Solving Linear Differential Equations to High Precision

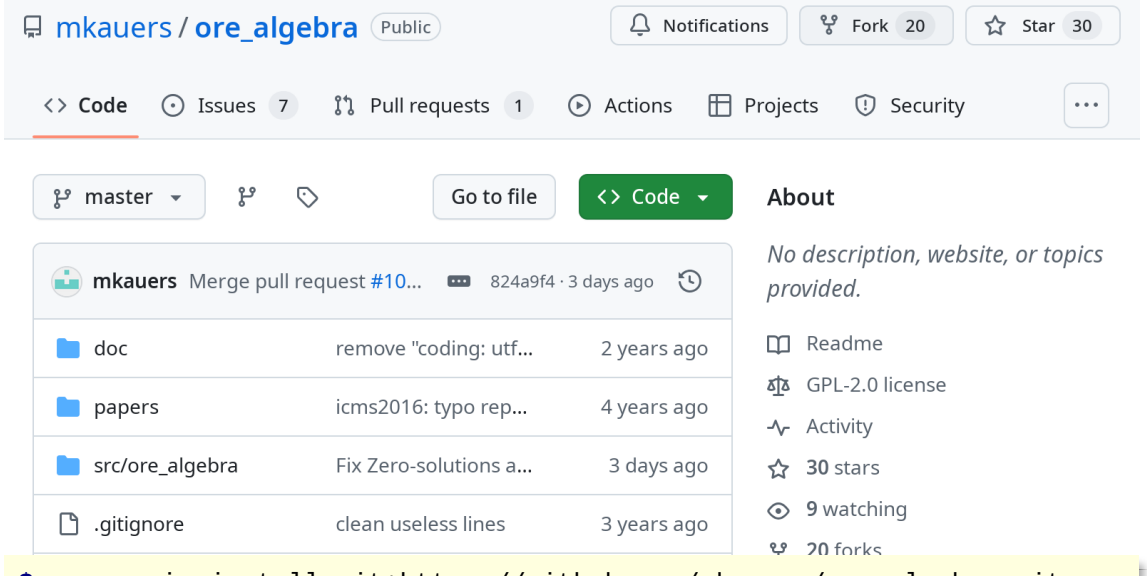
Marc Mezzarobba

CNRS, École polytechnique

Journées nationales de calcul formel 2025

Outline

Implementation




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

1 Solutions

Linear Ordinary Differential Equations

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

Complex time Arbitrary precision Singular points Error bounds

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

Linear Ordinary Differential Equations

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

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

Complex time Arbitrary precision Singular points Error bounds

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

Linear Ordinary Differential Equations

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

Complex time Arbitrary precision Singular points Error bounds

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

Linear Ordinary Differential Equations

$$Y'(x) = A(x)\,Y(x)$$

$a_i(x) \in \mathbb{C}[[x]]$

$a_i \in \mathbb{C}[x]$ $\quad a_i \in \mathbb{C}[x]$ $\quad a_i \in \mathbb{C}[x]$

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

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

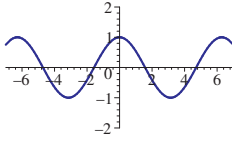
$$a_r(x)\,y^{(r)}(x) + \cdots + a_0(x)\,y(x) = b(x)$$

$y\colon \mathbb{Q}_p \to \mathbb{Q}_p$

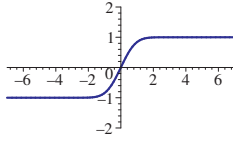
Complex time Arbitrary precision Singular points Error bounds

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

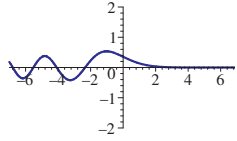
Special Functions



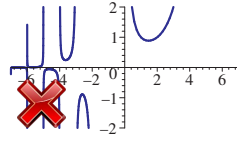
cos(x)



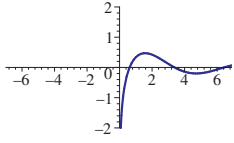
erf(x)



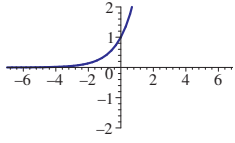
Ai(x)



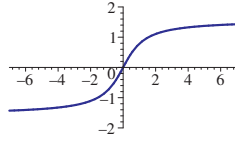
Γ(x)



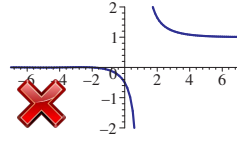
Ci(x)



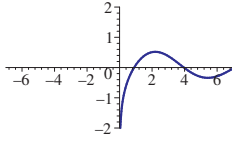
exp(x)



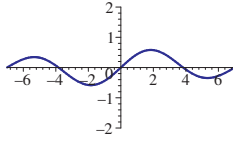
arctan(x)



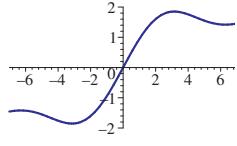
ζ(x)



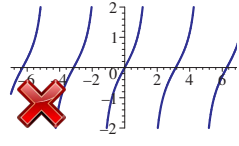
$Y_0(x)$



$J_1(x)$



Si(x)

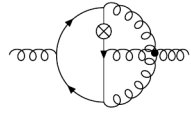


tan(x)

Iterated Integrals

[Ablinger, Blümlein, Raab, Schneider, 2014]

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



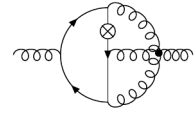
(with suitable branch choices)

```
sage: from ore_algebra.examples.iint import diffop, h, w, f
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, myini, 1e-500)
[0.9708046956249312405 ... 59027603834204946 +/- 9.05e-501]
```

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)

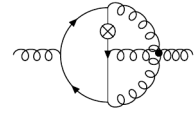
$$\begin{aligned}
&(4x^9-13x^8+15x^7-7x^6+x^5)\,I^{(6)}(x)\\
+\,&(54x^8-140x^7+120x^6-36x^5+2x^4)\,I^{(5)}(x)\\
+\,&(202x^7-397x^6+228x^5-34x^4+x^3)\,I^{(4)}(x)\\
+\,&(222x^6-303x^5+90x^4+3x^3-3x^2)\,I^{(3)}(x)\\
+\,&(48x^5-37x^4+x^3-6x^2+6x)\,I''(x)\\
+\,&(-x^2+6x-6)\,I'(x) \quad = 0
\end{aligned}$$

```
sage: from ore_algebra.examples.iint import diffop, h, w, f
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, myini, 1e-500)
[0.9708046956249312405 ... 59027603834204946 +/- 9.05e-501]
```

Iterated Integrals

[Ablinger, Blümlein, Raab, Schneider, 2014]

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

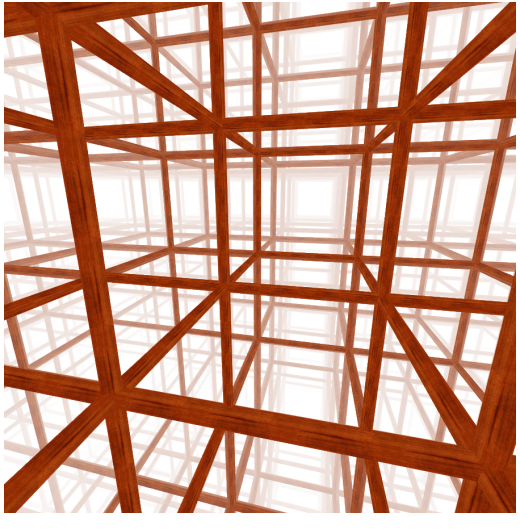


(with suitable branch choices)

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

```
sage: from ore_algebra.examples.iint import diffop, h, w, f
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, myini, 1e-500)
[0.9708046956249312405 ... 59027603834204946 +/- 9.05e-501]
```

Pólya walks



Generating series of walks on $\mathbb{Z}^d$ ($d \geqslant 3$):

$$w(x) = 1 + f(x)\, w(x)$$

$f(x)$: first return after n steps

$w(x)$: walks $0 \xrightarrow{\circlearrowleft} 0$

Return probability:

$$f(1/2d) = 1 - \frac{1}{w(1/2d)}$$

Differential equation (via diagonals):

$$L(w) = 0$$

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

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

$$((269276305858560000\,z^{30} - 80950584950784000\,z^{28} + 3629735201193984\,z^{26} - 57080630763520\,z^{24} + 409981265408\,z^{22} - 1499829760\,z^{20} + 2871232\,z^{18} - 2720\,z^{16} + z^{14})\,D^{15} + (605871688181760000000\,z^{29} - 16999622839664640000\,z^{27} + 70779836423282680\,z^{25} - 1027451357343600\,z^{23} + 67646908792320\,z^{21} - 224974464000\,z^{19} + 387616320\,z^{17} - 326400\,z^{15} + 105\,z^{13})\,D^{14} + (59375425441812480000000\,z^{28} - 1551046657578762240000\,z^{26} + 59789931630571929600\,z^{24} - 798278106947641344\,z^{22} + 4796169345443840\,z^{20} - 14416495588352\,z^{18} + 22181540160\,z^{16} - 16423904\,z^{14} + 4550\,z^{12})\,D^{13} + (33448156332221030400000\,z^{27} - 811319007569013964800000\,z^{25} + 28862642198504964096000\,z^{23} - 353037275694014545592\,z^{21} + 19259120590062080\,z^{19} - 5197874390691841\,z^{17} + 707833651200\,z^{15} - 454991264\,z^{13} + 106470\,z^{11})\,D^{12} + (12041336279599570944000000\,z^{26} - 270425045868247056384000\,z^{24} + 8847421868406446161920\,z^{22} - 98711300684195617996\,z^{20} + 486206711673234585 6\,z^{18} - 11694546001056256\,z^{16} + 13948983786816\,z^{14} - 7666274304\,z^{12} + 1479478\,z^{10})\,D^{11} + (291400337966309616844800000\,z^{25} - 6040389492409733480448000\,z^{23} + 181051617856618073715200\,z^{21} - 1833664793505926160384\,z^{19} + 8103382065529258784\,z^{17} - 1722144491873123200\,z^{15} + 177735512066112\,z^{13} - 8199432384\,z^{11} + 12662650\,z^{9})\,D^{10} + (485667229943849361408000000\,z^{24} - 9260868901892856348672000\,z^{22} + 253242086951399699345040 0\,z^{20} - 2315693541459216642048 00\,z^{18} + 9115977153370472463 36\,z^{16} - 16948125960137866 24\,z^{14} + 1491624282894720\,z^{12} - 564676102848\,z^{10} + 67128490\,z^{8})\,D^{9} + (561986366077882832486400000\,z^{23} - 9821600336936930947891200000\,z^{21} + 243895079188460801654784000\,z^{19} - 2001474041629081060638720\,z^{17} + 6961148053301631762432\,z^{15} - 1119041319794701363 2\,z^{13} + 8252818589658240\,z^{11} - 2491958589120\,z^{9} + 216627840\,z^{7})\,D^{8} + (449589092863062659891200000000\,z^{22} - 71725429608240620136038400000\,z^{20} + 160916181872552335717171200\,z^{18} - 11770087561727055499100160\,z^{16} + 358259377289463117250 56\,z^{14} - 491129274712818263 04\,z^{12} + 297048875446688 64\,z^{10} - 689824676\backslash 6112\,z^{8} + 408741333\,z^{6})\,D^{7} + (244776283891700078149632000000 0\,z^{21} - 354907365563571111670579200000\,z^{19} + 715247540548767602508832000\,z^{17} - 46268511513163153122263040\,z^{15} + 12189194366916395320934 4\,z^{13} - 140140992502944986112\,z^{11} + 67667087104034880\,z^{9} - 11517430544256\,z^{7} + 420693273\,z^{5})\,D^{6} + 88119462201012028133867520000000\,z^{20} - 1155571219953730314672537600000\,z^{18} + 207851129536136358620823555200\,z^{16} - 11785207154959595278172160\,z^{14} + 265216641208694799482880\,z^{12} - 2505598368965747251 20\,z^{10} + 93303962552021952\,z^{8} - 10897207743264\,z^{6} + 210766920\,z^{4})\,D^{5} + (200271505002300063940608000000 00\,z^{19} - 23625739365821343014584320000 00\,z^{17} + 37651520030507075820847104000\,z^{15} - 18518017537186339789209600\,z^{13} + 35042065122976245153792 0\,z^{11} - 26516934155002042368 0\,z^{9} + 72752913986864640\,z^{7} - 5304232959840\,z^{5} + 42355950\,z^{3})\,D^{4} + (26702867333640008525414400000000\,z^{18} - 28168195528972191997624320000000\,z^{16} + 39443088818742150876364800000\,z^{14} - 16621735034402967500 8819200\,z^{12} + 25944127922231096868 8640\,z^{10} - 152179348773380567040\,z^{8} + 28891552538695680\,z^{6} - 1138125841504\,z^{4} + 2375101\,z^{2})\,D^{3} + (184866004617507751329792000000000\,z^{17} - 17320803797906854080675840000000\,z^{15} + 21106053778472440493506560000\,z^{13} - 75098351566004361206169600\,z^{11} + 94387689349096767160320\,z^{9} - 41091696296267489280\,z^{7} + 4930837635294720\,z^{5} - 82548913344\,z^{3} + 16383\,z)\,D^{2} + (52818858462145071808512000000000\,z^{16} - 43621745324389943161651200000\,z^{14} + 45737881495077761895628800\,z^{12} - 13497729262079414265446400\,z^{10} + 13245706827488316948480\,z^{8} - 4031063063164477440\,z^{6} + 265858264373760\,z^{4} - 1204325184\,z^{2} + 1)\,D + 35212572308096714539008000000\,z^{15} - 25412342171473281024000000\,z^{13} + 22623197301788479429017600 0\,z^{11} - 5415746955623320780800 00\,z^{9} + 39833545741558076800\,z^{7} - 77667722566041600\,z^{5} + 2222757642240\,z^{3} - 983040\,z)$$

data courtesy of Bruno Salvy

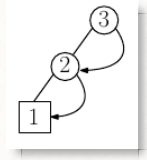
```
sage: from ore_algebra.examples import polya
sage: 1 - 1/polya.dop[15].numerical_solution([0]*14+[1], [0, 1/(2*15)], 1e-50).real()
[0.035869623125356514229404270 8 +/- 2.07e-29]
```

Compacted Binary Trees

[Genitrini, Gittenberger, Kauers, Wallner 2020]

Theorem 8.6 (Differential operators). *Let $(M_k)_{k\geq 0}$ be the family of differential operators given by*

$$M_0 = (1-z)D - 1,$$
$$M_1 = (1-2z)D^2 - (3-z)D,$$
$$M_k = M_{k-1}\cdot D - M_{k-2}\cdot\left(D^2\cdot z - zD\right), \qquad k \geq 2.$$



Then the exponential generating function $C_k(z)$ of compacted binary trees with right height at most k satisfies

$$M_k \cdot C_k = 0.$$

```
sage: from ore_algebra.examples import cbt
sage: cbt.dop[5]
(-4*z^3 + 10*z^2 - 6*z + 1)*Dz^6 + (9*z^3 - 58*z^2 + 75*z - 21)*Dz^5
+ (-6*z^3 + 78*z^2 - 184*z + 95)*Dz^4 + (z^3 - 33*z^2 + 141*z - 110)*Dz^3
+ (3*z^2 - 32*z + 40)*Dz^2 + (z - 3)*Dz
```

Asymptotics

Theorem 3.4 (Asymptotics of compacted trees with bounded right height)**.** *The number $c_{k,n}$ of compacted trees with right height at most k is for $n \to \infty$ asymptotically equivalent to*

$$c_{k,n} \sim \kappa_k n! \left(4\cos\left(\frac{\pi}{k+3}\right)\right)^n n^{-\frac{k}{2}-\frac{1}{k+3}-\left(\frac{1}{4}-\frac{1}{k+3}\right)\frac{1}{\cos^2\left(\frac{\pi}{k+3}\right)}},$$

where $\kappa_k \in \mathbb{R}$ is independent of n.

[Genitrini, Gittenberger, Kauers, Wallner 2020]

Asymptotic expansion + rigorous enclosure of κ_k + error bound — all automatic:

```
sage: ini = list(cbt.egf[:6]); ini
[1, 1, 3/2, 5/2, 37/8, 373/40]
sage: bound_coefficients(cbt.dop[5], ini, 'n', 1, 30, n0=100)
1.00000000*3.414213562373095?^n*(
([1.624857 +/- 1.35e-7] + [+/- 1.01e-7]*I)*n^(-2.771446609406727?)
+ B([62.8523859 +/- 3.82e-8]*n^(-3.771446609406727?), n >= 100))
```

[Dong, Melczer, M., 2024]

Irrationality of $\zeta(3)$

La suite u_n s'écrit $(1,5,73,1445,33001,\ldots)$

La suite v_n s'écrit $(0,6,\frac{351}{4},\frac{62531}{36},\ldots)$

Les deux suites vérifient la relation de récurrence

$$(n+1)^3 u_{n+1} - (34n^3+51n^2+27n+5)u_n + n^3 u_{n-1} = 0$$

$$\lambda = 17+12\sqrt{2}$$

[Apéry 1979]

Irrationality of $\zeta(3)$

La suite u_n s'écrit $(1,5,73,1445,33001,\ldots)$

La suite v_n s'écrit $(0,6, \frac{351}{4}, \frac{625}{3}$

Les deux suites vérifient la relation

$$(n+1)^3 u_{n+1} - (34n^3+51n^2+$$

$$\lambda = 17+$$

[Apéry 1979]

(at Princeton). For example, one's intuition is just wrong in feeling incredulity at the facts of (3). All that these report is that the differential equation

$$\frac{d}{dX}\left\{(X^4 - 34X^3 + X^2)\frac{d^3y}{dX^3} + (6X^3 - 103X^2 + 3X)\frac{d^2y}{dX^2} + (7X^2 - 112X + 1)\frac{dy}{dX} + (X-5)y - (u_1 - 5u_0)\right\} = 0$$

has two *G-function* solutions, namely $a(X) = 6X + a_2X^2 + \ldots$; $b(X) = 1 + b_1X + b_2X^2 + \ldots$; and $a(X) - \zeta(3)b(X)$ is regular (in fact vanishes) at $\alpha' = (1-\sqrt{2})^4$. This is interesting, but no longer incredible; and it is readily generalisable . . . All this too is an idea of Beukers. In keeping with the bizarre

[van der Poorten 1979]

Irrationality of $\zeta(3)$

La suite u_n s'écrit $(1,5,73,1445,33001,\ldots)$

La suite v_n s'écrit $(0,6,\frac{351}{4},\frac{625}{3}$ (at Princeton). For example, one's intuition is just wrong in feeling incredulity at the facts of (3). All that these report is that the differential equation

Les deux suites vérifient la relation

$$(n+1)^3 u_{n+1} - (34n^3+51n^2+ \quad \frac{d}{dX}\left\{(X^4-34X^3+X^2)\frac{d^3y}{dX^3}+\right.$$

$$\lambda = 17+ \qquad +(6X^3-103X^2+3X)\frac{d^2y}{dX^2}+$$

```
sage: dop = (Dx*((x^4 - 34*x^3 + x^2)*Dx^3 +
....:             (6*x^3 - 153*x^2 + 3*x)*Dx^2 +
....:             (7*x^2 - 112*x + 1)*Dx + (x - 5))
sage: lam = QQbar(17+12*sqrt(2))
sage: mat = dop.numerical_transition_matrix([0, 1/lam], 1e-30)
sage: 6*mat[1,3]/(mat[1,2] + 5*mat[1,3])
[1.2020569031595942853997381615 1 +/- 2.05e-30] + [+/- 2.59e-42]*I
sage: RealBallField(100)(3).zeta()
[1.2020569031595942853997381615 1 +/- 2.71e-30]
```

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; …]

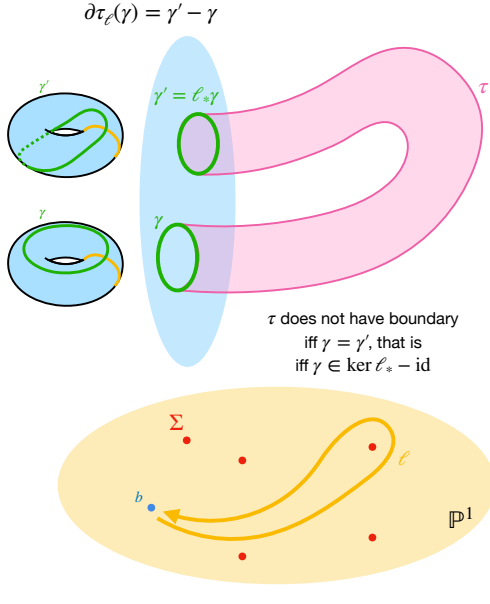
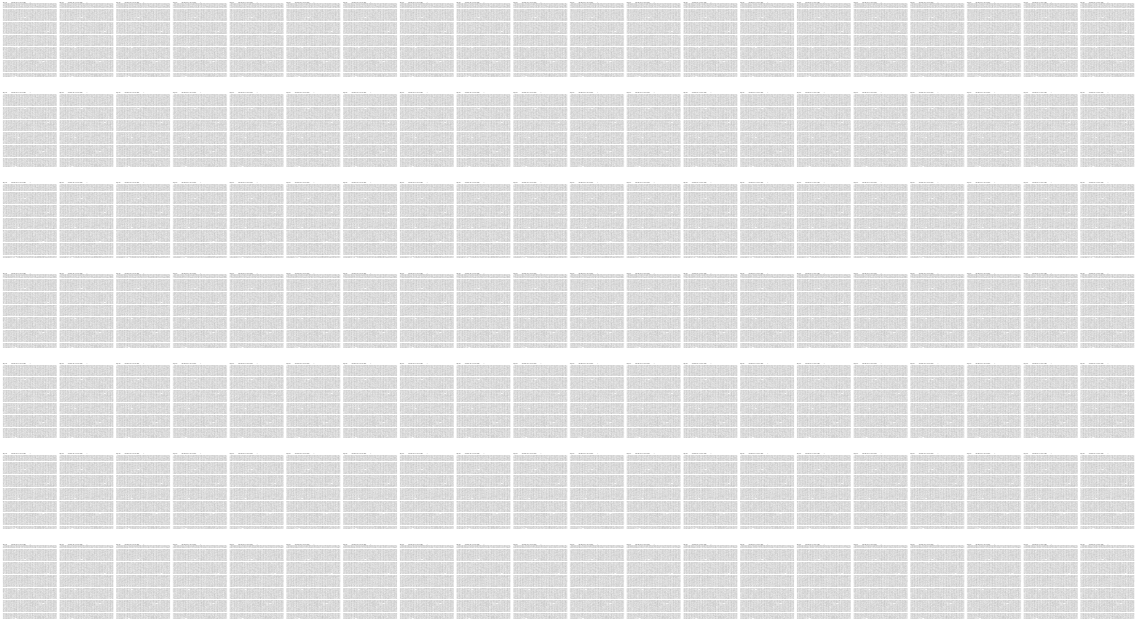
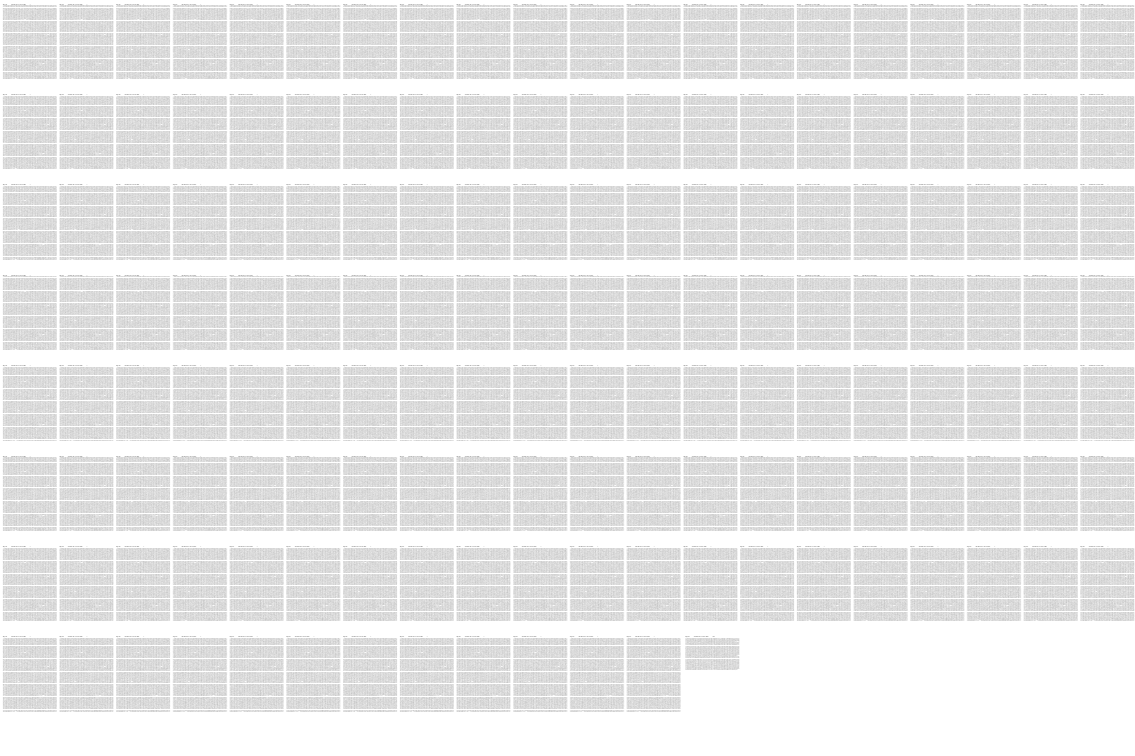


image courtesy of Eric Pichon-Pharabod

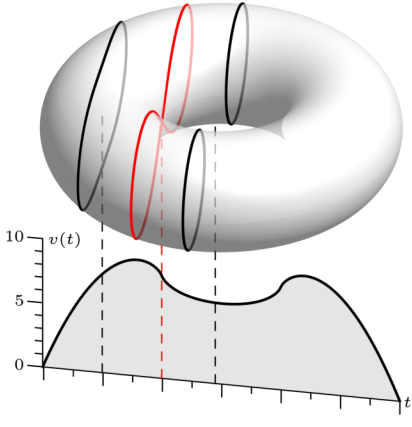
Equations can get large





Volumes of Semi-Algebraic Sets

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



For a compact s.a. set:

- 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

```
···· slice #2: ρ = 10866099/4849664
····· slice length = [3.95699242690042041342397892533404623584614411033674866606926914003 +/- 5.52e-66]
···· integrating PF equation over [1.0109061762643997, 2.989093823735602?]...
···· ...piece volume = [8.10844587166147220133178843300791539013253760904431939702317 34 +/- 8.50e-62]
··· slice volume = [24.858639122870438686966469615822549439813781340716313074232 20 +/- 5.78e-60]
·· integrating PF equation over [-1, 1]...
·· ...piece volume = [39.478417604357434475337963999504604541254797628963162506 +/- 6.38e-55]
[39.478417604357434475337963999504604541254797628963162506 +/- 6.38e-55]
```

Symbolic solutions

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

Reconstruct factorization of a differential operator from a basis of numerical solutions

```
sage: dop = (768*x^18 - 13824*x^17 + 112896*x^16 - 552960*x^15 + 1808640*x^14 - 4161024*x^13 + 6903552*x^12
- 8322048*x^11 + 7234560*x^10 - 4423680*x^9 + 1806336*x^8 - 442368*x^7 + 49152*x^6)*Dx^6 + (-76032*x^17 +
1279488*x^16 - 9763584*x^15 + 44682240*x^14 - 136631040*x^13 + 294237696*x^12 - 457868544*x^11 + 519124992*x^10
- 425948160*x^9 + 246865920*x^8 - 96006144*x^7 + 22511616*x^6 - 2408448*x^5)*Dx^5 + (2962944*x^16 -
45959360*x^15 + 323956800*x^14 - 1374850304*x^13 + 3921581184*x^12 - 7939259072*x^11 + 11723587392*x^10 -
12744194944*x^9 + 10130443776*x^8 - 5741996032*x^7 + 2200396800*x^6 - 510883840*x^5 + 54214656*x^4)*Dx^4 +
(-54535680*x^15 + 764912128*x^14 - 4916996224*x^13 + 19281508096*x^12 - 51702820352*x^11 + 100365985024*x^10
- 144942728064*x^9 + 156742564352*x^8 - 125508995072*x^7 + 72192137216*x^6 - 28154152960*x^5 + 6648872960*x^4
- 715751424*x^3)*Dx^3 + (405844992*x^14 - 5092110976*x^13 + 30303194448*x^12 - 114462393664*x^11
+ 306626051808*x^10 - 610586454848*x^9 + 918179036112*x^8 - 1040458705280*x^7 + 874119303552*x^6 -
526676536832*x^5 + 214313360640*x^4 - 52485566464*x^3 + 5814976512*x^2)*Dx^2 + (477398016*x^13 -
1428000768*x^12 - 25960369072*x^11 + 220548962784*x^10 - 831430058816*x^9 + 1944753068704*x^8 -
3176580284176*x^7 + 3812803669504*x^6 - 3386130527616*x^5 + 2160307930624*x^4 - 928232421632*x^3 +
238279360512*x^2 - 27408728064*x)*Dx - 15797846016*x^12 + 97889810688*x^11 - 187457537408*x^10 -64013239764*x^9
+ 1021695132036*x^8 - 2748033509204*x^7 + 4844154274236*x^6 - 6210610474824*x^5 + 5833453259
sage: dop.factor()
[(768*x^6 - 4608*x^5 + 9984*x^4 - 9216*x^3 + 3072*x^2)*Dx^2 + (-33024*x^5 + 172032*x^4 - 323328*x^3 +
261120*x^2 - 76800*x)*Dx + 122880*x^4 - 853568*x^3 + 2044608*x^2 - 1782400*x + 470016,
 (x^6 - 6*x^5 + 13*x^4 - 12*x^3 + 4*x^2)*Dx^2 + (-46*x^5 + 217*x^4 - 365*x^3 + 266*x^2 - 72*x)*Dx + 496*x^4 -
5223/4*x^3 + 5305/4*x^2 - 2127/2*x + 352,
 (x^6 - 6*x^5 + 13*x^4 - 12*x^3 + 4*x^2)*Dx^2 + (-46*x^5 + 217*x^4 - 365*x^3 + 266*x^2 - 72*x)*Dx + 496*x^4 -
5223/4*x^3 + 5305/4*x^2 - 2127/2*x + 352]
```

code by Alexandre Goyer

Existence of solutions

[Cauchy, ∼1839]

Theorem. Let $\alpha_1, \ldots, \alpha_r$ be analytic functions $\{|x| < \rho\} \to \mathbb{C}$.

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

$$y^{(r)}(x) + \alpha_{r-1}(x)\, y^{(r-1)}(x) + \cdots + \alpha_0(x)\, y(x) = 0$$

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

basis of solutions

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

Existence of solutions

[Cauchy, ∼1839]

Theorem. Let $\alpha_1, \dots, \alpha_r$ be analytic functions $\{|x| < \rho\} \to \mathbb{C}$.

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

$$y^{(r)}(x) + \alpha_{r-1}(x)\, y^{(r-1)}(x) + \cdots + \alpha_0(x)\, y(x) = 0$$

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

basis of solutions

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

Proof sketch. 1. Indeterminate coefficients $\Rightarrow$ formal solution $\hat{y} \in \mathbb{C}[[x]]$.

2. The formal series $\hat{y}$ converges for $|z| < \rho$. □

Existence of solutions

[Cauchy, ∼1839]

Theorem. Let $\alpha_1, \dots, \alpha_r$ be analytic functions $\{|x| < \rho\} \to \mathbb{C}$.

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

$$y^{(r)}(x) + \alpha_{r-1}(x)\, y^{(r-1)}(x) + \cdots + \alpha_0(x)\, y(x) = 0$$

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

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:

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

Proof sketch. 1. Indeterminate coefficients ⇒ formal solution $\hat{y} \in \mathbb{C}[[x]]$.

2. The formal series $\hat{y}$ converges for $|z| < \rho$. □

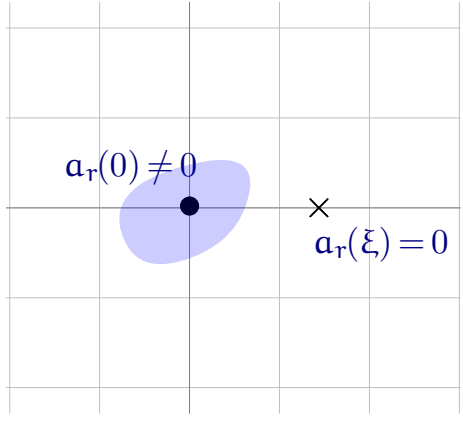
Singular points

Corollary. For an equation

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

with polynomial coefficients:

- whenever $a_r(x_0) \neq 0$, there is a full basis of analytic solutions around x_0,



Definition. The zeroes of the leading coefficient a_r are called the **singular points** of the equation.

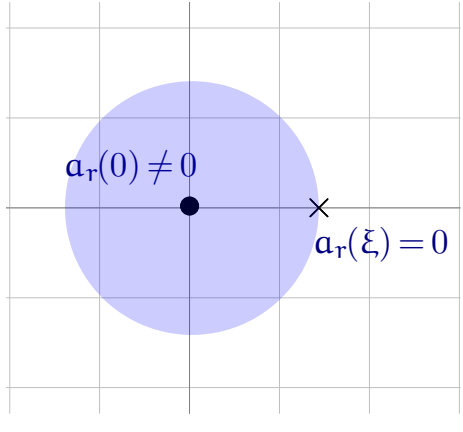
Singular points

Corollary. For an equation

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

with polynomial coefficients:

- whenever $a_r(x_0) \neq 0$, there is a full basis of analytic solutions around x_0,
- their Taylor expansions converge at least up to the closest root of a_r,



Definition. The zeroes of the leading coefficient a_r are called the **singular points** of the equation.

