)*Dz^8 + (-2370274099200*z^16 + 550176012288*z^14 - 17038986624*z^1
2 + 156601632*z^10 - 498696*z^8 + 462*z^6)*Dz^7 + (-38714476953600*z^15 + 78616610
79552*z^13 - 208388936448*z^11 + 1587838560*z^9 - 3984288*z^7 + 2646*z^5)*Dz^6 + (
-348430292582400*z^14 + 61302652637184*z^12 - 1371240707328*z^10 + 8466726048*z^8
- 16052916*z^6 + 6951*z^4)*Dz^5 + (-1742151462912000*z^13 + 262594912849920*z^11 -
 487272756800*z^9 + 23679448320*z^7 - 32006700*z^5 + 7770*z^3)*Dz^4 + (-464573723
4432000*z^12 + 592052526858240*z^10 - 8923673318400*z^8 + 32838948864*z^6 - 290544
76*z^4 + 3025*z^2)*Dz^3 + (-5973090729984000*z^11 + 633557545205760*z^9 - 75524219
59680*z^7 + 19955195136*z^5 - 10089816*z^3 + 255*z)*Dz^2 + (-2986545364992000*z^10
 + 258683438039040*z^8 - 2355318466560*z^6 + 4133152512*z^4 - 935592*z^2 + 1)*Dz -
 331838373888000*z^9 + 22924354682880*z^7 - 152023080960*z^5 + 156432384*z^3 - 921
6*z
sage: polya.dop[9].is_provably_irreducible()  # Maple DFactor: > 1 min 40 s
True
```

code by Alexandre Goyer