

Rigorous Numerics for Differentially Finite Functions

Marc Mezzarobba
CNRS, Sorbonne Université

Computing with D-Modules II, Leipzig
September 3, 2019



Try the implementation!

With Sage: <http://marc.mezzarobba.net/leipzig2019.ipynb>

Online: <http://marc.mezzarobba.net/oademo>

Introduction

ODE Solving from a Computer Algebra Perspective

Problem

Starting from a linear differential equation

$$p_r(z) \mathbf{y}^{(r)}(z) + \cdots + p_1(z) \mathbf{y}'(z) + p_0(z) \mathbf{y}(z) = 0$$

with polynomial coefficients p_0, \dots, p_r and initial values,
compute the solution at a given point.

I.e.: Evaluate a (univariate) holonomic function.

ODE Solving from a Computer Algebra Perspective

Problem

Starting from a linear differential equation

$$p_r(z) \mathbf{y}^{(r)}(z) + \dots + p_1(z) \mathbf{y}'(z) + p_0(z) \mathbf{y}(z) = 0$$

with polynomial coefficients p_0, \dots, p_r and initial values, compute the solution at a given point.

I.e.: Evaluate a (univariate) holonomic function.

Our requirements:

- ▶ Complex variables: $z \in \mathbb{C}$
- ▶ Arbitrary precision
- ▶ Rigorous error bounds (→ use in algebraic algorithms, in proofs)
- ▶ **Singular cases**

Applications

► Special functions

► Combinatorics

via **generating functions** and **singularity analysis**

random walks on lattices,
asymptotics of P-recursive sequences...

► Numerical (Real) Algebraic Geometry

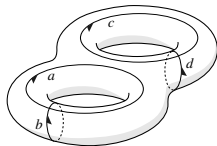
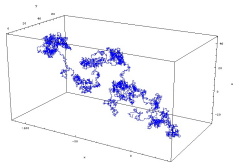
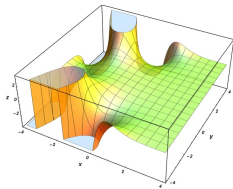
via **Picard-Fuchs equations**

periods of surfaces [Sertöz 2019, ...],
volumes of semi-algebraic sets [Lairez, M., Safey 2019]...

► “Numerical differential Galois theory”

via **connection / monodromy / Stokes matrices**

operator factoring, heuristic diff. Galois groups
[van der Hoeven 2007, ...]



$$g = \mathcal{L}(\hat{B}(\hat{g}))$$

Applications

► Special functions

► Combinatorics

via **generating functions** and **singularity analysis**

random walks on lattices,
asymptotics of P-recursive sequences...

► Numerical (Real) Algebraic Geometry

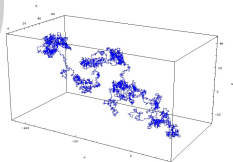
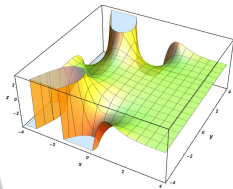
via **Picard-Fuchs equations**

periods of surfaces [Sertöz 2019, ...],
volumes of semi-algebraic sets [Lairez, M., Safey 2019]...

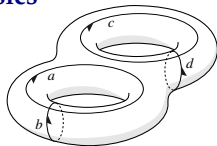
► “Numerical differential Galois theory”

via **connection / monodromy / Stokes matrices**

operator factoring, heuristic diff. Galois groups
[van der Hoeven 2007, ...]



Math. physics

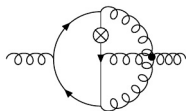


$$g = \mathcal{L}(\hat{B}(\hat{g}))$$

Iterated Integrals

arising from Feynman diagram calculations

[Ablinger, Blümlein, Raab, Schneider, 2014]



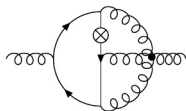
$$\int_0^1 \frac{x_5 dx_5}{x_5 - 1} \int_{x_5}^1 \frac{dx_4}{x_4 \sqrt{x_4 - \frac{1}{4}}} \int_{x_4}^1 \frac{dx_3}{x_3 \sqrt{x_3 - \frac{1}{4}}} \int_{x_3}^1 \frac{dx_2}{1 - x_2} \int_{x_2}^1 \frac{dx_1}{1 - x_1} = ?$$

(with suitable branch choices)

Iterated Integrals

arising from Feynman diagram calculations

[Ablinger, Blümlein, Raab, Schneider, 2014]



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

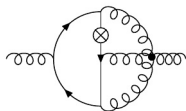
(with suitable branch choices)

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

Iterated Integrals

arising from Feynman diagram calculations

[Ablinger, Blümlein, Raab, Schneider, 2014]



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

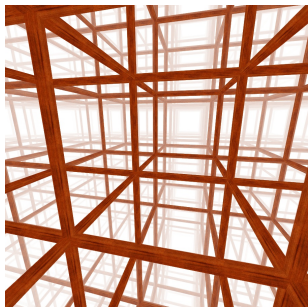
(with suitable branch choices)

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

`sage: iint_value(dop, myini, 1e-500)`

`[0.9708046956249312405 ... 59027603834204946 +/- 9.05e-501]`

Pólya Walks



For a random walk on \mathbb{Z}^d ($d \geq 3$) starting at 0:

$$\text{return probability} = 1 - \frac{1}{w(1/2d)}$$

where

$$w(z) = \sum_{n=0}^{\infty} w_n z^n$$

#walks of length n
ending at origin

satisfies an LODE with polynomial coefficients

Examples:

$$d=3 \quad z^2(4z^2-1)(36z^2-1)D^3 + (1296z^5 - 240z^3 + 3z)D^2 \\ + (2592z^4 - 288z^2 + 1)D + 864z^3 - 48z$$

$$d=4 \quad (1024z^7 - 80z^5 + z^3)D^4 + (14336z^6 - 800z^4 + 6z^2)D^3 \\ + (55296z^5 - 2048z^3 + 7z)D^2 + (61440z^4 - 1344z^2 + 1)D \\ + 12288z^3 - 128z$$

...

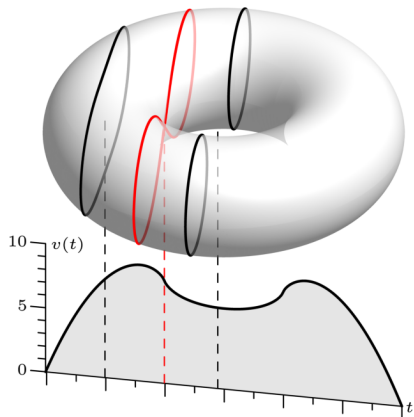
(thanks to B. Salvy)

First return after n steps:

$$f(z) = \sum_{n=0}^{\infty} f_n z^n$$
$$f\left(\frac{1}{2d}\right) = \sum_{n=0}^{\infty} \frac{f_n}{(2d)^n}$$
$$w(z) = 1 + f(z)w(z)$$

Volumes of Compact Semi-Algebraic Sets

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



- The “**slice volume**” function satisfies a Picard-Fuchs eqn
 - Except at **critical values** of the projection, it is analytic
- Compute initial values by recursive calls, integrate the equation

Cost for p digits = $\tilde{O}(p)$

Implementation

ore_algebra

mkauers / ore_algebra

Watch 7

Star 5

Fork 5

Code

Issues 1

Pull requests 0

Projects 0

Security

Insights



GNU GPL v2+

No description, website, or topics provided.