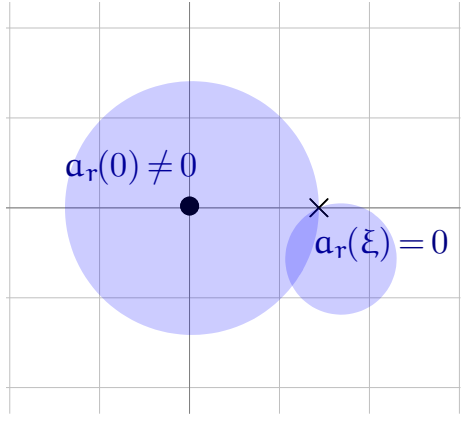
Singular points

Corollary. For an equation

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

with polynomial coefficients:

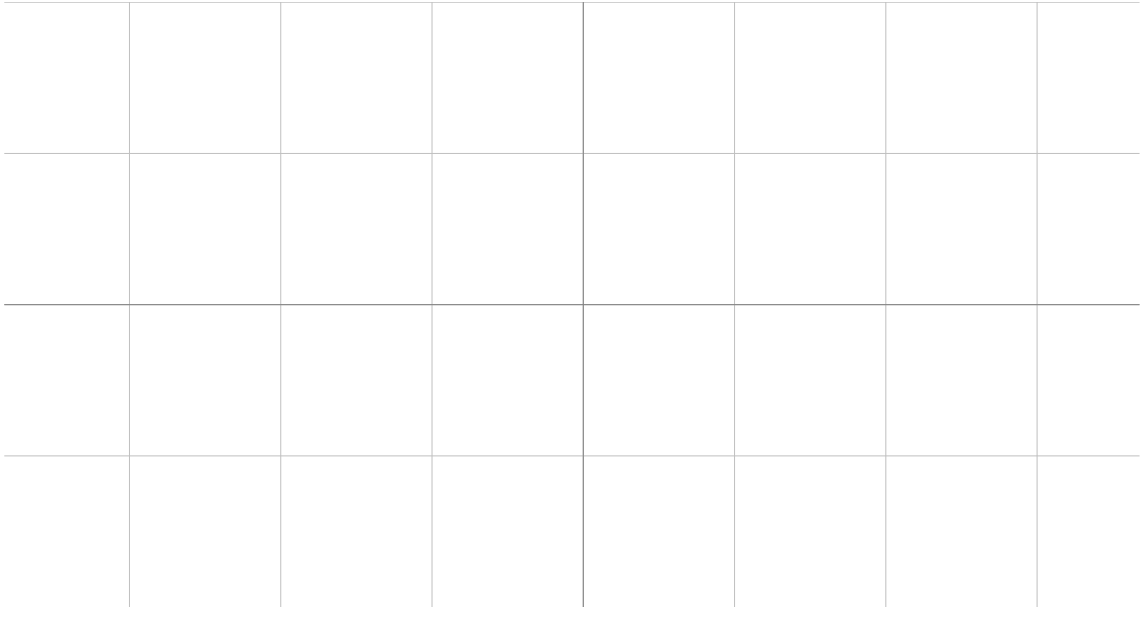
- whenever $a_r(x_0) \neq 0$, there is a full basis of analytic solutions around x_0,
- their Taylor expansions converge at least up to the closest root of a_r,
- the only place a solution can become singular is at a root of a_r.



Definition. The zeroes of the leading coefficient a_r are called the **singular points** of the equation.

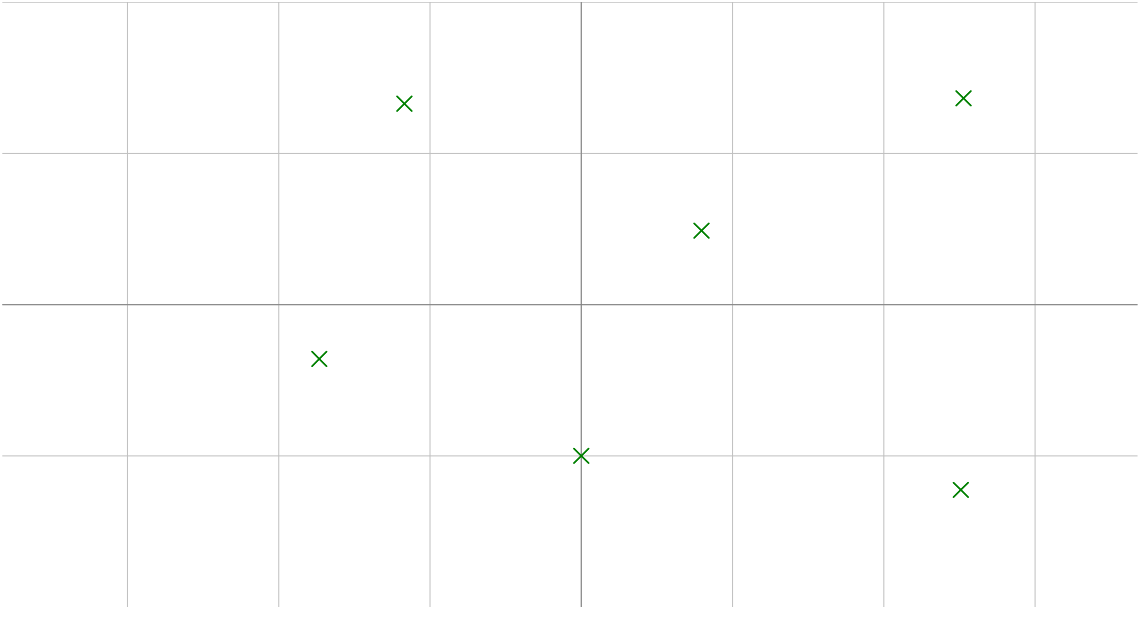
Values of solutions

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



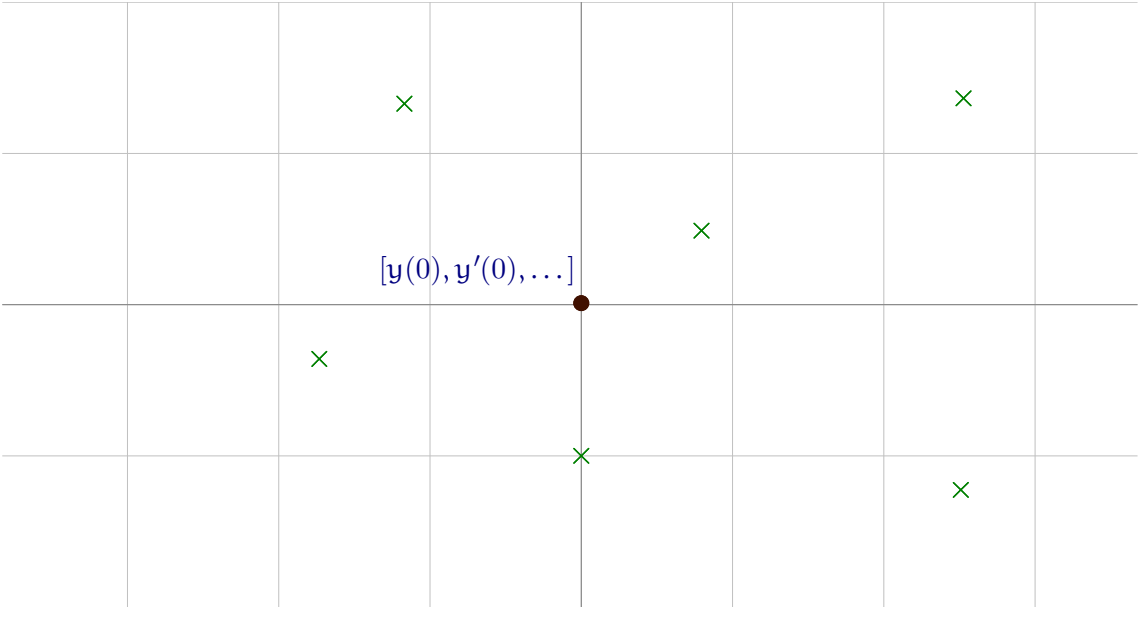
Values of solutions

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



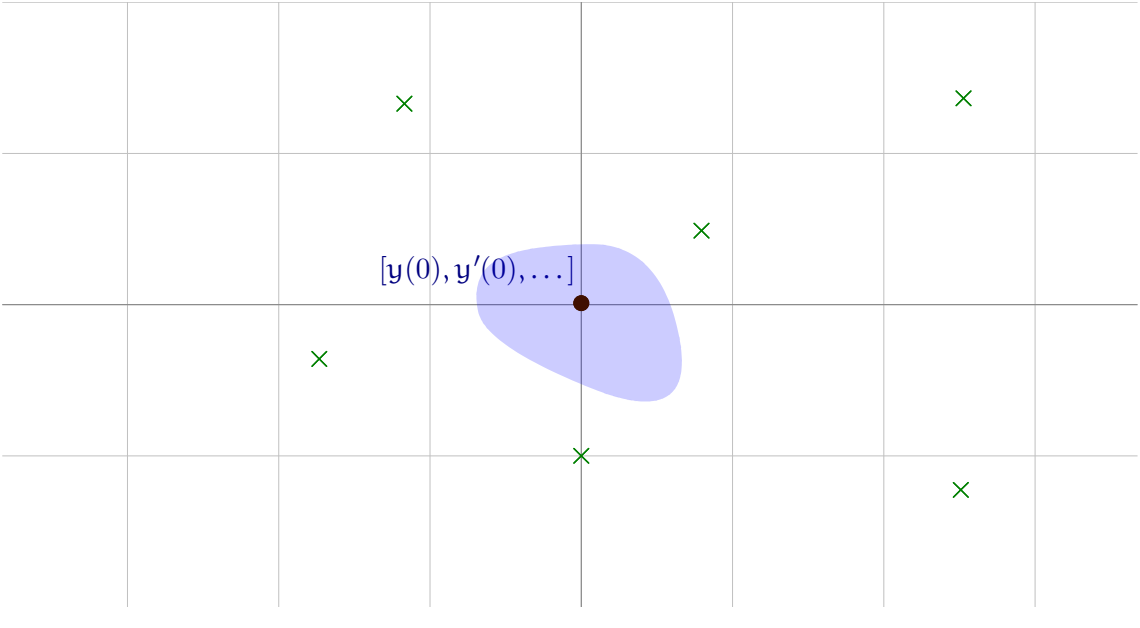
Values of solutions

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



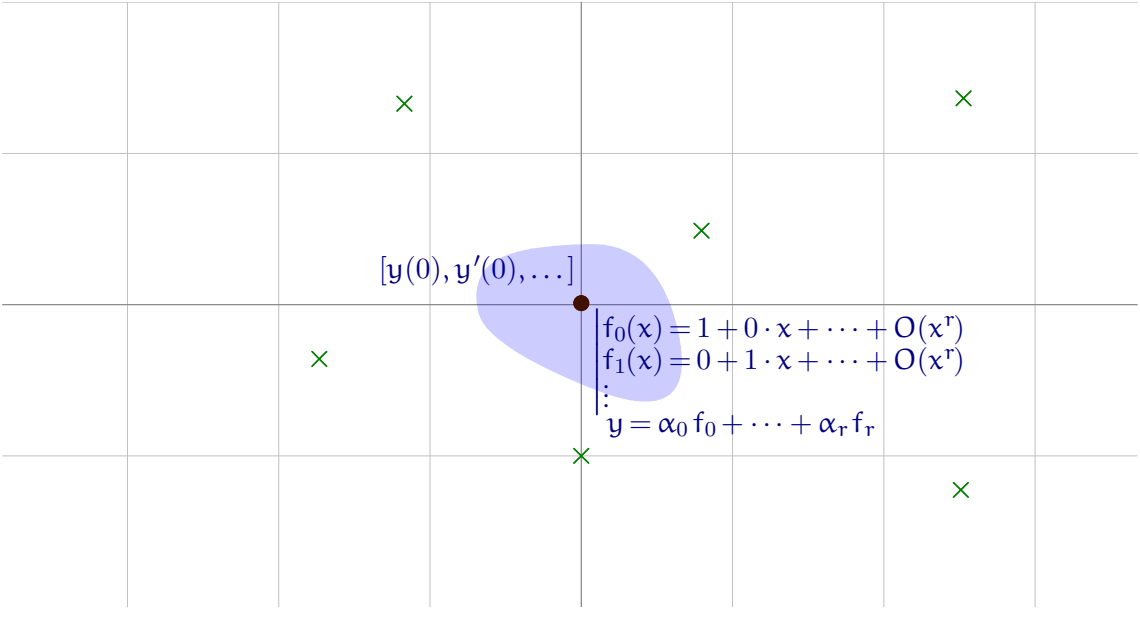
Values of solutions

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



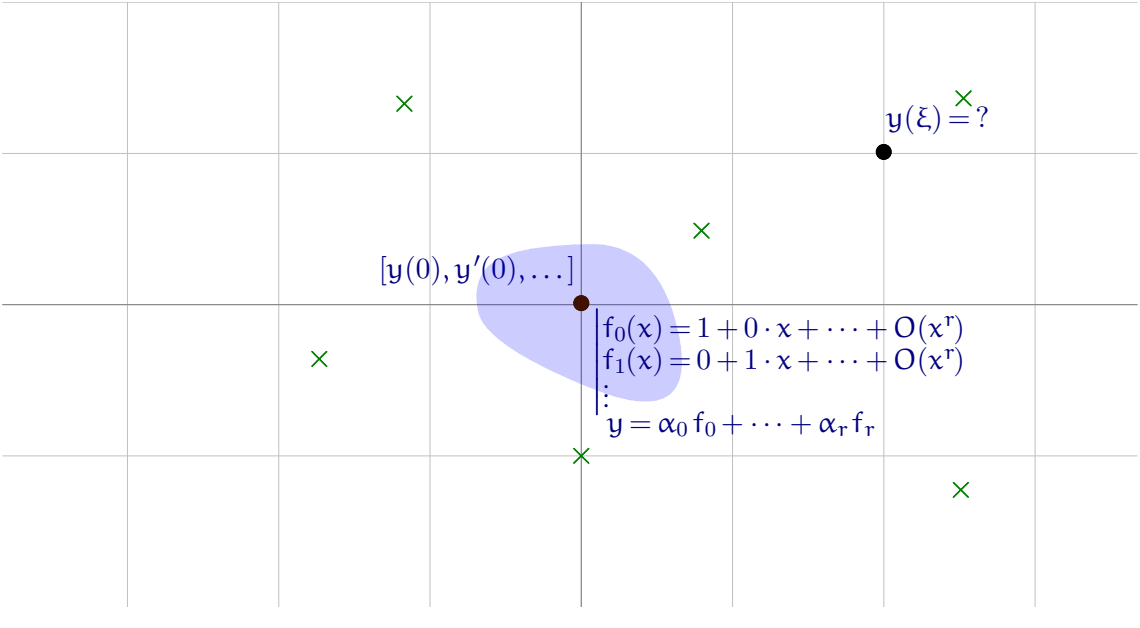
Values of solutions

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



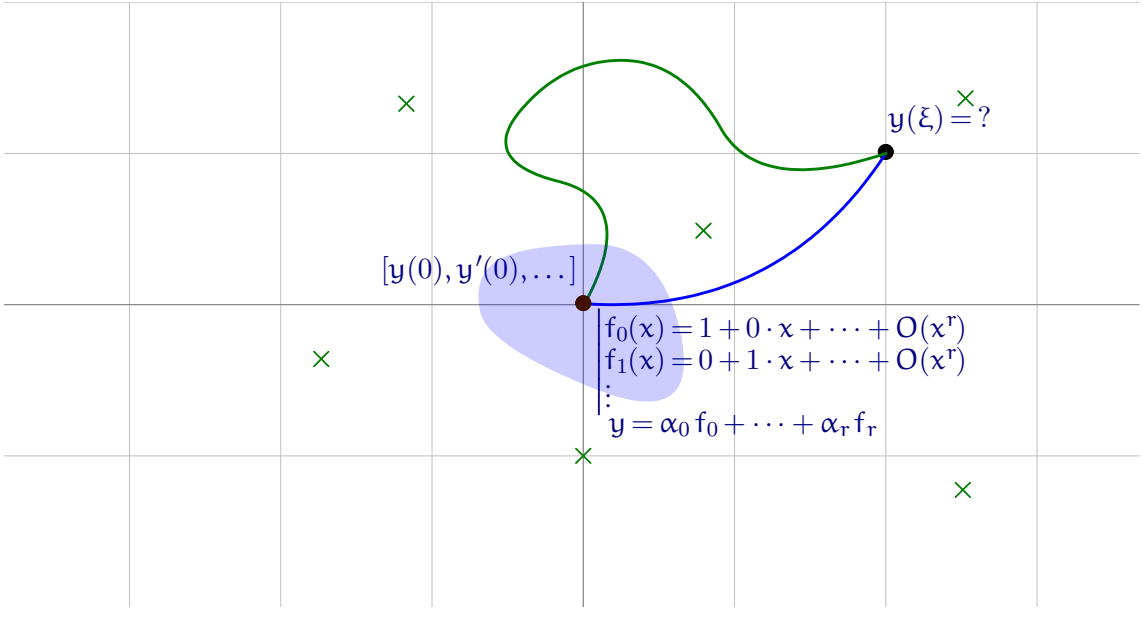
Values of solutions

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



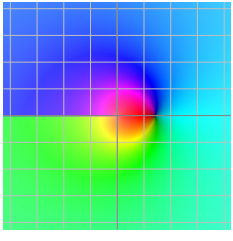
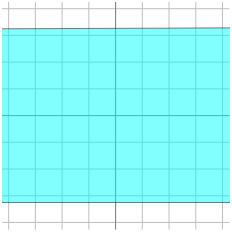
Values of solutions

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



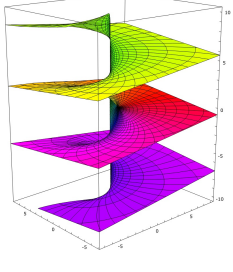
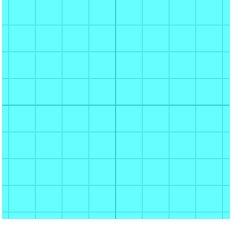
The complex logarithm

$x\,y''(x) + y'(x) = 0$

log:  $\rightarrow$ 

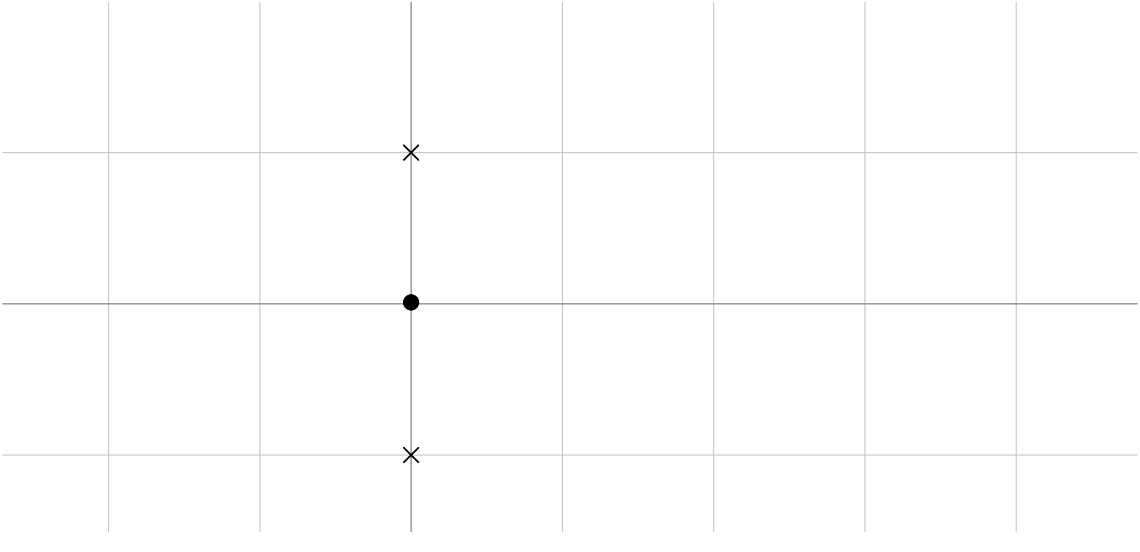
$\rho\, e^{i\theta} \mapsto \log \rho + i\,\theta \qquad -\pi < \theta \leqslant \pi$

---

log:  $\rightarrow$ 

Connection matrices

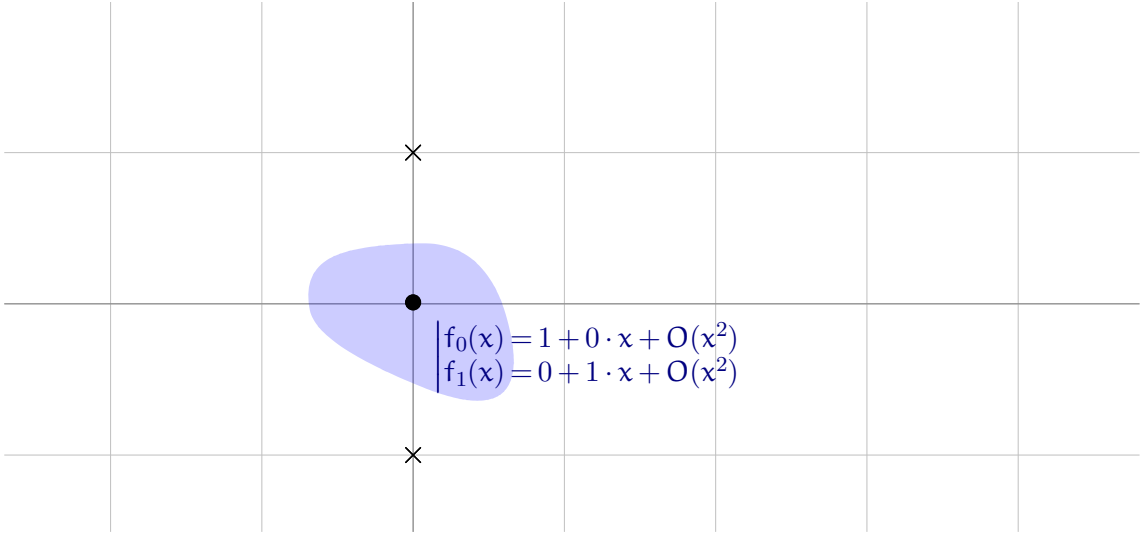
$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$



Connection defined along a path considered up to homotopy (or given fixed branch cuts)

Connection matrices

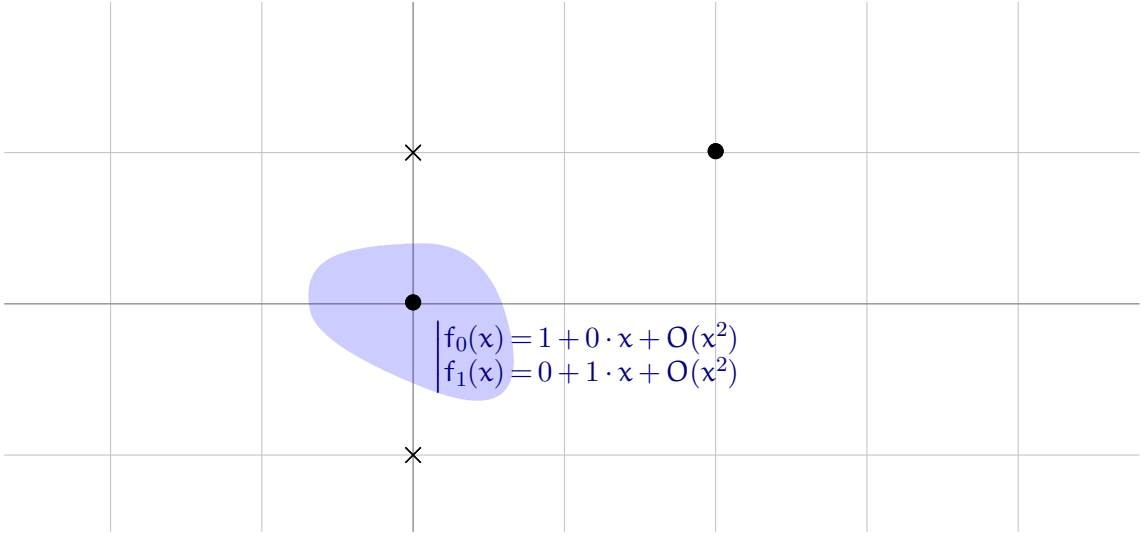
$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$



Connection defined along a path considered up to homotopy (or given fixed branch cuts)

Connection matrices

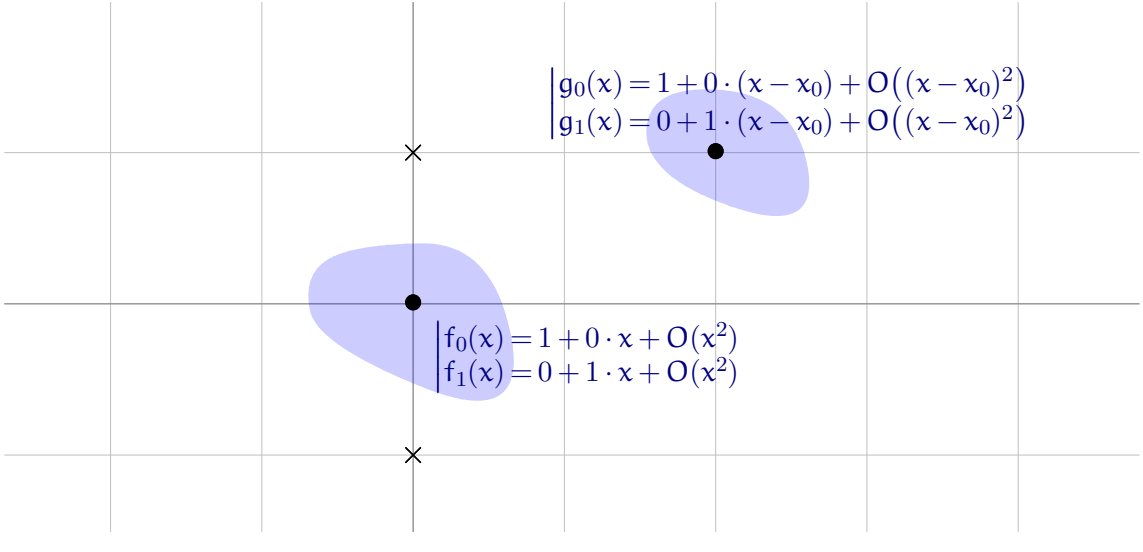
$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$



Connection defined along a path considered up to homotopy (or given fixed branch cuts)

Connection matrices

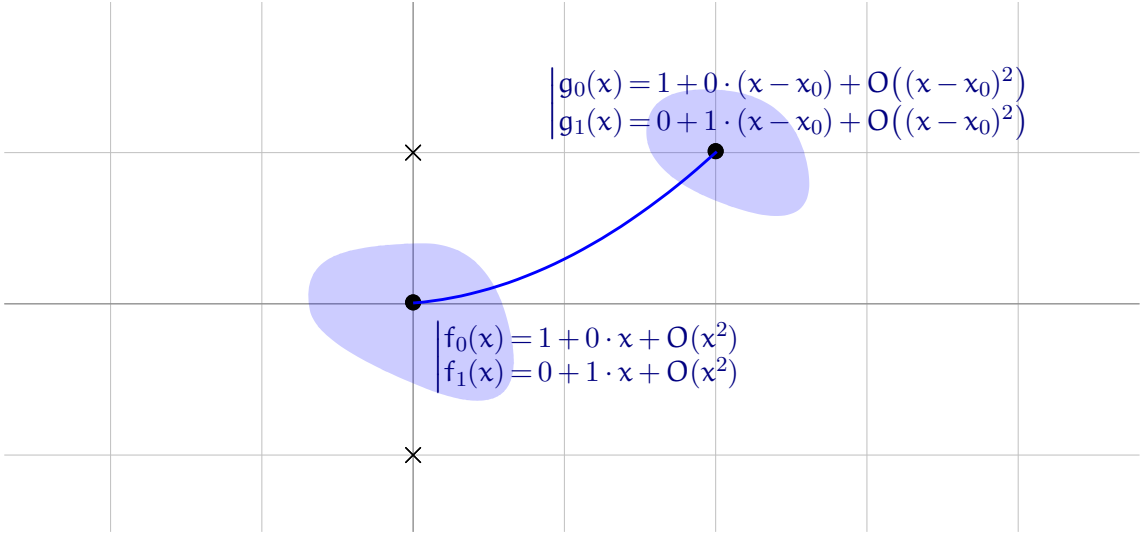
$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$



Connection defined along a path considered up to homotopy (or given fixed branch cuts)

Connection matrices

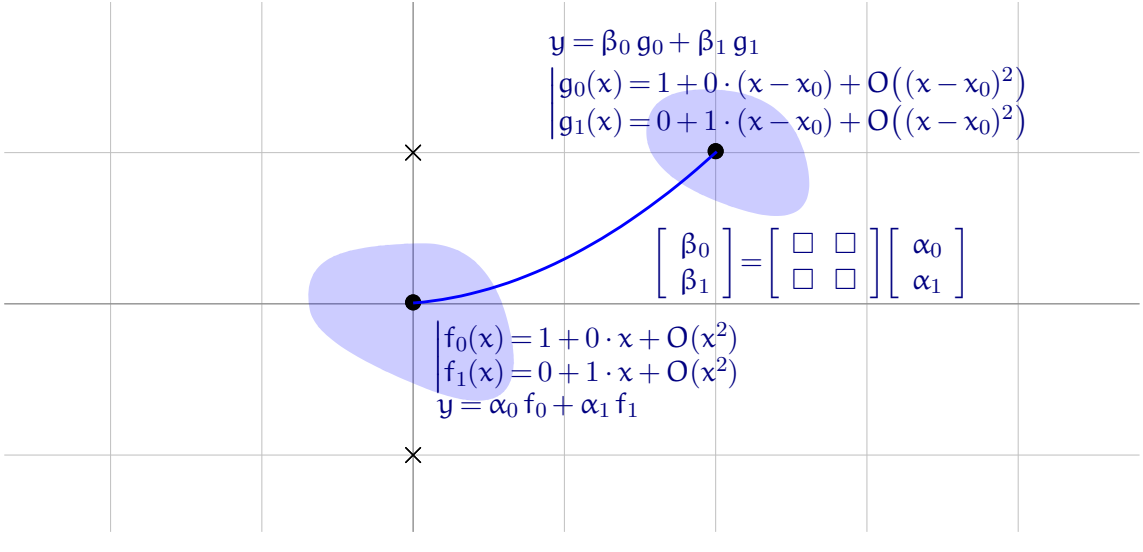
$$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$$



Connection defined along a path considered up to homotopy (or given fixed branch cuts)

Connection matrices

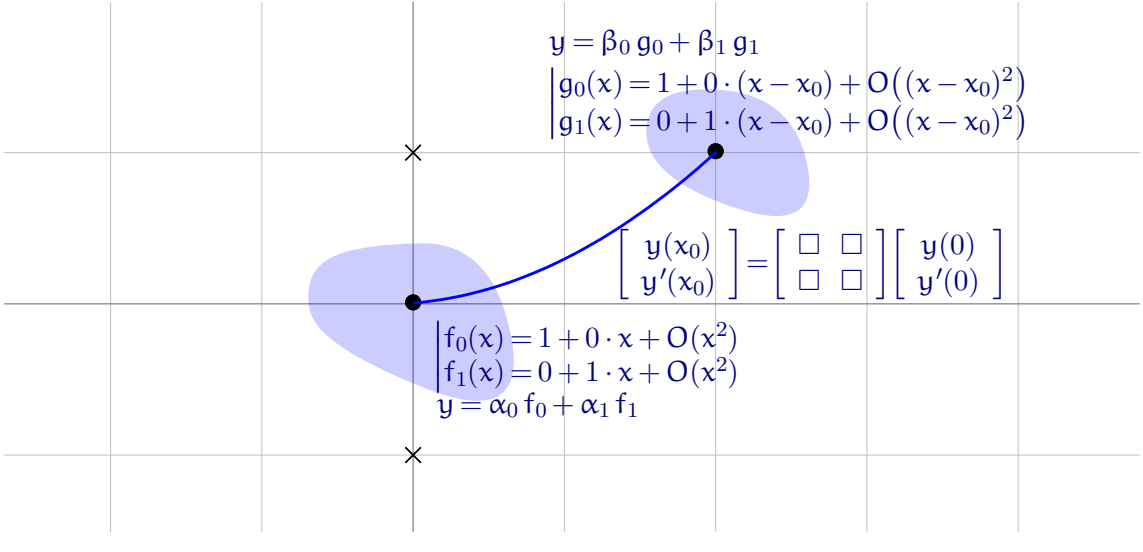
$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$



Connection defined along a path considered up to homotopy (or given fixed branch cuts)

Connection matrices

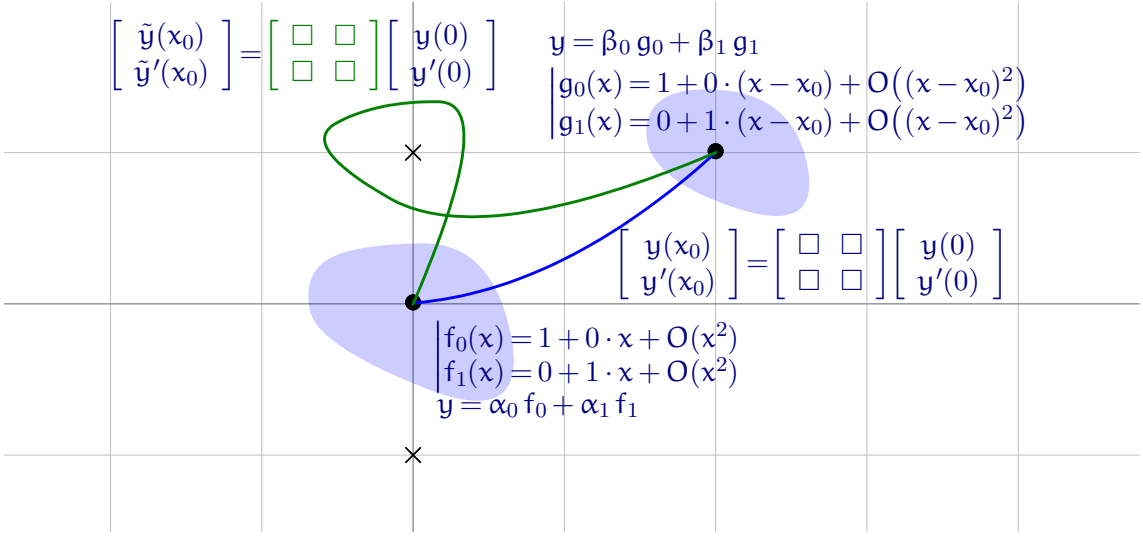
$$(x^2+1)\,y''(x)+2\,x\,y'(x)=0$$



Connection defined along a path considered up to homotopy (or given fixed branch cuts)

Connection matrices

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



Connection defined along a path considered up to homotopy (or given fixed branch cuts)

