Rigorous High-Precision Solution of Fuchsian ODEs and Applications

Marc Mezzarobba

CNRS, École polytechnique

Kolchin seminar, March 20, 2026

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

```
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
sage:
```

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

$$\uparrow$$

$$y\colon U \subseteq \mathbb{C} \to \mathbb{C}$$

```
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
sage:
```

$$\underset{\downarrow}{a_i \in \mathbb{C}[x]} \qquad \underset{\downarrow}{a_i \in \mathbb{C}[x]} \qquad \underset{\downarrow}{a_i \in \mathbb{C}[x]}$$

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

$$\underset{\uparrow}{y}$$

$$y : U \subseteq \mathbb{C} \to \mathbb{C}$$

```
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
sage:
```

$$\underset{\downarrow}{a_i \in \mathbb{C}[x]} \qquad \underset{\downarrow}{a_i \in \mathbb{C}[x]} \qquad \underset{\downarrow}{a_i \in \mathbb{C}[x]}$$

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

$$\underset{\uparrow}{y : U \subseteq \mathbb{C} \to \mathbb{C}}$$

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
sage:
```

$$\underset{\downarrow}{a_i \in \mathbb{C}[x]} \qquad \underset{\downarrow}{a_i \in \mathbb{C}[x]} \qquad \underset{\downarrow}{a_i \in \mathbb{C}[x]}$$

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

$$\underset{\uparrow}{y : U \subseteq \mathbb{C} \to \mathbb{C}}$$

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
sage:
```

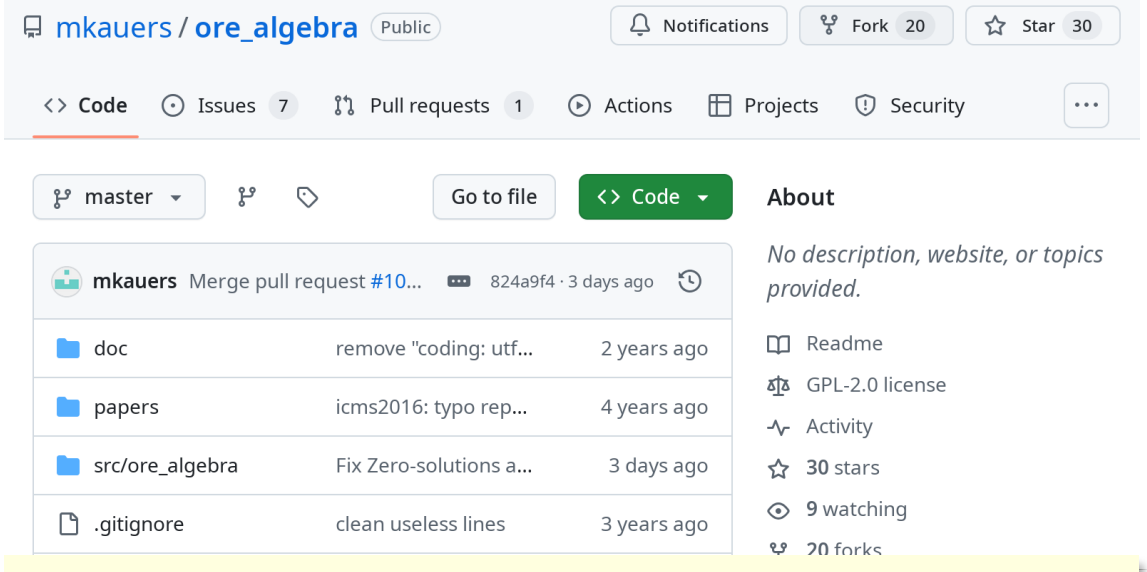
Rigorous High-Precision Solution of Fuchsian ODEs

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

($a_i \in \mathbb{C}[x]$ for each coefficient; $y : U \subseteq \mathbb{C} \to \mathbb{C}$)