952 commits

2 branches

3 releases

5 contributors

GPL-2.0

Branch: master

New pull request

mezzarobba	test fixes for the upcoming sage 8.8 release
doc	0.4
papers	Issac2019: typo
src/ore_algebra	test fixes for the upcoming sage 8.8 release
.gitignore	update .gitignore



Contributors

- **M. Kauers** – main author
- **M. Jarschek, F. Johansson** – initial implementation
- **MM** – numerics + misc
- **C. Hofstadler, S. Schwaiger** – D-finite function objects




```
$ sage -pip install \
git+https://github.com/mkauers/ore_algebra.git
```

ore_algebra: An Implementation of Ore Polynomials


Ore polynomials are **skew polynomials** that can be used to model linear functional operators.

Examples

► $K(z)\langle D \rangle = \{ \text{skew polynomials} \\ \text{in } D \text{ over } K(z) \\ \text{subject to } D z = z D + 1 \}$
 $\cong \{ \text{differential operators} \}$


$$[z f(z)]' = z f'(z) + f(z)$$

► $K(n)\langle S \rangle = \{ \text{skew polynomials} \\ \text{in } S \text{ over } K(n) \\ \text{subject to } S n = (n + 1) S \}$
 $\cong \{ \text{recurrence operators} \}$


$$[n u(n)]_{k+1} = [(n + 1) u(n + 1)]_k$$

Features

- Basic arithmetic (diff, shift, qdiff, qshift, custom)
- Gcrd, lcm, D-finite closure (incl. multivariate)
- Creative telescoping
- Polynomial, rational, generalized series solutions
- **Numerical connection** (diff.)
- Desingularization
- Guessing
- ...