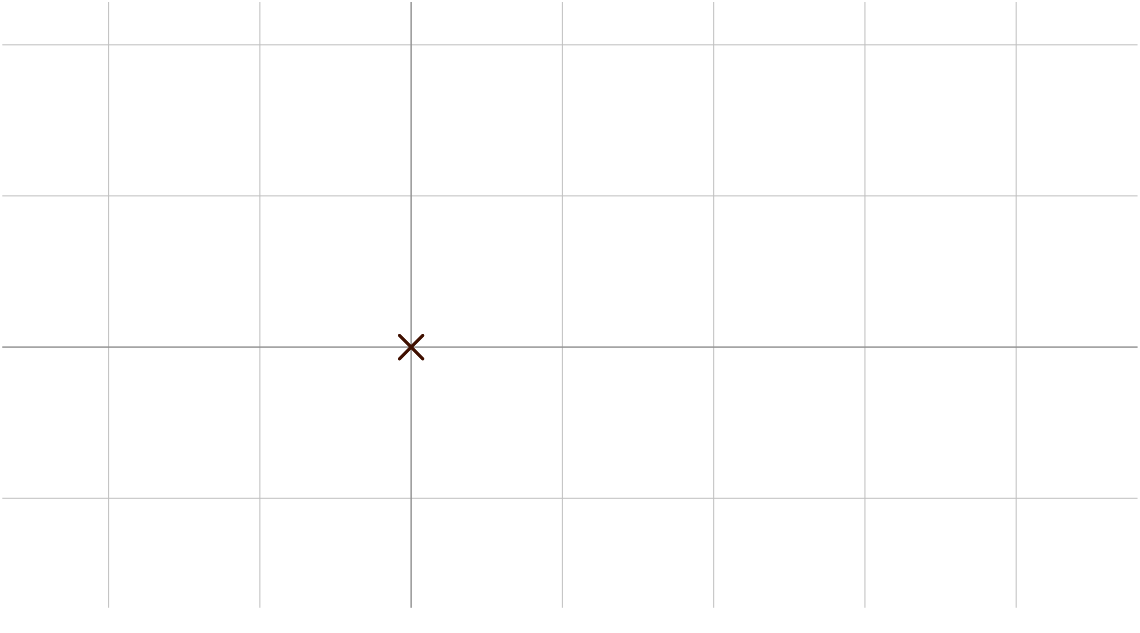
Connection matrices, more explicitly

$$\begin{bmatrix} y(x_1) \\ y'(x_1) \\ \vdots \\ \dfrac{y^{(r-1)}(x_1)}{(r-1)!} \end{bmatrix} = \underbrace{\begin{bmatrix} f_0(x_1) & \cdots & f_{r-1}(x_1) \\ f_0'(x_1) & & f_{r-1}'(x_1) \\ \vdots & & \vdots \\ \dfrac{f_0^{(r-1)}(x_1)}{(r-1)!} & \cdots & \dfrac{f_{r-1}^{(r-1)}(x_1)}{(r-1)!} \end{bmatrix}}_{=\mathcal{F}(x_1)} \begin{bmatrix} y(x_0) \\ y'(x_0) \\ \vdots \\ \dfrac{y^{(r-1)}(x_0)}{(r-1)!} \end{bmatrix}$$

$$\begin{cases} f_0(x_0+t) &= 1 + 0\,t + \cdots + 0\,t^{r-1} + \blacksquare\, t^r + \cdots \\ f_1(x_0+t) &= 0 + 1\,t + \cdots + 0\,t^{r-1} + \blacksquare\, t^r + \cdots \\ & \vdots \\ f_{r-1}(x_0+t) &= 0 + 0\,t + \cdots + 1\,t^{r-1} + \blacksquare\, t^r + \cdots \end{cases}$$

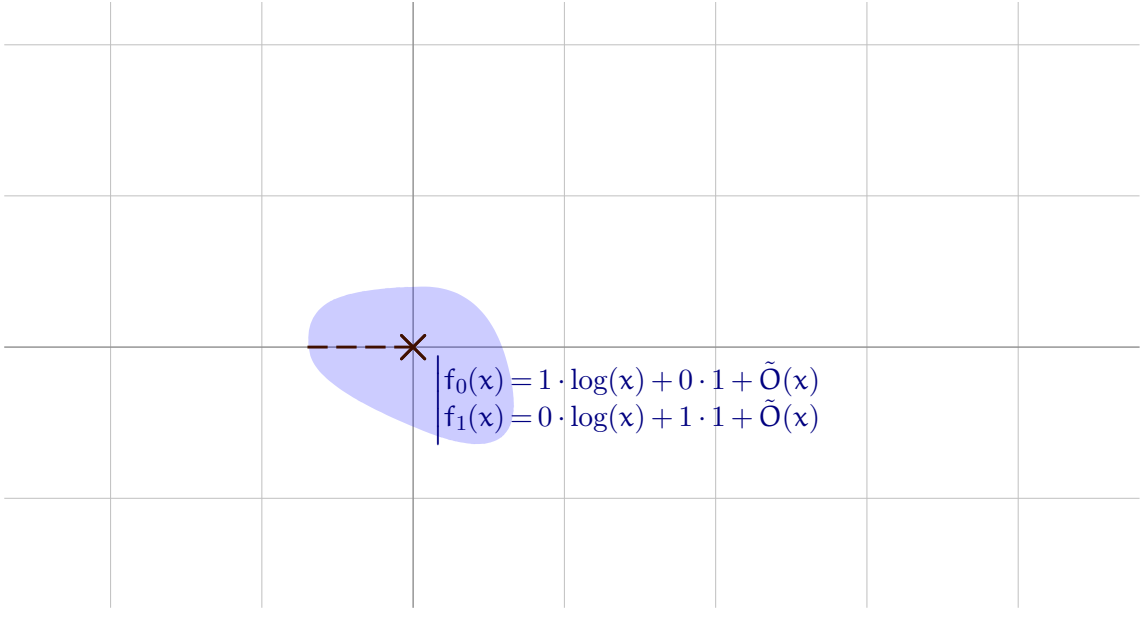
Initial conditions at singular points

$x\,y''(x) + y'(x) + y(x) = 0$ (Bessel$_0$)



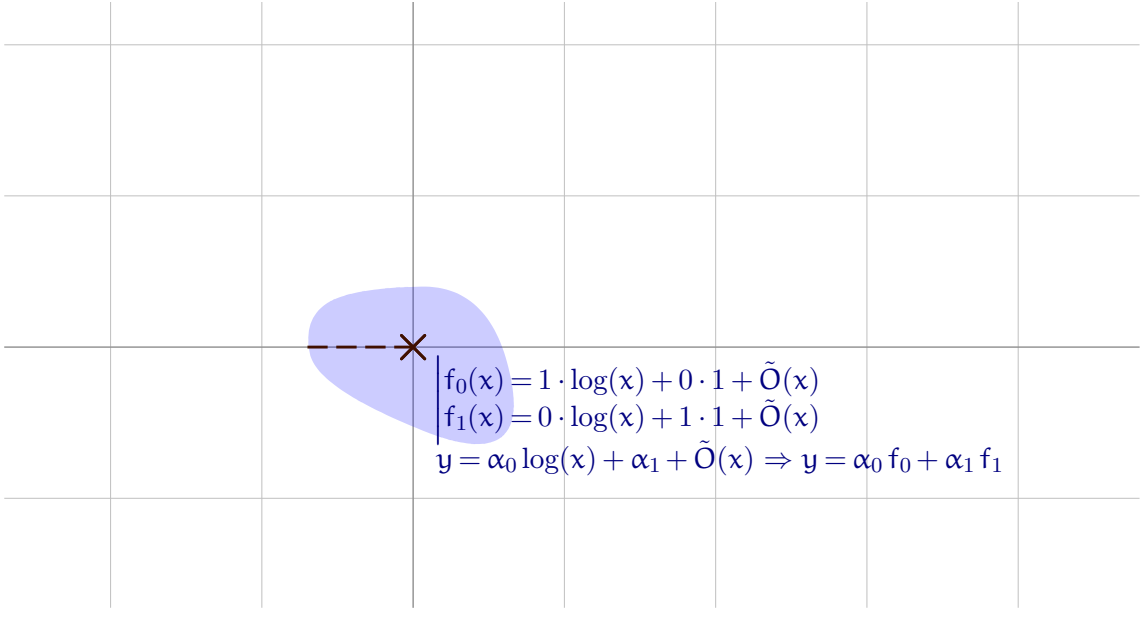
Initial conditions at singular points

$x\,y''(x) + y'(x) + y(x) = 0$ (Bessel$_0$)



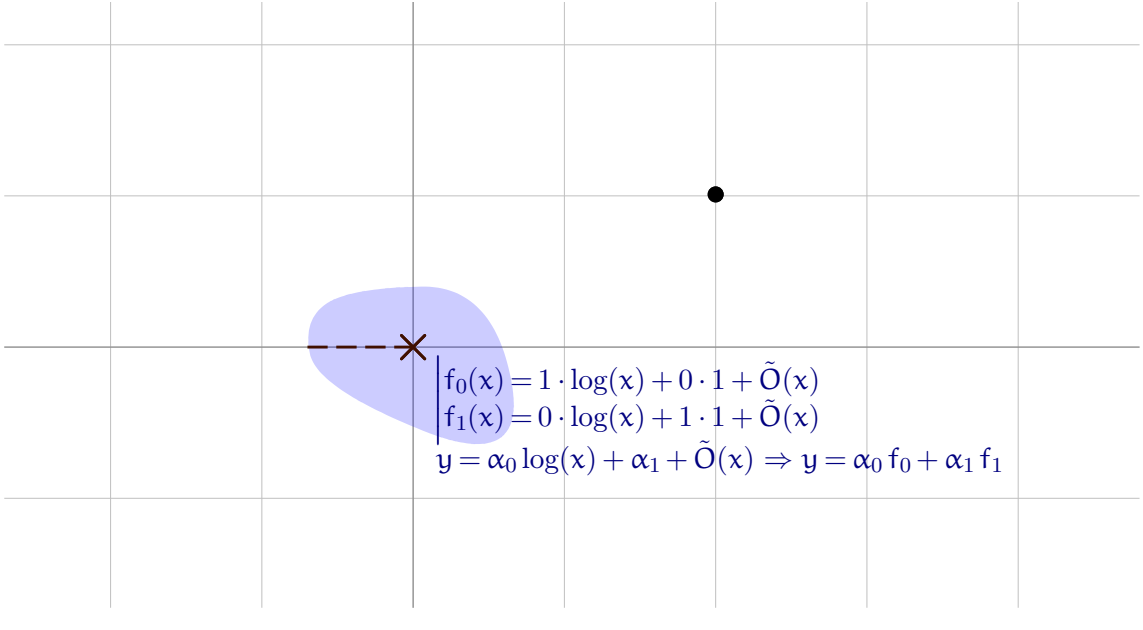
Initial conditions at singular points

$x\,y''(x) + y'(x) + y(x) = 0$ (Bessel$_0$)



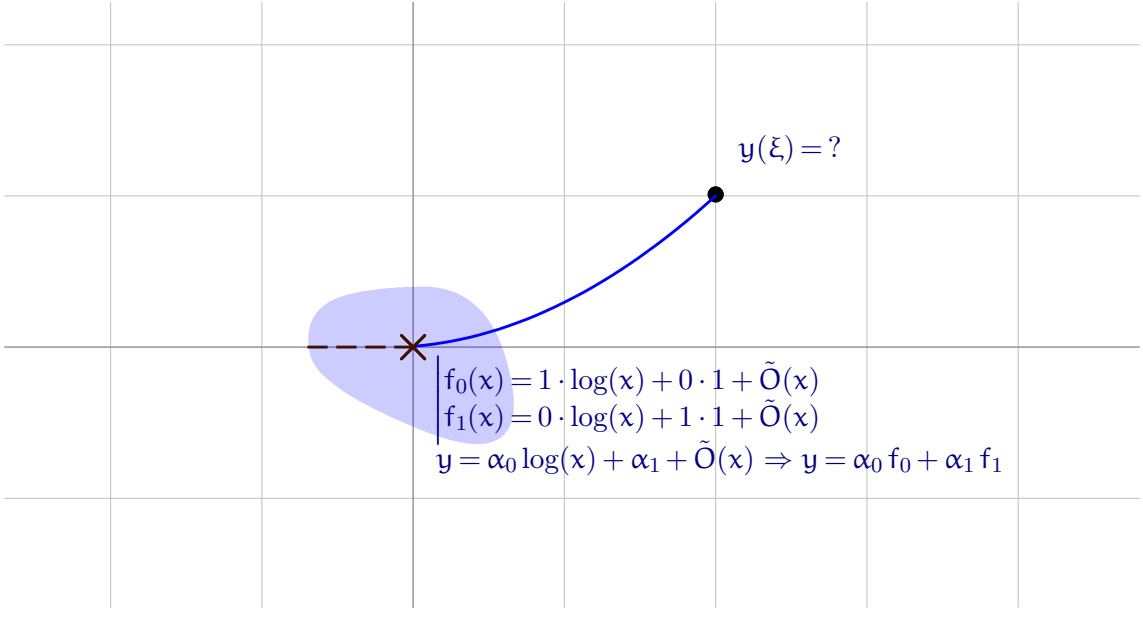
Initial conditions at singular points

$x\,y''(x) + y'(x) + y(x) = 0$ (Bessel$_0$)

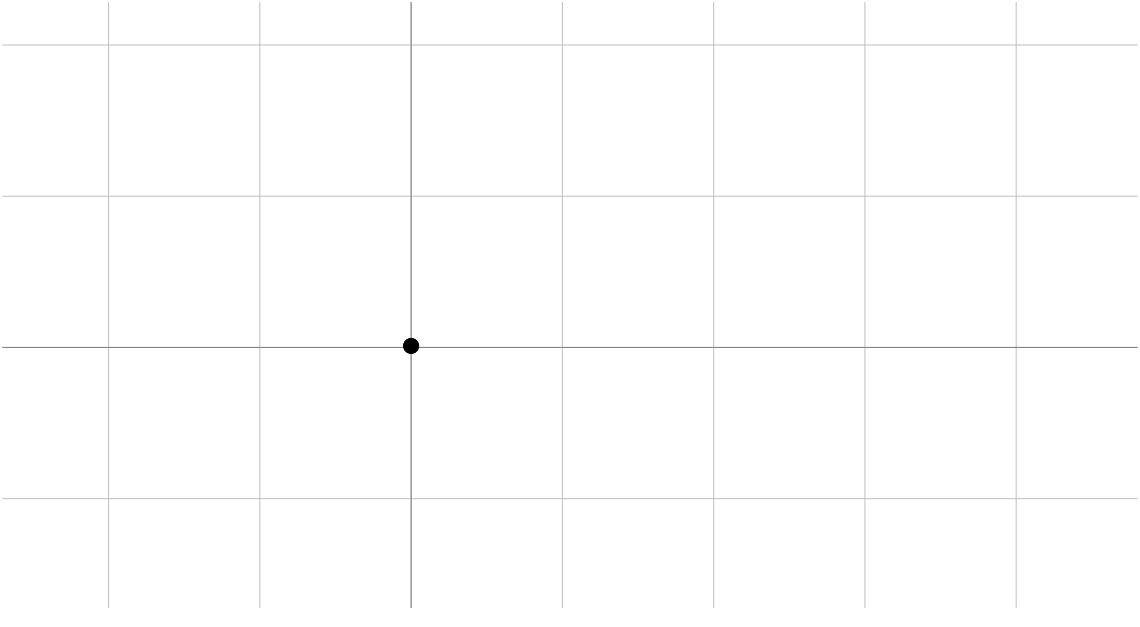


Initial conditions at singular points

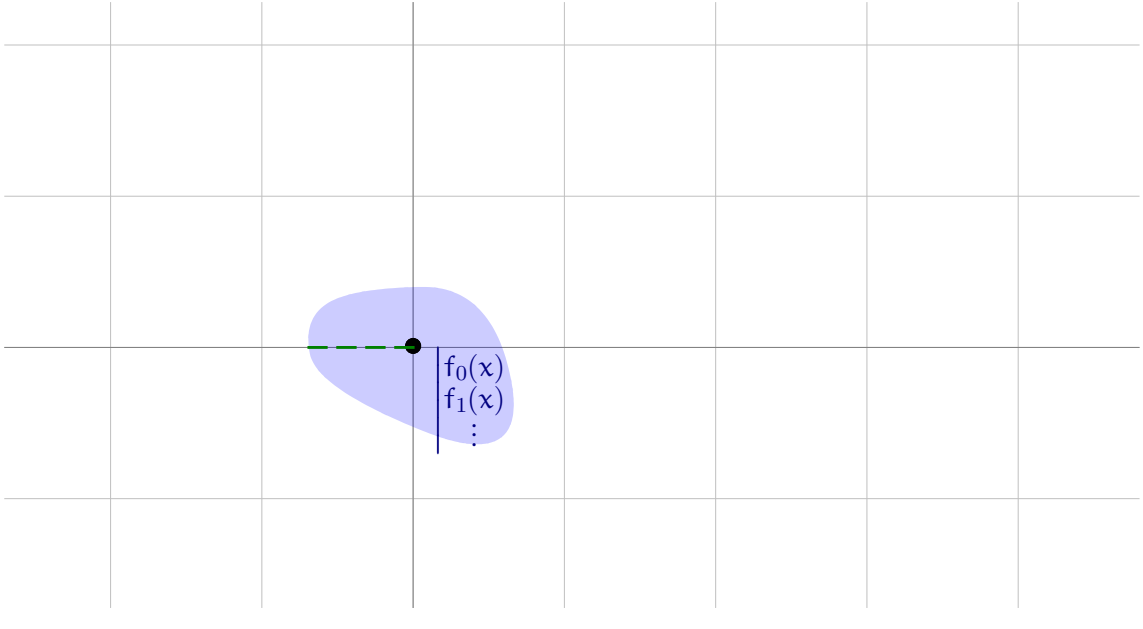
$x\,y''(x) + y'(x) + y(x) = 0$ (Bessel$_0$)



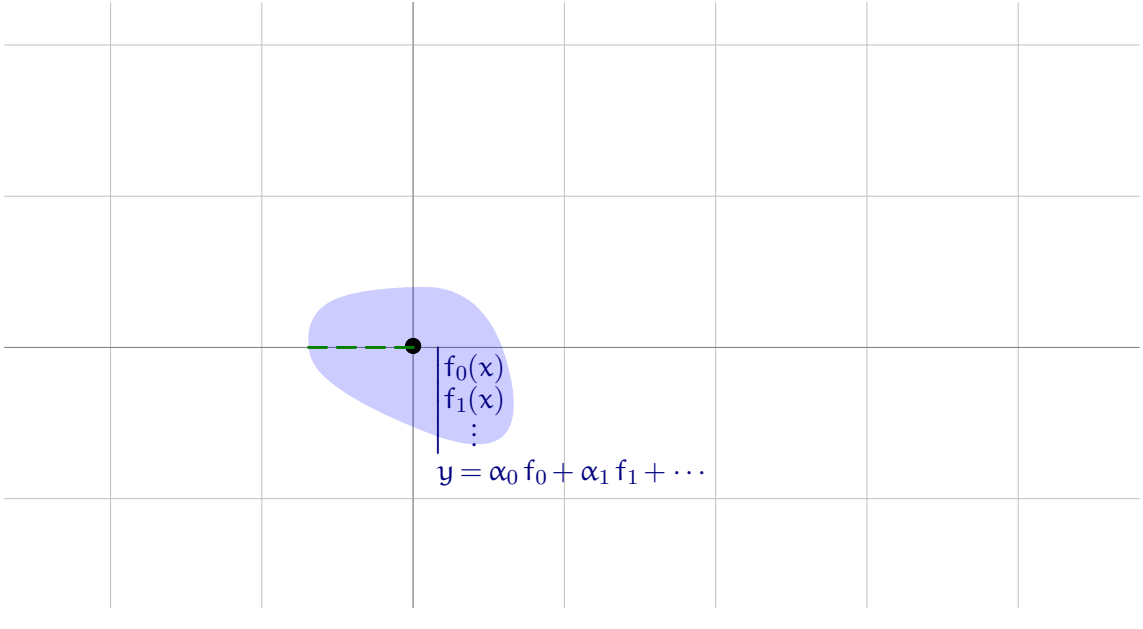
Singular connection matrices



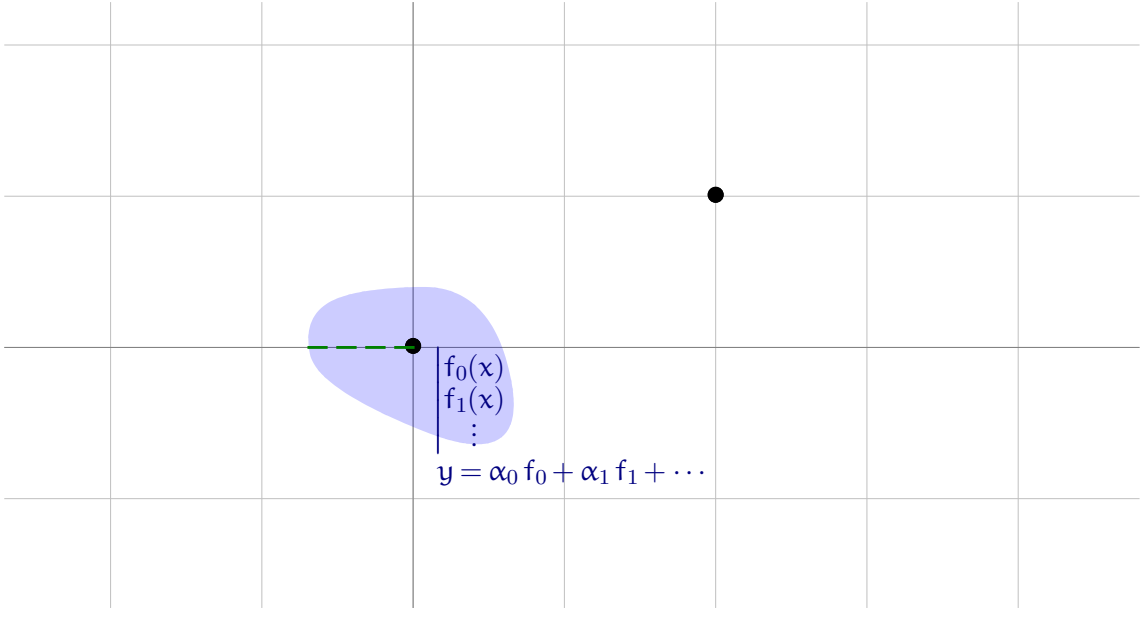
Singular connection matrices



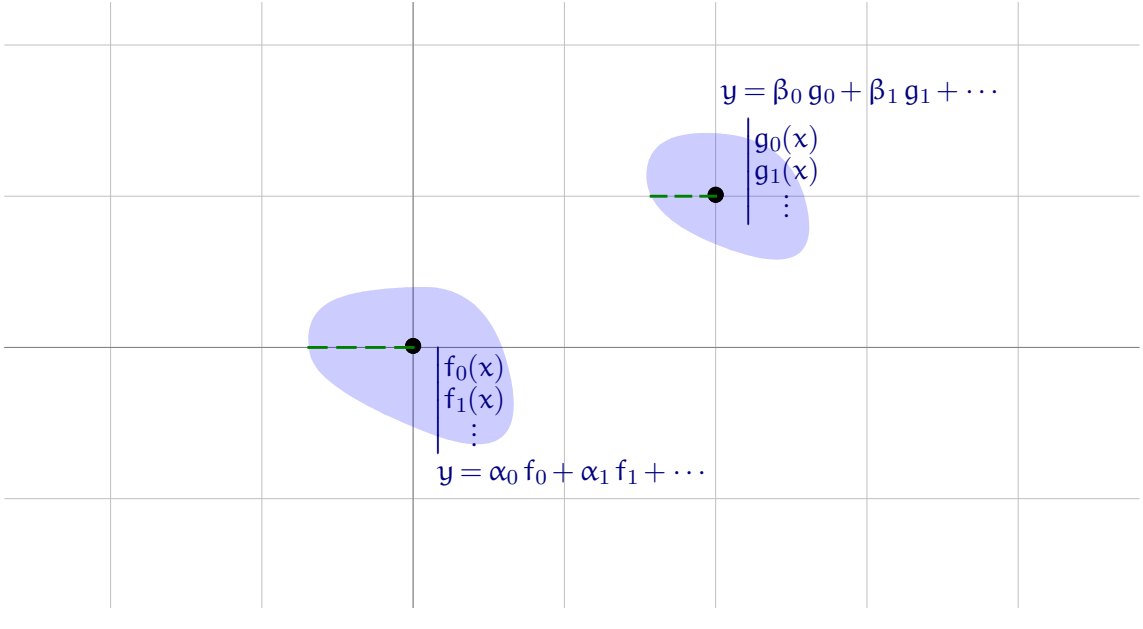
Singular connection matrices



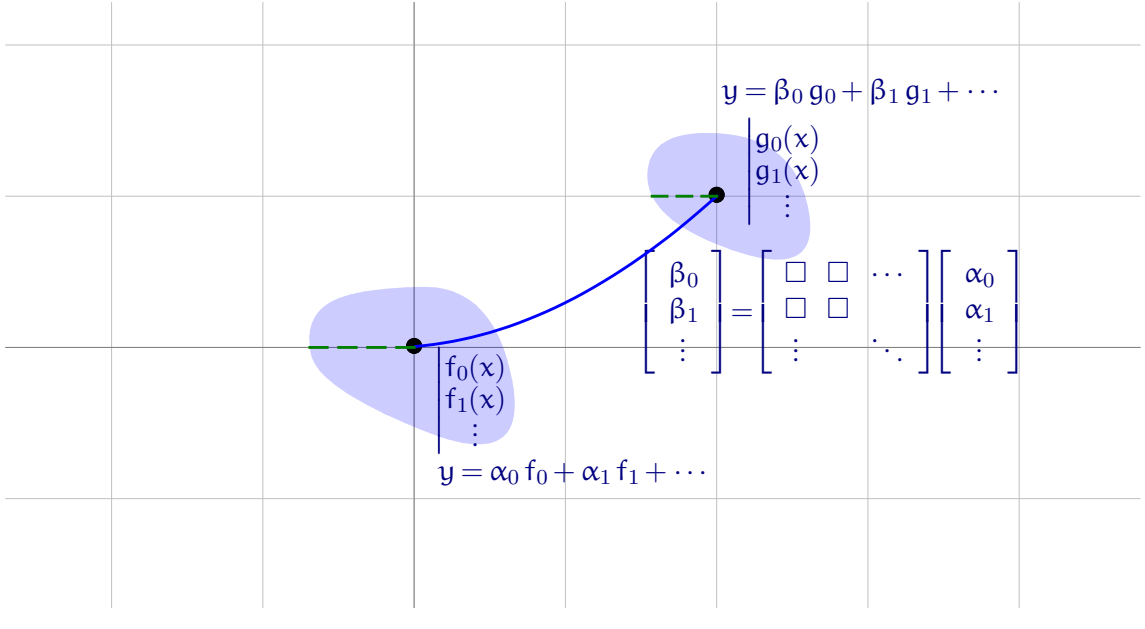
Singular connection matrices



Singular connection matrices



Singular connection matrices



Summary

Problem. Given

- 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 singular points,
- a working precision p,

compute a p-bit numerical approximation of the corresponding connection matrix.

2 Integration Methods

The Runge-Kutta method

$y'(x) = \Phi(x, y(x))$ for very general Φ
possibly given by code

- solution values on a dense grid
  ($\Rightarrow$ everywhere by low-degree interpolation)
- decrease step size to increase accuracy
- typical accuracy: $\sim 10^{-3}$ to 10^{-6}

The Runge-Kutta method

$y'(x) = \Phi(x, y(x))$ for very general Φ
possibly given by code

- solution values on a dense grid
  ($\Rightarrow$ everywhere by low-degree interpolation)
- decrease step size to increase accuracy
- typical accuracy: $\sim 10^{-3}$ to 10^{-6}

Algorithm. RK4 step for $y'(x) = a(x)\,y(x)$

1. $\alpha_1 := a(x)$
2. $\alpha_2 := a\left(x + \frac{h}{2}\right)\left(1 + \alpha_1 \frac{h}{2}\right)$
3. $\alpha_3 := a\left(x + \frac{h}{2}\right)\left(1 + \alpha_2 \frac{h}{2}\right)$
4. $\alpha_4 := a(x + h)\,(1 + \alpha_3 h)$
5. $y_{n+1} := \left(1 + \frac{\alpha_1 + 2\,\alpha_2 + 2\,\alpha_3 + \alpha_4}{6} h\right) y_n$

The Runge-Kutta method

$y'(x) = \Phi(x, y(x))$ for very general Φ
possibly given by code

- solution values on a dense grid
  ($\Rightarrow$ everywhere by low-degree interpolation)
- decrease step size to increase accuracy
- typical accuracy: $\sim 10^{-3}$ to 10^{-6}

Algorithm. RK4 step for $y'(x) = a(x)\,y(x)$

1. $\alpha_1 := a(x)$
2. $\alpha_2 := a\left(x + \frac{h}{2}\right)\left(1 + \alpha_1 \frac{h}{2}\right)$
3. $\alpha_3 := a\left(x + \frac{h}{2}\right)\left(1 + \alpha_2 \frac{h}{2}\right)$
4. $\alpha_4 := a(x + h)\,(1 + \alpha_3 h)$
5. $y_{n+1} := \left(1 + \frac{\alpha_1 + 2\,\alpha_2 + 2\,\alpha_3 + \alpha_4}{6} h\right) y_n$

Cost: $2 + o(1)$ eval + $O(1)$ add/mul

The Runge-Kutta method

$y'(x) = \Phi(x, y(x))$ for very general Φ
possibly given by code

- solution values on a dense grid
  ($\Rightarrow$ everywhere by low-degree interpolation)
- decrease step size to increase accuracy
- typical accuracy: $\sim 10^{-3}$ to 10^{-6}

Algorithm. RK4 step for $y'(x) = a(x)\,y(x)$

1. $\alpha_1 := a(x)$
2. $\alpha_2 := a\left(x + \frac{h}{2}\right)\left(1 + \alpha_1 \frac{h}{2}\right)$
3. $\alpha_3 := a\left(x + \frac{h}{2}\right)\left(1 + \alpha_2 \frac{h}{2}\right)$
4. $\alpha_4 := a(x + h)\,(1 + \alpha_3 h)$
5. $y_{n+1} := \left(1 + \frac{\alpha_1 + 2\,\alpha_2 + 2\,\alpha_3 + \alpha_4}{6} h\right) y_n$

Cost: $2 + o(1)$ eval + $O(1)$ add/mul

Mock theorem. For a fixed Φ satisfying a suitable Lipschitz condition and a fixed interval $[x_0, x_N]$, the solution $(\tilde{y}_n)_{n=0}^N$ computed by RK4 on a grid $(x_n)_{n=0}^N$ of max step $\leqslant h$ satisfies

$$\max_n |\tilde{y}_n - y(x_n)| = O(h^4) \quad \text{as } h \to 0.$$

The Runge-Kutta method

$y'(x) = \Phi(x, y(x))$ for very general Φ
possibly given by code

- solution values on a dense grid
  ($\Rightarrow$ everywhere by low-degree interpolation)
- decrease step size to increase accuracy
- typical accuracy: $\sim 10^{-3}$ to 10^{-6}

Algorithm. RK4 step for $y'(x) = a(x)\,y(x)$

1. $\alpha_1 := a(x)$
2. $\alpha_2 := a\big(x + \frac{h}{2}\big)\big(1 + \alpha_1 \frac{h}{2}\big)$
3. $\alpha_3 := a\big(x + \frac{h}{2}\big)\big(1 + \alpha_2 \frac{h}{2}\big)$
4. $\alpha_4 := a(x + h)\,(1 + \alpha_3 h)$
5. $y_{n+1} := \left(1 + \dfrac{\alpha_1 + 2\,\alpha_2 + 2\,\alpha_3 + \alpha_4}{6}\,h\right) y_n$

Cost: $2 + o(1)$ eval + $O(1)$ add/mul

Mock theorem. For a fixed Φ satisfying a suitable Lipschitz condition and a fixed interval $[x_0, x_N]$, the solution $(\tilde{y}_n)_{n=0}^N$ computed by RK4 on a grid $(x_n)_{n=0}^N$ of max step $\leqslant h$ satisfies

$$\max_n |\tilde{y}_n - y(x_n)| = O(h^4) \quad \text{as } h \to 0.$$

$$\begin{aligned} y(x+h) &= y(x) + y'(x)\,h + y''(x)\,\frac{h^2}{2} + \cdots \\ &= \left(1 + a(x)\,h + (a'(x) + a(x)^2)\,\frac{h^2}{2} + \cdots\right) y(x) \end{aligned}$$

$$y(x+h) = \frac{y_{n+1}}{y_n}\,y(x) + O(h^5)$$

Taylor methods

$$y'(x) = \Phi(x, y(x))$$

Alternative to RK4 with similar convergence:

$$y(x+h) \approx y(x) + y'(x)\,h + \cdots + y^{(4)}(x)\,\frac{h^4}{4!}$$

expressible in terms of $y(x)$ using partial derivatives of Φ

("4th order Taylor method")

- more complex and costly
- arbitrary order

(There are other numerical methods that can reach arbitrary order!)

Taylor methods

$y'(x) = \Phi(x, y(x))$

Alternative to RK4 with similar convergence:

$$y(x+h) \approx y(x) + y'(x)\,h + \cdots + y^{(4)}(x)\,\frac{h^4}{4!}$$

expressible in terms of $y(x)$ using partial derivatives of Φ

("4th order Taylor method")

- more complex and costly
- arbitrary order

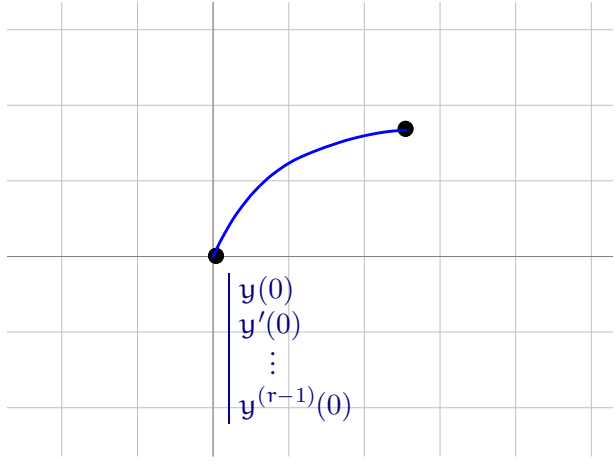
Error $\approx h^p$ means $2^{\sigma/p}$ steps for accuracy $2^{-\sigma}$

To keep #steps polynomial, need $p \approx \sigma$

Variable order
crucial for arbitrary precision

(There are other numerical methods that can reach arbitrary order!)