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
sage:
```




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

Theorem. [Cauchy, ~1839]

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

For any choice of $y(0), \dots, y^{(r-1)}(0) \in \mathbb{C}$, 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 $y\colon \{|x| < \rho\} \to \mathbb{C}$.

Theorem. [Cauchy, ~1839]

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

For any choice of $y(0), \dots, y^{(r-1)}(0) \in \mathbb{C}$, 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:

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

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 $y\colon \{|x| < \rho\} \to \mathbb{C}$.

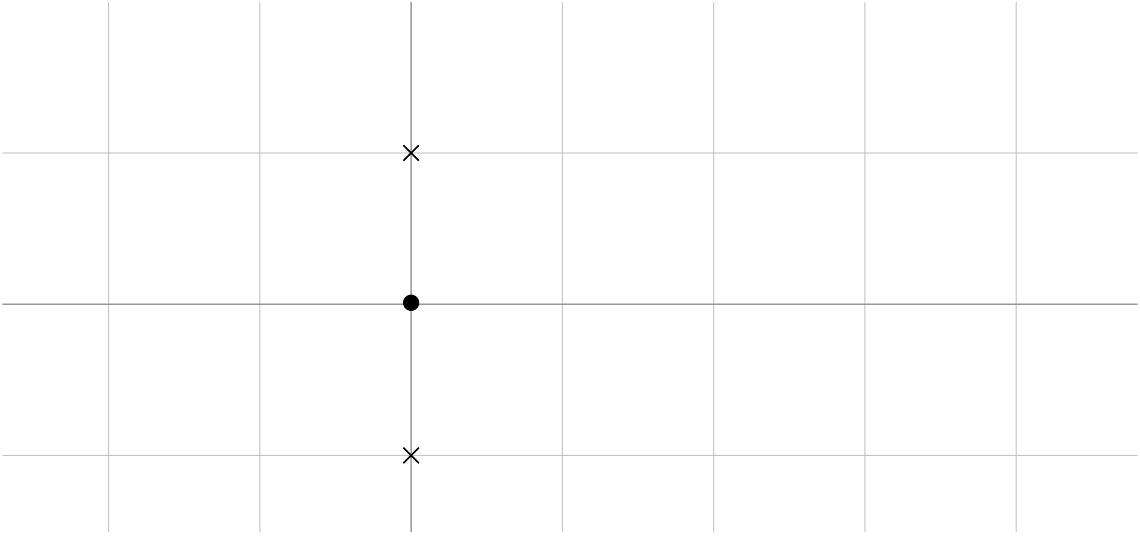
"Local" basis of solutions:

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

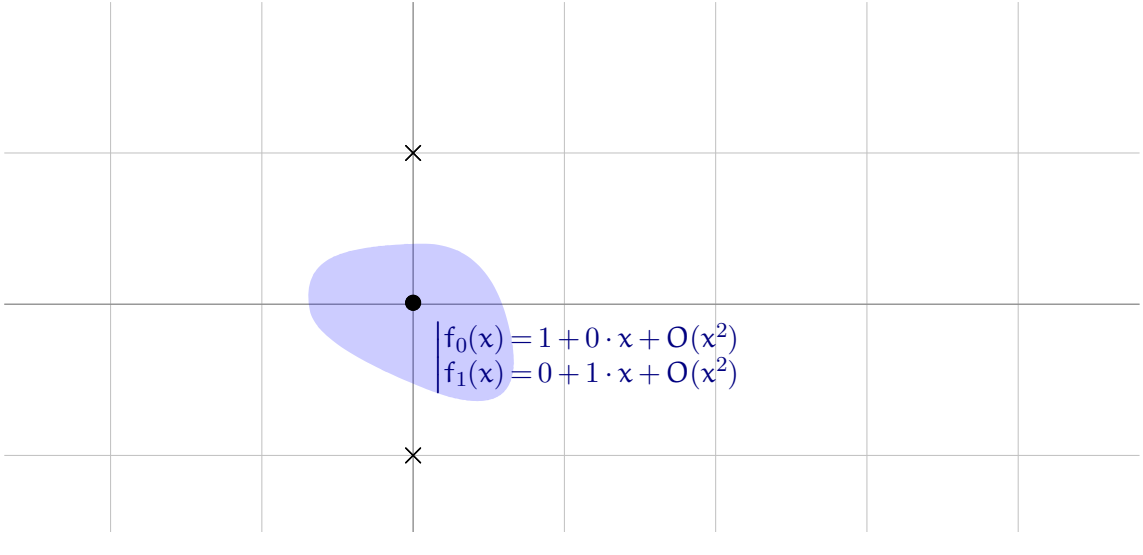
Fundamental matrix:

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

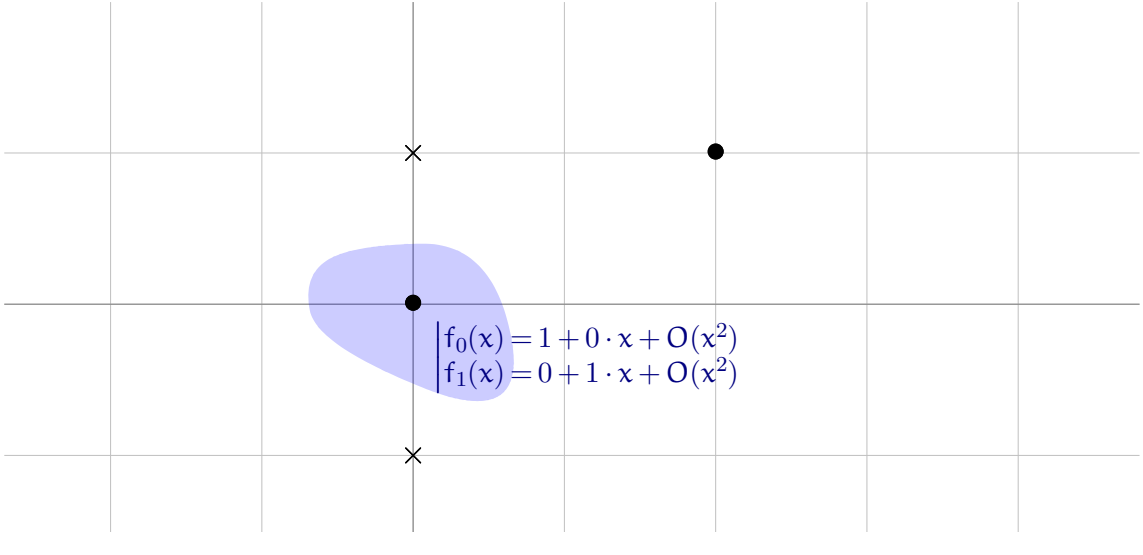
$$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$$



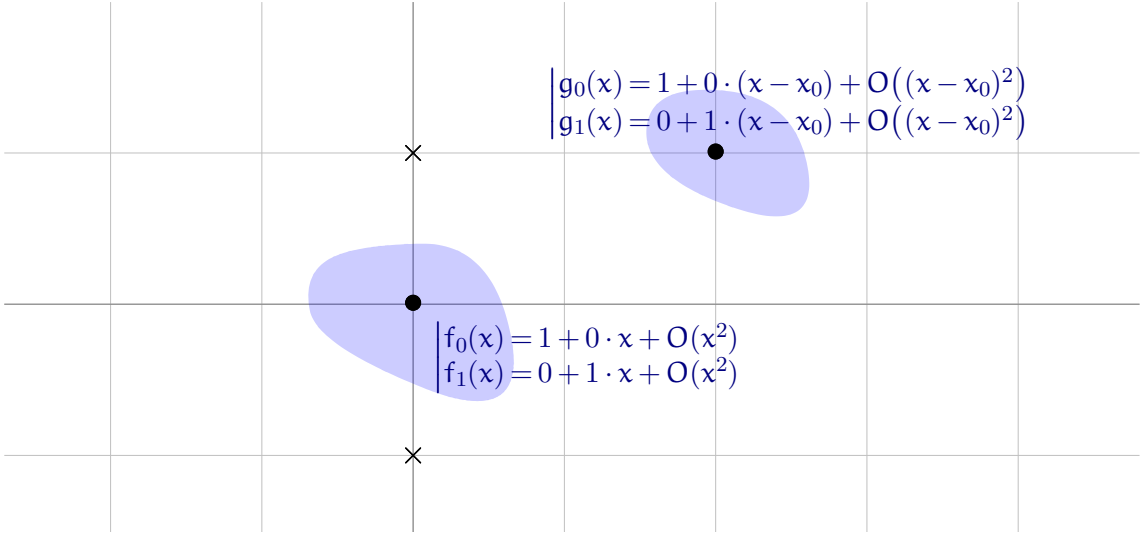
$$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$$



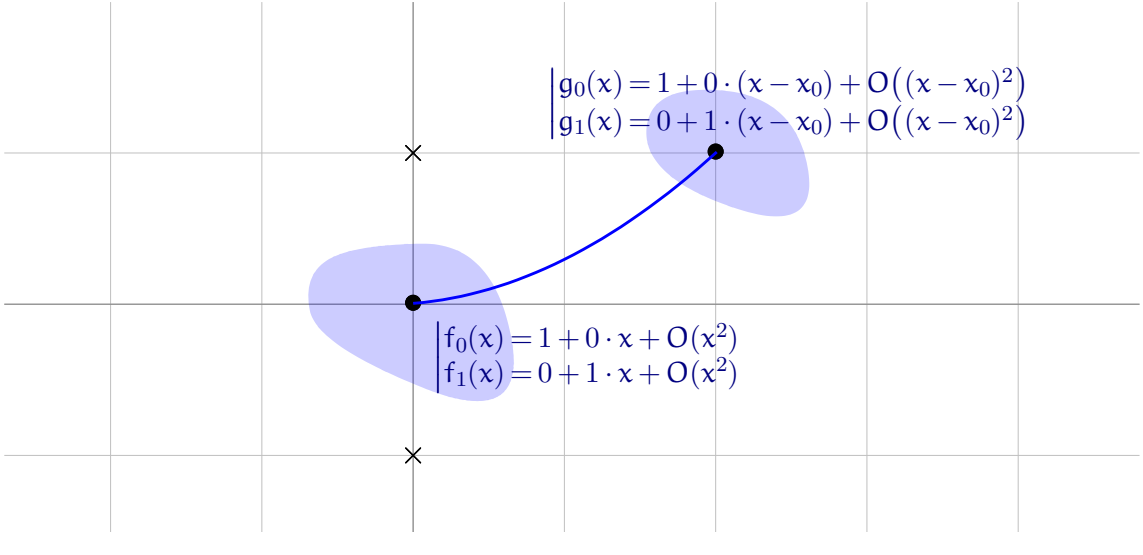
$$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$$



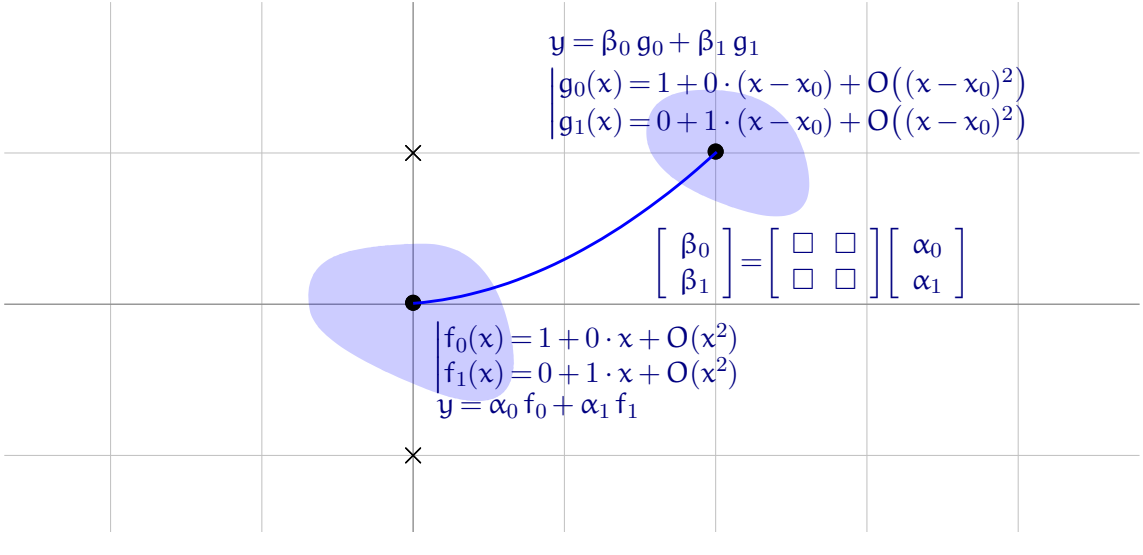
$$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$$


$g_0(x) = 1 + 0\cdot(x-x_0) + O\big((x-x_0)^2\big)$
$g_1(x) = 0 + 1\cdot(x-x_0) + O\big((x-x_0)^2\big)$
$f_0(x) = 1 + 0\cdot x + O(x^2)$
$f_1(x) = 0 + 1\cdot x + O(x^2)$

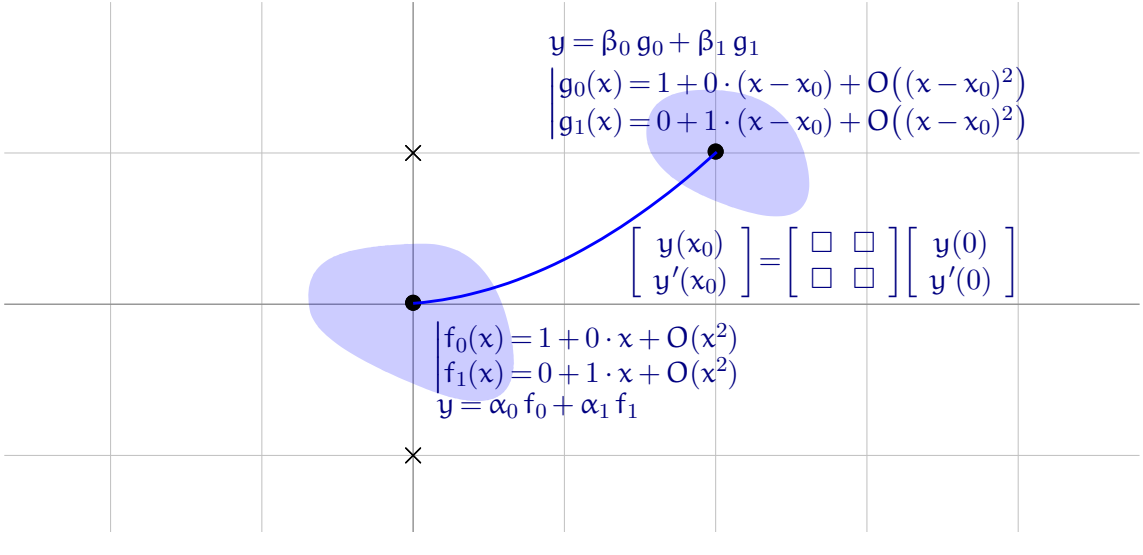
$$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$$


$g_0(x) = 1 + 0\cdot(x-x_0) + O\big((x-x_0)^2\big)$