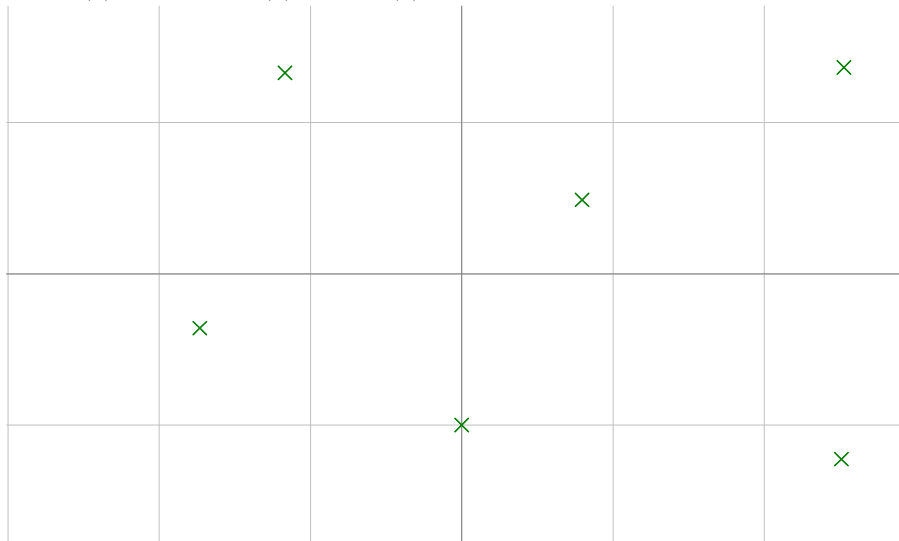
Analytic Continuation

$$\mathcal{L} = a_r(z) D_z^r + \cdots + a_1(z) D_z + a_0(z)$$



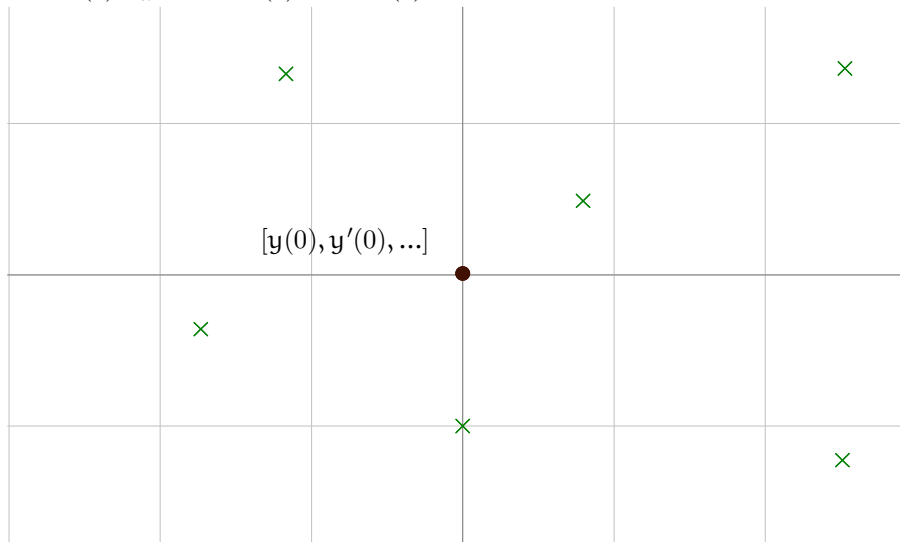
Analytic Continuation

$$\mathcal{L} = a_r(z) D_z^r + \cdots + a_1(z) D_z + a_0(z)$$



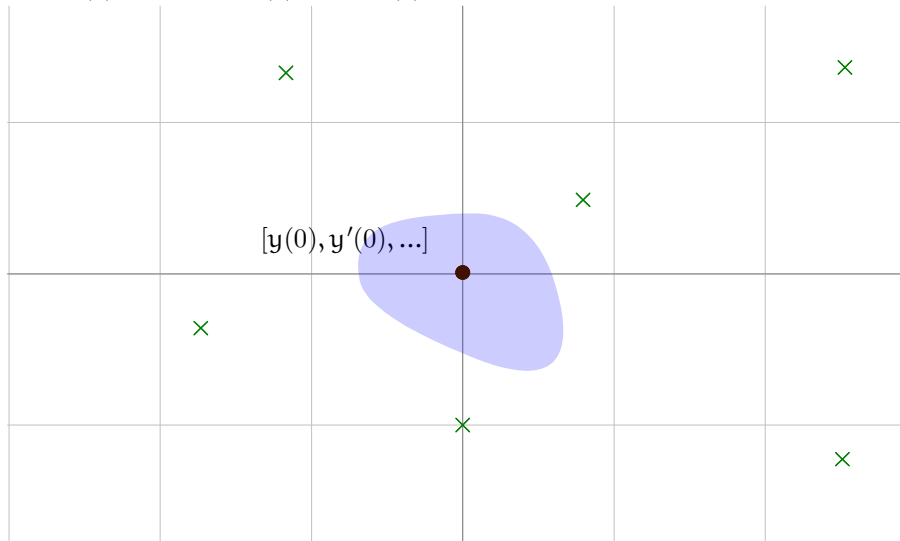
Analytic Continuation

$$\mathcal{L} = a_r(z) D_z^r + \cdots + a_1(z) D_z + a_0(z)$$



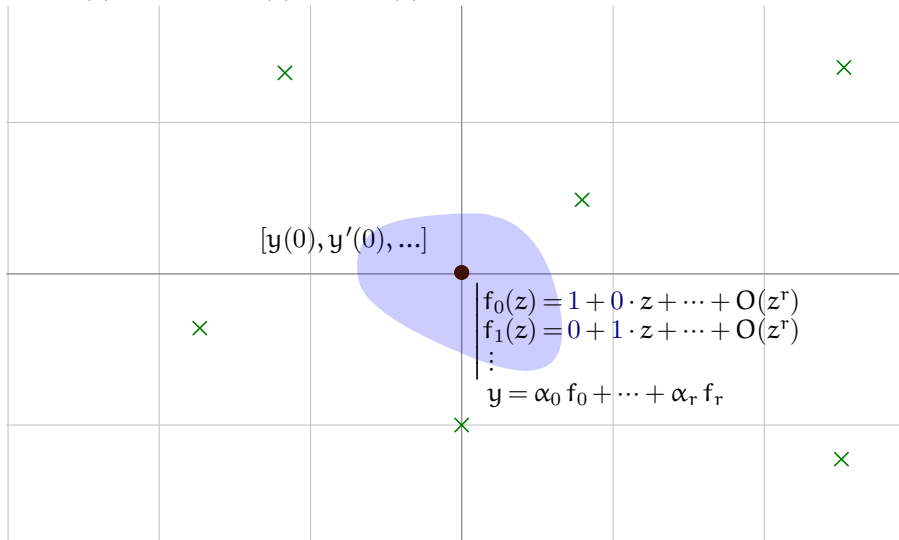
Analytic Continuation

$$\mathcal{L} = a_r(z) D_z^r + \cdots + a_1(z) D_z + a_0(z)$$



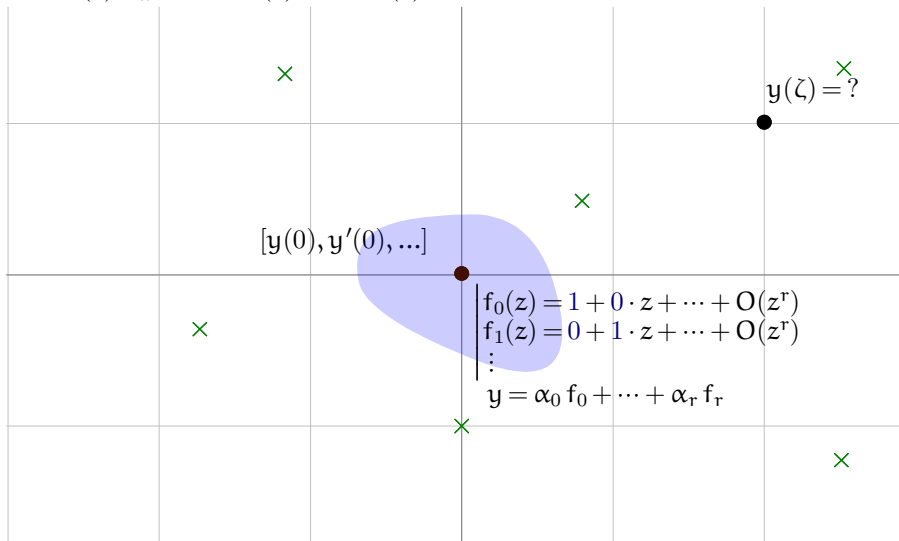
Analytic Continuation

$$\mathcal{L} = a_r(z) D_z^r + \cdots + a_1(z) D_z + a_0(z)$$



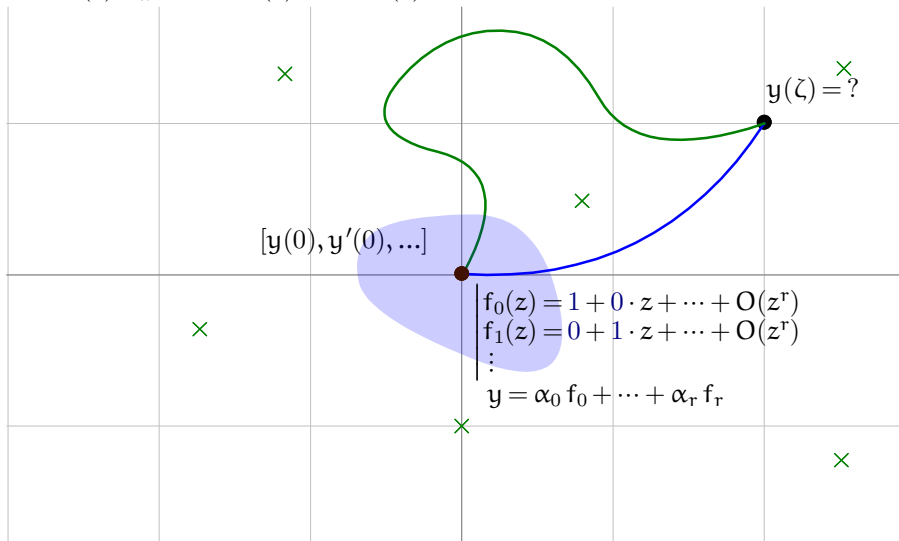
Analytic Continuation

$$\mathcal{L} = a_r(z) D_z^r + \dots + a_1(z) D_z + a_0(z)$$



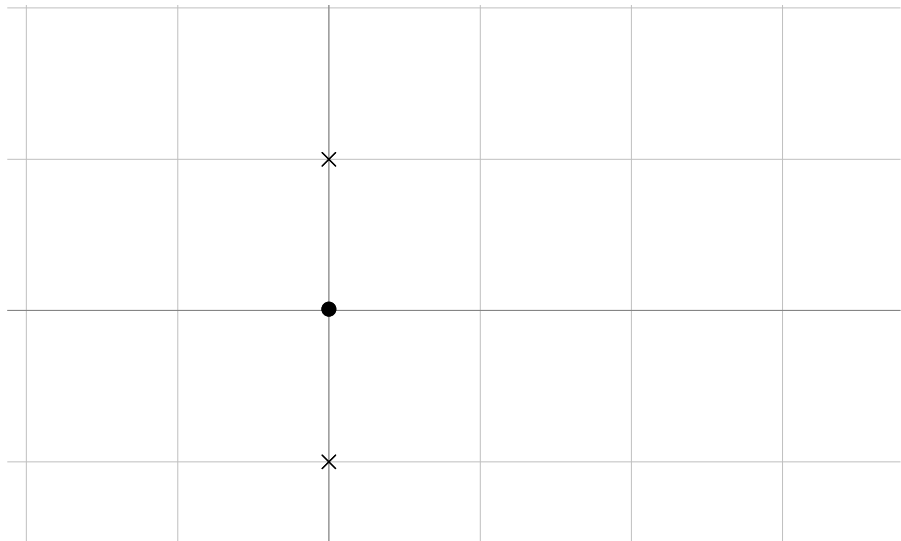
Analytic Continuation

$$\mathcal{L} = a_r(z) D_z^r + \dots + a_1(z) D_z + a_0(z)$$



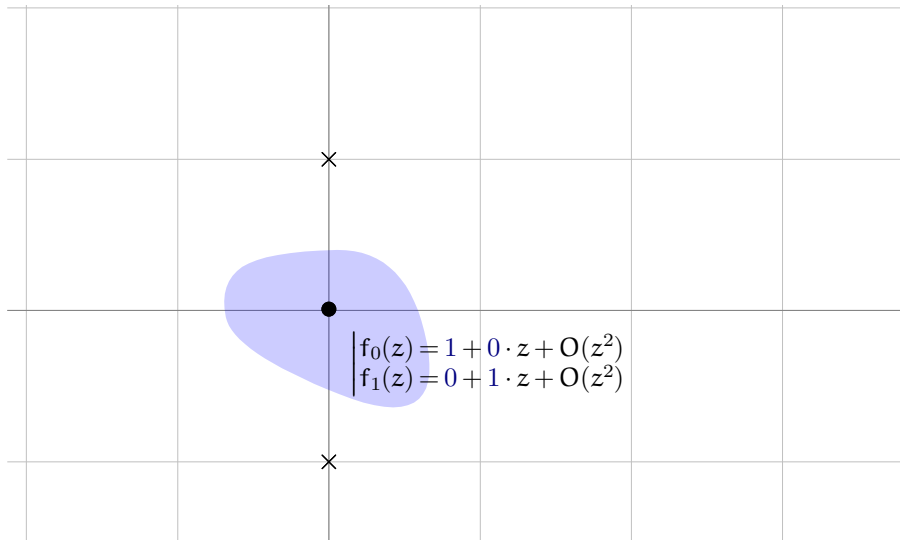
Transition Matrices

$$(z^2 + 1) y''(z) + 2z y'(z) = 0$$



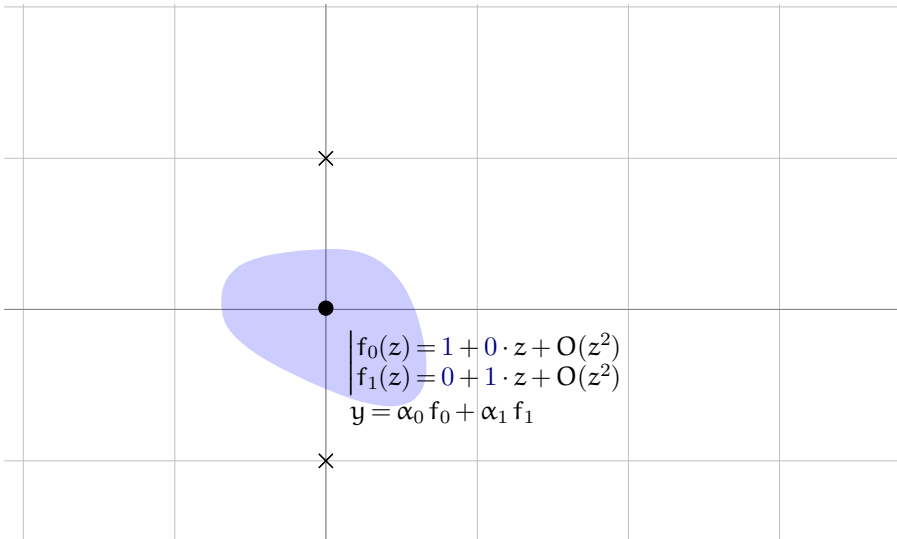