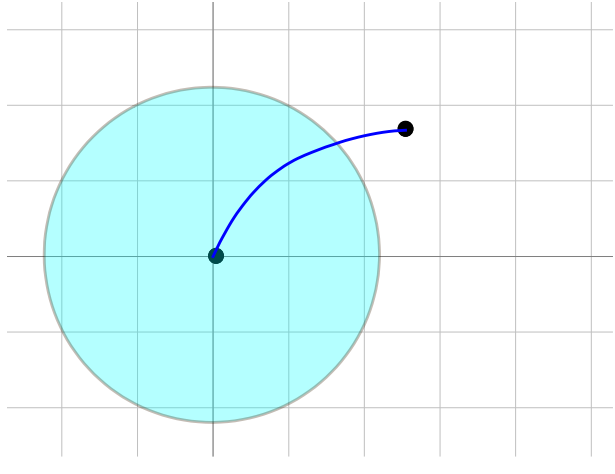
Order-adaptive, large-step Taylor



- Locally, the solutions are given by **convergent power series**
- **Sum the series** numerically to get "initial values" at a new point

(This is just analytic continuation à la Weierstraß.)

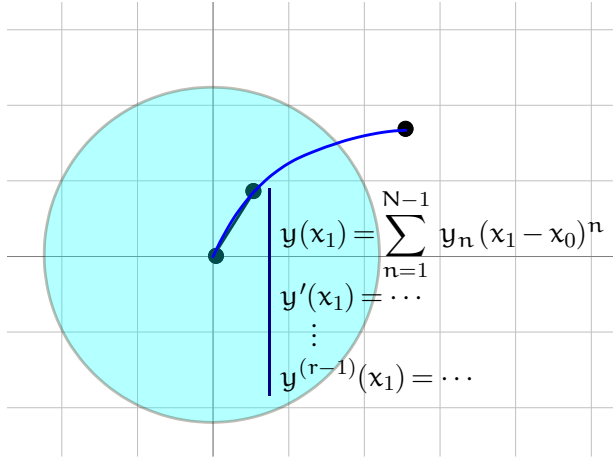
Order-adaptive, large-step Taylor



- Locally, the solutions are given by **convergent power series**
- **Sum the series** numerically to get "initial values" at a new point

(This is just analytic continuation à la Weierstraß.)

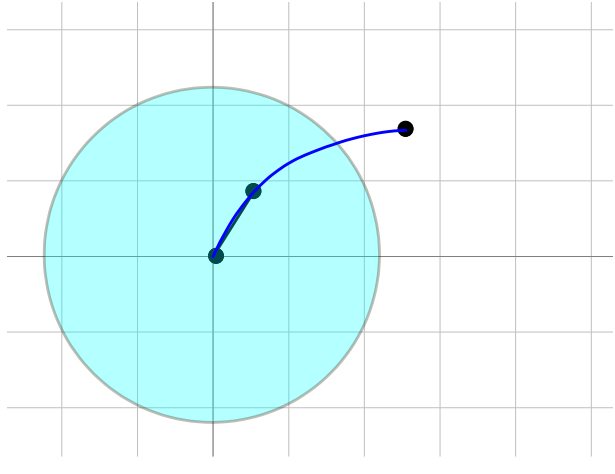
Order-adaptive, large-step Taylor



- Locally, the solutions are given by **convergent power series**
- **Sum the series** numerically to get "initial values" at a new point

(This is just analytic continuation à la Weierstraß.)

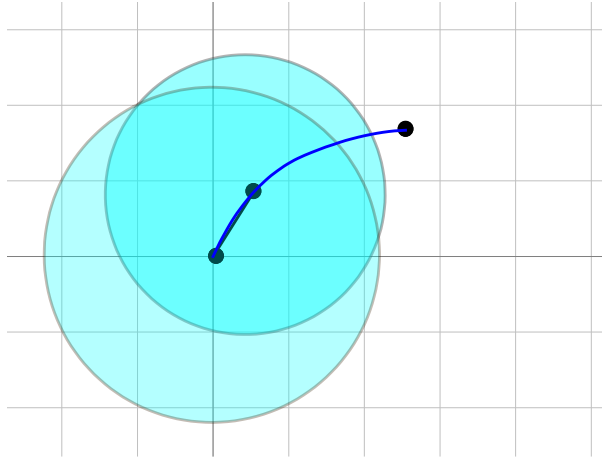
Order-adaptive, large-step Taylor



- Locally, the solutions are given by **convergent power series**
- **Sum the series** numerically to get "initial values" at a new point

(This is just analytic continuation à la Weierstraß.)

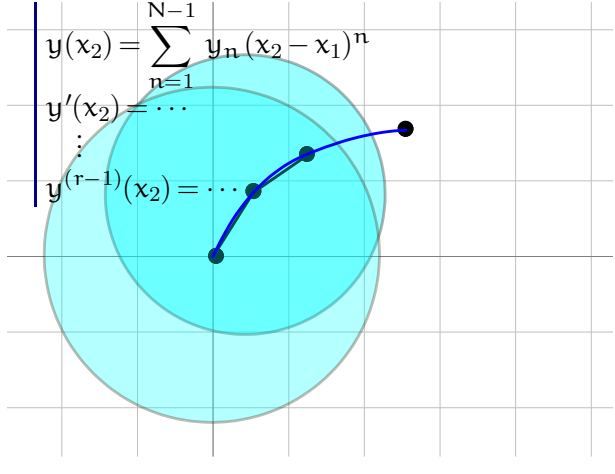
Order-adaptive, large-step Taylor



- Locally, the solutions are given by **convergent power series**
- **Sum the series** numerically to get "initial values" at a new point

(This is just analytic continuation à la Weierstraß.)

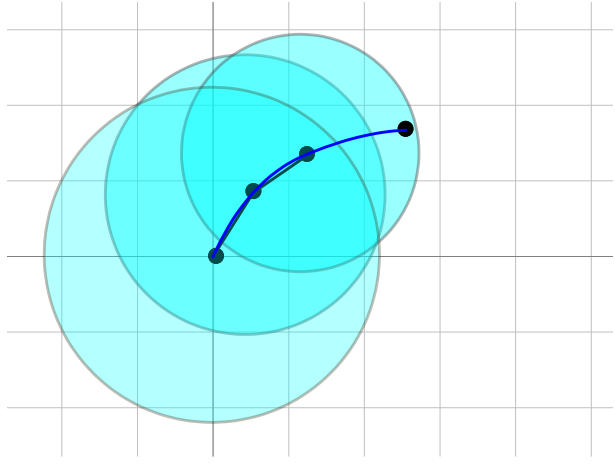
Order-adaptive, large-step Taylor



- Locally, the solutions are given by **convergent power series**
- **Sum the series** numerically to get "initial values" at a new point

(This is just analytic continuation à la Weierstraß.)

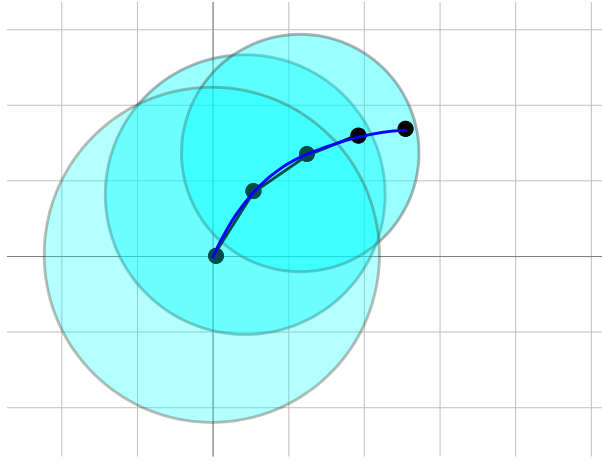
Order-adaptive, large-step Taylor



- Locally, the solutions are given by **convergent power series**
- **Sum the series** numerically to get "initial values" at a new point

(This is just analytic continuation à la Weierstraß.)

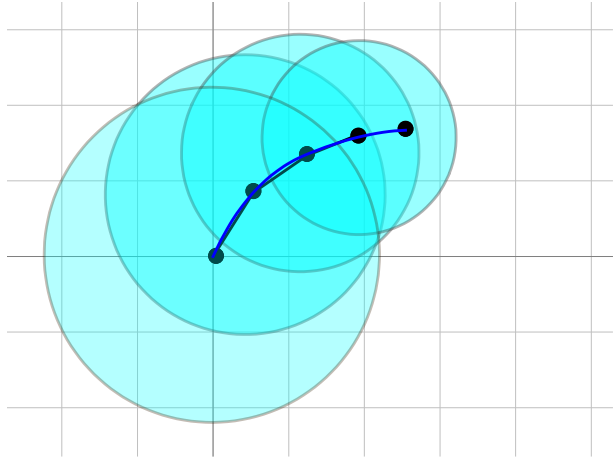
Order-adaptive, large-step Taylor



- Locally, the solutions are given by **convergent power series**
- **Sum the series** numerically to get "initial values" at a new point

(This is just analytic continuation à la Weierstraß.)

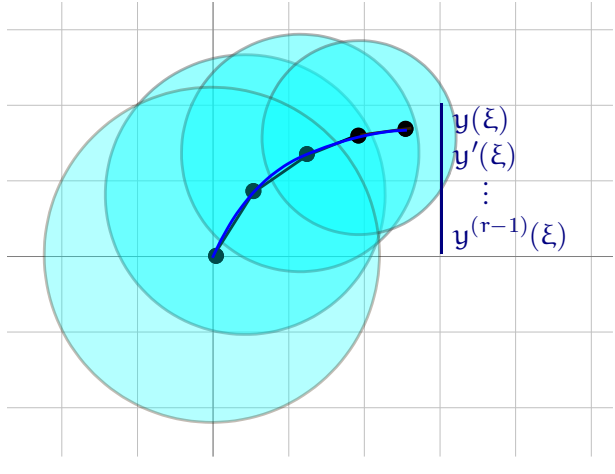
Order-adaptive, large-step Taylor



- Locally, the solutions are given by **convergent power series**
- **Sum the series** numerically to get "initial values" at a new point

(This is just analytic continuation à la Weierstraß.)

Order-adaptive, large-step Taylor



- Locally, the solutions are given by **convergent power series**
- **Sum the series** numerically to get "initial values" at a new point

(This is just analytic continuation à la Weierstraß.)

Speed of convergence and step size

Want:

$$\sum_{n=0}^{\infty} y_n x^n = \sum_{n=0}^{N-1} y_n x^n + \underbrace{\sum_{n=N}^{\infty} y_n x^n}_{|\cdot| \lesssim \varepsilon = 2^{-p}}$$

Speed of convergence and step size

Want:

$$\sum_{n=0}^{\infty} y_n x^n = \sum_{n=0}^{N-1} y_n x^n + \underbrace{\sum_{n=N}^{\infty} y_n x^n}_{|\cdot| \lesssim \varepsilon = 2^{-p}}$$

If the radius of convergence is $> \rho$, then $|y_n \rho^n| \leqslant 1$ for large n, so for $N \gg 0$

$$\left|\sum_{n=N}^{\infty} y_n x^n\right| \leqslant \sum_{n=N}^{\infty} |y_n \rho^n| \left|\frac{x}{\rho}\right|^n \leqslant \sum_{n=0}^{\infty} \left|\frac{x}{\rho}\right|^n \leqslant \left|\frac{x}{\rho}\right|^N \frac{1}{1-|x/\rho|}.$$

Speed of convergence and step size

Want:

$$\sum_{n=0}^{\infty} y_n x^n = \sum_{n=0}^{N-1} y_n x^n + \underbrace{\sum_{n=N}^{\infty} y_n x^n}_{|\cdot| \lesssim \varepsilon = 2^{-p}}$$

If the radius of convergence is $> \rho$, then $|y_n \rho^n| \leqslant 1$ for large n, so for $N \gg 0$

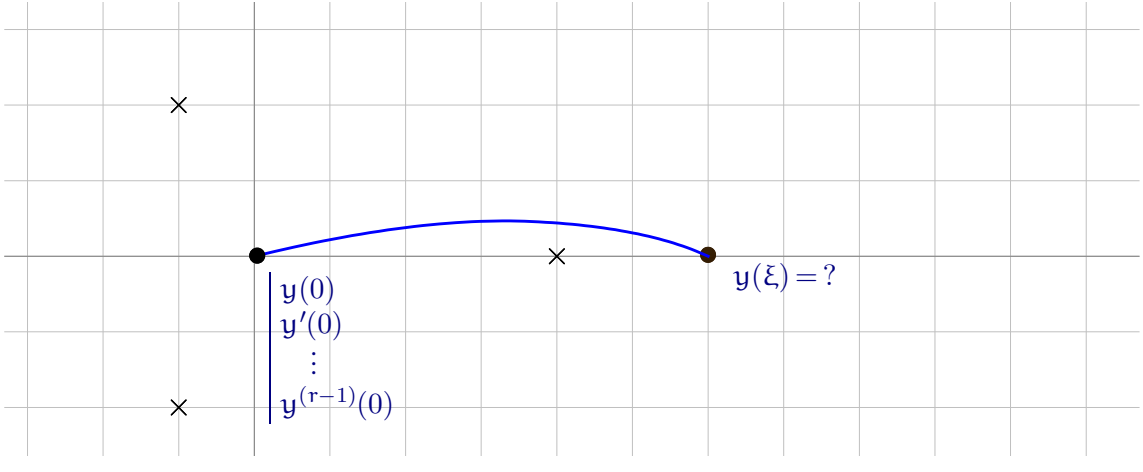
$$\left|\sum_{n=N}^{\infty} y_n x^n\right| \leqslant \sum_{n=N}^{\infty} |y_n \rho^n| \left|\frac{x}{\rho}\right|^n \leqslant \sum_{n=0}^{\infty} \left|\frac{x}{\rho}\right|^n \leqslant \left|\frac{x}{\rho}\right|^N \frac{1}{1-|x/\rho|}.$$

In particular, $\left|\sum_{n=N}^{\infty} y_n x^n\right| \leqslant 2^{-N+1}$ when $|x| \leqslant \rho/2$.

Asymptotically, we can make steps of size $\Omega(\rho)$ regardless of p by taking $N = O(p)$.

Linear case

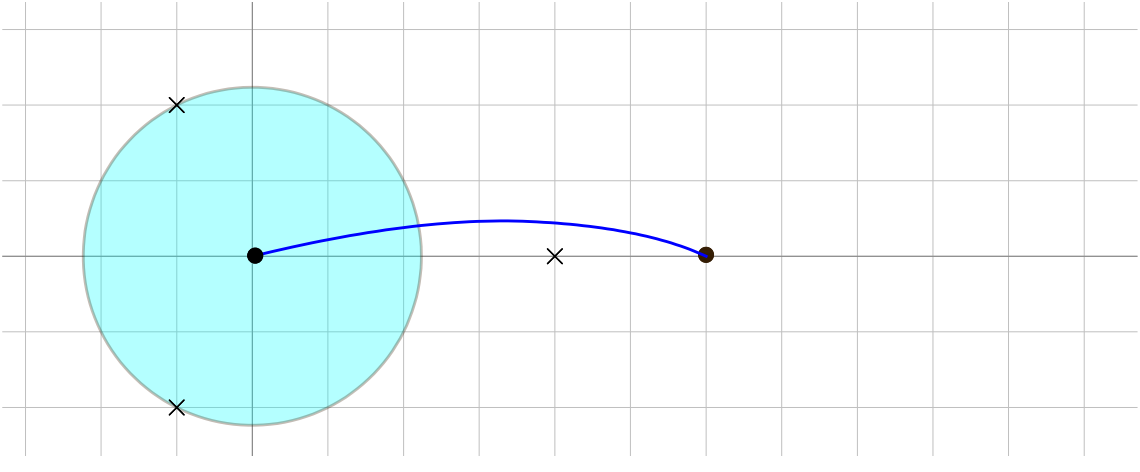
$$a_r(x)\,y^{(r)}(x) + \cdots + a_1(x)\,y'(x) + a_0(x)\,y(x) = 0$$



When computing a full basis: steps are independent
easy error propagation

Linear case

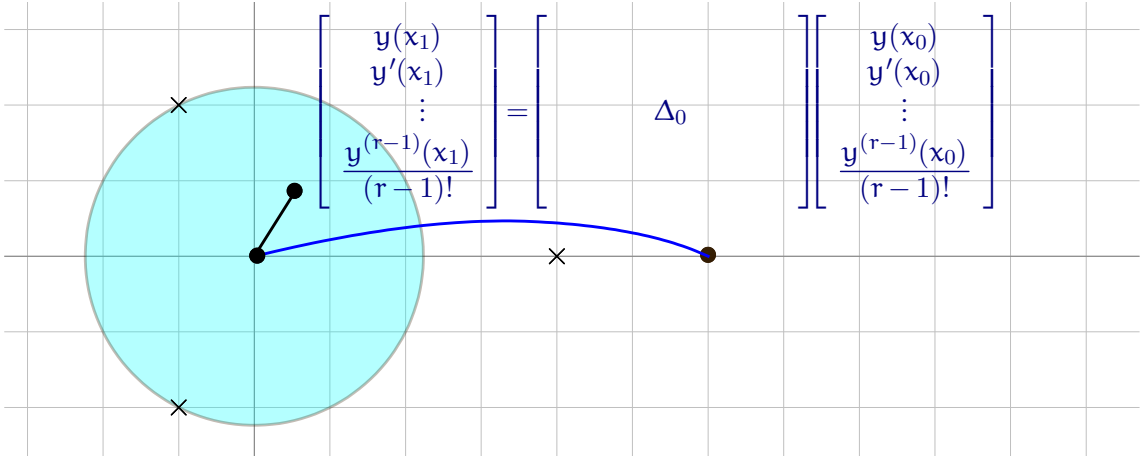
$$a_r(x)\,y^{(r)}(x) + \cdots + a_1(x)\,y'(x) + a_0(x)\,y(x) = 0$$



When computing a full basis: steps are independent
easy error propagation

Linear case

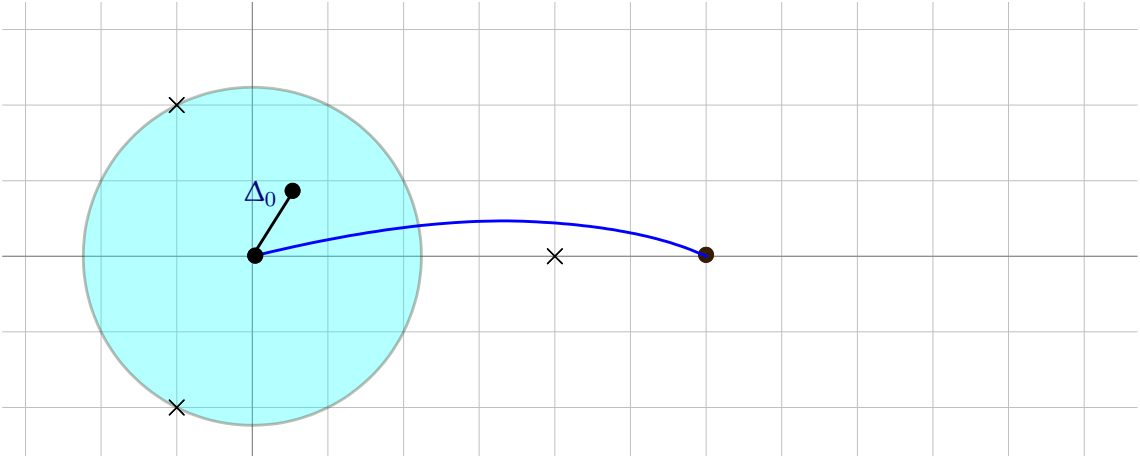
$$a_r(x)\,y^{(r)}(x) + \cdots + a_1(x)\,y'(x) + a_0(x)\,y(x) = 0$$



When computing a full basis: steps are independent
easy error propagation

Linear case

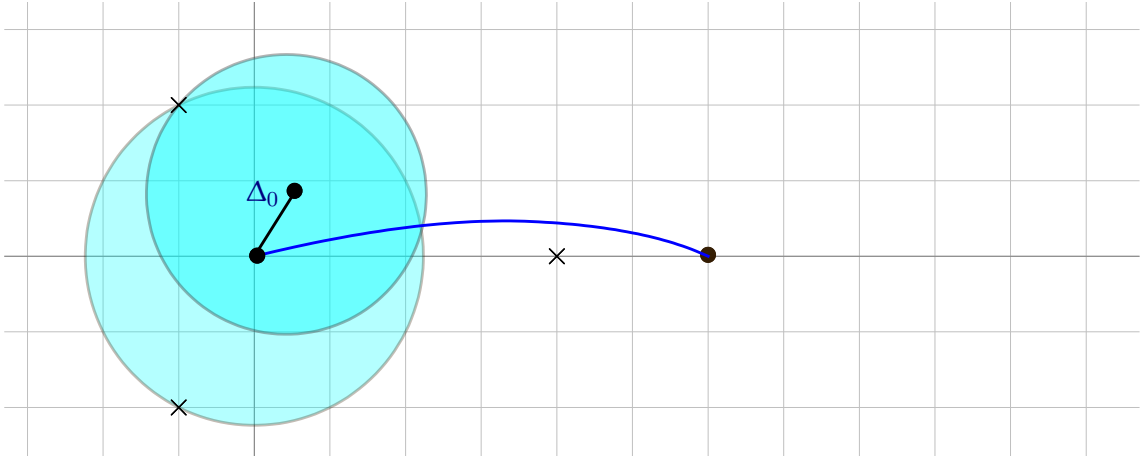
$$a_r(x)\,y^{(r)}(x) + \cdots + a_1(x)\,y'(x) + a_0(x)\,y(x) = 0$$



When computing a full basis: steps are independent
easy error propagation

Linear case

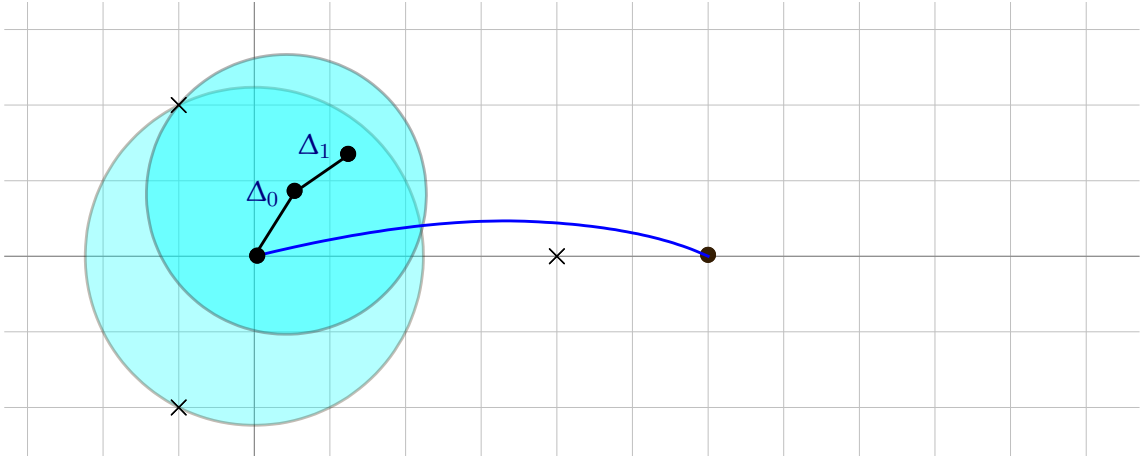
$$a_r(x)\,y^{(r)}(x) + \cdots + a_1(x)\,y'(x) + a_0(x)\,y(x) = 0$$



When computing a full basis: steps are independent
easy error propagation

Linear case

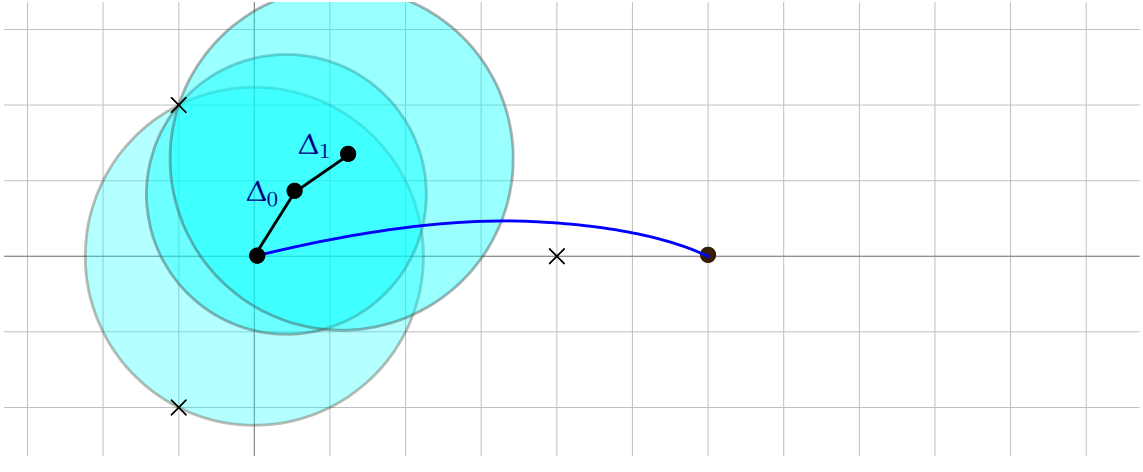
$$a_r(x)\,y^{(r)}(x) + \cdots + a_1(x)\,y'(x) + a_0(x)\,y(x) = 0$$



When computing a full basis: steps are independent
easy error propagation

Linear case

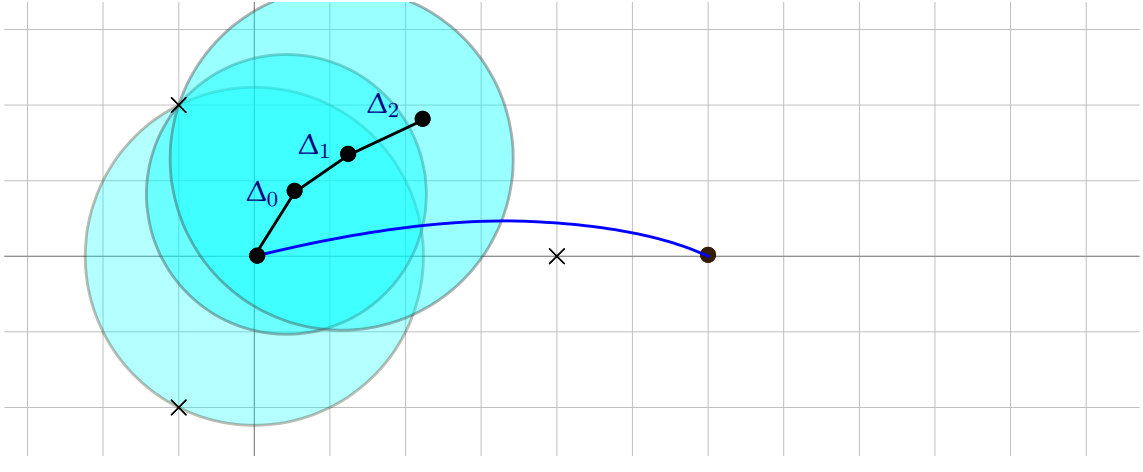
$$a_r(x)\,y^{(r)}(x) + \cdots + a_1(x)\,y'(x) + a_0(x)\,y(x) = 0$$



When computing a full basis: steps are independent
easy error propagation

Linear case

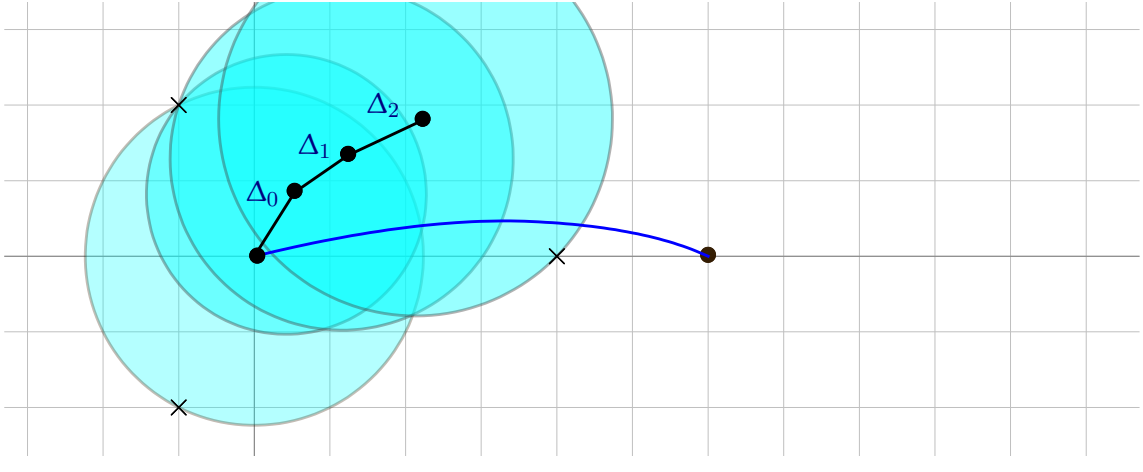
$$a_r(x)\,y^{(r)}(x) + \cdots + a_1(x)\,y'(x) + a_0(x)\,y(x) = 0$$



When computing a full basis: steps are independent
easy error propagation

Linear case

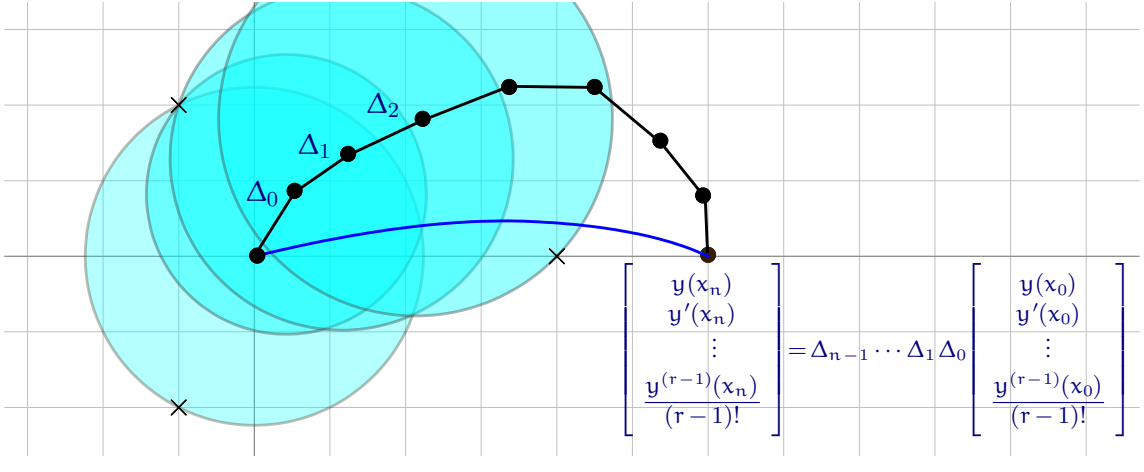
$$a_r(x)\,y^{(r)}(x) + \cdots + a_1(x)\,y'(x) + a_0(x)\,y(x) = 0$$



When computing a full basis: steps are independent
easy error propagation

Linear case

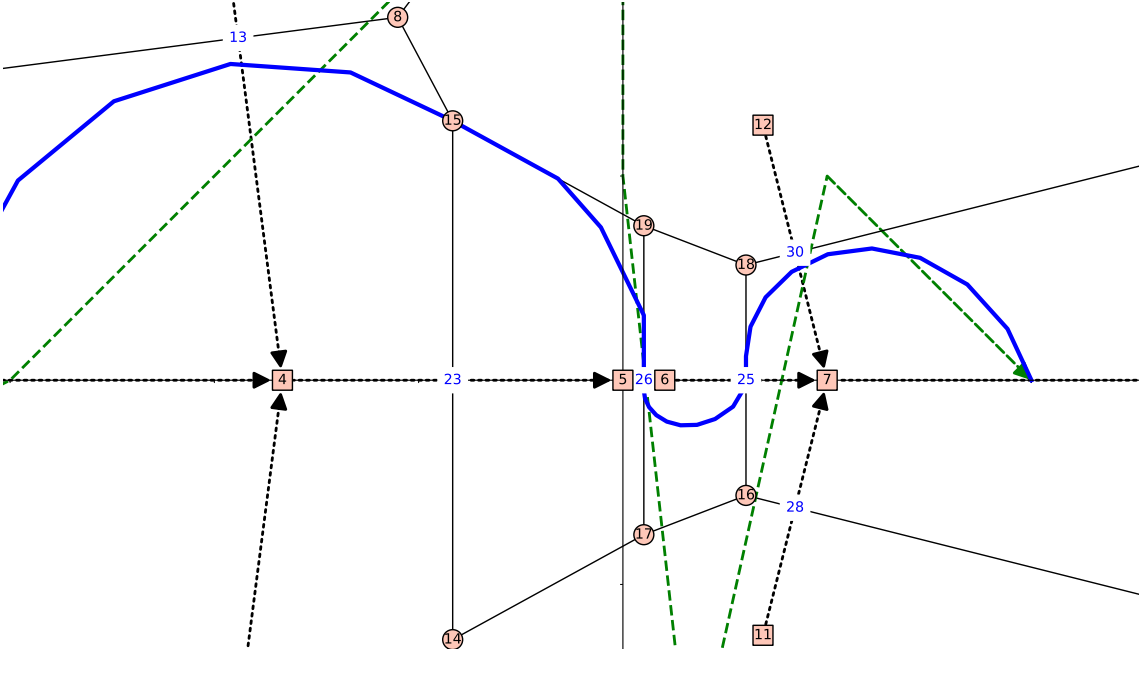
$$a_r(x)\,y^{(r)}(x) + \cdots + a_1(x)\,y'(x) + a_0(x)\,y(x) = 0$$