$g_1(x) = 0 + 1\cdot(x-x_0) + O\big((x-x_0)^2\big)$
$f_0(x) = 1 + 0\cdot x + O(x^2)$
$f_1(x) = 0 + 1\cdot x + O(x^2)$

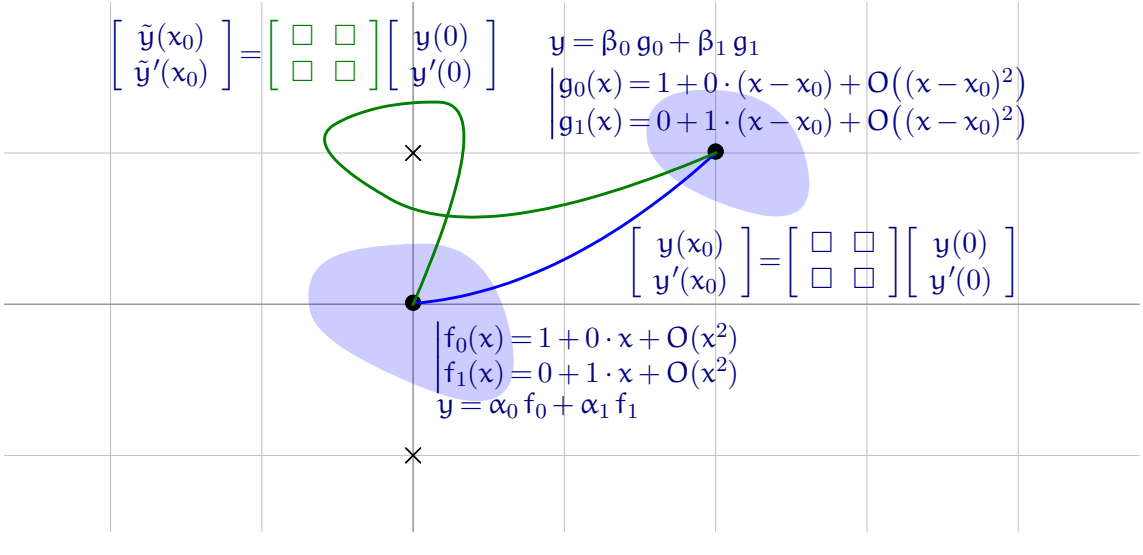
$$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$$


$y=\beta_0\,g_0+\beta_1\,g_1$
$g_0(x)=1+0\cdot(x-x_0)+O\big((x-x_0)^2\big)$
$g_1(x)=0+1\cdot(x-x_0)+O\big((x-x_0)^2\big)$
$\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix}=\begin{bmatrix}\square & \square\\ \square & \square\end{bmatrix}\begin{bmatrix}\alpha_0\\ \alpha_1\end{bmatrix}$
$f_0(x)=1+0\cdot x+O(x^2)$
$f_1(x)=0+1\cdot x+O(x^2)$
$y=\alpha_0\,f_0+\alpha_1\,f_1$

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


$y = \beta_0\, g_0 + \beta_1\, g_1$
$g_0(x) = 1 + 0\cdot(x-x_0) + O\big((x-x_0)^2\big)$
$g_1(x) = 0 + 1\cdot(x-x_0) + O\big((x-x_0)^2\big)$
$\begin{bmatrix} y(x_0) \\ y'(x_0) \end{bmatrix} = \begin{bmatrix} \square & \square \\ \square & \square \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$
$f_0(x) = 1 + 0\cdot x + O(x^2)$
$f_1(x) = 0 + 1\cdot x + O(x^2)$
$y = \alpha_0\, f_0 + \alpha_1\, f_1$

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


$\begin{bmatrix}\tilde{y}(x_0)\\ \tilde{y}'(x_0)\end{bmatrix}=\begin{bmatrix}\square & \square\\ \square & \square\end{bmatrix}\begin{bmatrix}y(0)\\ y'(0)\end{bmatrix}$
$y=\beta_0\,g_0+\beta_1\,g_1$
$g_0(x)=1+0\cdot(x-x_0)+O\big((x-x_0)^2\big)$
$g_1(x)=0+1\cdot(x-x_0)+O\big((x-x_0)^2\big)$
$\begin{bmatrix}y(x_0)\\ y'(x_0)\end{bmatrix}=\begin{bmatrix}\square & \square\\ \square & \square\end{bmatrix}\begin{bmatrix}y(0)\\ y'(0)\end{bmatrix}$
$f_0(x)=1+0\cdot x+O(x^2)$
$f_1(x)=0+1\cdot x+O(x^2)$
$y=\alpha_0\,f_0+\alpha_1\,f_1$

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])
sage:
```