Transition Matrices

$$(z^2 + 1) y''(z) + 2z y'(z) = 0$$



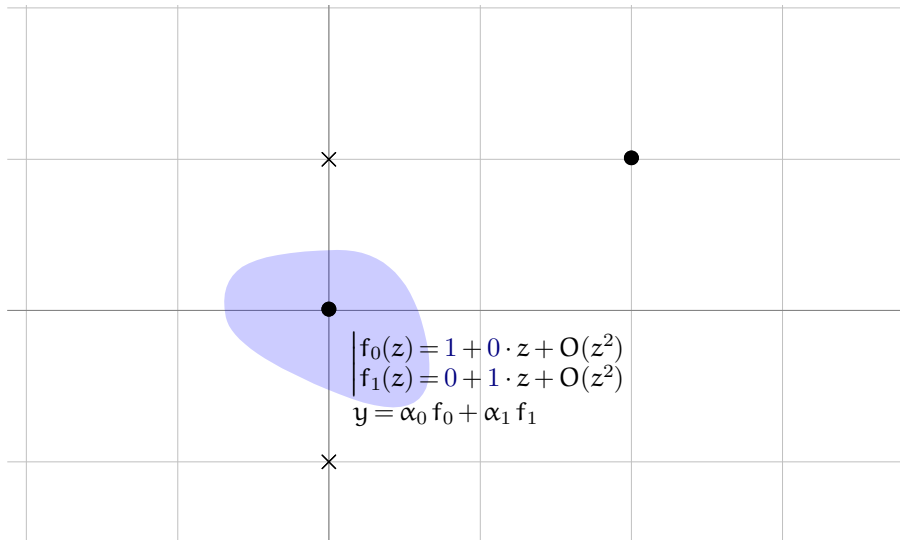
Transition Matrices

$$(z^2 + 1) y''(z) + 2z y'(z) = 0$$



Transition Matrices

$$(z^2 + 1) y''(z) + 2z y'(z) = 0$$



Transition Matrices

$$(z^2 + 1) y''(z) + 2z y'(z) = 0$$

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

$$\begin{cases} g_0(z) = 1 + 0 \cdot (z - z_0) + O((z - z_0)^2) \\ g_1(z) = 0 + 1 \cdot (z - z_0) + O((z - z_0)^2) \end{cases}$$

$$\begin{cases} f_0(z) = 1 + 0 \cdot z + O(z^2) \\ f_1(z) = 0 + 1 \cdot z + O(z^2) \end{cases}$$
$$y = \alpha_0 f_0 + \alpha_1 f_1$$

x

x

Transition Matrices

$$(z^2 + 1) y''(z) + 2z y'(z) = 0$$

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

$$\begin{cases} g_0(z) = 1 + 0 \cdot (z - z_0) + O((z - z_0)^2) \\ g_1(z) = 0 + 1 \cdot (z - z_0) + O((z - z_0)^2) \end{cases}$$

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

$$\begin{cases} f_0(z) = 1 + 0 \cdot z + O(z^2) \\ f_1(z) = 0 + 1 \cdot z + O(z^2) \end{cases}$$
$$y = \alpha_0 f_0 + \alpha_1 f_1$$