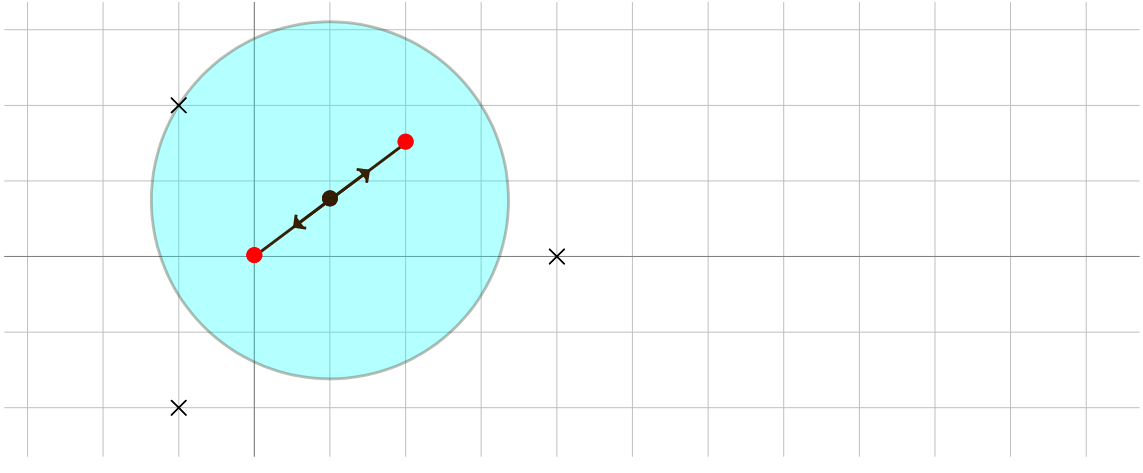
When computing a full basis: steps are independent
easy error propagation

Path optimization

□ = singular points



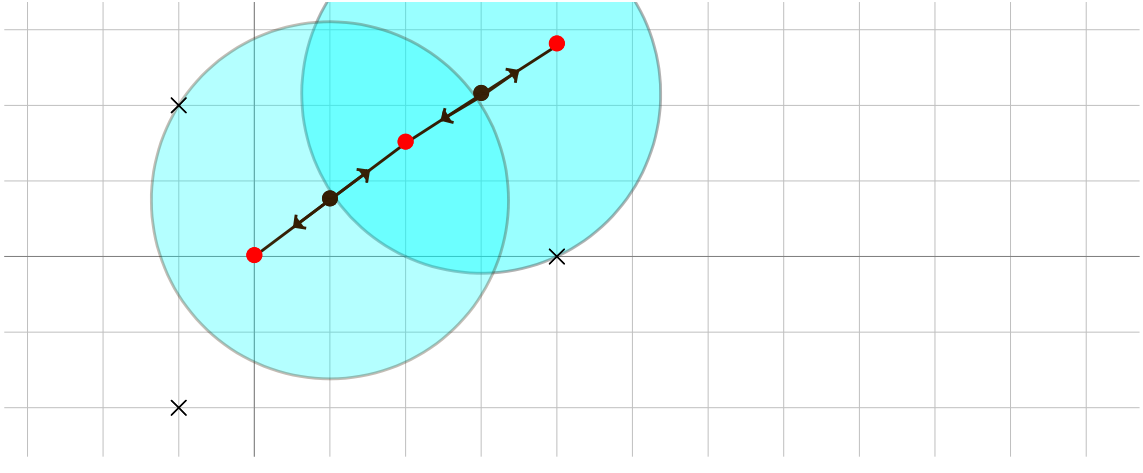
Two-point variant



Evaluate the same series at two points, invert one of the connection matrices

Alternate expansion points • and evaluation points •

Two-point variant



Evaluate the same series at two points, invert one of the connection matrices

Alternate expansion points • and evaluation points •

[corrupted slide deleted]

Recurrences

Lemma. Any differential operator

$$a_r(x)\,D^r + \cdots + a_1(x)\,D + a_0(x), \quad a_i \in \mathbb{K}[x]$$

order r
degree δ

can be rewritten in terms of $\theta = x\,D$ as

$$x^\nu\,[\alpha_r(x)\,\theta^r + \cdots + \alpha_1(x)\,\theta + \alpha_0(x)], \quad \alpha_i \in \mathbb{K}[x], \quad \nu \in \mathbb{Z}.$$

order r
degree $d \leqslant \delta + r$

Recurrences

Lemma. Any differential operator

$$a_r(x)\,D^r + \cdots + a_1(x)\,D + a_0(x), \quad a_i \in \mathbb{K}[x]$$

order r
degree δ

can be rewritten in terms of $\theta = x\,D$ as

$$x^\nu\,[\alpha_r(x)\,\theta^r + \cdots + \alpha_1(x)\,\theta + \alpha_0(x)], \quad \alpha_i \in \mathbb{K}[x], \quad \nu \in \mathbb{Z}.$$

order r
degree $d \leqslant \delta + r$

Proposition. The series $y \in \mathbb{K}((x))$ is solution to

$$L(x, \theta) \cdot y = 0$$

order r
degree $d \leqslant \delta + r$

if and only if the sequence $(y_n)_{n\in\mathbb{Z}}$ is solution to

$$L(S^{-1}, n) \cdot (y_n) = 0.$$

order d
degree r

$S^{-1} : (u_n)_{n\in\mathbb{Z}} \mapsto (u_{n-1})_{n\in\mathbb{Z}}$

Proof. One has $x \cdot \sum y_n x^n = \sum y_{n-1} x^n$ and $\theta \cdot \sum y_n x^n = \sum n y_n x^n$. □

Recurrences

Lemma. Any differential operator

$$a_r(x)\,D^r + \cdots + a_1(x)\,D + a_0(x), \quad a_i \in \mathbb{K}[x]$$

order r
degree δ

can be rewritten in terms of $\theta = x\,D$ as

$$x^\nu\,[\alpha_r(x)\,\theta^r + \cdots + \alpha_1(x)\,\theta + \alpha_0(x)], \quad \alpha_i \in \mathbb{K}[x], \quad \nu \in \mathbb{Z}.$$

order r
degree $d \leqslant \delta + r$

Proposition. The series $y \in \mathbb{K}((x))$ is solution to

$$L(x, \theta) \cdot y = 0$$

order r
degree $d \leqslant \delta + r$

if and only if the sequence $(y_n)_{n\in\mathbb{Z}}$ is solution to

$$L(S^{-1}, n) \cdot (y_n) = 0.$$

order d
degree r

$S^{-1}\colon (u_n)_{n\in\mathbb{Z}} \mapsto (u_{n-1})_{n\in\mathbb{Z}}$

Proof. One has $x \cdot \sum y_n x^n = \sum y_{n-1} x^n$ and $\theta \cdot \sum y_n x^n = \sum n y_n x^n$. □

Summary

Mock algorithm. Generic Taylor method for polynomial coefficients

Input: diff. op. L, path $x_0 \to x_{\text{fin}}$, tolerance ε *Output*: connection matrix $x_0 \to x_{\text{fin}}$

1. Deform the path to stay away from the singular points
2. For $n = 0, 1, \dots$ until $x_n = x_{\text{fin}}$
   a. Select x_{n+1} on the path at distance $\approx \frac{1}{2}\, d(x_n, \text{singular points})$
   b. Change x to $x_n + \xi$ in L
   c. For $f \in \{\text{canonical basis at } x_n\}$ and $0 \leqslant i < r$,
   compute $f^{(i)}(x_1 - x_0)$ using a partial sum of its series expansion s.t. tail $\lesssim \varepsilon$

   If convergence is too slow or numerical issues, retry with x_{n+1} closer to x_n.
3. Return $\Delta_n \cdots \Delta_1 \Delta_0$

3 Regular Singular Points

Solutions at singular points

Examples.

$L = \theta^2 - 2 = x^2 D^2 + x D - 1$ $\qquad$ $y(x) = \alpha x^{\sqrt{2}} + \beta x^{-\sqrt{2}}$

Solutions at singular points

$$x^{\lambda}$$

Examples.

$L = \theta^2 - 2 = x^2 D^2 + x D - 1$ $\qquad$ $y(x) = \alpha x^{\sqrt{2}} + \beta x^{-\sqrt{2}}$

Solutions at singular points

$$x^{\lambda}$$

Examples.

$L = \theta^2 - 2 = x^2 D^2 + x D - 1$ $\qquad y(x) = \alpha x^{\sqrt{2}} + \beta x^{-\sqrt{2}}$

$L = \theta^2 = x^2 D^2 + x D$ $\qquad y(x) = \alpha + \beta \log(x)$

Solutions at singular points

x^λ $\log x$

Examples.

$L = \theta^2 - 2 = x^2 D^2 + x D - 1$ $y(x) = \alpha x^{\sqrt{2}} + \beta x^{-\sqrt{2}}$

$L = \theta^2 = x^2 D^2 + x D$ $y(x) = \alpha + \beta \log(x)$

Solutions at singular points

x^λ $\log x$

Examples.

$L = \theta^2 - 2 = x^2 D^2 + x D - 1$	$y(x) = \alpha x^{\sqrt{2}} + \beta x^{-\sqrt{2}}$
$L = \theta^2 = x^2 D^2 + x D$	$y(x) = \alpha + \beta \log(x)$
$L = 2(x-1) x D + (x+1)$	$y(x) = x^{1/2}(1 + x + x^2 + \cdots)$

Solutions at singular points

$$x^{\lambda}\ f_0(x) \qquad \log x$$

Examples.

$$L = \theta^2 - 2 = x^2 D^2 + x D - 1 \qquad y(x) = \alpha\, x^{\sqrt{2}} + \beta\, x^{-\sqrt{2}}$$
$$L = \theta^2 = x^2 D^2 + x D \qquad y(x) = \alpha + \beta \log(x)$$
$$L = 2(x-1)\, x D + (x+1) \qquad y(x) = x^{1/2}(1 + x + x^2 + \cdots)$$

Solutions at singular points

$$x^{\lambda}\left(f_0(x)+f_1(x)\log x+\cdots+f_K(x)\log(x)^K\right)$$

Examples.

$L=\theta^2-2=x^2D^2+xD-1$	$y(x)=\alpha\,x^{\sqrt{2}}+\beta\,x^{-\sqrt{2}}$	
$L=\theta^2=x^2D^2+xD$	$y(x)=\alpha+\beta\log(x)$	regular
$L=2(x-1)xD+(x+1)$	$y(x)=x^{1/2}(1+x+x^2+\cdots)$	

Solutions at singular points

$$x^{\lambda}\left(f_0(x)+f_1(x)\log x+\cdots+f_K(x)\log(x)^K\right)$$

Examples.

$L=\theta^2-2=x^2D^2+xD-1$	$y(x)=\alpha\,x^{\sqrt{2}}+\beta\,x^{-\sqrt{2}}$	
$L=\theta^2=x^2D^2+xD$	$y(x)=\alpha+\beta\log(x)$	regular
$L=2(x-1)xD+(x+1)$	$y(x)=x^{1/2}(1+x+x^2+\cdots)$	
$L=x^2D-1$	$y(x)=\exp(-1/x)$	
$L=x^2D^2+(3x-1)D+1$	$y(x)=\sum_n n!\,x^n$	(formally)

Solutions at singular points

$$x^{\lambda}\left(f_0(x)+f_1(x)\log x+\cdots+f_K(x)\log(x)^K\right)$$

Examples.

$L=\theta^2-2=x^2D^2+xD-1$	$y(x)=\alpha\,x^{\sqrt{2}}+\beta\,x^{-\sqrt{2}}$	
$L=\theta^2=x^2D^2+xD$	$y(x)=\alpha+\beta\log(x)$	regular
$L=2(x-1)xD+(x+1)$	$y(x)=x^{1/2}(1+x+x^2+\cdots)$	
$L=x^2D-1$	$y(x)=\exp(-1/x)$	irregular
$L=x^2D^2+(3x-1)D+1$	$y(x)=\sum_n n!\,x^n$ (formally)	

The indicial polynomial

$$y(x) = x^{\lambda}\left(f_0(x) + \cdots + f_K(x)\log^K(x)\right)$$

Write the recurrence on $(y_n)_{n\in\mathbb{Z}}$ in the form

$$q_0(n)\,y_n + q_1(n)\,y_{n-1} + \cdots + q_s(n)\,y_{n-s} = 0.$$

The indicial polynomial

$$y(x) = x^{\lambda}\left(f_0(x) + \cdots + f_K(x)\log^K(x)\right)$$

Write the recurrence on $(y_n)_{n\in\mathbb{Z}}$ in the form

$$q_0(n)\,y_n + q_1(n)\,y_{n-1} + \cdots + q_s(n)\,y_{n-s} = 0.$$

- The valuation of any Laurent series solution must be a root of q_0.

The indicial polynomial

$$y(x) = x^{\lambda}\left(f_0(x) + \cdots + f_K(x)\log^K(x)\right)$$

Write the recurrence on $(y_n)_{n\in\mathbb{Z}}$ in the form

$$q_0(n)\,y_n + q_1(n)\,y_{n-1} + \cdots + q_s(n)\,y_{n-s} = 0.$$

Definition. The polynomial q_0 is called the **indicial polynomial** of L at 0.
The polynomial obtained in the same way after $x \leftarrow \xi + x$ is called the indicial polynomial at ξ.

- The valuation of any Laurent series solution must be a root of q_0.

The indicial polynomial

$$y(x) = x^{\lambda}\left(f_0(x) + \cdots + f_K(x)\log^K(x)\right)$$

Write the recurrence on $(y_n)_{n\in\mathbb{Z}}$ in the form

$$q_0(n)\,y_n + q_1(n)\,y_{n-1} + \cdots + q_s(n)\,y_{n-s} = 0.$$

Definition. The polynomial q_0 is called the **indicial polynomial** of L at 0.
The polynomial obtained in the same way after $x \leftarrow \xi + x$ is called the indicial polynomial at ξ.

- The valuation of any Laurent series solution must be a root of q_0.
  If there is no larger integer root, the recurrence defines a formal series solution.

The indicial polynomial

$$y(x) = x^\lambda \left(f_0(x) + \cdots + f_K(x) \log^K(x) \right)$$

Write the recurrence on $(y_n)_{n\in\mathbb{Z}}$ in the form

$$q_0(n)\, y_n + q_1(n)\, y_{n-1} + \cdots + q_s(n)\, y_{n-s} = 0.$$

Definition. The polynomial q_0 is called the **indicial polynomial** of L at 0.
The polynomial obtained in the same way after $x \leftarrow \xi + x$ is called the indicial polynomial at ξ.

- The valuation of any Laurent series solution must be a root of q_0.
  If there is no larger integer root, the recurrence defines a formal series solution.
- When $q_0(\lambda) = 0$ for some $\lambda \notin \mathbb{Z}$, look for solutions $x^\lambda f(x)$ with $f(x) \in \mathbb{C}[[x]]$.
  The recurrence remains valid.

The indicial polynomial

$$y(x) = x^\lambda \left(f_0(x) + \cdots + f_K(x) \log^K(x) \right)$$

Write the recurrence on $(y_n)_{n \in \mathbb{Z}}$ in the form

$$q_0(n)\, y_n + q_1(n)\, y_{n-1} + \cdots + q_s(n)\, y_{n-s} = 0.$$

Definition. The polynomial q_0 is called the **indicial polynomial** of L at 0.
The polynomial obtained in the same way after $x \leftarrow \xi + x$ is called the indicial polynomial at ξ.

- The valuation of any Laurent series solution must be a root of q_0.
  If there is no larger integer root, the recurrence defines a formal series solution.
- When $q_0(\lambda) = 0$ for some $\lambda \notin \mathbb{Z}$, look for solutions $x^\lambda f(x)$ with $f(x) \in \mathbb{C}[[x]]$.
  The recurrence remains valid.
- In the presence of integer root differences,
  consider solutions $x^\lambda \left(f_0(x) + f_1(x) \log x + \cdots + f_{r-1}(x) \log(x)^{r-1} \right)$.

This results in $\deg q_0$ linearly independent solutions.

Regular singular points

$L = a_r(x)\,D^r + \cdots + a_0(x)$ [Fuchs, 1866]

Definition. A singular point where the indicial polynomial has degree r (=eq. order) is called a **regular singular point**.

Theorem. A point $\xi \in \mathbb{C}$ is a regular singular point of L iff every solution y of $L(y) = 0$ satisfies, for some $c \in \mathbb{Z}$,

$$y(x) = O(|x - \xi|^{-c}) \quad \text{as } x \to \xi.$$

Theorem. Suppose that 0 is a regular singular point of L.

- L admits r linearly independent formal solutions of the form

$$x^\lambda \left(f_0(x) + f_1(x) \log x + \cdots + f_K(x) \log(x)^K \right), \qquad f_k \in \mathbb{K}(\lambda)[[x]],$$

  with $K < r$ and λ among the roots of the indicial polynomial.
- The f_k converge in a disk extending at least to the closest other singular point.

Example

$L = x D^2 - D + 1$

$$y(x) = \sum u_n x^n + \sum v_n x^n \log(x)$$

Example

$$L = x\,D^2 - D + 1$$

$$\begin{aligned} y(x) &= \sum u_n x^n + \sum v_n x^n \log(x) \\ y'(x) &= \sum n\,u_n x^{n-1} + \sum n\,v_n x^{n-1} \log(x) + \sum v_n x^{n-1} \end{aligned}$$

Example

$L = x D^2 - D + 1$

$$
\begin{aligned}
y(x) &= \sum u_n x^n + \sum v_n x^n \log(x) \\
y'(x) &= \sum n u_n x^{n-1} + \sum n v_n x^{n-1} \log(x) + \sum v_n x^{n-1} \\
&= \sum (n u_n + v_n)\, x^{n-1} + \sum n v_n\, x^{n-1} \log(x)
\end{aligned}
$$

Example

$$L = x\,D^2 - D + 1$$

$$\begin{aligned}
y(x) &= \sum u_n x^n + \sum v_n x^n \log(x) \\
y'(x) &= \sum n\,u_n x^{n-1} + \sum n\,v_n x^{n-1}\log(x) + \sum v_n x^{n-1} \\
&= \sum (n\,u_n + v_n)\, x^{n-1} + \sum n\,v_n\, x^{n-1}\log(x) \\
y''(x) &= \sum \underbrace{[(n-1)\,(n\,u_n + v_n) + n\,v_n]}_{(n^2-n)u_n + (2n-1)v_n} x^{n-2} + \sum (n-1)\,n\,v_n\, x^{n-2}\log(x)
\end{aligned}$$

Example

$$L = x\,D^2 - D + 1$$

$$\begin{aligned}
y(x) &= \sum u_n x^n + \sum v_n x^n \log(x) \\
y'(x) &= \sum n\,u_n x^{n-1} + \sum n\,v_n x^{n-1}\log(x) + \sum v_n x^{n-1} \\
&= \sum (n\,u_n + v_n)\, x^{n-1} + \sum n\,v_n\, x^{n-1}\log(x) \\
y''(x) &= \sum \underbrace{[(n-1)\,(n\,u_n + v_n) + n\,v_n]}_{(n^2-n)u_n + (2n-1)v_n} x^{n-2} + \sum (n-1)\, n\,v_n\, x^{n-2}\log(x)
\end{aligned}$$

$$L\,y(x) = \sum \underbrace{[n\,(n-2)\,u_n + u_{n-1} + 2\,n\,v_n]}_{\mathrm{REC}_0} x^{n-1} + \sum \underbrace{[n\,(n-2)\,v_n + v_{n-1}]}_{\mathrm{REC}_1} x^{n-1}\log(x)$$

Example

$$L = x\,D^2 - D + 1$$

$$\begin{aligned}
y(x) &= \sum u_n x^n + \sum v_n x^n \log(x) \\
y'(x) &= \sum n\,u_n x^{n-1} + \sum n\,v_n x^{n-1}\log(x) + \sum v_n x^{n-1} \\
&= \sum (n\,u_n + v_n)\, x^{n-1} + \sum n\,v_n\, x^{n-1}\log(x) \\
y''(x) &= \sum \underbrace{[(n-1)\,(n\,u_n + v_n) + n\,v_n]}_{(n^2-n)u_n + (2n-1)v_n} x^{n-2} + \sum (n-1)\, n\,v_n\, x^{n-2}\log(x)
\end{aligned}$$

$$Ly(x) = \sum \underbrace{[n\,(n-2)\,u_n + u_{n-1} + 2\,n\,v_n]}_{\mathrm{REC}_0} x^{n-1} + \sum \underbrace{[n\,(n-2)\,v_n + v_{n-1}]}_{\mathrm{REC}_1} x^{n-1}\log(x)$$

v_n	0						$\cdots$
u_n	1						$\cdots$
n	0	1	2	3	4	5	$\cdots$

Example

$$L = x\,D^2 - D + 1$$

$$\begin{aligned}
y(x) &= \sum u_n x^n + \sum v_n x^n \log(x) \\
y'(x) &= \sum n\,u_n x^{n-1} + \sum n\,v_n x^{n-1}\log(x) + \sum v_n x^{n-1} \\
&= \sum (n\,u_n + v_n)\, x^{n-1} + \sum n\,v_n\, x^{n-1}\log(x) \\
y''(x) &= \sum \underbrace{[(n-1)\,(n\,u_n + v_n) + n\,v_n]}_{(n^2-n)u_n + (2n-1)v_n} x^{n-2} + \sum (n-1)\, n\,v_n\, x^{n-2}\log(x)
\end{aligned}$$

$$Ly(x) = \sum \underbrace{[n\,(n-2)\,u_n + u_{n-1} + 2\,n\,v_n]}_{\mathrm{REC}_0} x^{n-1} + \sum \underbrace{[n\,(n-2)\,v_n + v_{n-1}]}_{\mathrm{REC}_1} x^{n-1}\log(x)$$

v_n	0	0					$\cdots$
u_n	1						$\cdots$
n	0	1	2	3	4	5	$\cdots$

Example

$$L = x\,D^2 - D + 1$$

$$\begin{aligned}
y(x) &= \sum u_n x^n + \sum v_n x^n \log(x) \\
y'(x) &= \sum n\,u_n x^{n-1} + \sum n\,v_n x^{n-1}\log(x) + \sum v_n x^{n-1} \\
&= \sum (n\,u_n + v_n)\, x^{n-1} + \sum n\,v_n\, x^{n-1}\log(x) \\
y''(x) &= \sum \underbrace{[(n-1)\,(n\,u_n + v_n) + n\,v_n]}_{(n^2-n)u_n + (2n-1)v_n} x^{n-2} + \sum (n-1)\, n\,v_n\, x^{n-2}\log(x)
\end{aligned}$$

$$Ly(x) = \sum \underbrace{[n\,(n-2)\,u_n + u_{n-1} + 2\,n\,v_n]}_{\mathrm{REC}_0} x^{n-1} + \sum \underbrace{[n\,(n-2)\,v_n + v_{n-1}]}_{\mathrm{REC}_1} x^{n-1}\log(x)$$

v_n	0	0					$\cdots$
u_n	1	1					$\cdots$
n	0	1	2	3	4	5	$\cdots$

Example

$$L = x\,D^2 - D + 1$$

$$\begin{aligned}
y(x) &= \sum u_n x^n + \sum v_n x^n \log(x) \\
y'(x) &= \sum n\,u_n x^{n-1} + \sum n\,v_n x^{n-1}\log(x) + \sum v_n x^{n-1} \\
&= \sum (n\,u_n + v_n)\, x^{n-1} + \sum n\,v_n\, x^{n-1}\log(x) \\
y''(x) &= \sum \underbrace{[(n-1)\,(n\,u_n + v_n) + n\,v_n]}_{(n^2-n)u_n + (2n-1)v_n} x^{n-2} + \sum (n-1)\, n\,v_n\, x^{n-2}\log(x)
\end{aligned}$$

$$Ly(x) = \sum \underbrace{[n\,(n-2)\,u_n + u_{n-1} + 2\,n\,v_n]}_{\mathrm{REC}_0} x^{n-1} + \sum \underbrace{[n\,(n-2)\,v_n + v_{n-1}]}_{\mathrm{REC}_1} x^{n-1}\log(x)$$

v_n	0	0	$-\frac{1}{2}$				$\cdots$
u_n	1	1					$\cdots$
n	0	1	2	3	4	5	$\cdots$

Example

$$L = x\,D^2 - D + 1$$

$$\begin{aligned}
y(x) &= \sum u_n x^n + \sum v_n x^n \log(x) \\
y'(x) &= \sum n\,u_n x^{n-1} + \sum n\,v_n x^{n-1}\log(x) + \sum v_n x^{n-1} \\
&= \sum (n\,u_n + v_n)\, x^{n-1} + \sum n\,v_n\, x^{n-1}\log(x) \\
y''(x) &= \sum \underbrace{[(n-1)\,(n\,u_n + v_n) + n\,v_n]}_{(n^2-n)u_n + (2n-1)v_n} x^{n-2} + \sum (n-1)\,n\,v_n\, x^{n-2}\log(x)
\end{aligned}$$

$$Ly(x) = \sum \underbrace{[n\,(n-2)\,u_n + u_{n-1} + 2\,n\,v_n]}_{\mathrm{REC}_0} x^{n-1} + \sum \underbrace{[n\,(n-2)\,v_n + v_{n-1}]}_{\mathrm{REC}_1} x^{n-1}\log(x)$$

v_n	0	0	$-\frac{1}{2}$				$\cdots$
u_n	1	1	0				$\cdots$
n	0	1	2	3	4	5	$\cdots$

Example

$$L = x\,D^2 - D + 1$$

$$\begin{aligned}
y(x) &= \sum u_n x^n + \sum v_n x^n \log(x) \\
y'(x) &= \sum n\,u_n x^{n-1} + \sum n\,v_n x^{n-1}\log(x) + \sum v_n x^{n-1} \\
&= \sum (n\,u_n + v_n)\, x^{n-1} + \sum n\,v_n\, x^{n-1}\log(x) \\
y''(x) &= \sum \underbrace{[(n-1)\,(n\,u_n + v_n) + n\,v_n]}_{(n^2-n)u_n + (2n-1)v_n} x^{n-2} + \sum (n-1)\, n\,v_n\, x^{n-2}\log(x)
\end{aligned}$$

$$Ly(x) = \sum \underbrace{[n\,(n-2)\,u_n + u_{n-1} + 2\,n\,v_n]}_{\mathrm{REC}_0} x^{n-1} + \sum \underbrace{[n\,(n-2)\,v_n + v_{n-1}]}_{\mathrm{REC}_1} x^{n-1}\log(x)$$

v_n	0	0	$-\frac{1}{2}$	$\frac{1}{6}$			$\cdots$
u_n	1	1	0				$\cdots$
n	0	1	2	3	4	5	$\cdots$

Example

$$L = x\,D^2 - D + 1$$

$$\begin{aligned}
y(x) &= \sum u_n x^n + \sum v_n x^n \log(x) \\
y'(x) &= \sum n\,u_n x^{n-1} + \sum n\,v_n x^{n-1}\log(x) + \sum v_n x^{n-1} \\
&= \sum (n\,u_n + v_n)\, x^{n-1} + \sum n\,v_n\, x^{n-1}\log(x) \\
y''(x) &= \sum \underbrace{[(n-1)\,(n\,u_n + v_n) + n\,v_n]}_{(n^2-n)u_n + (2n-1)v_n} x^{n-2} + \sum (n-1)\, n\,v_n\, x^{n-2}\log(x)
\end{aligned}$$

$$Ly(x) = \sum \underbrace{[n\,(n-2)\,u_n + u_{n-1} + 2\,n\,v_n]}_{\mathrm{REC}_0} x^{n-1} + \sum \underbrace{[n\,(n-2)\,v_n + v_{n-1}]}_{\mathrm{REC}_1} x^{n-1}\log(x)$$

v_n	0	0	$-\frac{1}{2}$	$\frac{1}{6}$			$\cdots$
u_n	1	1	0	$-\frac{2}{9}$			$\cdots$
n	0	1	2	3	4	5	$\cdots$

Example

$$L = x\,D^2 - D + 1$$

$$\begin{aligned}
y(x) &= \sum u_n x^n + \sum v_n x^n \log(x) \\
y'(x) &= \sum n\,u_n x^{n-1} + \sum n\,v_n x^{n-1}\log(x) + \sum v_n x^{n-1} \\
&= \sum (n\,u_n + v_n)\, x^{n-1} + \sum n\,v_n\, x^{n-1}\log(x) \\
y''(x) &= \sum \underbrace{[(n-1)\,(n\,u_n + v_n) + n\,v_n]}_{(n^2-n)u_n + (2n-1)v_n} x^{n-2} + \sum (n-1)\, n\,v_n\, x^{n-2}\log(x)
\end{aligned}$$

$$Ly(x) = \sum \underbrace{[n\,(n-2)\,u_n + u_{n-1} + 2\,n\,v_n]}_{\mathrm{REC}_0} x^{n-1} + \sum \underbrace{[n\,(n-2)\,v_n + v_{n-1}]}_{\mathrm{REC}_1} x^{n-1}\log(x)$$

v_n	0	0	$-\frac{1}{2}$	$\frac{1}{6}$	$-\frac{1}{48}$		$\cdots$
u_n	1	1	0	$-\frac{2}{9}$	$\frac{25}{576}$		$\cdots$
n	0	1	2	3	4	5	$\cdots$

Example

$$L = x\,D^2 - D + 1$$

$$\begin{aligned}
y(x) &= \sum u_n x^n + \sum v_n x^n \log(x) \\
y'(x) &= \sum n\,u_n x^{n-1} + \sum n\,v_n x^{n-1}\log(x) + \sum v_n x^{n-1} \\
&= \sum (n\,u_n + v_n)\, x^{n-1} + \sum n\,v_n\, x^{n-1}\log(x) \\
y''(x) &= \sum \underbrace{[(n-1)\,(n\,u_n + v_n) + n\,v_n]}_{(n^2-n)u_n + (2n-1)v_n} x^{n-2} + \sum (n-1)\, n\,v_n\, x^{n-2}\log(x)
\end{aligned}$$

$$Ly(x) = \sum \underbrace{[n\,(n-2)\,u_n + u_{n-1} + 2\,n\,v_n]}_{\mathrm{REC}_0} x^{n-1} + \sum \underbrace{[n\,(n-2)\,v_n + v_{n-1}]}_{\mathrm{REC}_1} x^{n-1}\log(x)$$

v_n	0	0	$-\frac{1}{2}$	$\frac{1}{6}$	$-\frac{1}{48}$	$\frac{1}{720}$	$\cdots$
u_n	1	1	0	$-\frac{2}{9}$	$\frac{25}{576}$	$-\frac{157}{43200}$	$\cdots$
n	0	1	2	3	4	5	$\cdots$

Recurrences at regular singular points

[∼ Heffter 1890]

Proposition. Let $\theta = x\,D$. The series

$$\sum_{\nu\in\lambda+\mathbb{Z}} \sum_{k=0}^{\infty} y_{\nu,k}\, x^{\nu} \frac{\log(x)^k}{k!} \qquad (\lambda \in \mathbb{C})$$

is solution to $[\theta : y(x) \mapsto x\,y'(x)]$

$$L(x,\theta)\cdot y = 0$$

if and only if the sequence $(y_{\nu,k})_{\substack{\nu\in\lambda+\mathbb{Z}\\ k\in\mathbb{N}}}$ is solution to

$$L(S_\nu^{-1}, \nu + S_k)\cdot (y_{n,k}) = 0.$$

Here

$$\begin{aligned} S_\nu^{-1}\colon\ & (u_{\nu,k}) \ \mapsto\ (u_{\nu-1,k}) \\ S_k\colon\ & (u_{\nu,k}) \ \mapsto\ (u_{\nu,k+1}). \end{aligned}$$

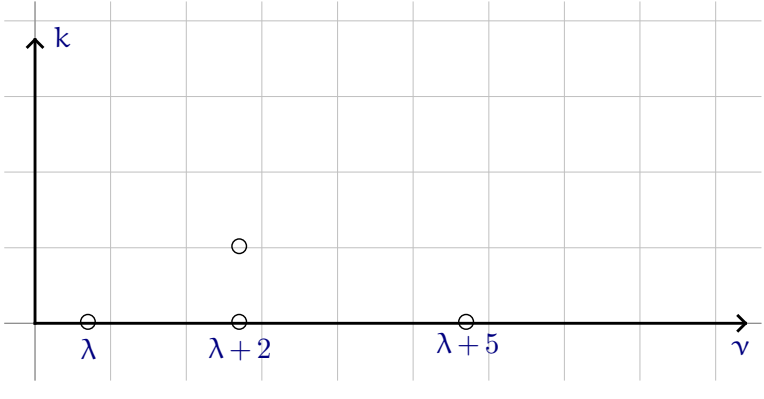
Proof. Same as before (look at the action of x and θ on $x^\nu \log(x)^k/k!$). □