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])
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])
sage:
```

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])
sage:
```

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])
[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])
sage:
```

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])
[0.78539816339744831 +/- 5.40e-19]
sage: dop.numerical_solution(ini=[0,1], path=[0,2+I])
[1.1780972450961725 +/- 3.58e-17] + [0.17328679513998633 +/- 2.68e-18]*I
sage: dop.numerical_solution(ini=[0,1], path=[0,1+I,2*I,I-1,0,2+I])
sage:
```

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])
[0.78539816339744831 +/- 5.40e-19]
sage: dop.numerical_solution(ini=[0,1], path=[0,2+I])
[1.1780972450961725 +/- 3.58e-17] + [0.17328679513998633 +/- 2.68e-18]*I
sage: dop.numerical_solution(ini=[0,1], path=[0,1+I,2*I,I-1,0,2+I])
[4.3196898986859657 +/- 3.33e-18] + [0.17328679513998633 +/- 3.00e-18]*I
sage:
```

Automatic bounds on truncation errors in series expansions

Interval arithmetic for error propagation

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

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

High precision requires order-adaptive methods, large steps.

```
sage: (Dx - 1).numerical_solution(ini=[1], path=[0,1], eps=1e-1000)
[2.71828182845904523536028747135266249775724709369995957496696762772407663035354759457138217852516642742746639193200305992181741359662904357290033429526059563073813232862794349076323382988075319525101901157383418793070215408914993488416750924476146066808226480016847741185374234544243710753907774499206955170276183860626133138458300075204493382656029760673711320070932870912744374704723069697720931014169283681902551510865746377211125238978442505695369677078544996996794686445490598793163688923009879312773617821542499922957635148220826989519366803318252886939849646510582093923982948879332036250944311730123819706841614039701983767932068328237646480429531180232878250981945581530175671736133206981125099618188159304169035159888851934580727386673858942287922849892086805825749279610484198444363463244968487560233624827041978623209002160990235304369941849146314093431738143640546253152096183690888707016768396424378140592714563549061303107208510383750510115747704171898610687396965521267154688957035035 40 +/- 5.53e-1002]