Transition Matrices

$$(z^2 + 1) y''(z) + 2z y'(z) = 0$$

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

$$\begin{cases} g_0(z) = 1 + 0 \cdot (z - z_0) + O((z - z_0)^2) \\ g_1(z) = 0 + 1 \cdot (z - z_0) + O((z - z_0)^2) \end{cases}$$

$$\begin{bmatrix} y(z_0) \\ y'(z_0) \end{bmatrix} = \begin{bmatrix} \square & \square \\ \square & \square \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

$$\begin{cases} f_0(z) = 1 + 0 \cdot z + O(z^2) \\ f_1(z) = 0 + 1 \cdot z + O(z^2) \end{cases}$$
$$y = \alpha_0 f_0 + \alpha_1 f_1$$

x

x

Transition Matrices

$$(z^2 + 1) y''(z) + 2z y'(z) = 0$$

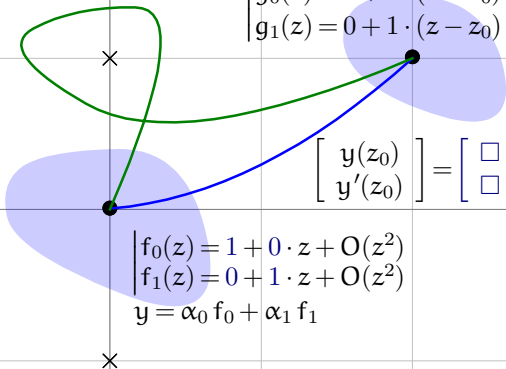
$$\begin{bmatrix} \tilde{y}(z_0) \\ \tilde{y}'(z_0) \end{bmatrix} = \begin{bmatrix} \square & \square \\ \square & \square \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

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

$$\begin{cases} g_0(z) = 1 + 0 \cdot (z - z_0) + O((z - z_0)^2) \\ g_1(z) = 0 + 1 \cdot (z - z_0) + O((z - z_0)^2) \end{cases}$$

$$\begin{bmatrix} y(z_0) \\ y'(z_0) \end{bmatrix} = \begin{bmatrix} \square & \square \\ \square & \square \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

$$\begin{cases} f_0(z) = 1 + 0 \cdot z + O(z^2) \\ f_1(z) = 0 + 1 \cdot z + O(z^2) \end{cases}$$
$$y = \alpha_0 f_0 + \alpha_1 f_1$$



Regular Singular Points

Theorem

[Fuchs, 1866]

Assume that 0 is a regular singular point. Then, for some neighborhood D of 0, there exists a basis of solutions defined on $D \setminus \mathbb{R}_{\leq 0}$ of the form

$$z^\lambda (y_0(z) + y_1(z) \log z + \cdots + y_t(z) \log^t z), \quad \lambda \in \bar{\mathbb{Q}}, \quad y_i \text{ analytic on } D.$$

Thus $y(z) = \sum_{\nu \in \Lambda} \sum_{k=0}^t \mathbf{y}_{\nu,k} z^\nu \frac{\log^k z}{k!}$ where $\Lambda = (\lambda_1 + \mathbb{N}) \cup \cdots \cup (\lambda_\ell + \mathbb{N}) \subset \mathbb{C}$

“Canonical” basis: $\left[z^\nu \frac{\log^k z}{k!} \right]^\vee$, ν root of mult. $> k$ of indicial polynomial
(compare $[z^n]^\vee$, $0 \leq n < r$, at ordinary points)

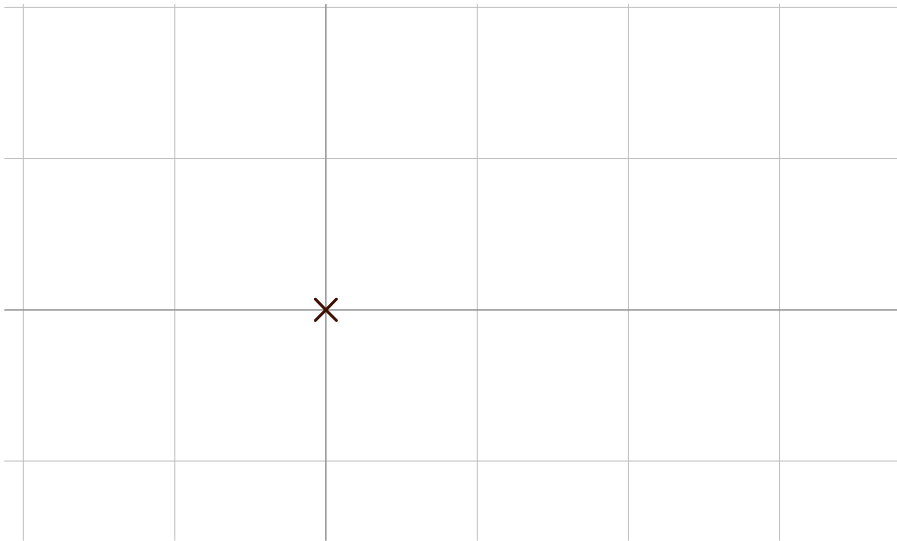
✓ $y(z) \sim z^{-3/2} \log z$

✓ $y(z) \sim z^{i\sqrt{2}}$

✗ $y(z) \sim e^{\pm 1/z}$

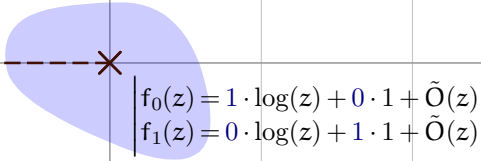
Initial Conditions at Regular Singular Points

$$\mathcal{L} = z D_z^2 + D_z + z \quad (J_0, Y_0)$$



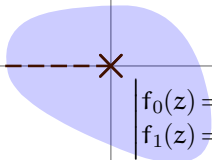
Initial Conditions at Regular Singular Points

$$\mathcal{L} = z D_z^2 + D_z + z \quad (J_0, Y_0)$$


$$\left| \begin{array}{l} f_0(z) = 1 \cdot \log(z) + 0 \cdot 1 + \tilde{O}(z) \\ f_1(z) = 0 \cdot \log(z) + 1 \cdot 1 + \tilde{O}(z) \end{array} \right.$$

Initial Conditions at Regular Singular Points

$$\mathcal{L} = z D_z^2 + D_z + z \quad (J_0, Y_0)$$



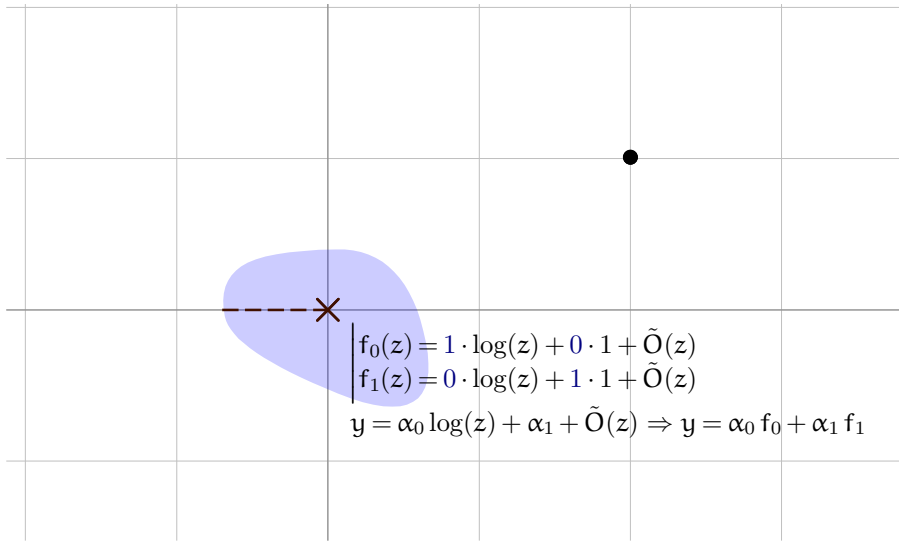
$$f_0(z) = 1 \cdot \log(z) + 0 \cdot 1 + \tilde{O}(z)$$

$$f_1(z) = 0 \cdot \log(z) + 1 \cdot 1 + \tilde{O}(z)$$

$$y = \alpha_0 \log(z) + \alpha_1 + \tilde{O}(z) \Rightarrow y = \alpha_0 f_0 + \alpha_1 f_1$$