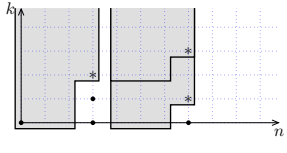
Recurrences and the shape of solutions

$$y(x) = \sum_{\nu,k} y_{\nu,k}\, x^{\nu} \frac{\log(x)^k}{k!}$$

$$q_0(\nu) = (\nu - \lambda)\,(\nu - \lambda - 2)\,(\nu - \lambda - 5)\,(\cdots)$$

no roots in $\lambda + \mathbb{Z}$



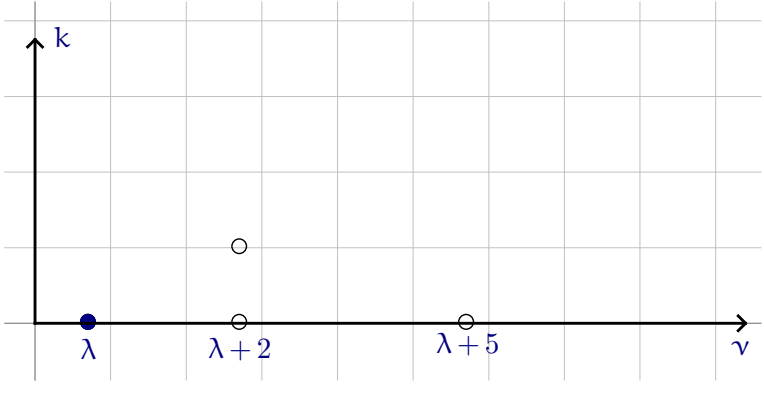


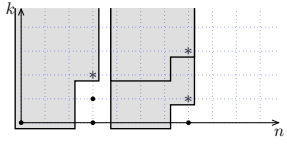
Recurrences and the shape of solutions

$$y(x) = \sum_{\nu,k} y_{\nu,k}\, x^{\nu} \frac{\log(x)^k}{k!}$$

$$q_0(\nu) = (\nu - \lambda)\,(\nu - \lambda - 2)\,(\nu - \lambda - 5)\,(\cdots)$$

no roots in $\lambda + \mathbb{Z}$



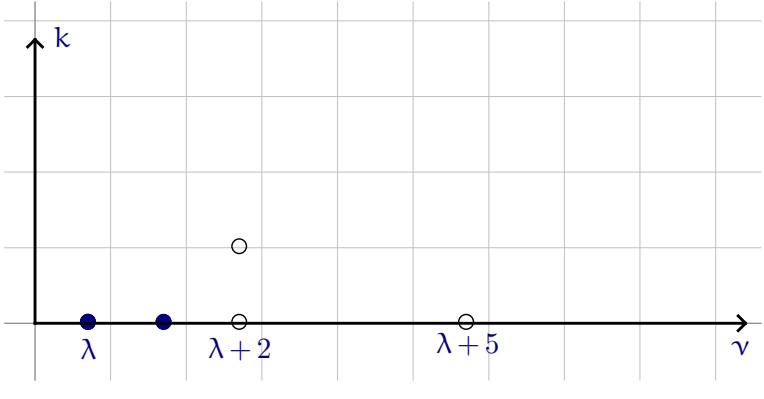


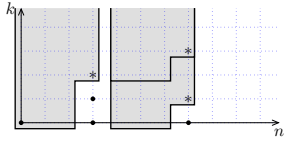
Recurrences and the shape of solutions

$$y(x) = \sum_{\nu,k} y_{\nu,k}\, x^{\nu} \frac{\log(x)^k}{k!}$$

$$q_0(\nu) = (\nu - \lambda)\,(\nu - \lambda - 2)\,(\nu - \lambda - 5)\,(\cdots)$$

no roots in $\lambda + \mathbb{Z}$



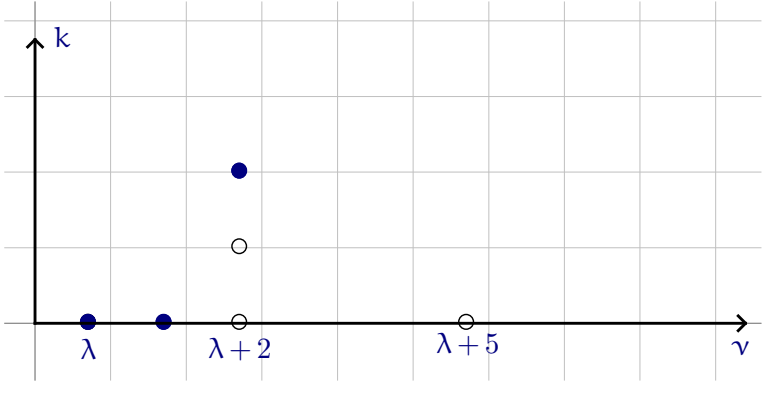


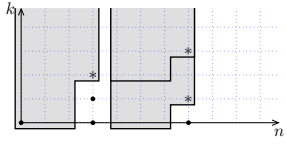
Recurrences and the shape of solutions

$$y(x) = \sum_{\nu,k} y_{\nu,k} x^{\nu} \frac{\log(x)^k}{k!}$$

$$q_0(\nu) = (\nu - \lambda)\,(\nu - \lambda - 2)\,(\nu - \lambda - 5)\,(\cdots)$$

no roots in $\lambda + \mathbb{Z}$



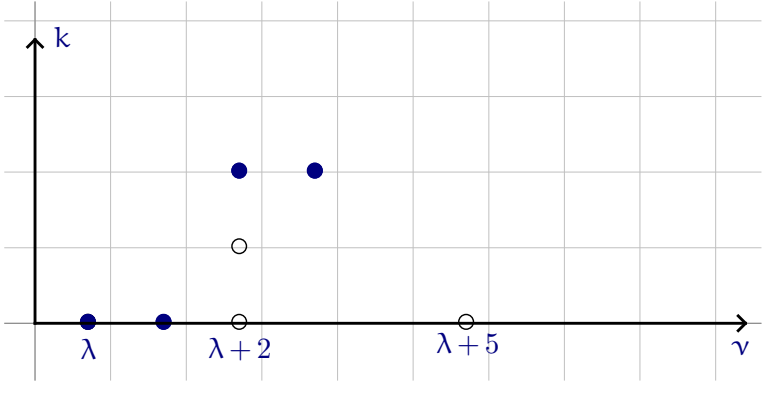


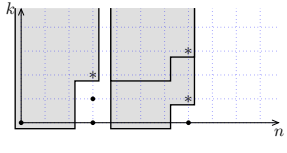
Recurrences and the shape of solutions

$$y(x) = \sum_{\nu,k} y_{\nu,k}\, x^{\nu} \frac{\log(x)^k}{k!}$$

$$q_0(\nu) = (\nu - \lambda)\,(\nu - \lambda - 2)\,(\nu - \lambda - 5)\,(\cdots)$$

no roots in $\lambda + \mathbb{Z}$



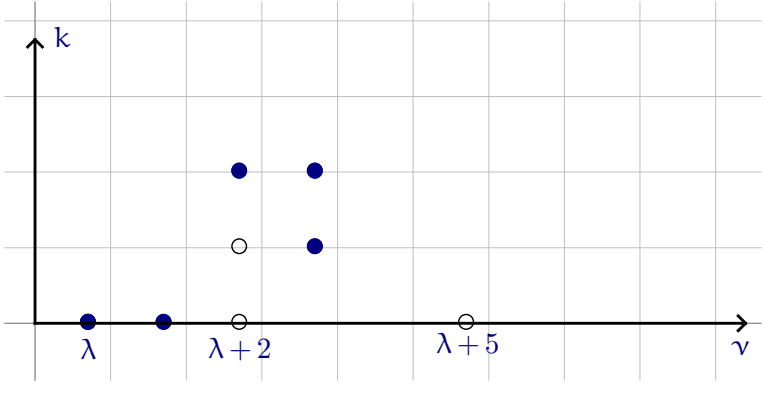


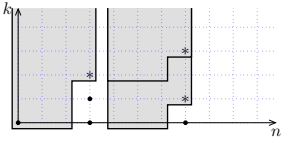
Recurrences and the shape of solutions

$$y(x) = \sum_{\nu,k} y_{\nu,k}\, x^{\nu} \frac{\log(x)^k}{k!}$$

$$q_0(\nu) = (\nu - \lambda)\,(\nu - \lambda - 2)\,(\nu - \lambda - 5)\,(\cdots)$$

no roots in $\lambda + \mathbb{Z}$



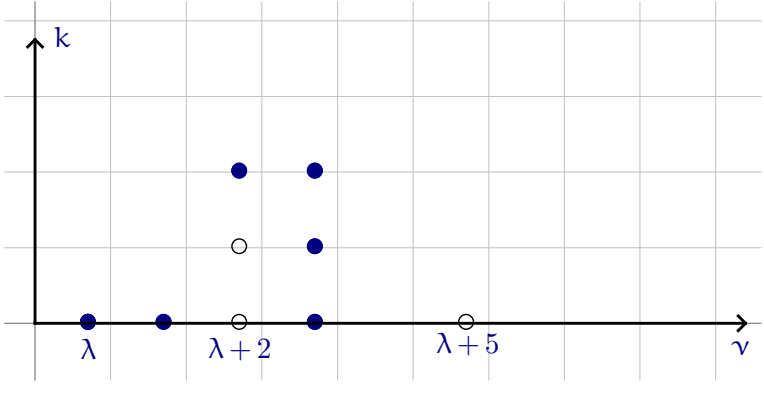


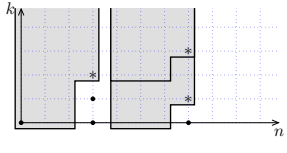
Recurrences and the shape of solutions

$$y(x) = \sum_{\nu,k} y_{\nu,k}\, x^{\nu} \frac{\log(x)^k}{k!}$$

$$q_0(\nu) = (\nu - \lambda)\,(\nu - \lambda - 2)\,(\nu - \lambda - 5)\,(\cdots)$$

no roots in $\lambda + \mathbb{Z}$



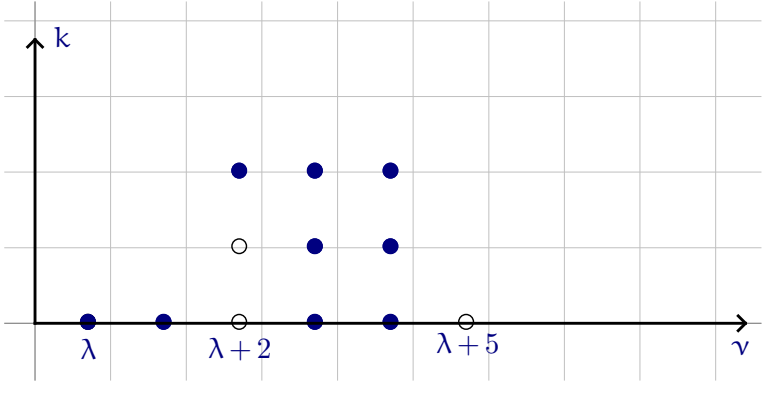


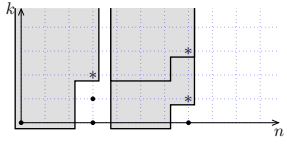
Recurrences and the shape of solutions

$$y(x) = \sum_{\nu,k} y_{\nu,k}\, x^{\nu} \frac{\log(x)^k}{k!}$$

$$q_0(\nu) = (\nu - \lambda)\,(\nu - \lambda - 2)\,(\nu - \lambda - 5)\,(\cdots)$$

no roots in $\lambda + \mathbb{Z}$



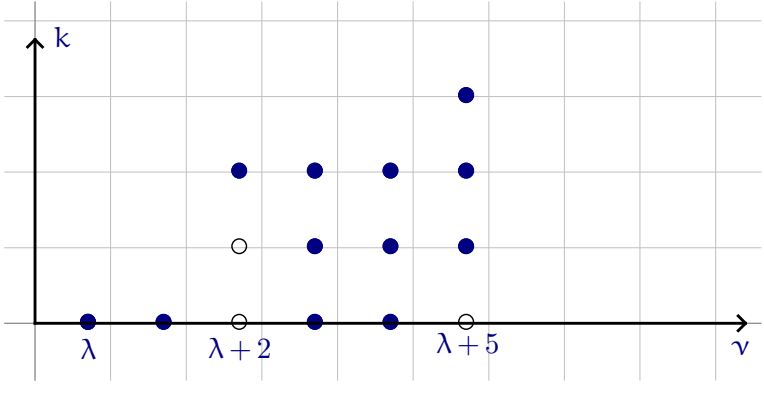


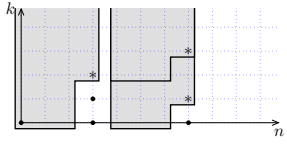
Recurrences and the shape of solutions

$$y(x) = \sum_{\nu,k} y_{\nu,k} x^{\nu} \frac{\log(x)^k}{k!}$$

$$q_0(\nu) = (\nu - \lambda)\,(\nu - \lambda - 2)\,(\nu - \lambda - 5)\,(\cdots)$$

no roots in $\lambda + \mathbb{Z}$



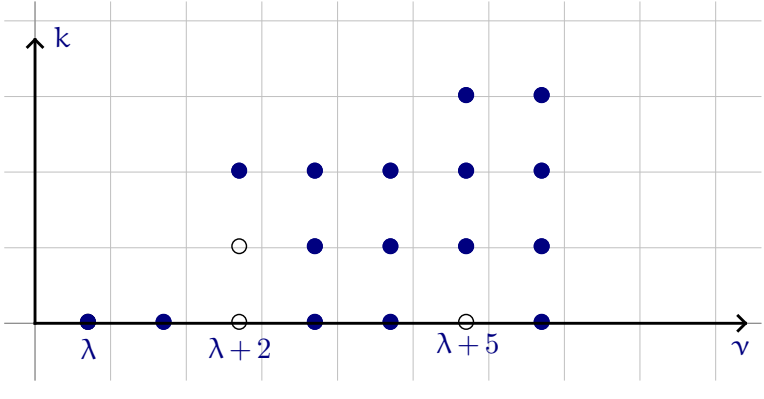


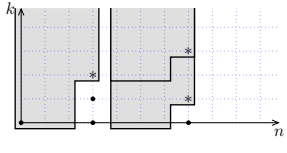
Recurrences and the shape of solutions

$$y(x) = \sum_{\nu,k} y_{\nu,k}\, x^{\nu} \frac{\log(x)^k}{k!}$$

$$q_0(\nu) = (\nu - \lambda)\,(\nu - \lambda - 2)\,(\nu - \lambda - 5)\,(\cdots)$$

no roots in $\lambda + \mathbb{Z}$



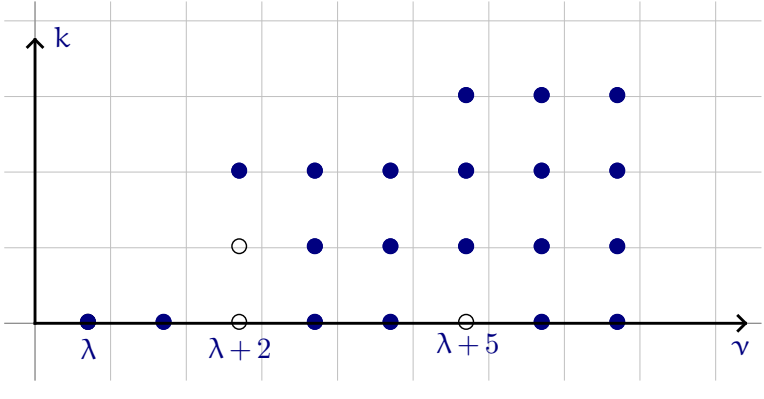


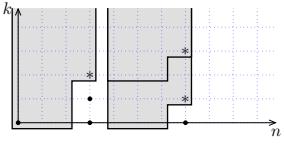
Recurrences and the shape of solutions

$$y(x) = \sum_{\nu,k} y_{\nu,k}\, x^{\nu} \frac{\log(x)^k}{k!}$$

$$q_0(\nu) = (\nu - \lambda)\,(\nu - \lambda - 2)\,(\nu - \lambda - 5)\,(\cdots)$$

no roots in $\lambda + \mathbb{Z}$



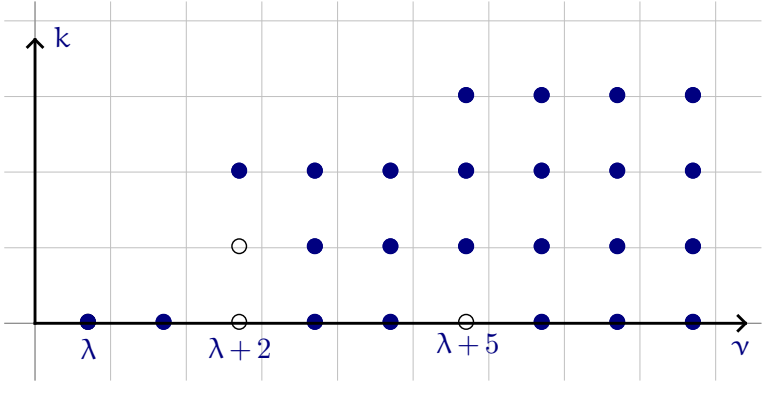


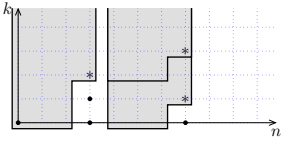
Recurrences and the shape of solutions

$$y(x) = \sum_{\nu,k} y_{\nu,k} x^{\nu} \frac{\log(x)^k}{k!}$$

$$q_0(\nu) = (\nu - \lambda)\,(\nu - \lambda - 2)\,(\nu - \lambda - 5)\,(\cdots)$$

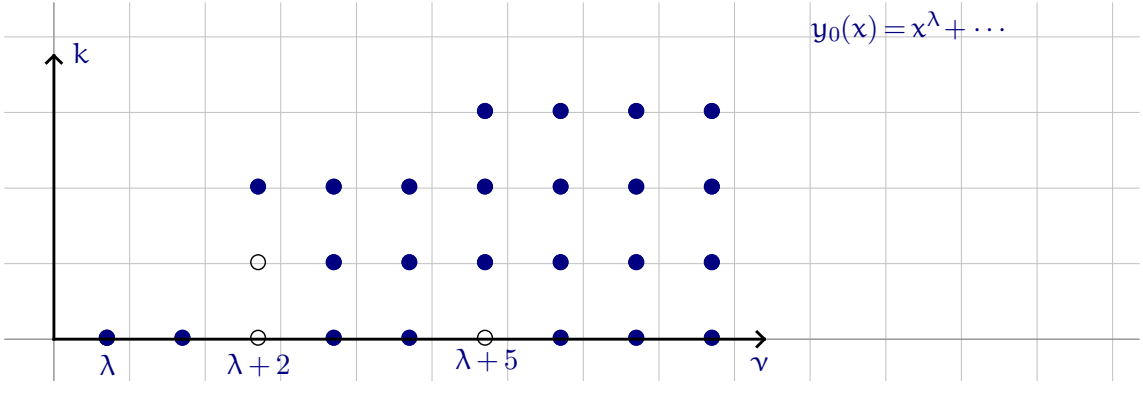
no roots in $\lambda + \mathbb{Z}$





The echelonized basis

$$y(x) = \sum_{\nu,k} y_{\nu,k} x^{\nu} \frac{\log(x)^k}{k!}$$

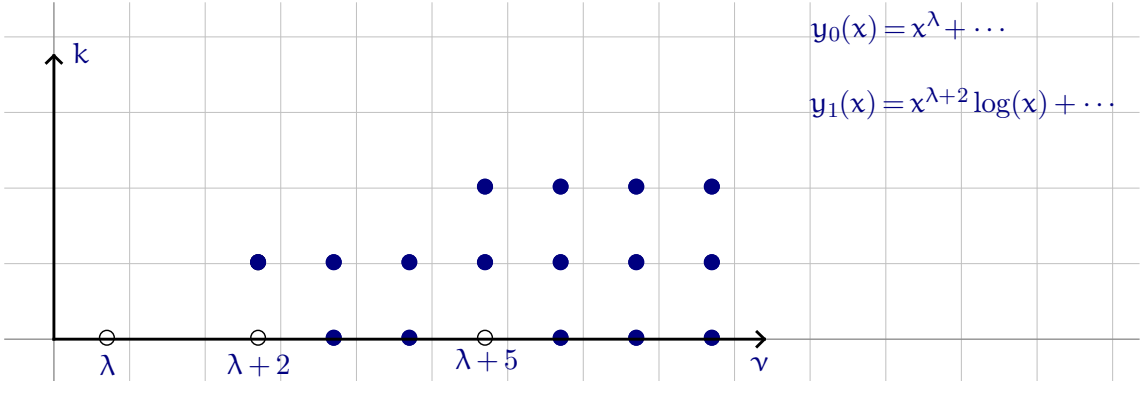


Echelonized basis of solutions:

"initial conditions" $= y_{\nu,k}$ where ν root of q_0 and $k < \mathrm{mult}(\nu)$

The echelonized basis

$$y(x) = \sum_{\nu,k} y_{\nu,k} x^{\nu} \frac{\log(x)^k}{k!}$$

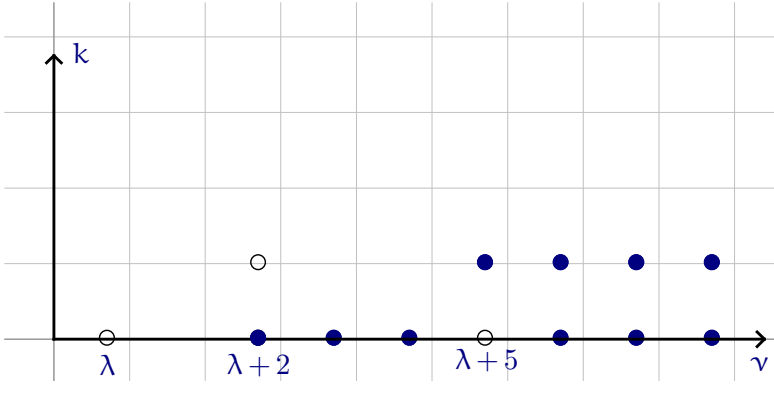


Echelonized basis of solutions:

"initial conditions" $= y_{\nu,k}$ where ν root of q_0 and $k < \mathrm{mult}(\nu)$

The echelonized basis

$$y(x) = \sum_{\nu,k} y_{\nu,k} x^{\nu} \frac{\log(x)^k}{k!}$$



$$y_0(x) = x^{\lambda} + \cdots$$

$$y_1(x) = x^{\lambda+2}\log(x) + \cdots$$

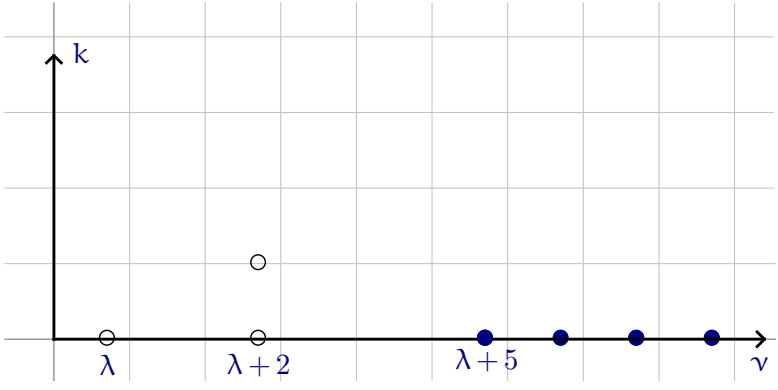
$$y_2(x) = x^{\lambda+2} + \cdots$$

Echelonized basis of solutions:

"initial conditions" $= y_{\nu,k}$ where ν root of q_0 and $k < \mathrm{mult}(\nu)$

The echelonized basis

$$y(x) = \sum_{\nu,k} y_{\nu,k} x^{\nu} \frac{\log(x)^k}{k!}$$



$$y_0(x) = x^{\lambda} + \cdots$$

$$y_1(x) = x^{\lambda+2}\log(x) + \cdots$$

$$y_2(x) = x^{\lambda+2} + \cdots$$

$$y_3(x) = x^{\lambda+5} + \cdots$$

Echelonized basis of solutions:

"initial conditions" $= y_{\nu,k}$ where ν root of q_0 and $k < \mathrm{mult}(\nu)$

Redundancies and the Frobenius basis

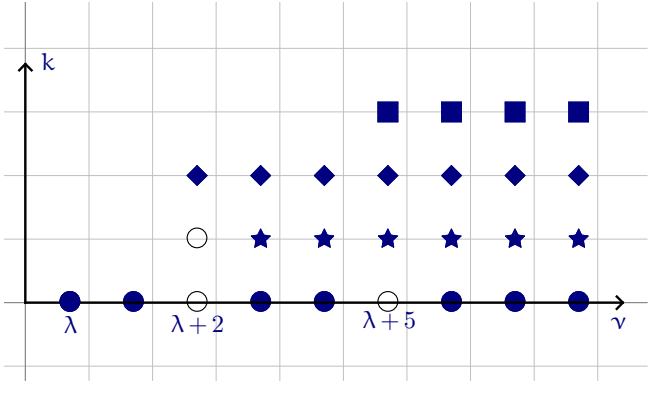
Proposition. If

$$\sum_{\nu} \sum_{k \geqslant 0} y_{\nu,k}\, x^{\nu} \frac{\log(x)^k}{k!}$$

is a solution, then

$$\sum_{\nu} \sum_{k \geqslant 0} y_{\nu,\,k+1}\, x^{\nu} \frac{\log(x)^k}{k!}$$

too.



Frobenius basis:

$$\begin{aligned} y_0(x) &= f_0(x) \\ y_1(x) &= f_0(x)\log(x) + f_1(x) \\ y_2(x) &= f_0(x)\log(x)^2 + 2\,f_1(x)\log(x) + f_2(x) \\ &\vdots \end{aligned}$$

$f_i \in z^{\lambda}\,\mathbb{C}[[x]]$

Consequence: "On average" only one row, like for plain series

Redundancies and the Frobenius basis

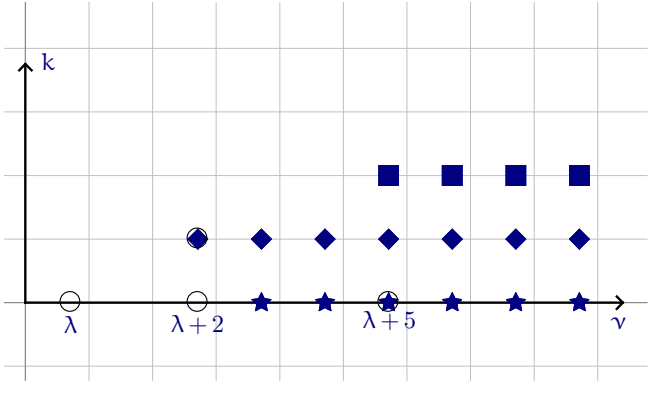
Proposition. If

$$\sum_{\nu} \sum_{k \geqslant 0} y_{\nu,k}\, x^{\nu} \frac{\log(x)^k}{k!}$$

is a solution, then

$$\sum_{\nu} \sum_{k \geqslant 0} y_{\nu,\,k+1}\, x^{\nu} \frac{\log(x)^k}{k!}$$

too.



Frobenius basis:

$$\begin{aligned} y_0(x) &= f_0(x) \\ y_1(x) &= f_0(x)\log(x) + f_1(x) \\ y_2(x) &= f_0(x)\log(x)^2 + 2\,f_1(x)\log(x) + f_2(x) \\ &\vdots \end{aligned}$$

$f_i \in z^{\lambda}\,\mathbb{C}[[x]]$

Consequence: "On average" only one row, like for plain series

Redundancies and the Frobenius basis

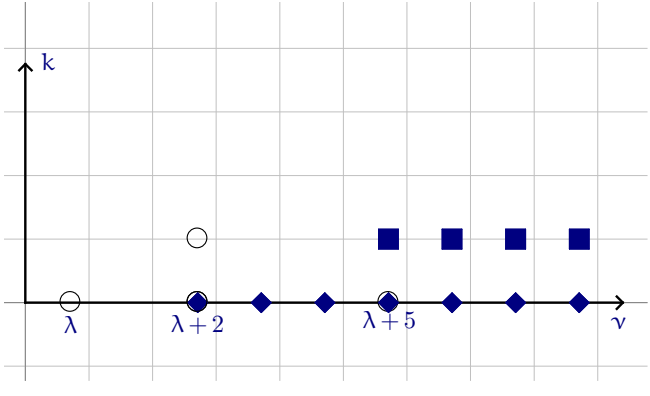
Proposition. If

$$\sum_{\nu} \sum_{k \geqslant 0} y_{\nu,k}\, x^{\nu} \frac{\log(x)^k}{k!}$$

is a solution, then

$$\sum_{\nu} \sum_{k \geqslant 0} y_{\nu,\,k+1}\, x^{\nu} \frac{\log(x)^k}{k!}$$

too.



Frobenius basis:

$$\begin{aligned} y_0(x) &= f_0(x) \\ y_1(x) &= f_0(x)\log(x) + f_1(x) \\ y_2(x) &= f_0(x)\log(x)^2 + 2\,f_1(x)\log(x) + f_2(x) \\ &\vdots \end{aligned}$$

$f_i \in z^{\lambda}\,\mathbb{C}[[x]]$

Consequence: "On average" only one row, like for plain series

Redundancies and the Frobenius basis

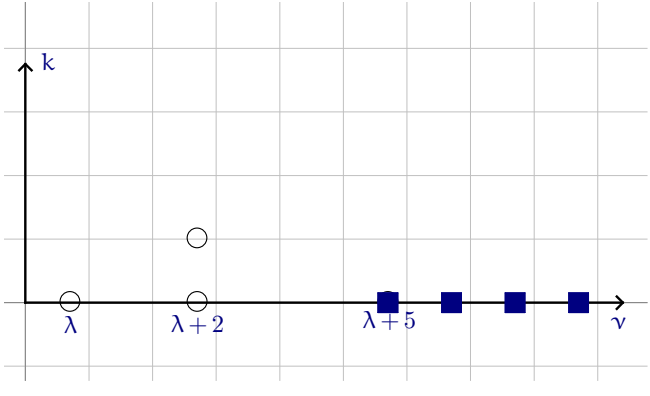
Proposition. If

$$\sum_{\nu} \sum_{k \geqslant 0} y_{\nu,k}\, x^{\nu} \frac{\log(x)^k}{k!}$$

is a solution, then

$$\sum_{\nu} \sum_{k \geqslant 0} y_{\nu,\,k+1}\, x^{\nu} \frac{\log(x)^k}{k!}$$

too.



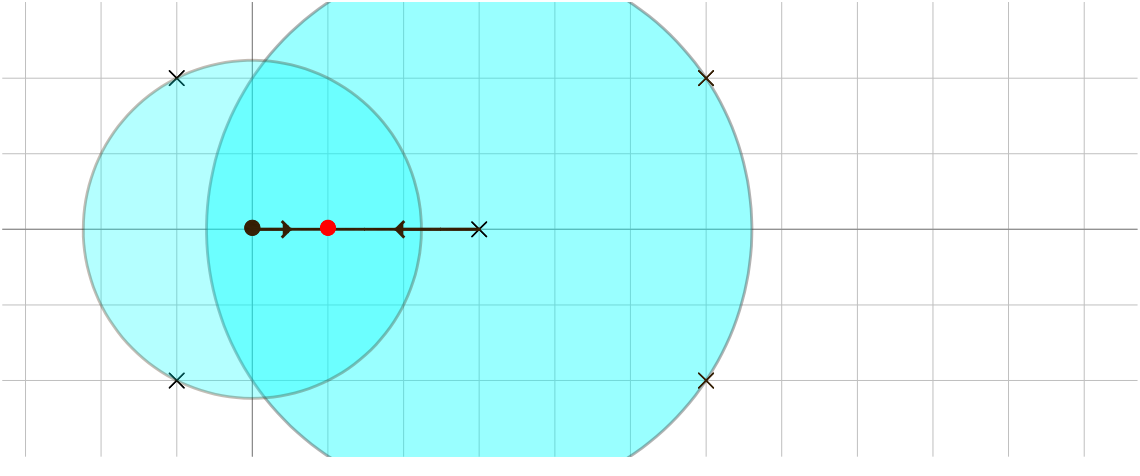
Frobenius basis:

$$\begin{aligned} y_0(x) &= f_0(x) \\ y_1(x) &= f_0(x)\log(x) + f_1(x) \\ y_2(x) &= f_0(x)\log(x)^2 + 2\,f_1(x)\log(x) + f_2(x) \\ &\vdots \end{aligned}$$

$f_i \in z^{\lambda}\,\mathbb{C}[[x]]$

Consequence: "On average" only one row, like for plain series

Connection to a regular singular point



To go “into” a regular singular point,

- evaluate a basis of solutions (+ derivatives) at a point • nearby,
- invert the resulting connection matrix

Solving Linear Differential Equations to High Precision

Part II

Marc Mezzarobba

CNRS, École polytechnique

Journées nationales de calcul formel 2025

Recap

Differential equations with polynomial coefficients

$$a_r(x)\,y^{(r)}(x) + a_{r-1}(x)\,y^{(r-1)}(x) + \cdots + a_0(x)\,y(x) = 0, \qquad a_i \in \mathbb{C}[x]$$

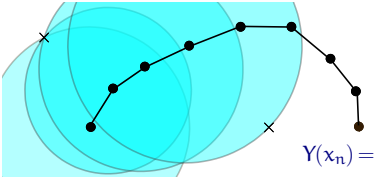
Connection problem

$$y = \sum \mu_i f_i = \sum_i \nu_i f_j$$

$\mu_i f_i$: basis at ξ_0; $\nu_i f_j$: basis at ξ_1

$$Y(\xi_1) = \Delta\, Y(\xi_0)$$

Taylor method, large steps



$$Y(x_n) = \Delta_{n-1} \cdots \Delta_1\, \Delta_0\, Y(x_0)$$

Singular points

- Regular: minor technicalities
- Irregular: genuine new phenomena

Local problem

$$\sum_{n=0}^{\infty} \blacksquare\, (x_{i+1} - x_i)^n$$

$\blacksquare$: local basis, derivatives ($\Rightarrow$ redundancies)

The local problem

$$y(\xi) = \sum_{k=0}^{K-1} \sum_{n=0}^{\infty} y_{n,j}\, \xi^{\lambda+n} \frac{\log^k \xi}{k!} \qquad \text{where} \qquad [\alpha_r(x)\,\theta^r + \cdots + \alpha_1(x)\,\theta + \alpha_0(x)] \cdot y(x) = 0$$

$$\theta = x\,\frac{d}{dx}$$

Good primitive in practice:

- several solutions with the same λ
- possibly with logs (no need for special-casing ordinary points)
- several derivatives
- at several points

We will focus on $\lambda = 0$ and $K = 1$ for simplicity ($\sim$ everything generalizes)

Outline

4 Low precision

“Low” precision?

$\alpha_r(x)\,\theta^r + \cdots + \alpha_1(x)\,\theta + \alpha_0(x)$

r — order
d — degree
h — integer height
p — precision (bits)

ω = complexity exponent of linear algebra

$M_{\mathbb{Z}}(n)$ = cost of integer multiplication

"Low" precision?

$\alpha_r(x)\,\theta^r + \cdots + \alpha_1(x)\,\theta + \alpha_0(x)$

- Low compared to the size of the α_i

  say $p \lesssim h\left(\frac{d}{r}\right)^{\omega}$.

  This can mean $p \approx 10000$.

r — order
d — degree
h — integer height
p — precision (bits)

- I.e.: Regime where algorithms of cost $O(p\,M_{\mathbb{Z}}(p))$ make sense.

ω = complexity exponent of linear algebra

$M_{\mathbb{Z}}(n)$ = cost of integer multiplication

"Low" precision?

$\alpha_r(x)\,\theta^r + \cdots + \alpha_1(x)\,\theta + \alpha_0(x)$

- Low compared to the size of the α_i

  say $p \lesssim h\left(\frac{d}{r}\right)^{\omega}$.

  This can mean $p \approx 10000$.

r — order
d — degree
h — integer height
p — precision (bits)

- I.e.: Regime where algorithms of cost $O(p\,M_{\mathbb{Z}}(p))$ make sense.

computing the sum $\approx$ computing the series coefficients

complexity model: operations in $\mathbb{C}$

(No chance of reaching $o(p^2)$ while computing all coefficients!)

ω = complexity exponent of linear algebra

$M_{\mathbb{Z}}(n)$ = cost of integer multiplication

"Low" precision?

$\alpha_r(x)\,\theta^r + \cdots + \alpha_1(x)\,\theta + \alpha_0(x)$

- Low compared to the size of the α_i

  say $p \lesssim h\left(\frac{d}{r}\right)^{\omega}$.