sage:
```

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

High precision requires order-adaptive methods, large steps.

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

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

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

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

$$(x-\xi)^{\lambda}\big(f_0(x) + f_1(x)\log(x-\xi) + \cdots + f_K(x)\log(x-\xi)^K\big).$$

$\lambda \in \mathbb{C}$; $K \in \mathbb{N}$; f_0, f_1: convergent power series

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

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

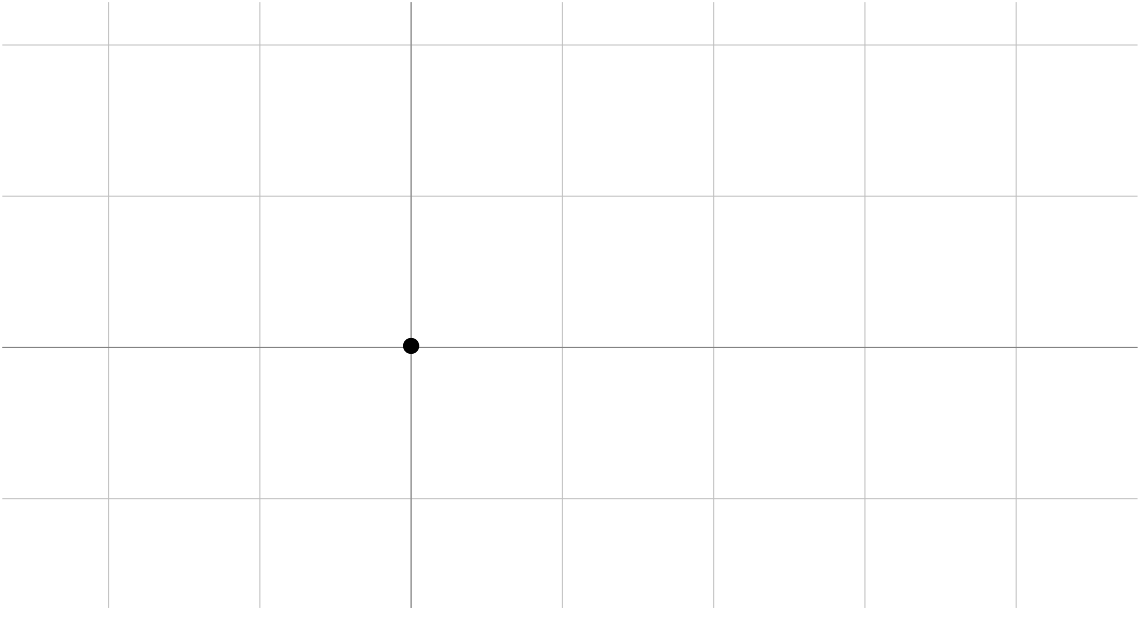
Definition. A **regular singular point** of L is a point $\xi \in \mathbb{C}$ where $a_r(\xi) = 0$ but there is a full basis of solutions of the form

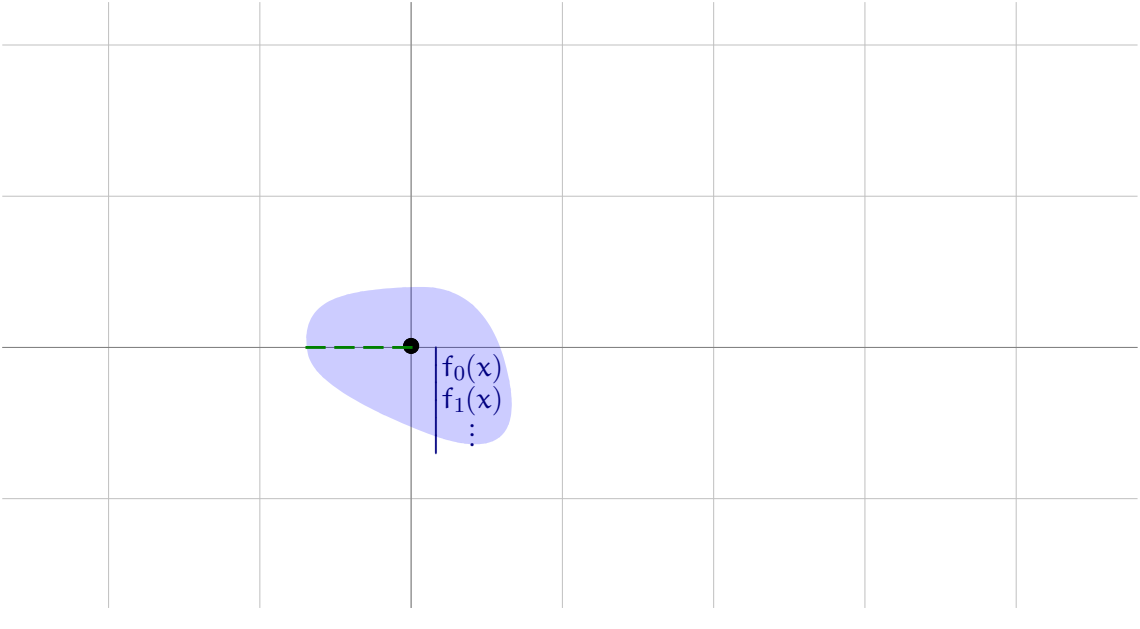
$$(x-\xi)^{\lambda}\big(f_0(x) + f_1(x)\log(x-\xi) + \cdots + f_K(x)\log(x-\xi)^K\big).$$

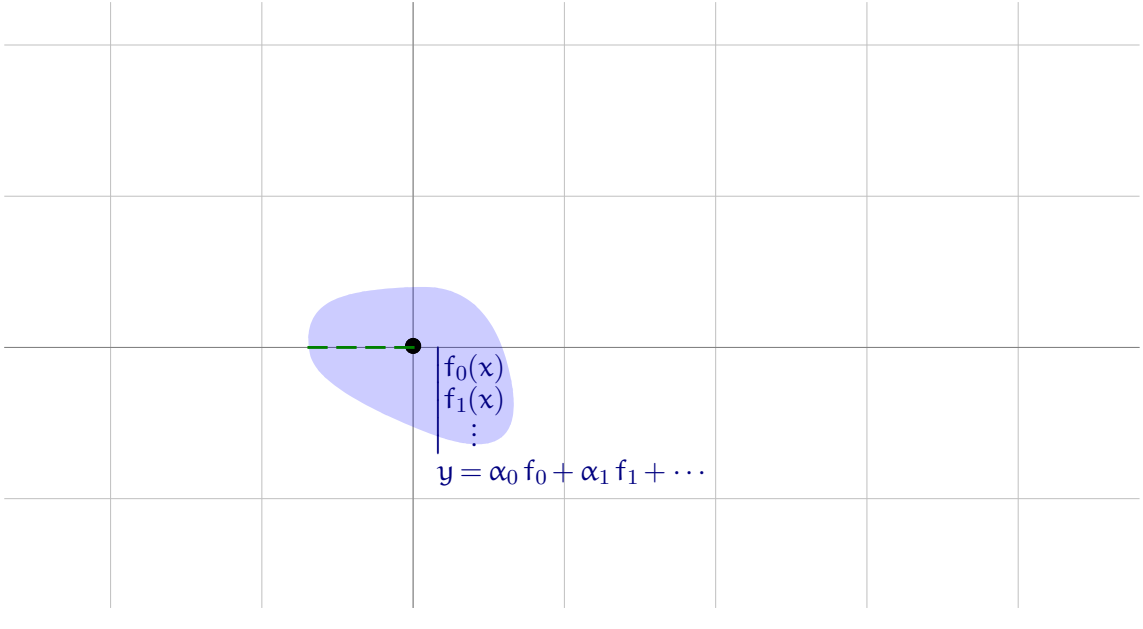
$\lambda \in \mathbb{C}$; $K \in \mathbb{N}$; f_0, f_1: convergent power series

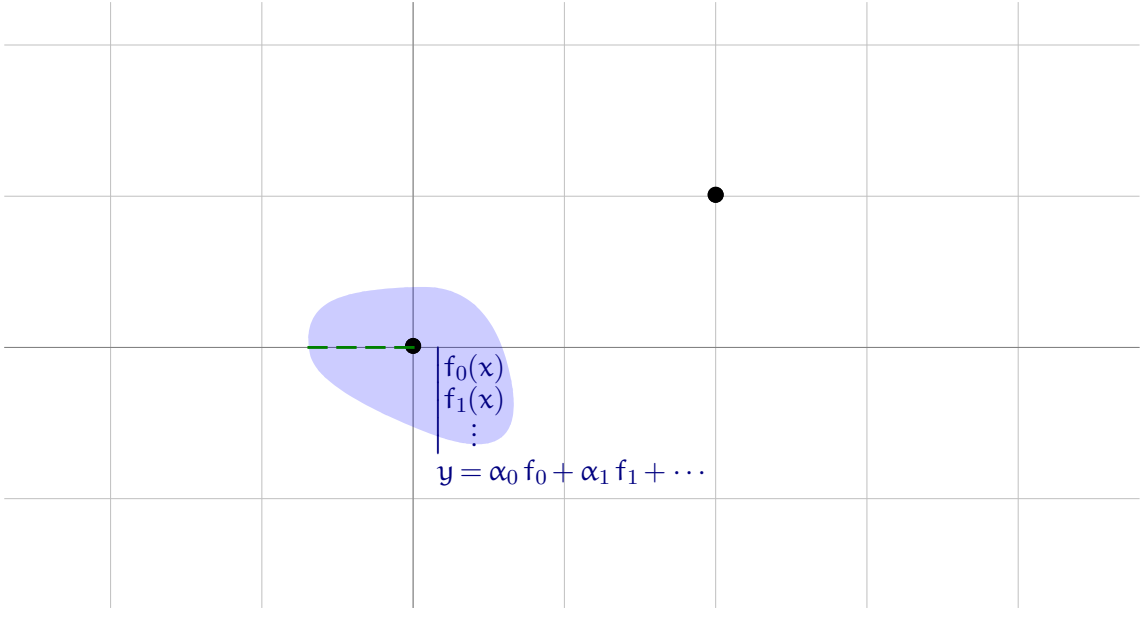
An equation is **Fuchsian** if all its singular points in $\mathbb{P}^1(\mathbb{C})$ are regular.

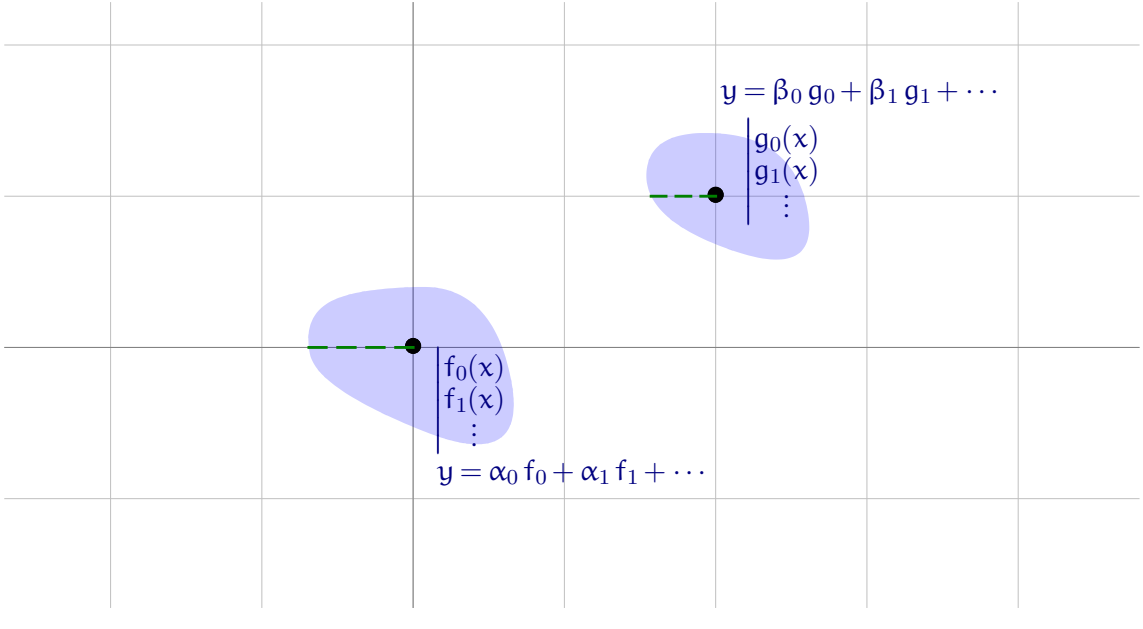
Examples: differential equations satisfied by algebraic functions
periods of algebraic varieties

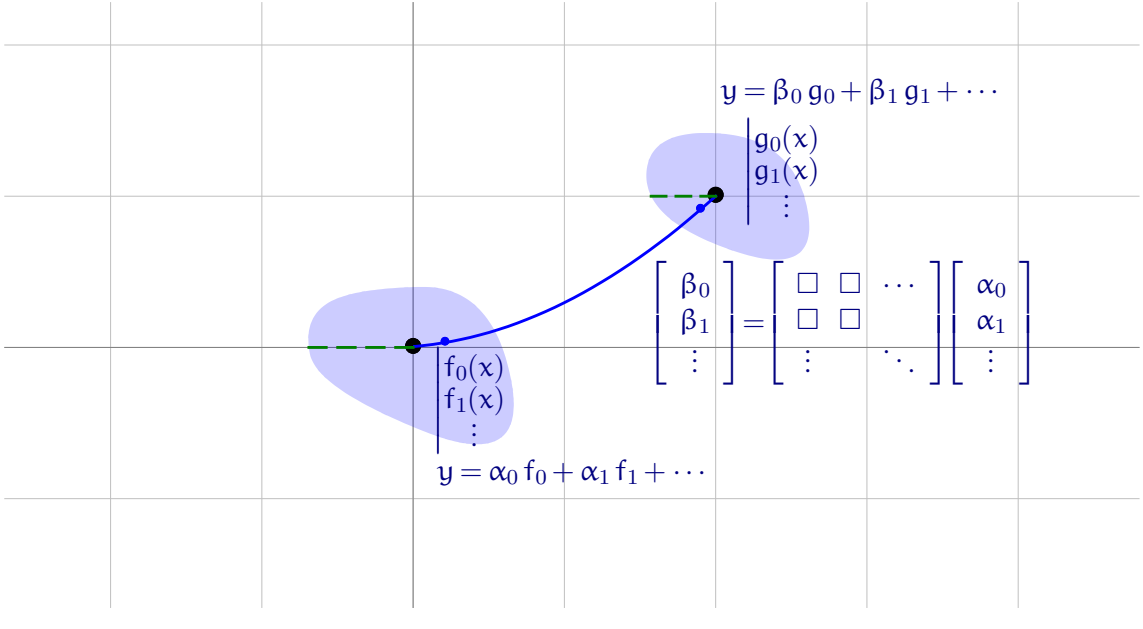



$f_0(x)$
$f_1(x)$
$\vdots$


$f_0(x)$
$f_1(x)$
$\vdots$
$y = \alpha_0 f_0 + \alpha_1 f_1 + \cdots$


$f_0(x)$
$f_1(x)$
$\vdots$
$y = \alpha_0 f_0 + \alpha_1 f_1 + \cdots$


$y = \beta_0\, g_0 + \beta_1\, g_1 + \cdots$
$g_0(x)$
$g_1(x)$
$\vdots$
$f_0(x)$
$f_1(x)$
$\vdots$
$y = \alpha_0\, f_0 + \alpha_1\, f_1 + \cdots$


$y = \beta_0\, g_0 + \beta_1\, g_1 + \cdots$
$g_0(x)$
$g_1(x)$
$\vdots$
$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \end{bmatrix} = \begin{bmatrix} \square & \square & \cdots \\ \square & \square & \\ \vdots & & \ddots \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \end{bmatrix}$
$f_0(x)$
$f_1(x)$
$\vdots$
$y = \alpha_0\, f_0 + \alpha_1\, f_1 + \cdots$