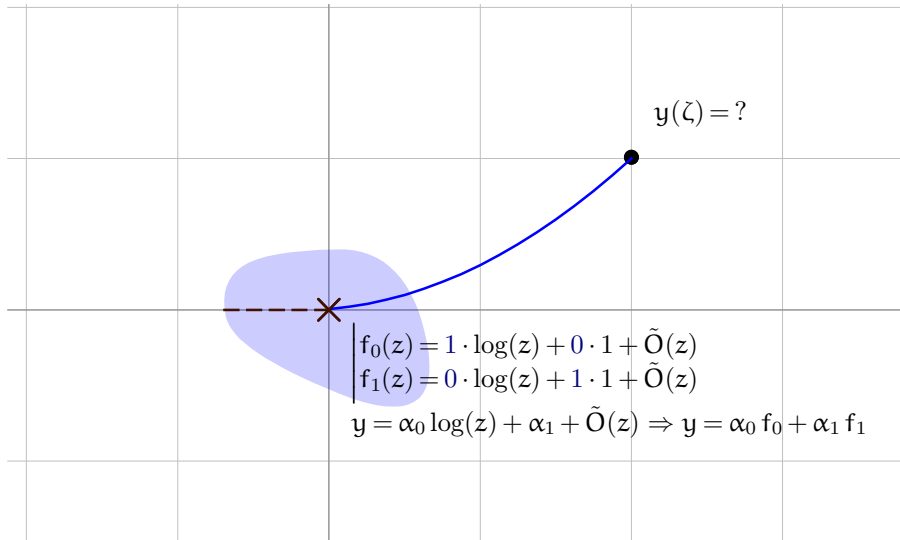
Initial Conditions at Regular Singular Points

$$\mathcal{L} = z D_z^2 + D_z + z \quad (J_0, Y_0)$$

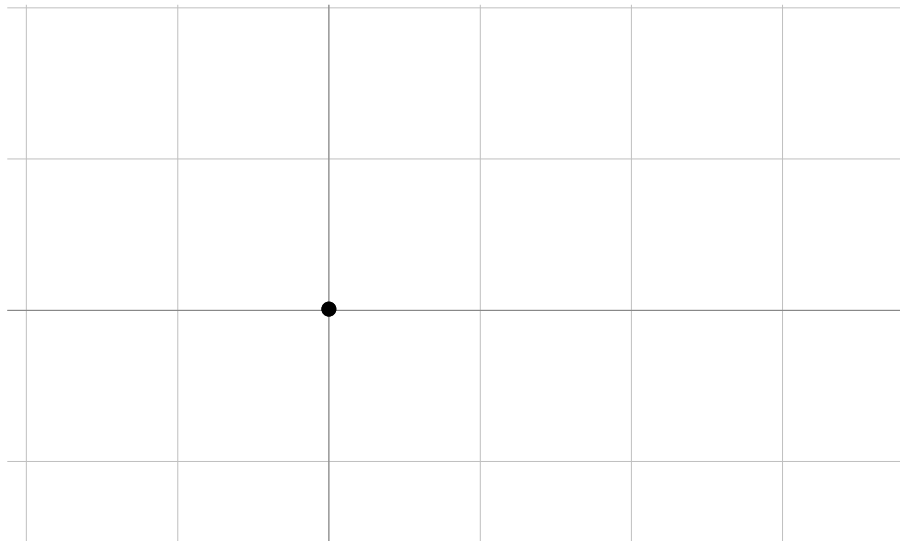


Initial Conditions at Regular Singular Points

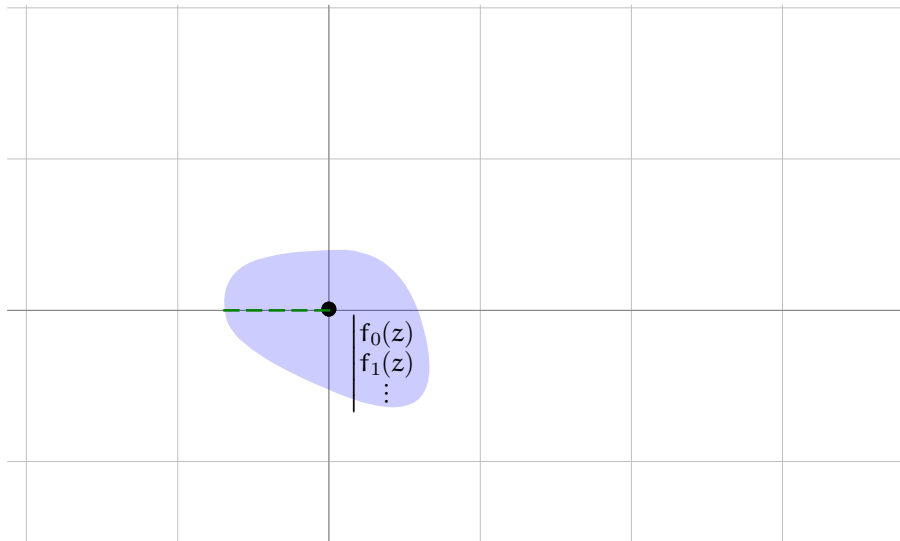
$$\mathcal{L} = z D_z^2 + D_z + z \quad (J_0, Y_0)$$