  This can mean $p \approx 10000$.

r — order
d — degree
h — integer height
p — precision (bits)

- I.e.: Regime where algorithms of cost $O(p\,M_{\mathbb{Z}}(p))$ make sense.

computing the sum $\approx$ computing the series coefficients

complexity model: operations in $\mathbb{C}$

(No chance of reaching $o(p^2)$ while computing all coefficients!)

- True low precision ($p \lesssim 100$): techniques closer to numerical analysis
  + low-level tricks

ω = complexity exponent of linear algebra

$M_{\mathbb{Z}}(n)$ = cost of integer multiplication

Direct recurrence unrolling

$$q_0(n)\,y_n + \cdots + q_d(n)\,y_{n+d} = 0$$

r — order (diff) degree (rec)
d — degree (diff) order (rec)

Algorithm.

For $n = r, r+1, \ldots, N-1$

$\beta_0 \leftarrow q_0(n)$; ... ; $\beta_d \leftarrow q_d(n)$

$y_n := -\beta_0^{-1}\,(\beta_1\,y_{n-1} + \cdots + \beta_d\,y_{n-d})$

Cost for one solution:

Direct recurrence unrolling

$$q_0(n)\,y_n + \cdots + q_d(n)\,y_{n+d} = 0$$

r — order (diff) degree (rec)
d — degree (diff) order (rec)

Algorithm.

For $n = r, r+1, \ldots, N-1$

$\beta_0 \leftarrow q_0(n)$; ... ; $\beta_d \leftarrow q_d(n)$ $\qquad O(d\,r)$

$y_n := -\beta_0^{-1}\,(\beta_1\,y_{n-1} + \cdots + \beta_d\,y_{n-d})$

Cost for one solution:

Direct recurrence unrolling

$$q_0(n)\,y_n + \cdots + q_d(n)\,y_{n+d} = 0$$

r — order (diff) degree (rec)
d — degree (diff) order (rec)

Algorithm.

For $n = r, r+1, \ldots, N-1$

$\beta_0 \leftarrow q_0(n)$; ... ; $\beta_d \leftarrow q_d(n)$ $\quad O(dr)$

$y_n := -\beta_0^{-1}(\beta_1 y_{n-1} + \cdots + \beta_d y_{n-d})$ $\quad O(d)$

Cost for one solution:

Direct recurrence unrolling

$$q_0(n)\,y_n + \cdots + q_d(n)\,y_{n+d} = 0$$

r — order (diff) degree (rec)
d — degree (diff) order (rec)

Algorithm.

For $n = r, r+1, \ldots, N-1$

$\beta_0 \leftarrow q_0(n)$; ... ; $\beta_d \leftarrow q_d(n)$ $\qquad O(d\,r)$

$y_n := -\beta_0^{-1}(\beta_1 y_{n-1} + \cdots + \beta_d y_{n-d})$ $\qquad O(d)$

Cost for one solution: $O(d\,r\,N)$ ops

Direct recurrence unrolling

$$q_0(n)\,y_n + \cdots + q_d(n)\,y_{n+d} = 0$$

r — order (diff) degree (rec)
d — degree (diff) order (rec)

Algorithm.

For $n = r, r+1, \ldots, N-1$

$\beta_0 \leftarrow q_0(n)$; ... ; $\beta_d \leftarrow q_d(n)$ $\quad O(dr)$

$y_n := -\beta_0^{-1}(\beta_1 y_{n-1} + \cdots + \beta_d y_{n-d})$ $\quad O(d)$

Cost for one solution: $O(drN)$ ops

r solutions: $O(drN)$ ops

Direct recurrence unrolling

$$q_0(n)\,y_n + \cdots + q_d(n)\,y_{n+d} = 0$$

r — order (diff) degree (rec)
d — degree (diff) order (rec)

Algorithm.

For $n = r, r+1, \ldots, N-1$

$\beta_0 \leftarrow q_0(n)\ ;\ \ldots\ ;\ \beta_d \leftarrow q_d(n)$ $\quad O(dr)$

$y_n := -\beta_0^{-1}\,(\beta_1\,y_{n-1} + \cdots + \beta_d\,y_{n-d})$ $\quad O(d)$

	naively	fast extrapolation
Cost for one solution:	$O(drN)$ ops	$O(d\,\lambda(r)\,N)$ ops
r solutions:	$O(drN)$ ops	

$$\lambda(r) = \frac{M(r)}{r}$$

[Shoup, 1991; Gerhard, 2000; Bostan, Schost, 2005; Bostan, Gaudry, Schost, 2007]

$M(n)$ = cost of polynomial multiplication

Direct recurrence unrolling

$$q_0(n)\,y_n + \cdots + q_d(n)\,y_{n+d} = 0$$

r — order (diff) degree (rec)
d — degree (diff) order (rec)

Algorithm.

For $n = r, r+1, \ldots, N-1$

$\beta_0 \leftarrow q_0(n)$; ... ; $\beta_d \leftarrow q_d(n)$ $\quad O(dr)$

$y_n := -\beta_0^{-1}(\beta_1 y_{n-1} + \cdots + \beta_d y_{n-d})$ $\quad O(d)$

		naively	fast extrapolation
Cost for	one solution:	$O(drN)$ ops	$O(d\lambda(r)N)$ ops
	r solutions:	$O(drN)$ ops	

$$\lambda(r) = \frac{M(r)}{r}$$

[Shoup, 1991; Gerhard, 2000; Bostan, Schost, 2005; Bostan, Gaudry, Schost, 2007]

fund. mat., ν pts: idem $+ O(\nu r^2 N)$ ops

$M(n)$ = cost of polynomial multiplication

Complexity results for power series coefficients

polynomial coefficients of high degree $\approx$ power series coefficients

r — order (diff)
d — degree (diff)
N — #terms

	1 solution	r solutions	
recurrence	$O(r\,N^2)$	$O(r\,N^2)$	
D&C	$O(r\,M(N)\log N)$	$O(r^2\,M(N)\log N)$	[BCOS³, 2007]
relaxed	$O(r\,N\,(\log N)^{1+o(1)})$	$O(r^{\omega-1}\,N\log N + r\,M(N)\log^2 N)$	[vdH, 2002, 2007, 2016]
Newton		$O(r^r\,M(N))$	[Brent, Kung, 1978; vdH, 2002]
		$O(r^\omega\,M(N))$	[BCOS³, 2007]
		$O(r^\omega\,N + r^2\,M(N))$ (*)	
block Newton	$O(r\,M(N))$ (*)	$O(r^2\,M(N))$ (*)	[vdH, 2010]

(*) using $O(M(N))$ evaluation & interpolation

standard assumption: $M(n)/n \nearrow$

BCOS³ = Bostan, Chyzak, Ollivier, Salvy, Schost, Sedoglavic; vdH = van der Hoeven

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} + y_m x^m + \cdots + y_{N-1} x^{N-1} + \cdots \big) = 0$:

- Solve $L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} \,\big) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1} x^{N-1} + O(x^N) = L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} \,\big)$
- Solve $L\big(\, y_m x^m + y_{m+1} x^{m+1} + \cdots + y_{N-1} x^{N-1} \,\big) = -\big(\, h_m x^m + \cdots + h_{N-1} x^{N-1} \,\big) + O(x^N)$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\,x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1} + y_m x^m + \cdots + y_{N-1}x^{N-1} + \cdots) = 0$:

- Solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1}x^{N-1} + O(x^N) = L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,)$
- Solve $L(\,y_m x^m + y_{m+1}x^{m+1} + \cdots + y_{N-1}x^{N-1}\,) = -(\,h_m x^m + \cdots + h_{N-1}x^{N-1}\,) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\,x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} + y_m x^m + \cdots + y_{N-1} x^{N-1} + \cdots \big) = 0$:

- Solve $L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} \,\big) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1} x^{N-1} + O(x^N) = L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} \,\big)$
- Solve $L\big(\, y_m x^m + y_{m+1} x^{m+1} + \cdots + y_{N-1} x^{N-1} \,\big) = -\big(\, h_m x^m + \cdots + h_{N-1} x^{N-1} \,\big) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + O(x^6)$$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\,x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} + y_m x^m + \cdots + y_{N-1} x^{N-1} + \cdots \big) = 0$:

- Solve $L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} \,\big) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1} x^{N-1} + O(x^N) = L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} \,\big)$
- Solve $L\big(\, y_m x^m + y_{m+1} x^{m+1} + \cdots + y_{N-1} x^{N-1} \,\big) = -\big(\, h_m x^m + \cdots + h_{N-1} x^{N-1} \,\big) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + O(x^6)$$

$$L(y_4 x^4) = g_4 x^4 + O(x^5)$$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\,x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$$

Idea. To solve $L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} + y_m x^m + \cdots + y_{N-1} x^{N-1} + \cdots \big) = 0$:

- Solve $L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} \,\big) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1} x^{N-1} + O(x^N) = L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} \,\big)$
- Solve $L\big(\, y_m x^m + y_{m+1} x^{m+1} + \cdots + y_{N-1} x^{N-1} \,\big) = -\big(\, h_m x^m + \cdots + h_{N-1} x^{N-1} \,\big) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + O(x^6)$$

$$L(y_4 x^4) = g_4 x^4 + O(x^5)$$

$$q_0(4)\, y_4 = g_4$$

$$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$$

Lemma. $L(x^n) = q_0(n)\, x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$$

Idea. To solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1} + y_m x^m + \cdots + y_{N-1}x^{N-1} + \cdots) = 0$:

- Solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1}x^{N-1} + O(x^N) = L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,)$
- Solve $L(\,y_m x^m + y_{m+1}x^{m+1} + \cdots + y_{N-1}x^{N-1}\,) = -(\,h_m x^m + \cdots + h_{N-1}x^{N-1}\,) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + O(x^6)$$

$$L(y_4 x^4) = g_4 x^4 + O(x^5)$$

$$q_0(4)\,y_4 = g_4$$

$$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$$

Lemma. $L(x^n) = q_0(n)\,x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1} + y_m x^m + \cdots + y_{N-1}x^{N-1} + \cdots) = 0$:

- Solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1}x^{N-1} + O(x^N) = L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,)$
- Solve $L(\,y_m x^m + y_{m+1}x^{m+1} + \cdots + y_{N-1}x^{N-1}\,) = -(\,h_m x^m + \cdots + h_{N-1}x^{N-1}\,) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + O(x^6)$$

$$L(y_4 x^4) = g_4 x^4 + h_5 x^5 + \cdots$$

$$q_0(4)\, y_4 = g_4$$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\,x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1} + y_m x^m + \cdots + y_{N-1}x^{N-1} + \cdots) = 0$:

- Solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1}x^{N-1} + O(x^N) = L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,)$
- Solve $L(\,y_m x^m + y_{m+1}x^{m+1} + \cdots + y_{N-1}x^{N-1}\,) = -(\,h_m x^m + \cdots + h_{N-1}x^{N-1}\,) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + O(x^6)$$

$L(y_4 x^4) = g_4 x^4 + h_5 x^5 + \cdots$ $\quad L(y_5 x^5) = (g_5 - h_5)x^5 + O(x^6)$

$q_0(4)\,y_4 = g_4$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\,x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L\big(\ y_0 + y_1 x + \cdots + y_{m-1}x^{m-1} + y_m x^m + \cdots + y_{N-1}x^{N-1} + \cdots\ \big) = 0$:

- Solve $L\big(\ y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\ \big) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1}x^{N-1} + O(x^N) = L\big(\ y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\ \big)$
- Solve $L\big(\ y_m x^m + y_{m+1}x^{m+1} + \cdots + y_{N-1}x^{N-1}\ \big) = -\big(\ h_m x^m + \cdots + h_{N-1}x^{N-1}\ \big) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + O(x^6)$$

$L(y_4 x^4) = g_4 x^4 + h_5 x^5 + \cdots$ $\quad$ $L(y_5 x^5) = (g_5 - h_5)x^5 + O(x^6)$

$q_0(4)\,y_4 = g_4$ $\qquad$ $q_0(5)\,y_5 = g_5 - h_5$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\,x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1} + y_m x^m + \cdots + y_{N-1}x^{N-1} + \cdots) = 0$:

- Solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1}x^{N-1} + O(x^N) = L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,)$
- Solve $L(\,y_m x^m + y_{m+1}x^{m+1} + \cdots + y_{N-1}x^{N-1}\,) = -(\,h_m x^m + \cdots + h_{N-1}x^{N-1}\,) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + O(x^6)$$

$L(y_4 x^4) = g_4 x^4 + h_5 x^5 + \cdots$ $\quad L(y_5 x^5) = (g_5 - h_5)x^5 + O(x^6)$

$q_0(4)\,y_4 = g_4$ $\qquad$ $q_0(5)\,y_5 = g_5 - h_5$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\,x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1} + y_m x^m + \cdots + y_{N-1}x^{N-1} + \cdots) = 0$:

- Solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1}x^{N-1} + O(x^N) = L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,)$
- Solve $L(\,y_m x^m + y_{m+1}x^{m+1} + \cdots + y_{N-1}x^{N-1}\,) = -(\,h_m x^m + \cdots + h_{N-1}x^{N-1}\,) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$$

$L(y_4 x^4) = g_4 x^4 + h_5 x^5 + \cdots$ $\quad$ $L(y_5 x^5) = (g_5 - h_5)x^5 + O(x^6)$

$q_0(4)\,y_4 = g_4$ $\quad$ $q_0(5)\,y_5 = g_5 - h_5$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\,x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1} + y_m x^m + \cdots + y_{N-1}x^{N-1} + \cdots) = 0$:

- Solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1}x^{N-1} + O(x^N) = L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,)$
- Solve $L(\,y_m x^m + y_{m+1}x^{m+1} + \cdots + y_{N-1}x^{N-1}\,) = -(\,h_m x^m + \cdots + h_{N-1}x^{N-1}\,) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_6 x^6 + y_7 x^7) = \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_4 x^4) = g_4 x^4 + h_5 x^5 + \cdots$ $\quad L(y_5 x^5) = (g_5 - h_5)x^5 + O(x^6)$

$q_0(4)\,y_4 = g_4$ $\qquad q_0(5)\,y_5 = g_5 - h_5$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\,x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1} + y_m x^m + \cdots + y_{N-1}x^{N-1} + \cdots) = 0$:

- Solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1}x^{N-1} + O(x^N) = L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,)$
- Solve $L(\,y_m x^m + y_{m+1}x^{m+1} + \cdots + y_{N-1}x^{N-1}\,) = -(\,h_m x^m + \cdots + h_{N-1}x^{N-1}\,) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_6 x^6 + y_7 x^7) = \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_4 x^4) = g_4 x^4 + h_5 x^5 + \cdots$ $\quad L(y_5 x^5) = (g_5 - h_5)x^5 + O(x^6)$ $\quad L(y_6 x^6) = \blacksquare\, x^6 + O(x^6)$

$q_0(4)\, y_4 = g_4$ $\qquad q_0(5)\, y_5 = g_5 - h_5$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\,x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L(\ y_0 + y_1 x + \cdots + y_{m-1}x^{m-1} + y_m x^m + \cdots + y_{N-1}x^{N-1} + \cdots\ ) = 0$:

- Solve $L(\ y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\ ) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1}x^{N-1} + O(x^N) = L(\ y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\ )$
- Solve $L(\ y_m x^m + y_{m+1}x^{m+1} + \cdots + y_{N-1}x^{N-1}\ ) = -(\ h_m x^m + \cdots + h_{N-1}x^{N-1}\ ) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_6 x^6 + y_7 x^7) = \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_4 x^4) = g_4 x^4 + h_5 x^5 + \cdots$ $\quad$ $L(y_5 x^5) = (g_5 - h_5)x^5 + O(x^6)$ $\quad$ $L(y_6 x^6) = \blacksquare\, x^6 + O(x^6)$

$q_0(4)\, y_4 = g_4$ $\quad$ $q_0(5)\, y_5 = g_5 - h_5$ $\quad$ $q_0(6)\, y_6 = \blacksquare$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\,x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1} + y_m x^m + \cdots + y_{N-1}x^{N-1} + \cdots) = 0$:

- Solve $L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1}x^{N-1} + O(x^N) = L(\,y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\,)$
- Solve $L(\,y_m x^m + y_{m+1}x^{m+1} + \cdots + y_{N-1}x^{N-1}\,) = -(\,h_m x^m + \cdots + h_{N-1}x^{N-1}\,) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_6 x^6 + y_7 x^7) = \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_4 x^4) = g_4 x^4 + h_5 x^5 + \cdots$ $\quad q_0(4)\, y_4 = g_4$

$L(y_5 x^5) = (g_5 - h_5)x^5 + O(x^6)$ $\quad q_0(5)\, y_5 = g_5 - h_5$

$L(y_6 x^6) = \blacksquare\, x^6 + O(x^6)$ $\quad q_0(6)\, y_6 = \blacksquare$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\,x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L\big(\ y_0 + y_1 x + \cdots + y_{m-1}x^{m-1} + y_m x^m + \cdots + y_{N-1}x^{N-1} + \cdots\ \big) = 0$:

- Solve $L\big(\ y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\ \big) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1}x^{N-1} + O(x^N) = L\big(\ y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\ \big)$
- Solve $L\big(\ y_m x^m + y_{m+1}x^{m+1} + \cdots + y_{N-1}x^{N-1}\ \big) = -\big(\ h_m x^m + \cdots + h_{N-1}x^{N-1}\ \big) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$ $\qquad$ $L(y_6 x^6 + y_7 x^7) = \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_4 x^4) = g_4 x^4 + h_5 x^5 + \cdots$ $\quad$ $L(y_5 x^5) = (g_5 - h_5)x^5 + O(x^6)$ $\quad$ $L(y_6 x^6) = \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$q_0(4)\, y_4 = g_4$ $\qquad$ $q_0(5)\, y_5 = g_5 - h_5$ $\qquad$ $q_0(6)\, y_6 = \blacksquare$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\, x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} + y_m x^m + \cdots + y_{N-1} x^{N-1} + \cdots \big) = 0$:

- Solve $L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} \,\big) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1} x^{N-1} + O(x^N) = L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} \,\big)$
- Solve $L\big(\, y_m x^m + y_{m+1} x^{m+1} + \cdots + y_{N-1} x^{N-1} \,\big) = -\big(\, h_m x^m + \cdots + h_{N-1} x^{N-1} \,\big) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$ $\qquad$ $L(y_6 x^6 + y_7 x^7) = \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_4 x^4) = g_4 x^4 + h_5 x^5 + \cdots$ $\quad$ $L(y_5 x^5) = (g_5 - h_5) x^5 + O(x^6)$ $\quad$ $L(y_6 x^6) = \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$ $\quad$ $L(y_7 x^7) = \blacksquare\, x^7 + O(x^8)$

$q_0(4)\, y_4 = g_4$ $\qquad$ $q_0(5)\, y_5 = g_5 - h_5$ $\qquad$ $q_0(6)\, y_6 = \blacksquare$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\, x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L(\ y_0 + y_1 x + \cdots + y_{m-1}x^{m-1} + y_m x^m + \cdots + y_{N-1}x^{N-1} + \cdots\ ) = 0$:

- Solve $L(\ y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\ ) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1}x^{N-1} + O(x^N) = L(\ y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\ )$
- Solve $L(\ y_m x^m + y_{m+1}x^{m+1} + \cdots + y_{N-1}x^{N-1}\ ) = -(\ h_m x^m + \cdots + h_{N-1}x^{N-1}\ ) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_6 x^6 + y_7 x^7) = \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_4 x^4) = g_4 x^4 + h_5 x^5 + \cdots$ $\quad$ $L(y_5 x^5) = (g_5 - h_5)x^5 + O(x^6)$ $\quad$ $L(y_6 x^6) = \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$ $\quad$ $L(y_7 x^7) = \blacksquare\, x^7 + O(x^8)$

$q_0(4)\, y_4 = g_4$ $\quad$ $q_0(5)\, y_5 = g_5 - h_5$ $\quad$ $q_0(6)\, y_6 = \blacksquare$ $\quad$ $q_0(7)\, y_7 = \blacksquare$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\, x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L(\ y_0 + y_1 x + \cdots + y_{m-1}x^{m-1} + y_m x^m + \cdots + y_{N-1}x^{N-1} + \cdots) = 0$:

- Solve $L(\ y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\ ) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1}x^{N-1} + O(x^N) = L(\ y_0 + y_1 x + \cdots + y_{m-1}x^{m-1}\ )$
- Solve $L(\ y_m x^m + y_{m+1}x^{m+1} + \cdots + y_{N-1}x^{N-1}\ ) = -(\ h_m x^m + \cdots + h_{N-1}x^{N-1}\ ) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$$

$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_6 x^6 + y_7 x^7) = \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_4 x^4) = g_4 x^4 + h_5 x^5 + \cdots$ $\quad$ $L(y_5 x^5) = (g_5 - h_5)x^5 + O(x^6)$ $\quad$ $L(y_6 x^6) = \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$ $\quad$ $L(y_7 x^7) = \blacksquare\, x^7 + O(x^8)$

$q_0(4)\, y_4 = g_4$ $\quad$ $q_0(5)\, y_5 = g_5 - h_5$ $\quad$ $q_0(6)\, y_6 = \blacksquare$ $\quad$ $q_0(7)\, y_7 = \blacksquare$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\,x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

A divide-and-conquer algorithm

$L = \alpha_r(x)\,\theta^r + \cdots + \alpha_0(x)$

Idea. To solve $L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} + y_m x^m + \cdots + y_{N-1} x^{N-1} + \cdots \big) = 0$:

- Solve $L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} \big) = O(x^m)$
- Compute the residual $h_m x^m + \cdots + h_{N-1} x^{N-1} + O(x^N) = L\big(\, y_0 + y_1 x + \cdots + y_{m-1} x^{m-1} \big)$
- Solve $L\big(\, y_m x^m + y_{m+1} x^{m+1} + \cdots + y_{N-1} x^{N-1} \big) = -\big(h_m x^m + \cdots + h_{N-1} x^{N-1} \big) + O(x^N)$

$$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + g_8 x^8 + g_9 x^9 + \cdots$$

$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_6 x^6 + y_7 x^7) = \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$

$L(y_4 x^4) = g_4 x^4 + h_5 x^5 + \cdots$ $\quad$ $L(y_5 x^5) = (g_5 - h_5) x^5 + O(x^6)$ $\quad$ $L(y_6 x^6) = \blacksquare\, x^6 + \blacksquare\, x^7 + O(x^8)$ $\quad$ $L(y_7 x^7) = \blacksquare\, x^7 + O(x^8)$

$q_0(4)\, y_4 = g_4$ $\quad$ $q_0(5)\, y_5 = g_5 - h_5$ $\quad$ $q_0(6)\, y_6 = \blacksquare$ $\quad$ $q_0(7)\, y_7 = \blacksquare$

$q_0(\theta) = \alpha_r(0)\,\theta^r + \cdots + \alpha_1(0)\,\theta + \alpha_0(0)$

Lemma. $L(x^n) = q_0(n)\, x^n + \blacksquare\, x^{n+1} + \cdots$

For $N \geqslant r$ (ordinary point), $q_0(N) \neq 0$, hence $L(\tilde{y}) = O(x^N) \Leftrightarrow$ first N terms of $\tilde{y}$ correct.

Complexity

$L(y) = 0$ of order r, N terms

Algorithm. *Input*: L, $g = -L \cdot y_{0:n} + O(x^{n+2\ell})$, n, m *Output*: $y_{n:n+2\ell}$

1. If $m = 1$, return $q_0(n)^{-1} y_n x^n$ — $O(r)$ ($O(M(r)/r)$ amortized)
2. Write $m = 2\ell$ and $g_{\text{low}} = g_{n:n+\ell}$, $g_{\text{high}} = g_{n+\ell:n+2\ell}$
3. Solve $L \cdot y_{\text{low}} = g_{\text{low}} + O(x^{n+\ell})$ for $y_{\text{low}} \in \mathbb{C}[x]_{n:n+\ell}$
4. Compute $h_{\text{high}} := \big(L(y_{\text{low}})\big)_{n+\ell:n+2\ell}$ — $O(r\,M(\ell))$
5. Solve $L \cdot y_{\text{high}} = g_{\text{high}} - h_{\text{high}} + O(x^{n+2\ell})$ for $y_{\text{high}} \in \mathbb{C}[x]_{n+\ell:n+2\ell}$
6. Return $y_{\text{low}} + y_{\text{high}}$

$L(y_4 x^4 + y_5 x^5 + y_6 x^6 + y_7 x^7) = g_4 x^4 + g_5 x^5 + g_6 x^6 + g_7 x^7 + O(x^8)$

$L(y_4 x^4 + y_5 x^5) = g_4 x^4 + g_5 x^5 + \blacksquare x^6 + \blacksquare x^7 + O(x^8)$ $\quad$ $L(y_6 x^6 + y_7 x^7) = \blacksquare x^6 + \blacksquare x^7 + O(x^8)$

$L(y_4 x^4) = g_4 x^4 + h_5 x^5 + O(x^6)$ $\quad$ $L(y_5 x^5) = \blacksquare x^5 + O(x^6)$ $\quad$ $L(y_6 x^6) = \blacksquare x^6 + \blacksquare x^7 + O(x^8)$ $\quad$ $L(y_7 x^7) = \blacksquare x^7 + O(x^8)$

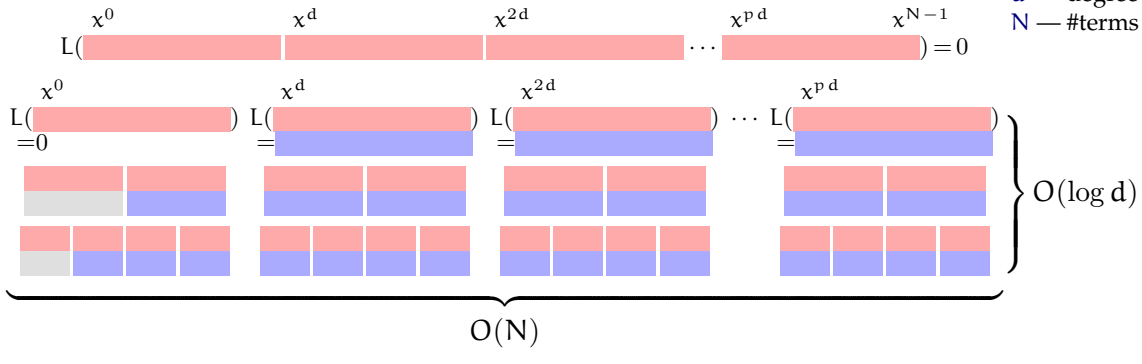
$O(\log N)$

$O(N)$

Whole tree: $O(r\,M(N)\log N)$

Polynomial coefficients of high degree

r — order
d — degree
N — #terms



Cost: $O\left(\frac{N}{d} r M(d) \log d\right)$

With schoolbook multiplication: $O(N r d)$

Naïve algorithm: $O(N r d)$ *for r solutions*

Large ℓ

$$\underbrace{\left[\begin{array}{l}(\alpha_{0,d}\,x^d+\cdots+\alpha_{0,\ell}\,x^{2\ell}+\alpha_{0,2\ell-1}\,x^{2\ell-1}+\cdots+\alpha_{0,0})\\ +\,(\alpha_{1,d}\,x^d+\cdots+\alpha_{1,\ell}\,x^{2\ell}+\alpha_{1,2\ell-1}\,x^{2\ell-1}+\cdots+\alpha_{1,0})\;\theta\\ +\qquad\qquad\cdots\\ +\,(\alpha_{r,d}\,x^d+\cdots+\alpha_{r,\ell}\,x^{2\ell}+\alpha_{r,2\ell-1}\,x^{2\ell-1}+\cdots+\alpha_{r,0})\;\theta^r\end{array}\right]}_{L}\cdot(y_n x^n+\cdots+y_{n+\ell-1}x^{n+\ell-1})$$

Cost: $O(r\,M(\ell))$

$$=\ \blacksquare\,x^n\ +\cdots+\blacksquare\,x^{n+\ell-1}$$
$$+\blacksquare\,x^{n+\ell}+\cdots+\blacksquare\,x^{n+2\ell-1}$$
$$+O(x^{2\ell})$$

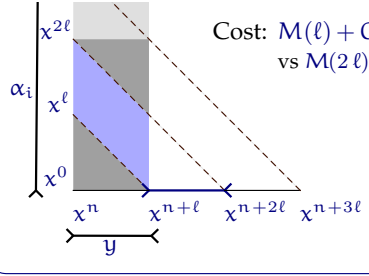
Large ℓ

$$\underbrace{\begin{bmatrix} (\alpha_{0,d}\,x^d + \cdots + \alpha_{0,\ell}\,x^{2\ell} + \alpha_{0,2\ell-1}\,x^{2\ell-1} + \cdots + \alpha_{0,0}) \\ + (\alpha_{1,d}\,x^d + \cdots + \alpha_{1,\ell}\,x^{2\ell} + \alpha_{1,2\ell-1}\,x^{2\ell-1} + \cdots + \alpha_{1,0})\ \theta \\ + \qquad\qquad \cdots \\ + (\alpha_{r,d}\,x^d + \cdots + \alpha_{r,\ell}\,x^{2\ell} + \alpha_{r,2\ell-1}\,x^{2\ell-1} + \cdots + \alpha_{r,0})\ \theta^r \end{bmatrix}}_{L} \cdot (y_n x^n + \cdots + y_{n+\ell-1} x^{n+\ell-1})$$

Cost: $O(r\,M(\ell))$

$$= \blacksquare\, x^n + \cdots + \blacksquare\, x^{n+\ell-1} + \blacksquare\, x^{n+\ell} + \cdots + \blacksquare\, x^{n+2\ell-1} + O(x^{2\ell})$$

Middle product



Cost: $M(\ell) + O(\ell)$
vs $M(2\,\ell)$

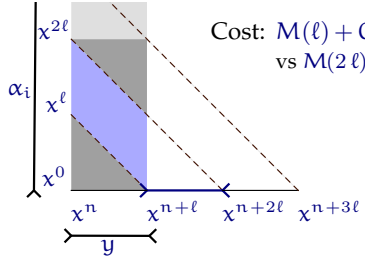
Large ℓ

$$\underbrace{\left[\begin{array}{l}(\alpha_{0,d}\,x^d+\cdots+\alpha_{0,\ell}\,x^{2\ell}+\alpha_{0,2\ell-1}\,x^{2\ell-1}+\cdots+\alpha_{0,0})\\ +\,(\alpha_{1,d}\,x^d+\cdots+\alpha_{1,\ell}\,x^{2\ell}+\alpha_{1,2\ell-1}\,x^{2\ell-1}+\cdots+\alpha_{1,0})\;\theta\\ +\quad\cdots\\ +\,(\alpha_{r,d}\,x^d+\cdots+\alpha_{r,\ell}\,x^{2\ell}+\alpha_{r,2\ell-1}\,x^{2\ell-1}+\cdots+\alpha_{r,0})\;\theta^r\end{array}\right]}_{L}\cdot(y_n x^n+\cdots+y_{n+\ell-1}x^{n+\ell-1})$$

Cost: $O(r\,M(\ell))$

$$=\ \blacksquare\,x^n\ +\cdots+\blacksquare\,x^{n+\ell-1}+\blacksquare\,x^{n+\ell}+\cdots+\blacksquare\,x^{n+2\ell-1}+O(x^{2\ell})$$

Middle product



Cost: $M(\ell)+O(\ell)$ vs $M(2\,\ell)$

FFT model

- naïve ⇨ $3\,r$ transforms (of length $3\,\ell$)
- reuse of $\mathrm{DFT}(\alpha_i)$; sum in transformed domain ⇨ $r\,\frac{d}{N}+r+1$ transforms

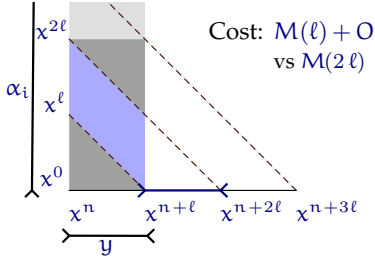
Large ℓ

$$\underbrace{\begin{bmatrix} (\alpha_{0,d}\,x^d + \cdots + \alpha_{0,\ell}\,x^{2\ell} + \alpha_{0,2\ell-1}\,x^{2\ell-1} + \cdots + \alpha_{0,0}) \\ + (\alpha_{1,d}\,x^d + \cdots + \alpha_{1,\ell}\,x^{2\ell} + \alpha_{1,2\ell-1}\,x^{2\ell-1} + \cdots + \alpha_{1,0})\ \theta \\ + \qquad\qquad \cdots \\ + (\alpha_{r,d}\,x^d + \cdots + \alpha_{r,\ell}\,x^{2\ell} + \alpha_{r,2\ell-1}\,x^{2\ell-1} + \cdots + \alpha_{r,0})\ \theta^r \end{bmatrix}}_{L} \cdot (y_n x^n + \cdots + y_{n+\ell-1} x^{n+\ell-1})$$

Cost: $O(r\,M(\ell))$

$$= \blacksquare\, x^n + \cdots + \blacksquare\, x^{n+\ell-1} + \blacksquare\, x^{n+\ell} + \cdots + \blacksquare\, x^{n+2\ell-1} + O(x^{2\ell})$$

Middle product



Cost: $M(\ell) + O(\ell)$
vs $M(2\,\ell)$

FFT model

- naïve ⇨ $3\,r$ transforms (of length $3\,\ell$)
- $\begin{cases} \text{reuse of } \mathrm{DFT}(\alpha_i) \\ \text{sum in transformed domain} \end{cases}$ ⇨ $r\frac{d}{N} + r + 1$ transforms

Fast algorithms

$O(r\,M(\ell)\log\ell + r^{\omega-1}\ell)$ for r values of y when $\ell \geqslant r$

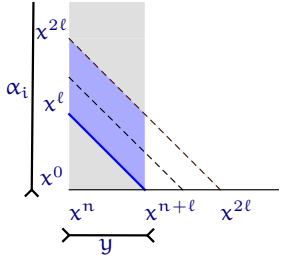
[Bostan, Benoit, van der Hoeven, 2012; van der Hoeven, 2016]