```
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()]
sage:
```

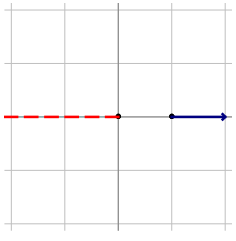
```
sage: dop = 2*x^2*(x-1)*Dx^2 + 3*x*(x-1)*Dx + x + 1

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()]

sage:
```

```
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()]

sage:
```

```
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]]

sage:
```

A Toy Example

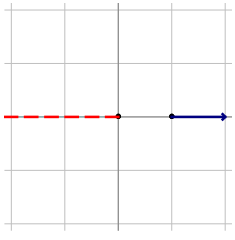
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
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])
sage:
```

A Toy Example

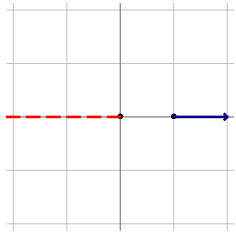
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
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])
sage:
```

A Toy Example

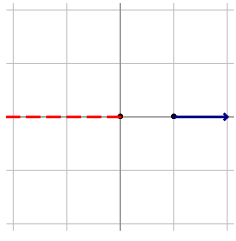
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
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] [0.69314718055994531 +/- 5.83e-19]]
[                   [+/- 5.28e-29]    [0.50000000000000000 +/- 1e-22]]
sage: log(2.)
sage: dop.numerical_transition_matrix([1, -1])
sage:
```

A Toy Example

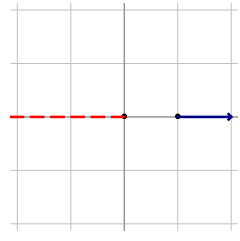
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
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] [0.69314718055994531 +/- 5.83e-19]]
[                    [+/- 5.28e-29]     [0.50000000000000000 +/- 1e-22]]
sage: log(2.)
0.693147180559945
sage: dop.numerical_transition_matrix([1, -1])
sage:
```

A Toy Example

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



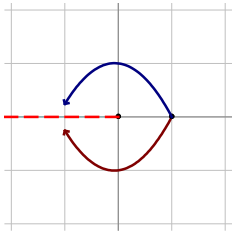
```
sage: dop = x*Dx^2 + Dx
sage: dop.local_basis_expansions(1)
[1, (x - 1) - 1/2*(x - 1)^2 + 1/3*(x - 1)^3 - 1/4*(x - 1)^4]
sage: dop.numerical_transition_matrix([1, 2])
[    [1.0000000000000000 +/- 1e-21] [0.69314718055994531 +/- 5.83e-19]]
[                   [+/- 5.28e-29]    [0.50000000000000000 +/- 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)
```

```
sage:
```

Branches of Log

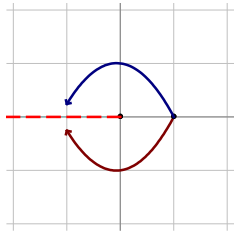
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
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)
sage:
```

Branches of Log

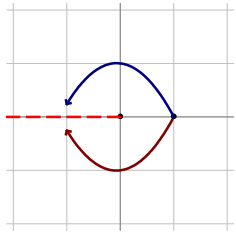
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
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)
sage:
```

Branches of Log

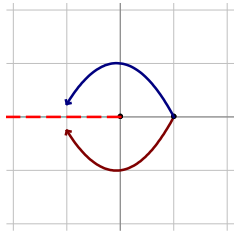
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
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)
sage:
```

Branches of Log

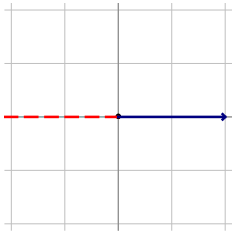
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
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]
sage:
```

Connection to Singular Points

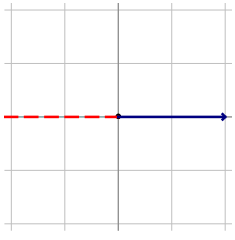
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
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)
sage:
```

Connection to Singular Points

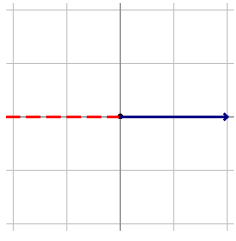
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
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)
sage:
```

Connection to Singular Points

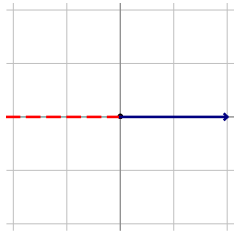
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
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])
sage: dop.local_basis_expansions(1)
sage:
```

Connection to Singular Points

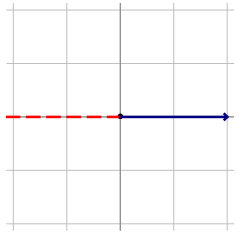
$$x\,y''(x) + y'(x) = 0 \qquad y(x) = a + b\log x$$



```
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)
sage:
```

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)
[1, (x - 1) - 1/2*(x - 1)^2 + 1/3*(x - 1)^3 - 1/4*(x - 1)^4]
sage:
```

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 $\to$ 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)
sage:
```

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)

$$\begin{aligned}
&(4x^9-13x^8+15x^7-7x^6+x^5)I^{(6)}(x)+(54x^8-140x^7+120x^6-36x^5+2x^4)I^{(5)}(x)\\
+&(202x^7-397x^6+228x^5-34x^4+x^3)I^{(4)}(x)+(222x^6-303x^5+90x^4+3x^3-3x^2)I^{(3)}(x)\\
+&(48x^5-37x^4+x^3-6x^2+6x)I''(x)+(-x^2+6x-6)I'(x) = 0
\end{aligned}$$

```
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)
sage:
```

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)

$$\begin{aligned}
&(4x^9-13x^8+15x^7-7x^6+x^5)I^{(6)}(x)+(54x^8-140x^7+120x^6-36x^5+2x^4)I^{(5)}(x)\\
+&(202x^7-397x^6+228x^5-34x^4+x^3)I^{(4)}(x)+(222x^6-303x^5+90x^4+3x^3-3x^2)I^{(3)}(x)\\
+&(48x^5-37x^4+x^3-6x^2+6x)I''(x)+(-x^2+6x-6)I'(x) = 0
\end{aligned}$$

```
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)
sage:
```

Iterated Integrals

[Ablinger, Blümlein, Raab, Schneider, 2014]

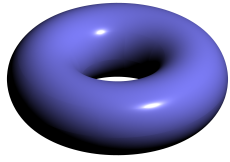
$$\mathrm{I}(x) = \int_x^1 \frac{x_5\,dx_5}{x_5-1} \int_{x_5}^1 \frac{dx_4}{x_4\sqrt{x_4-\frac{1}{4}}} \int_{x_4}^1 \frac{dx_3}{x_3\sqrt{x_3-\frac{1}{4}}} \int_{x_3}^1 \frac{dx_2}{1-x_2} \int_{x_2}^1 \frac{dx_1}{1-x_1} = ?$$

(with suitable branch choices)

$$\begin{aligned}
&(4x^9-13x^8+15x^7-7x^6+x^5)\mathrm{I}^{(6)}(x)+(54x^8-140x^7+120x^6-36x^5+2x^4)\mathrm{I}^{(5)}(x)\\
+&(202x^7-397x^6+228x^5-34x^4+x^3)\mathrm{I}^{(4)}(x)+(222x^6-303x^5+90x^4+3x^3-3x^2)\mathrm{I}^{(3)}(x)\\
+&(48x^5-37x^4+x^3-6x^2+6x)\mathrm{I}''(x)+(-x^2+6x-6)\mathrm{I}'(x) = 0
\end{aligned}$$