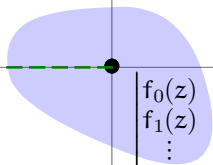
Regular Singular Transition Matrices



Regular Singular Transition Matrices



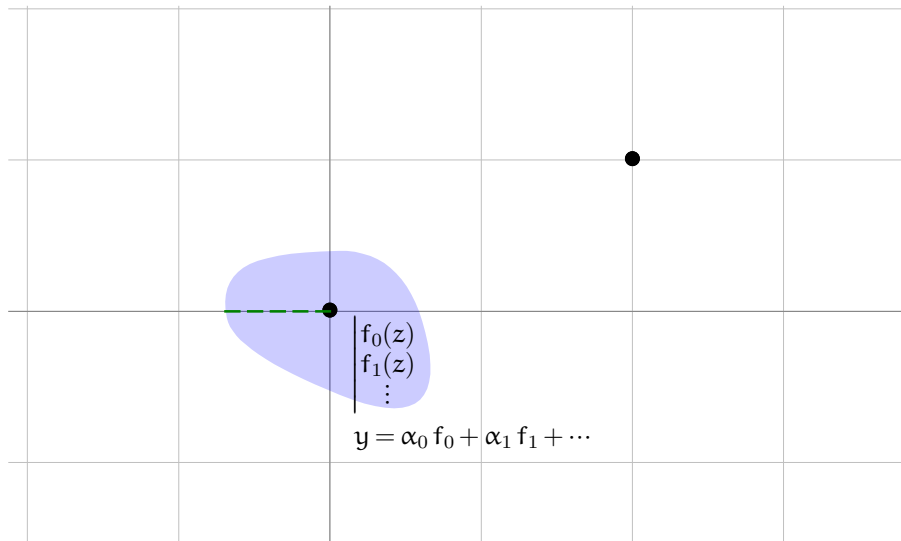
Regular Singular Transition Matrices



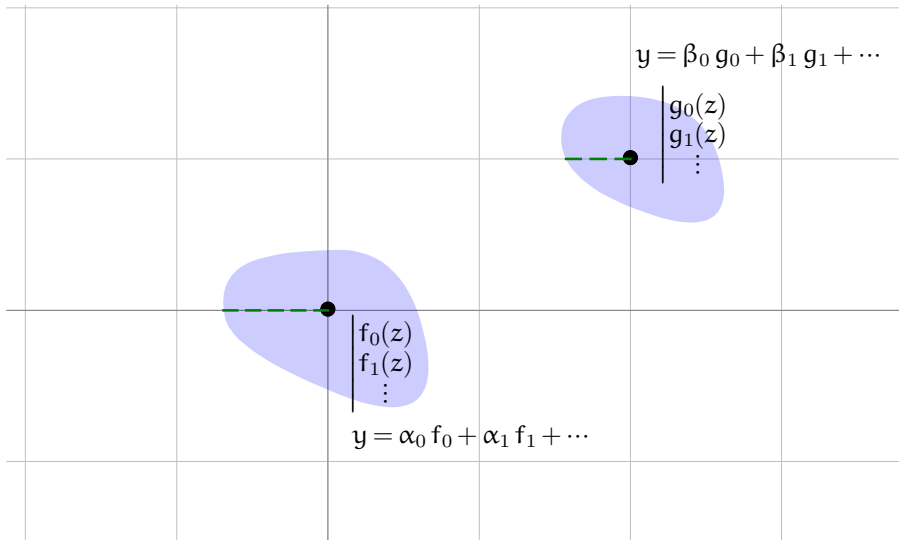
$$\begin{array}{|l} f_0(z) \\ f_1(z) \\ \vdots \end{array}$$

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

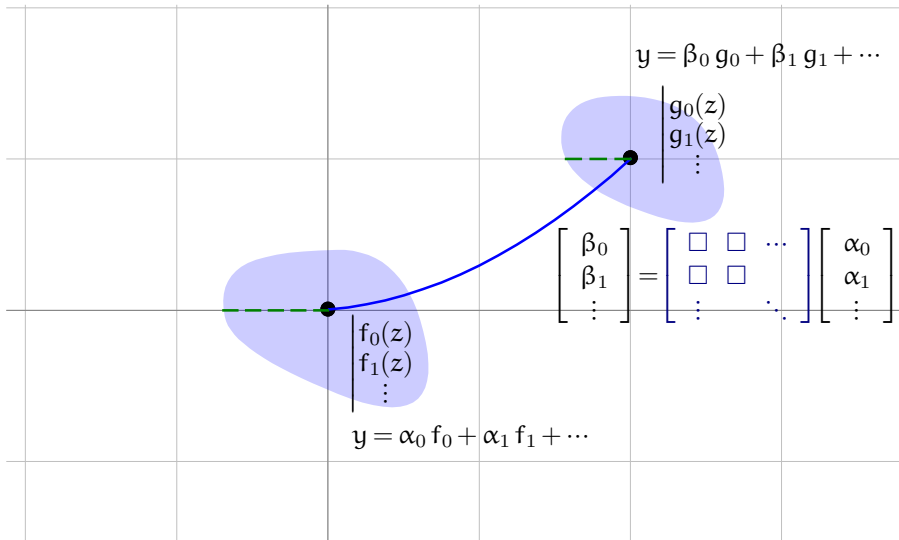
Regular Singular Transition Matrices



Regular Singular Transition Matrices

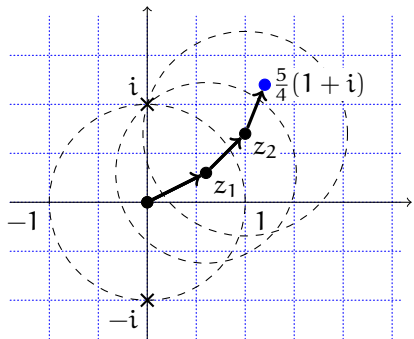


Regular Singular Transition Matrices



Algorithms

A Taylor Series Method



$$\arctan\left(\frac{5}{4}(1+i)\right) ?$$

$$\begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix} = \begin{bmatrix} 1 & 0.57\dots + 0.22\dots \\ 0 & 0.72\dots - 0.20\dots \end{bmatrix} \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix}$$

$$\begin{bmatrix} y(z_2) \\ y'(z_2) \end{bmatrix} = \begin{bmatrix} 1 & 0.39\dots + 0.24\dots \\ 0 & 0.57\dots - 0.29\dots \end{bmatrix} \begin{bmatrix} y(z_1) \\ y'(z_1) \end{bmatrix}$$

...

- ▶ Locally, the solutions are given by **convergent power series** (Cauchy)
- ▶ **Sum the series** numerically to get “initial values” at a new point
- ▶ Large steps (\propto radius of convergence)
- ▶ Extends to the regular singular case

Recurrences

The **Taylor coefficients** of a D-finite function $y(z) = \sum_{n=0}^{\infty} y_n z^n$ obey a linear **recurrence relation** with polynomial coefficients:

$$b_s(n) y_{n+s} + \cdots + b_1(n) y_{n+1} + b_0(n) y_n = 0.$$

(And conversely, for D-finite formal power series.)

Recurrences

The **Taylor coefficients** of a D-finite function $y(z) = \sum_{n=0}^{\infty} y_n z^n$ obey a linear **recurrence relation** with polynomial coefficients:

$$b_s(n) y_{n+s} + \cdots + b_1(n) y_{n+1} + b_0(n) y_n = 0.$$

(And conversely, for D-finite formal power series.)

Proof.

$$y = \sum_{n=-\infty}^{\infty} y_n z^n \quad \leftrightarrow \quad Y = (y_n)_{n \in \mathbb{Z}}$$

$$\mathbf{D} \cdot y = \sum_{n=-\infty}^{\infty} (n+1) y_{n+1} z^n \quad \leftrightarrow \quad (\mathbf{S}n) \cdot Y$$

$$\mathbf{z} \cdot y = \sum_{n=-\infty}^{\infty} y_{n-1} z^n \quad \leftrightarrow \quad \mathbf{S}^{-1} \cdot Y$$

Recurrences

The **Taylor coefficients** of a D-finite function $y(z) = \sum_{n=0}^{\infty} y_n z^n$ obey a linear **recurrence relation** with polynomial coefficients:

$$b_s(n) y_{n+s} + \dots + b_1(n) y_{n+1} + b_0(n) y_n = 0.$$

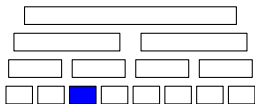
(And conversely, for D-finite formal power series.)

- ▶ Easy to generate
- ▶ Leads to **fast algorithms**

Best **boolean** complexity:

time $\mathbf{O}(\mathbf{M}(\mathbf{p} \log^2 \mathbf{p}))$, space $\mathbf{O}(\mathbf{p})$

for fixed z and $\varepsilon = 2^{-p}$



[Schroepel 1972; Brent 1976; Chudnovsky & Chudnovsky 1988;
van der Hoeven 1999, 2001; M. 2010, 2012]

Recurrences

The **coefficients** of a D-finite function $\sum_{\nu \in \lambda + \mathbb{Z}} \sum_{k=0}^K \mathbf{y}_{\nu, k} z^{\nu} \frac{\log(z)^k}{k!}$ obey a linear **recurrence relation** with polynomial coefficients:

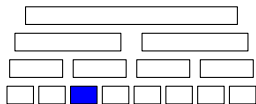
$$[b_s(\nu + S_k) \cdot S_{\nu}^s + \dots + b_1(\nu + S_k) S_{\nu} + b_0(\nu + S_k)] \cdot (\mathbf{y}_{\nu, k}) = 0.$$

- ▶ Easy to generate
- ▶ Leads to **fast algorithms**

Best **boolean** complexity:

time $\mathbf{O}(M(p \log^2 p))$, space $\mathbf{O}(p)$

for fixed z and $\varepsilon = 2^{-p}$



[Schroepel 1972; Brent 1976; Chudnovsky & Chudnovsky 1988;
van der Hoeven 1999, 2001; M. 2010, 2012]

Error Bounds

Rounding Errors

Real & complex arithmetic based on **Arb**

[Johansson 2012-]

({Real, Complex}BallField in Sage)

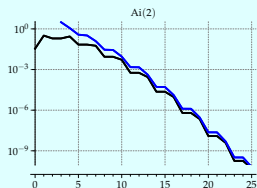
[Heuberger, M. & others]

- ▶ More generally: takes care of error propagation
- ▶ Arb supports truncated power series (→ derivatives, reg. sing. points)
- ▶ Manual error analysis still useful when intervals blow up

Truncation Errors

$$\sum_{n=0}^{\infty} u_n z^n = \underbrace{\sum_{n=0}^{N-1} u_n z^n}_{\text{known}} + \underbrace{\sum_{n=N}^{\infty} u_n z^n}_{|\cdot| \leq ?}$$

- ▶ Majorant series
- ▶ "Adaptive" bounds using residuals



[M. 2019]



Summary

Numerical solution of linear ODEs with polynomial coefficients

- full support for regular singular points (incl. algebraic, resonant...)
- arbitrary precision
- rigorous error bounds

Based on: Taylor series, an. continuation, recurrences, ball arithmetic, majorants...



Code available at

https://github.com/mkauers/ore_algebra/



Perspectives

- Irregular singular case
- Performance improvements (algorithms, code optimization)
- Convenience features (singularity analysis, monodromy...)

Bug reports, feature requests, examples welcome!

The Method of Majorants

[Cauchy 1842]

- ▶ Instead of directly bounding $|\sum_{n \geq N} u_n \zeta^n|$, compute a **majorant series**:

$$\sum \hat{u}_n z^n \in \mathbb{R}_{\geq 0}[[z]] \quad \text{s.t.} \quad \forall n, \quad |u_n| \leq \hat{u}_n$$

- ▶ To do that, "replace" L with a simple **model equation**:

$$L(z, d/dz) \cdot u = 0 \quad \ll \quad \hat{u}'(z) - \hat{a}(z) \hat{u}(z) = 0$$

"bounded by" for us: always 1st order

- ▶ Solve the model equation and study the solutions:

$$\hat{u}(z) = \exp \int^z \hat{a}(w) dw \quad \left| \sum_{n=N}^{+\infty} u_n z^n \right| \leq \sum_{n=N}^{+\infty} \hat{u}_n |z|^n \leq \dots$$

Adaptive Bounds

Problem. Computing majorants in a (too) naive way leads to catastrophic overestimations

Idea. Take into account the last computed / first neglected terms of the series

Analogy. Residuals of linear systems

$$A x = b$$

$$A \in GL_n(\mathbb{C}), \quad \|A^{-1}\| \leq M$$

$$A \tilde{x} = \tilde{b}$$

← computed approx.

$$\|x - \tilde{x}\| \leq M \cdot \underbrace{\|b - \tilde{b}\|}_{\text{known}}$$

← residual (\approx 1st negl. term)

Adaptive Majorants

$$L(z, D_z) \cdot u = 0$$

Residual: $q(z) := L(z, D_z) \cdot \tilde{u}$

$$u(z) = \sum_{n=0}^{\infty} u_n z^n = \underbrace{\sum_{n=0}^{N-1} u_n z^n}_{\tilde{u}(z)} + \sum_{n=N}^{\infty} u_n z^n$$

► Model equation

$$\begin{aligned} q(z) &\ll \hat{q}(z) \\ L(z, D_z) \cdot (\tilde{u} - u) = q &\ll \hat{L}(z, D_z) \cdot v = \hat{q} \end{aligned}$$

► Majorant property:

$$(\forall n \leq n_0) \quad |u_n| \leq v_n \quad \Rightarrow \quad (\forall n) \quad |u_n| \leq v_n$$

► Solving the model equation

$$v(z) = h(z) \left(\underset{\substack{\nearrow \\ \text{choose } \text{cst} = 0}}{\text{cst}} + \int^z \frac{t^{-1} \hat{q}(t)}{h(t)} dt \right)$$

$= O(z^N)$

$$\text{where } h(z) = \exp \int^z t^{-1} \hat{a}(t) dt$$

Image Credits

- ▶ “Work in progress” icon from https://commons.wikimedia.org/wiki/File:Work_in_progress_icon.svg
Copyright © Sławek Borewicz, licensed under the following terms:
This work is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; version 2. This work is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the version 2 of the GNU General Public License for more details.
- ▶ Other large color icons used in this document are from the Oxygen icon set (<http://www.kde.org/>). They can be copied under the GNU LGPLv3 (<http://www.gnu.org/copyleft/lesser.html>).
See also <https://techbase.kde.org/Projects/Oxygen/>.
- ▶ Surface plot of the real part of the Airy function by User:Inductiveload from https://commons.wikimedia.org/wiki/File:AiryAi_Real_Surface.png, released into the public domain by its author.
- ▶ 3D random walk from https://commons.wikimedia.org/wiki/File:Walk3d_2.png, Copyright © Zweistein~commonswiki, CC BY-SA 3.0.
- ▶ Two-torus with cycles from <https://commons.wikimedia.org/wiki/File:Two-toruscycles1.svg>
Copyright © Steelpillow, CC BY-SA 4.0.
- ▶ Benz diagram from: Jakob Ablinger, Johannes Bluemlein, Clemens Raab, Carsten Schneider, Fabian Weißbrock (2014). Calculating Massive 3-loop Graphs for Operator Matrix Elements by the Method of Hyperlogarithms. *Nuclear Physics B*. 885. 409-447. doi:10.1016/j.nuclphysb.2014.04.007.
- ▶ Cubic honeycomb from https://commons.wikimedia.org/wiki/File:Cubic_honeycomb.png, generated by User:Tomruen Curved Spaces v1.9 by Jeff Weeks and released into the public domain by its author.