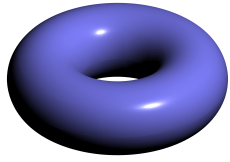
```
sage: from ore_algebra.examples.iint import diffop, h, w, f, iint_value

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)
[0.97080469562493124056011987537954344846933233520382808407829772093922068543247104
23969385436449133256949084800222962570840173198633096151138759803578207094031467 5
54659656351421824770757248719747088484824170249666207669173184105527085266856220513
37686622614078217356740250747402336641534712789952289437018411298758782819620119 36
94828021941213770178928611033372730671293830609745855925384012780946303920149774580
52310942925657352529316761168810798144011 69 +/- 9.45e-452]
sage:
```

Volumes of Semi-Algebraic Sets



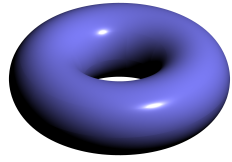
```
sage: import volume  # partial implementation with many limitations
sage: P.<X,Y,Z> = QQ[]
sage: volume.volume1(X^2+Y^2+Z^2-(1 - 2^10*(X^2*Y^2+Y^2*Z^2+Z^2*X^2)), 1000)
sage: volume.volume1((X^2+Y^2+Z^2+3)^2 - 16 * (X^2+Y^2), 1000)
sage: n((2*pi)^2)
```

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)
[0.10857542146036093773950339599420761981091787444660747544447582299328536067303292819494347441406406613662423462795980877810349323467815682039679678371920510986067273812128016753424636587445690849908126947530491722819737558442192533940241131055414354736564257531342253838423061375866757194540940154 2 +/- 3.56e-298]

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

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

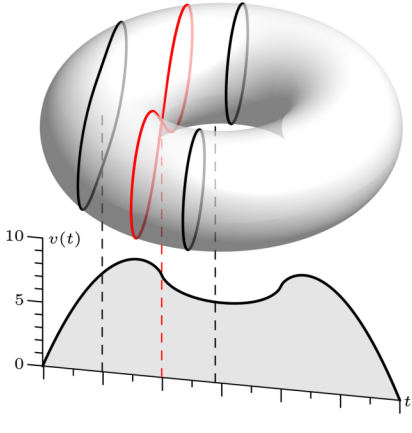
Volumes of Semi-Algebraic Sets



```
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)

[0.10857542146036093773950339599420761981091787444660747544447582299328536067303292819494347441406406613662423462795980877810349323467815682039679678371920510986067273812128016753424636587445690849908126947530491722819737558442192533940241131055414354736564257531342253838423061375866757194540940154 2 +/- 3.56e-298]

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

[39.47841760435743447533796399950460454125479762896316250565339750488017928967682097200709361487420892729610365509609257631108939248812018691044247071194079064396159942482630252228602416493136131512347741107736865975866628607617815494104711765528103304232677365006236819323927549308013230506672 +/- 2.47e-290]

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

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

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; …]

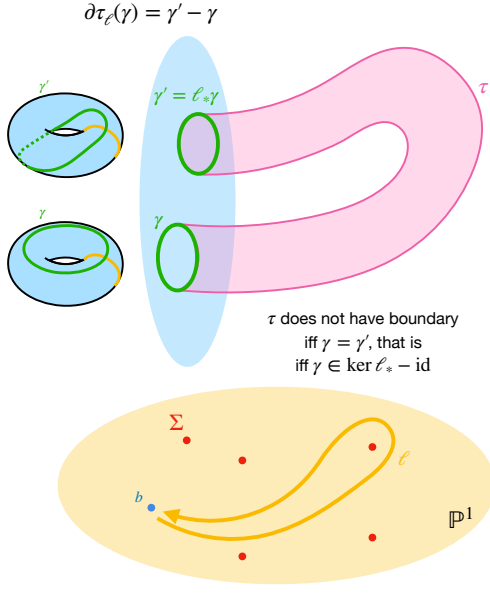
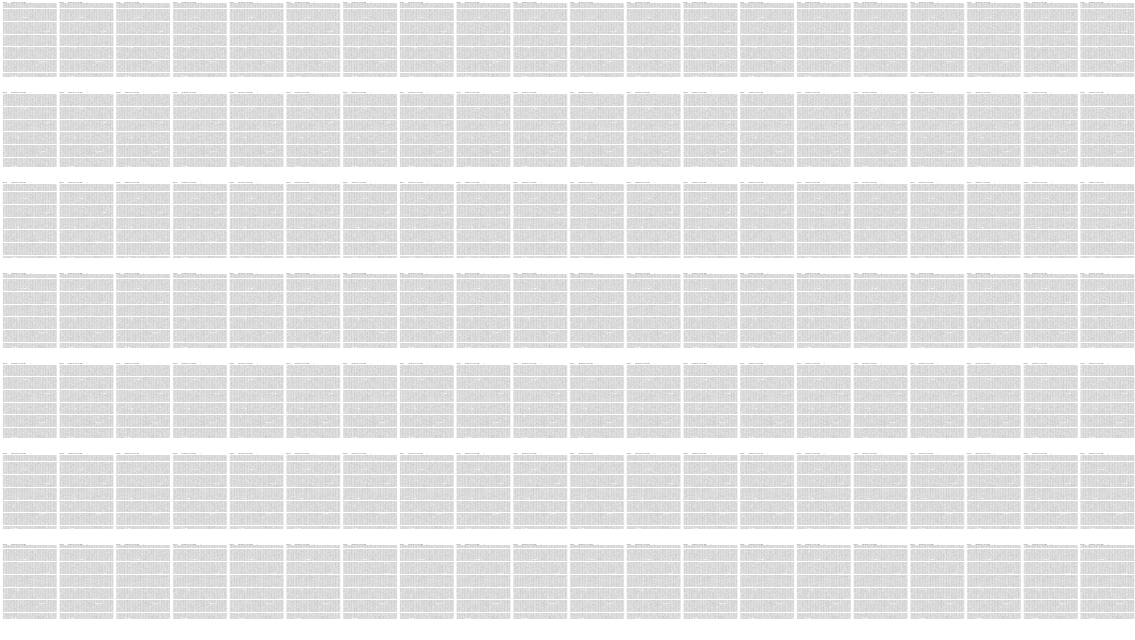
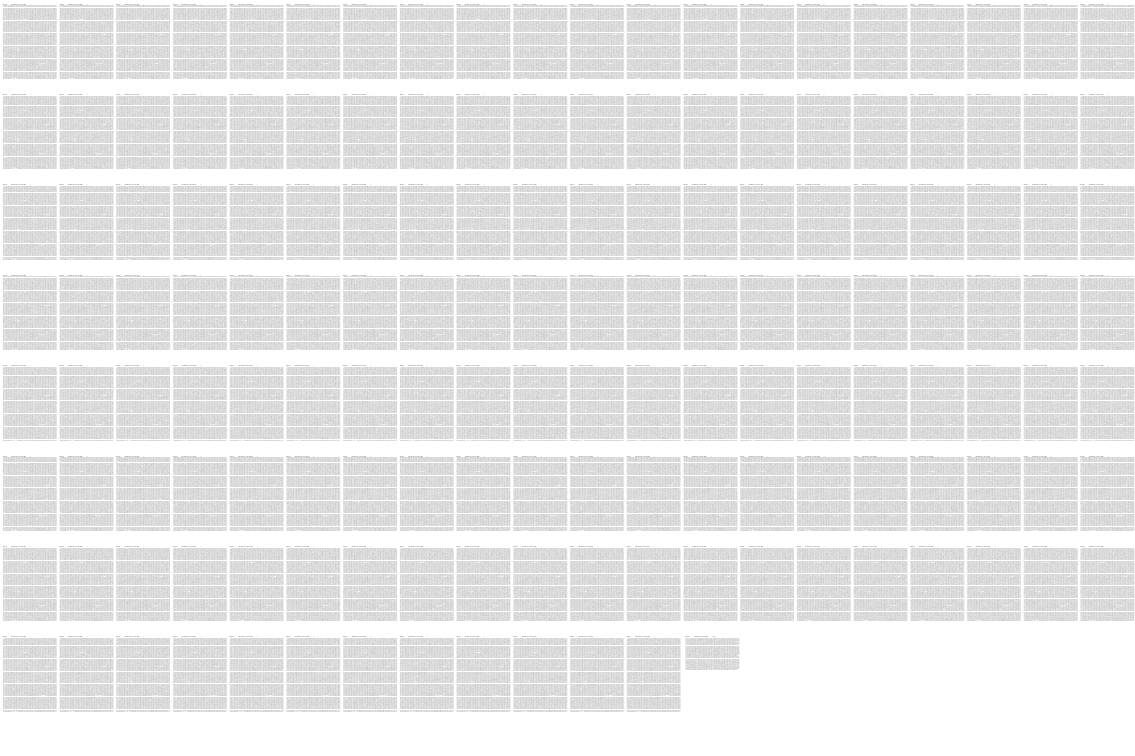


image courtesy of Eric Pichon-Pharabod

Equations can get large





Symbolic-Numeric Factoring

Theorem. [Schlesinger, 1895]

The differential Galois group of a Fuchsian differential operator

$$L = a_r(x)\, D_x^r + \cdots + a_1(x)\, D_x + a_0(x)$$

is the algebraic group generated by the monodromy matrices

(viewed as elements of $GL_r(\mathbb{C})$ acting on local solutions at a common base point x_0).

Non-Fuchsian L: add the exponential torus and the Stokes matrices. [Ramis, 1985]

Symbolic-Numeric Factoring

Theorem. [Schlesinger, 1895]

The differential Galois group of a Fuchsian differential operator

$$L = a_r(x)\, D_x^r + \cdots + a_1(x)\, D_x + a_0(x)$$

is the algebraic group generated by the monodromy matrices

(viewed as elements of $GL_r(\mathbb{C})$ acting on local solutions at a common base point x_0).

Non-Fuchsian L: add the exponential torus and the Stokes matrices. [Ramis, 1985]

Algorithm: [van der Hoeven, 2007]

- Compute numerical approximations of the monodromy matrices;
- Search for common invariant subspaces and
  - → reconstruct a right-hand factor, check exact divisibility
  - → or use error bounds to prove that none exists.

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
sage:
```

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 - 1)*Dx - 1, (-2*x - 2)*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
sage:
```

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 - 1)*Dx - 1, (-2*x - 2)*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 -
  4872727756800*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 + 229243546828 80*z^7 - 152023080960*z^5 + 156432384*z^3 - 921
6*z
sage: polya.dop[9].is_provably_irreducible()  # Maple DFactor: > 1 min 40 s
sage:
```

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 - 1)*Dx - 1, (-2*x - 2)*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 + (-387144769536000*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 -
  4872727756800*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 + 229243546828 80*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
sage:
```

code by Alexandre Goyer

Stokes Matrices

[Loday-Richaud, M., Remy, 2026]

For irregular singularities of pure level 1, computing the Stokes matrices reduces to regular singular connection problems.

```
sage: from ore_algebra.analytic.stokes import stokes_dict
sage: dop = 2*(x^2*Dx)^3 + (x-5)*(x^2*Dx)^2 + (2*x+2)*(x^2*Dx) - 2*x
sage: dop.generalized_series_solutions()
sage: stokes = stokes_dict(dop)
sage: stokes.keys()
sage: [list(row) for row in stokes[1]]
sage:
```

Stokes Matrices

[Loday-Richaud, M., Remy, 2026]

For irregular singularities of pure level 1, computing the Stokes matrices reduces to regular singular connection problems.

```
sage: from ore_algebra.analytic.stokes import stokes_dict

sage: dop = 2*(x^2*Dx)^3 + (x-5)*(x^2*Dx)^2 + (2*x+2)*(x^2*Dx) - 2*x

sage: dop.generalized_series_solutions()
[exp(-1/2*x^(-1))*x^(-1/2)*(1 - 2/3*x - 1/3*x^2 - 4/9*x^3 - 25/27*x^4 + O(x^5)),
 exp(-2*x^(-1))*x^(-1)*(1 + x + O(x^5)),
 x*(1 + 4*x + 45/2*x^2 + 333/2*x^3 + 3087/2*x^4 + O(x^5))]

sage: stokes = stokes_dict(dop)

sage: stokes.keys()

sage: [list(row) for row in stokes[1]]

sage:
```

Stokes Matrices

[Loday-Richaud, M., Remy, 2026]

For irregular singularities of pure level 1, computing the Stokes matrices reduces to regular singular connection problems.

```
sage: from ore_algebra.analytic.stokes import stokes_dict

sage: dop = 2*(x^2*Dx)^3 + (x-5)*(x^2*Dx)^2 + (2*x+2)*(x^2*Dx) - 2*x

sage: dop.generalized_series_solutions()
[exp(-1/2*x^(-1))*x^(-1/2)*(1 - 2/3*x - 1/3*x^2 - 4/9*x^3 - 25/27*x^4 + O(x^5)),
 exp(-2*x^(-1))*x^(-1)*(1 + x + O(x^5)),
 x*(1 + 4*x + 45/2*x^2 + 333/2*x^3 + 3087/2*x^4 + O(x^5))]

sage: stokes = stokes_dict(dop)

sage: stokes.keys()
dict_keys([1, -1])

sage: [list(row) for row in stokes[1]]

sage:
```

Stokes Matrices

[Loday-Richaud, M., Remy, 2026]

For irregular singularities of pure level 1, computing the Stokes matrices reduces to regular singular connection problems.

```
sage: from ore_algebra.analytic.stokes import stokes_dict

sage: dop = 2*(x^2*Dx)^3 + (x-5)*(x^2*Dx)^2 + (2*x+2)*(x^2*Dx) - 2*x

sage: dop.generalized_series_solutions()
[exp(-1/2*x^(-1))*x^(-1/2)*(1 - 2/3*x - 1/3*x^2 - 4/9*x^3 - 25/27*x^4 + O(x^5)),
 exp(-2*x^(-1))*x^(-1)*(1 + x + O(x^5)),
 x*(1 + 4*x + 45/2*x^2 + 333/2*x^3 + 3087/2*x^4 + O(x^5))]

sage: stokes = stokes_dict(dop)

sage: stokes.keys()
dict_keys([1, -1])

sage: [list(row) for row in stokes[1]]
[[1.0000000000000000000000, 0, 0],
 [[+/- 6.39e-23] + [-4.4562280437884453376 +/- 3.50e-20]*I,
  1.0000000000000000000000,
  0],
 [[4.8367983046245809349 +/- 3.46e-20] + [+/- 2.84e-23]*I,
  [+/- 1.28e-23] + [2.17080376367480297809 +/- 3.50e-21]*I,
  1.0000000000000000000000]]

sage:
```

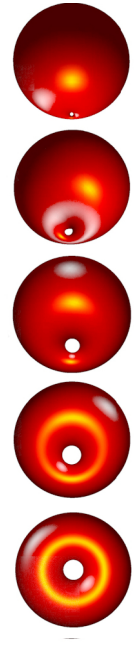
Sequence Positivity via Asymptotics

- **Yu–Chen** (2022) reduced a certain problem about minimal surfaces to the **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)

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



Sequence Positivity via Asymptotics

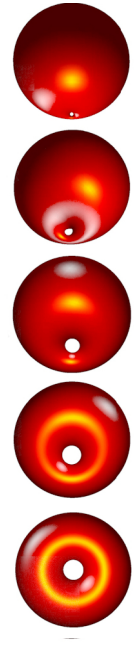
- **Yu–Chen** (2022) reduced a certain problem about minimal surfaces to the **complete positivity** of a sequence

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

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

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

- 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) reduced a certain problem about minimal surfaces to the **complete positivity** of a sequence

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

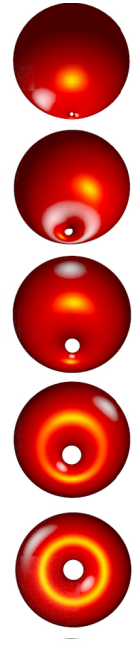
given by an **explicit recurrence relation** (equivalently, ODE)

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

- 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 $n \geqslant 1000$.



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)

sage:
```

original 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)

1.000000000000000*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))

sage:
```

original code by Ruiwen Dong