Small ℓ

$$\underbrace{\left[\begin{array}{l} (\cdots + \alpha_{0,\ell}\, x^{\ell} + \alpha_{0,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{0,1}\, x + \alpha_{0,0}) \\ + (\cdots + \alpha_{1,\ell}\, x^{\ell} + \alpha_{1,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{1,1}\, x + \alpha_{1,0})\ \theta \\ + \qquad\qquad \cdots \\ + (\cdots + \alpha_{r,\ell}\, x^{\ell} + \alpha_{1,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{r,1}\, x + \alpha_{r,0})\ \theta^{r} \end{array}\right]}_{L} \cdot (y_n x^n + y_{n+1} x^{n+1} + \cdots + y_{n+\ell-1} x^{n+\ell-1})$$

schoolbook mul: $O(r\,\ell^2)$

$$= \blacksquare\, x^n + \blacksquare\, x^{n+1} + \cdots + \blacksquare\, x^{n+\ell-1} + g_{n+\ell}\, x^{n+\ell} + \blacksquare\, x^{n+\ell+1} + \cdots + \blacksquare\, x^{n+2\ell-1} + O(x^{2d})$$



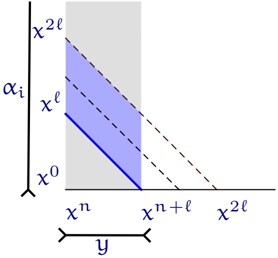
$g_{n+\ell} =$

Small ℓ

$$\underbrace{\begin{aligned}\big[\quad & (\cdots + \alpha_{0,\ell}\, x^{\ell} + \alpha_{0,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{0,1}\, x + \alpha_{0,0}) \\ + \; & (\cdots + \alpha_{1,\ell}\, x^{\ell} + \alpha_{1,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{1,1}\, x + \alpha_{1,0})\ \theta \\ + \; & \qquad\qquad\qquad \cdots \\ + \; & (\cdots + \alpha_{r,\ell}\, x^{\ell} + \alpha_{1,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{r,1}\, x + \alpha_{r,0})\ \theta^{r} \ \big]\end{aligned}}_{L} \cdot (y_n x^n + y_{n+1} x^{n+1} + \cdots + y_{n+\ell-1} x^{n+\ell-1})$$

schoolbook mul: $O(r\,\ell^2)$

$$\begin{aligned} = \; & \blacksquare\, x^n + \blacksquare\, x^{n+1} + \cdots + \blacksquare\, x^{n+\ell-1} \\ & + g_{n+\ell}\, x^{n+\ell} + \blacksquare\, x^{n+\ell+1} + \cdots + \blacksquare\, x^{n+2\ell-1} \\ & + O(x^{2d}) \end{aligned}$$



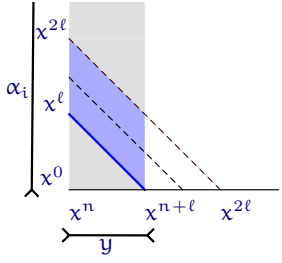
$$g_{n+\ell} = \big(\alpha_{0,\ell} \qquad\qquad\qquad \big) \qquad \times y_n$$

Small ℓ

$$\underbrace{\begin{bmatrix} (\cdots + \alpha_{0,\ell}\, x^{\ell} + \alpha_{0,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{0,1}\, x + \alpha_{0,0}) \\ + (\cdots + \alpha_{1,\ell}\, x^{\ell} + \alpha_{1,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{1,1}\, x + \alpha_{1,0})\ \theta \\ + \qquad\qquad \cdots \\ + (\cdots + \alpha_{r,\ell}\, x^{\ell} + \alpha_{1,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{r,1}\, x + \alpha_{r,0})\ \theta^{r} \end{bmatrix}}_{L} \cdot (y_n x^n + y_{n+1} x^{n+1} + \cdots + y_{n+\ell-1} x^{n+\ell-1})$$

schoolbook mul: $O(r\,\ell^2)$

$$= \blacksquare\, x^n + \blacksquare\, x^{n+1} + \cdots + \blacksquare\, x^{n+\ell-1} + g_{n+\ell}\, x^{n+\ell} + \blacksquare\, x^{n+\ell+1} + \cdots + \blacksquare\, x^{n+2\ell-1} + O(x^{2d})$$



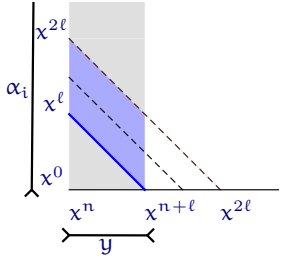
$g_{n+\ell} = (\alpha_{0,\ell} + \alpha_{1,\ell} \quad n \quad) \times y_n$

Small ℓ

$$\underbrace{\begin{bmatrix} (\cdots + \alpha_{0,\ell}\, x^{\ell} + \alpha_{0,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{0,1}\, x + \alpha_{0,0}) \\ + (\cdots + \alpha_{1,\ell}\, x^{\ell} + \alpha_{1,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{1,1}\, x + \alpha_{1,0})\ \theta \\ + \quad \cdots \\ + (\cdots + \alpha_{r,\ell}\, x^{\ell} + \alpha_{1,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{r,1}\, x + \alpha_{r,0})\ \theta^{r} \end{bmatrix}}_{L} \cdot (y_n x^n + y_{n+1} x^{n+1} + \cdots + y_{n+\ell-1} x^{n+\ell-1})$$

schoolbook mul: $O(r\,\ell^2)$

$$= \blacksquare\, x^n + \blacksquare\, x^{n+1} + \cdots + \blacksquare\, x^{n+\ell-1} + g_{n+\ell}\, x^{n+\ell} + \blacksquare\, x^{n+\ell+1} + \cdots + \blacksquare\, x^{n+2\ell-1} + O(x^{2d})$$



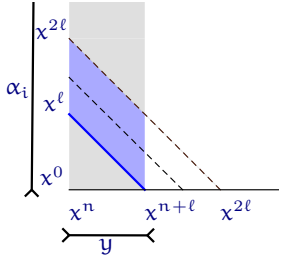
$$g_{n+\ell} = (\alpha_{0,\ell} + \alpha_{1,\ell}\ n + \cdots + \alpha_{r,\ell}\ n^r) \times y_n$$

Small ℓ

$$\underbrace{\begin{bmatrix} (\cdots + \alpha_{0,\ell}\, x^{\ell} + \alpha_{0,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{0,1}\, x + \alpha_{0,0}) \\ + (\cdots + \alpha_{1,\ell}\, x^{\ell} + \alpha_{1,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{1,1}\, x + \alpha_{1,0})\ \theta \\ + \qquad\qquad \cdots \\ + (\cdots + \alpha_{r,\ell}\, x^{\ell} + \alpha_{1,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{r,1}\, x + \alpha_{r,0})\ \theta^{r} \end{bmatrix}}_{L} \cdot (y_n x^n + y_{n+1} x^{n+1} + \cdots + y_{n+\ell-1} x^{n+\ell-1})$$

schoolbook mul: $O(r\,\ell^2)$

$$= \blacksquare\, x^n + \blacksquare\, x^{n+1} + \cdots + \blacksquare\, x^{n+\ell-1} + g_{n+\ell}\, x^{n+\ell} + \blacksquare\, x^{n+\ell+1} + \cdots + \blacksquare\, x^{n+2\ell-1} + O(x^{2d})$$



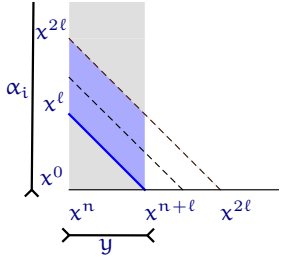
$$g_{n+\ell} = \begin{matrix} (\alpha_{0,\ell} + \alpha_{1,\ell}\, n + \cdots + \alpha_{r,\ell}\, n^r) & \times y_n \\ + (\alpha_{0,\ell-1} + \alpha_{1,\ell-1}(n+1) + \cdots + \alpha_{r,\ell-1}(n+1)^r) & \times y_{n+1} \end{matrix}$$

Small ℓ

$$\underbrace{\begin{bmatrix} (\cdots + \alpha_{0,\ell}\, x^{\ell} + \alpha_{0,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{0,1}\, x + \alpha_{0,0}) \\ + (\cdots + \alpha_{1,\ell}\, x^{\ell} + \alpha_{1,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{1,1}\, x + \alpha_{1,0})\ \theta \\ + \qquad\qquad \cdots \\ + (\cdots + \alpha_{r,\ell}\, x^{\ell} + \alpha_{1,\ell-1}\, x^{\ell-1} + \cdots + \alpha_{r,1}\, x + \alpha_{r,0})\ \theta^{r} \end{bmatrix}}_{L} \cdot (y_n x^n + y_{n+1} x^{n+1} + \cdots + y_{n+\ell-1} x^{n+\ell-1})$$

schoolbook mul: $O(r\,\ell^2)$

$$= \blacksquare\, x^n + \blacksquare\, x^{n+1} + \cdots + \blacksquare\, x^{n+\ell-1} + g_{n+\ell}\, x^{n+\ell} + \blacksquare\, x^{n+\ell+1} + \cdots + \blacksquare\, x^{n+2\ell-1} + O(x^{2d})$$



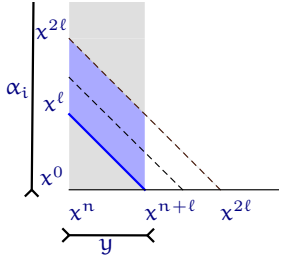
$$g_{n+\ell} = \underbrace{\begin{matrix} (\alpha_{0,\ell} + \alpha_{1,\ell}\ n + \cdots + \alpha_{r,\ell}\ n^r) & \times y_n \\ + (\alpha_{0,\ell-1} + \alpha_{1,\ell-1}(n+1) + \cdots + \alpha_{r,\ell-1}(n+1)^r) & \times y_{n+1} \\ + \qquad\qquad \cdots \\ + (\alpha_{0,1} + \alpha_{1,1}(n+\ell-1) + \cdots + \alpha_{r,\ell-1}(n+\ell-1)^r) & \times y_{n+\ell-1} \end{matrix}}_{\uparrow}$$

Small ℓ

$$\underbrace{\begin{bmatrix} (\cdots+\alpha_{0,\ell}x^{\ell}+\alpha_{0,\ell-1}x^{\ell-1}+\cdots+\alpha_{0,1}x+\alpha_{0,0}) \\ +\ (\cdots+\alpha_{1,\ell}x^{\ell}+\alpha_{1,\ell-1}x^{\ell-1}+\cdots+\alpha_{1,1}x+\alpha_{1,0})\ \theta \\ + \qquad\qquad \cdots \\ +\ (\cdots+\alpha_{r,\ell}x^{\ell}+\alpha_{1,\ell-1}x^{\ell-1}+\cdots+\alpha_{r,1}x+\alpha_{r,0})\ \theta^{r} \end{bmatrix}}_{L} \cdot (y_n x^n + y_{n+1}x^{n+1} + \cdots + y_{n+\ell-1}x^{n+\ell-1})$$

schoolbook mul: $O(r\,\ell^2)$

$$= \blacksquare\, x^n + \blacksquare\, x^{n+1} + \cdots + \blacksquare\, x^{n+\ell-1} + g_{n+\ell}\, x^{n+\ell} + \blacksquare\, x^{n+\ell+1} + \cdots + \blacksquare\, x^{n+2\ell-1} + O(x^{2d})$$



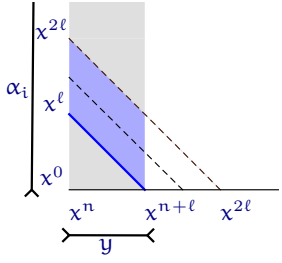
$$g_{n+\ell} = \underbrace{\begin{matrix} (\alpha_{0,\ell} + \alpha_{1,\ell}\ n + \cdots + \alpha_{r,\ell}\ n^r) & \times y_n \\ +\,(\alpha_{0,\ell-1} + \alpha_{1,\ell-1}(n+1) + \cdots + \alpha_{r,\ell-1}(n+1)^r) & \times y_{n+1} \\ + \qquad\qquad \cdots \\ +\,(\alpha_{0,1} + \alpha_{1,1}(n+\ell-1) + \cdots + \alpha_{r,\ell-1}(n+\ell-1)^r) & \times y_{n+\ell-1} \end{matrix}}_{r\ell \text{ (cheap) ops independent of } y}$$

Small ℓ

$$\underbrace{\begin{bmatrix} (\cdots + \alpha_{0,\ell}\,x^{\ell} + \alpha_{0,\ell-1}\,x^{\ell-1} + \cdots + \alpha_{0,1}\,x + \alpha_{0,0}) \\ +\ (\cdots + \alpha_{1,\ell}\,x^{\ell} + \alpha_{1,\ell-1}\,x^{\ell-1} + \cdots + \alpha_{1,1}\,x + \alpha_{1,0})\ \theta \\ + \qquad\qquad \cdots \\ +\ (\cdots + \alpha_{r,\ell}\,x^{\ell} + \alpha_{1,\ell-1}\,x^{\ell-1} + \cdots + \alpha_{r,1}\,x + \alpha_{r,0})\ \theta^{r} \end{bmatrix}}_{L} \cdot (y_n x^n + y_{n+1} x^{n+1} + \cdots + y_{n+\ell-1} x^{n+\ell-1})$$

schoolbook mul: $O(r\,\ell^2)$

$$= \blacksquare\, x^n + \blacksquare\, x^{n+1} + \cdots + \blacksquare\, x^{n+\ell-1} + g_{n+\ell}\, x^{n+\ell} + \blacksquare\, x^{n+\ell+1} + \cdots + \blacksquare\, x^{n+2\ell-1} + O(x^{2d})$$



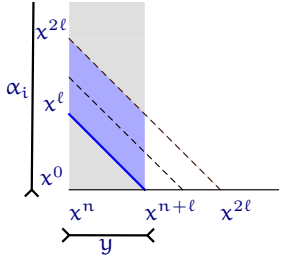
$$g_{n+\ell} = \underbrace{\begin{matrix} (\alpha_{0,\ell} + \alpha_{1,\ell}\ n + \cdots + \alpha_{r,\ell}\ n^r) & \times y_n \\ +\,(\alpha_{0,\ell-1} + \alpha_{1,\ell-1}\,(n+1) + \cdots + \alpha_{r,\ell-1}\,(n+1)^r) & \times y_{n+1} \\ + \qquad\qquad \cdots \\ +\,(\alpha_{0,1} + \alpha_{1,1}\,(n+\ell-1) + \cdots + \alpha_{r,\ell-1}\,(n+\ell-1)^r) & \times y_{n+\ell-1} \end{matrix}}_{r\ell \text{ (cheap) ops independent of } y} \quad \underbrace{\uparrow}_{\ell \text{ (costly) ops depending on } y}$$

Small ℓ

$$\underbrace{\begin{bmatrix} (\cdots + \alpha_{0,\ell}\,x^{\ell} + \alpha_{0,\ell-1}\,x^{\ell-1} + \cdots + \alpha_{0,1}\,x + \alpha_{0,0}) \\ + (\cdots + \alpha_{1,\ell}\,x^{\ell} + \alpha_{1,\ell-1}\,x^{\ell-1} + \cdots + \alpha_{1,1}\,x + \alpha_{1,0})\ \theta \\ + \qquad\qquad \cdots \\ + (\cdots + \alpha_{r,\ell}\,x^{\ell} + \alpha_{1,\ell-1}\,x^{\ell-1} + \cdots + \alpha_{r,1}\,x + \alpha_{r,0})\ \theta^{r} \end{bmatrix}}_{L} \cdot (y_n x^n + y_{n+1} x^{n+1} + \cdots + y_{n+\ell-1} x^{n+\ell-1})$$

schoolbook mul: $O(r\,\ell^2)$

$$= \blacksquare\, x^n + \blacksquare\, x^{n+1} + \cdots + \blacksquare\, x^{n+\ell-1} + g_{n+\ell}\, x^{n+\ell} + \blacksquare\, x^{n+\ell+1} + \cdots + \blacksquare\, x^{n+2\ell-1} + O(x^{2d})$$



$$g_{n+\ell} = \underbrace{\begin{matrix} (\alpha_{0,\ell} + \alpha_{1,\ell}\ n + \cdots + \alpha_{r,\ell}\ n^r) & \times y_n \\ + (\alpha_{0,\ell-1} + \alpha_{1,\ell-1}\,(n+1) + \cdots + \alpha_{r,\ell-1}\,(n+1)^r) & \times y_{n+1} \\ + \qquad\qquad \cdots \\ + (\alpha_{0,1} + \alpha_{1,1}\,(n+\ell-1) + \cdots + \alpha_{r,\ell-1}\,(n+\ell-1)^r) & \times y_{n+\ell-1} \end{matrix}}_{r\ell \text{ (cheap) ops independent of } y} \quad \underbrace{\uparrow}_{\ell \text{ (costly) ops depending on } y}$$

Total for r series:

/node: $r\,\ell^2$

/recursion tree: $r\,d^2$

over whole algorithm: $N\,r\,d$ ops

Beyond algebraic complexities

- All multiplications for computing the y_n were in size

$$(h + O(r \log N)) \quad \times \quad p$$

$\rightarrow$ cost $\approx p \dfrac{M_{\mathbb{Z}}(h)}{h}$ rather than $M_{\mathbb{Z}}(p)$

r — order (diff. eq.)
d — degree
h — integer height
p — precision
N — terms ($\approx p$)

- However, $\Theta(N)$ full-size muls $y_n \times \xi^n$ for deducing the sum(s)

Beyond algebraic complexities

- All multiplications for computing the y_n were in size

$$(h + O(r \log N)) \quad \times \quad p$$

$\rightarrow$ cost $\approx p \dfrac{M_{\mathbb{Z}}(h)}{h}$ rather than $M_{\mathbb{Z}}(p)$

r — order (diff. eq.)
d — degree
h — integer height
p — precision
N — terms ($\approx p$)

- However, $\Theta(N)$ full-size muls $y_n \times \xi^n$ for deducing the sum(s)

- E.g., with the divide-and-conquer algorithm

$$N\, r \frac{M(d)}{d} \log d \quad \text{“cheap” muls} \quad + \quad \underbrace{N\, r \quad \text{full-size muls}}_{\text{dominant if } p \gg r, d, h}$$

Beyond algebraic complexities

- All multiplications for computing the y_n were in size

$$(h + O(r \log N)) \quad \times \quad p$$

$\rightarrow$ cost $\approx p \dfrac{M_{\mathbb{Z}}(h)}{h}$ rather than $M_{\mathbb{Z}}(p)$

r — order (diff. eq.)
d — degree
h — integer height
p — precision
N — terms ($\approx p$)

- However, $\Theta(N)$ full-size muls $y_n \times \xi^n$ for deducing the sum(s)

- E.g., with the divide-and-conquer algorithm

$$N\, r \frac{M(d)}{d} \log d \quad \text{“cheap” muls} \quad + \quad \underbrace{N\, r \quad \text{full-size muls}}_{\text{dominant if } p \gg r, d, h}$$

- $y_n\, x^n = 2^{-\Omega(n)}$ $\rightarrow$ progressively decrease the working precision as n grows

5 High Precision

Setting

$$\underbrace{\left[\alpha_r(x)\,\theta^r + \cdots + \alpha_1(x)\,\theta + \alpha_0(x)\right]}_{\text{now } \in \mathbb{Z}[x] \text{ with coefficients of } \leqslant h \text{ bits}} \cdot y(x) = 0$$

r — order (diff. eq.)
d — degree (diff. eq.)
h — int. coeff. size
p — precision
N — terms

Want: $\sum_{n=0}^{N-1} y_n x^n$ for some given x

For simplicity: $h = O(1)$

Setting

$$\big[\alpha_r(x)\,\theta^r + \cdots + \alpha_1(x)\,\theta + \alpha_0(x)\big]\cdot y(x) = 0$$

$\alpha_r(x)$: now $\in \mathbb{Z}[x]$ with coefficients of $\leqslant h$ bits

r — order (diff. eq.)
d — degree (diff. eq.)
h — int. coeff. size
p — precision
N — terms

Want: $\sum_{n=0}^{N-1} y_n x^n$ for some given x

For simplicity: $h = O(1)$

Remark. General h:

$$\tilde{O}(d^\omega r^{1/2} p^{3/2}) \text{ bits ops}$$

using fast multipoint evaluation

(But high overhead,
poor numerical stability.)

[D. & G. Chudnovsky, 1987, 1990]

Setting

$$\left[\alpha_r(x)\,\theta^r + \cdots + \alpha_1(x)\,\theta + \alpha_0(x)\right] \cdot y(x) = 0$$

$\alpha_r(x)$: now $\in \mathbb{Z}[x]$ with coefficients of $\leqslant h$ bits

r — order (diff. eq.)
d — degree (diff. eq.)
h — int. coeff. size
p — precision
N — terms

Want: $\sum_{n=0}^{N-1} y_n x^n$ for some given x

For simplicity: $h = O(1)$

Remark. General h:

$\tilde{O}(d^\omega r^{1/2} p^{3/2})$ bits ops

using fast multipoint evaluation

(But high overhead,
poor numerical stability.)

[D. & G. Chudnovsky, 1987, 1990]

Example.

$$y(x) = \exp(x) \approx \sum_{n=0}^{N-1} \frac{1}{n!}\, x^n$$

Equation:
$(\theta - x) \cdot y = 0$

$r = d = 1$

$N \approx \dfrac{p}{\log p} \to \infty$

Exponential by rectangular splitting

[Smith 1989]

$$\sum_{n=0}^{N-1} \frac{x^n}{n!} = \left[1 + x + \cdots + \frac{1}{(\ell-1)!} x^{\ell-1} \right] x^0 \qquad N = \ell^2$$

$$+ \left[\frac{1}{\ell!} + \frac{1}{(\ell+1)!} x + \cdots + \frac{1}{(2\ell-1)!} x^{\ell-1} \right] x^{\ell}$$

$$+ \cdots$$

$$+ \left[\frac{1}{(\ell^2-\ell)!} + \frac{1}{(\ell^2-\ell+1)!} x + \cdots + \frac{1}{(\ell^2-1)!} x^{\ell-1} \right] x^{(\ell-1)\ell}$$

Exponential by rectangular splitting

[Smith 1989]

$$\sum_{n=0}^{N-1} \frac{x^n}{n!} = \left[1 + x + \cdots + \frac{1}{(\ell-1)!} x^{\ell-1} \right] x^0 \qquad N = \ell^2$$

$$+ \left[\frac{1}{\ell!} + \frac{1}{(\ell+1)!} x + \cdots + \frac{1}{(2\ell-1)!} x^{\ell-1} \right] x^{\ell}$$

$$+ \cdots$$

$$+ \left[\frac{1}{(\ell^2-\ell)!} + \frac{1}{(\ell^2-\ell+1)!} x + \cdots + \frac{1}{(\ell^2-1)!} x^{\ell-1} \right] x^{(\ell-1)\ell}$$

Exponential by rectangular splitting

[Smith 1989]

$$\sum_{n=0}^{N-1} \frac{x^n}{n!} = \left[1 + x + \cdots + \frac{1}{(\ell-1)!} x^{\ell-1} \right] x^0 \qquad N = \ell^2$$

$$+ \frac{1}{\ell!} \left[1 + \frac{\ell!}{(\ell+1)!} x + \cdots + \frac{\ell!}{(2\ell-1)!} x^{\ell-1} \right] x^{\ell}$$

$$+ \cdots$$

$$+ \frac{1}{(\ell^2-\ell)!} \left[1 + \frac{(\ell^2-\ell)!}{(\ell^2-\ell+1)!} x + \cdots + \frac{(\ell^2-\ell)!}{(\ell^2-1)!} x^{\ell-1} \right] x^{(\ell-1)\ell}$$

Exponential by rectangular splitting

[Smith 1989]

$$\sum_{n=0}^{N-1} \frac{x^n}{n!} = \left[\, 1 + x + \cdots + \frac{1}{(\ell-1)!}x^{\ell-1} \,\right] x^0 \qquad N = \ell^2$$
$$+ \frac{1}{\ell!}\left[\, 1 + \frac{\ell!}{(\ell+1)!}x + \cdots + \frac{\ell!}{(2\ell-1)!}x^{\ell-1} \,\right] x^{\ell}$$
$$+ \cdots$$
$$+ \frac{1}{(\ell^2-\ell)!}\left[\, 1 + \frac{(\ell^2-\ell)!}{(\ell^2-\ell+1)!}x + \cdots + \frac{(\ell^2-\ell)!}{(\ell^2-1)!}x^{\ell-1} \,\right] x^{(\ell-1)\ell}$$

Algorithm.

1. Compute $x, x^2, \ldots, x^{\ell-1}$ and $x^{\ell}, x^{2\ell}, \ldots, x^{(\ell-1)\ell}$ — $2\,\ell$ general mul
2. Compute the inner sums: e.g., — ℓ^2 scalar mul/div

$$1 + x + \cdots + \frac{1}{(\ell-1)!}x^{\ell-1} = \left(1 + \frac{1}{1}\left(x + \frac{1}{2}\left(x^2 + \frac{1}{3}\left(\cdots + \frac{1}{\ell-1}x^{\ell-1}\right)\right)\right)\right)$$

3. Compute the outer sum as — $(\ell$ general $+ \ell^2$ scalar$)$ mul

$$[\,\ldots\,]_0\, x^0 + \frac{1}{\ell!}\left([\,\ldots\,]_1\, x^{\ell} + \frac{\ell!}{(2\,\ell)!}\left(\cdots + \frac{(\ell^2-2\,\ell)!}{(\ell^2-\ell!)}[\,\ldots\,]_{\ell}\, x^{(\ell-1)\ell}\right)\right)$$

Rectangular splitting: comments

- Total cost

$$O(N^{1/2} M(p) + N p) = O(p^{1/2} M(p) + p^2) \text{ bit ops}$$

- Readily generalizes to **hypergeometric series**

$$y(x) = y_0 + y_1 x + y_2 x^2 + \cdots, \qquad \frac{y_{n+1}}{y_n} = \frac{q_1(n)}{q_0(n)}$$

- Partial analogue for differential equations [Johansson, 2014]

$$O\left(d^{\omega} \left[p^{1/2} M(p) + \frac{M(p^{1/2} \log p)}{p^{1/2} \log p} p^2 \right] \right) \text{ bit ops}$$

Also works, e.g., for $\sum_n \frac{1}{x(x+1)\cdots(x+n)}$

(now $M(n) =$ cost of integer multiplication)

Exponential by binary splitting

[Schroeppel, ~1972; Brent, 1976]

Sum version: $x \in \mathbb{Z}$ $\quad N = 2\ell$

$$\Sigma_N = \sum_{n=0}^{N-1} \frac{x^n}{n!} = \underbrace{\left(\frac{1}{0!} + \frac{x}{1!} + \cdots + \frac{x^{\ell-1}}{(\ell-1)!}\right)}_{} + \underbrace{\left(\frac{x^\ell}{\ell!} + \frac{x^{\ell+1}}{(\ell+1)!} + \cdots + \frac{x^{2\ell-1}}{(2\ell-1)!}\right)}_{}$$

Exponential by binary splitting

[Schroeppel, ~1972; Brent, 1976]

Sum version: $x \in \mathbb{Z}$ $\quad N = 2\ell$

$$\Sigma_N = \sum_{n=0}^{N-1} \frac{x^n}{n!} = \underbrace{\left(\frac{1}{0!} + \frac{x}{1!} + \cdots + \frac{x^{\ell-1}}{(\ell-1)!}\right)}_{} + \frac{x^\ell}{\ell!} \underbrace{\left(1 + \frac{x}{\ell+1} + \cdots + \frac{x^{\ell-1}}{(\ell+1)\cdots(2\ell-1)!}\right)}_{}$$

Exponential by binary splitting

[Schroeppel, ~1972; Brent, 1976]

Sum version: $x \in \mathbb{Z}$ $\quad N = 2\ell$

$$\Sigma_N = \sum_{n=0}^{N-1} \frac{x^n}{n!} = \underbrace{\left(\frac{1}{0!} + \frac{x}{1!} + \cdots + \frac{x^{\ell-1}}{(\ell-1)!}\right)}_{= \frac{T(0,\ell)}{Q(0,\ell)} \text{ (recurse)}} + \frac{x^\ell}{Q(0,\ell)} \underbrace{\left(1 + \frac{x}{\ell+1} + \cdots + \frac{x^{\ell-1}}{(\ell+1)\cdots(2\ell-1)!}\right)}_{= \frac{T(\ell,N)}{Q(\ell,N)} \text{ (recurse)}}$$

Exponential by binary splitting

[Schroeppel, ~1972; Brent, 1976]

Sum version: $x \in \mathbb{Z}$ $\quad N = 2\ell$

$$\Sigma_N = \sum_{n=0}^{N-1} \frac{x^n}{n!} = \underbrace{\left(\frac{1}{0!} + \frac{x}{1!} + \cdots + \frac{x^{\ell-1}}{(\ell-1)!}\right)}_{= \frac{T(0,\ell)}{Q(0,\ell)} \text{ (recurse)}} + \frac{x^\ell}{Q(0,\ell)} \underbrace{\left(1 + \frac{x}{\ell+1} + \cdots + \frac{x^{\ell-1}}{(\ell+1)\cdots(2\ell-1)!}\right)}_{= \frac{T(\ell,N)}{Q(\ell,N)} \text{ (recurse)}}$$

$$= \frac{Q(\ell,N)\,T(0,\ell) + x^\ell T(\ell,N)}{Q(0,\ell)\,Q(\ell,N)}$$

Exponential by binary splitting

[Schroeppel, ~1972; Brent, 1976]

Sum version: $x \in \mathbb{Z}$ $\quad N = 2\ell$

$$\Sigma_N = \sum_{n=0}^{N-1} \frac{x^n}{n!} = \underbrace{\left(\frac{1}{0!} + \frac{x}{1!} + \cdots + \frac{x^{\ell-1}}{(\ell-1)!}\right)}_{= \frac{T(0,\ell)}{Q(0,\ell)} \text{ (recurse)}} + \frac{x^\ell}{Q(0,\ell)} \underbrace{\left(1 + \frac{x}{\ell+1} + \cdots + \frac{x^{\ell-1}}{(\ell+1)\cdots(2\ell-1)!}\right)}_{= \frac{T(\ell,N)}{Q(\ell,N)} \text{ (recurse)}}$$

$$= \frac{Q(\ell,N)\,T(0,\ell) + x^\ell T(\ell,N)}{Q(0,\ell)\,Q(\ell,N)} = \frac{T(0,N)}{Q(0,N)}$$

Exponential by binary splitting

[Schroeppel, ~1972; Brent, 1976]

Sum version: $x \in \mathbb{Z}$ $\quad N = 2\ell$

$$\Sigma_N = \sum_{n=0}^{N-1} \frac{x^n}{n!} = \underbrace{\left(\frac{1}{0!} + \frac{x}{1!} + \cdots + \frac{x^{\ell-1}}{(\ell-1)!}\right)}_{= \frac{T(0,\ell)}{Q(0,\ell)} \text{ (recurse)}} + \frac{x^\ell}{Q(0,\ell)} \underbrace{\left(1 + \frac{x}{\ell+1} + \cdots + \frac{x^{\ell-1}}{(\ell+1)\cdots(2\ell-1)!}\right)}_{= \frac{T(\ell,N)}{Q(\ell,N)} \text{ (recurse)}}$$

$$= \frac{Q(\ell,N)\,T(0,\ell) + x^\ell T(\ell,N)}{Q(0,\ell)\,Q(\ell,N)} = \frac{T(0,N)}{Q(0,N)}$$

Product version:

$$u_n := \frac{x^n}{n!} = \frac{x}{n} u_{n-1}$$

$$\begin{pmatrix} u_n \\ \Sigma_n \end{pmatrix} = \begin{pmatrix} x/n & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} u_{n-1} \\ \Sigma_{n-1} \end{pmatrix}$$

Exponential by binary splitting

[Schroeppel, ~1972; Brent, 1976]

Sum version: $x \in \mathbb{Z} \qquad N = 2\ell$

$$\Sigma_N = \sum_{n=0}^{N-1} \frac{x^n}{n!} = \underbrace{\left(\frac{1}{0!} + \frac{x}{1!} + \cdots + \frac{x^{\ell-1}}{(\ell-1)!}\right)}_{= \frac{T(0,\ell)}{Q(0,\ell)} \text{ (recurse)}} + \frac{x^\ell}{Q(0,\ell)} \underbrace{\left(1 + \frac{x}{\ell+1} + \cdots + \frac{x^{\ell-1}}{(\ell+1)\cdots(2\ell-1)!}\right)}_{= \frac{T(\ell,N)}{Q(\ell,N)} \text{ (recurse)}}$$

$$= \frac{Q(\ell,N)\,T(0,\ell) + x^\ell T(\ell,N)}{Q(0,\ell)\,Q(\ell,N)} = \frac{T(0,N)}{Q(0,N)}$$

Product version:

$$u_n := \frac{x^n}{n!} = \frac{x}{n} u_{n-1}$$

$$\begin{pmatrix} u_n \\ \Sigma_n \end{pmatrix} = \frac{1}{n} \begin{pmatrix} x & 0 \\ n & n \end{pmatrix} \begin{pmatrix} u_{n-1} \\ \Sigma_{n-1} \end{pmatrix}$$

Exponential by binary splitting

[Schroeppel, ~1972; Brent, 1976]

Sum version: $x \in \mathbb{Z} \qquad N = 2\ell$

$$\Sigma_N = \sum_{n=0}^{N-1} \frac{x^n}{n!} = \underbrace{\left(\frac{1}{0!} + \frac{x}{1!} + \cdots + \frac{x^{\ell-1}}{(\ell-1)!}\right)}_{= \frac{T(0,\ell)}{Q(0,\ell)} \text{ (recurse)}} + \frac{x^\ell}{Q(0,\ell)} \underbrace{\left(1 + \frac{x}{\ell+1} + \cdots + \frac{x^{\ell-1}}{(\ell+1)\cdots(2\ell-1)!}\right)}_{= \frac{T(\ell,N)}{Q(\ell,N)} \text{ (recurse)}}$$

$$= \frac{Q(\ell,N)\,T(0,\ell) + x^\ell T(\ell,N)}{Q(0,\ell)\,Q(\ell,N)} = \frac{T(0,N)}{Q(0,N)}$$

Product version:

$$u_n := \frac{x^n}{n!} = \frac{x}{n} u_{n-1}$$

$$\begin{pmatrix} u_n \\ \Sigma_n \end{pmatrix} = \begin{pmatrix} x & 0 \\ N & N \end{pmatrix} \cdots \begin{pmatrix} x & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} u_0 \\ \Sigma_0 \end{pmatrix}$$

Exponential by binary splitting

[Schroeppel, ~1972; Brent, 1976]

Sum version: $x \in \mathbb{Z}$ $\quad N = 2\ell$

$$\Sigma_N = \sum_{n=0}^{N-1} \frac{x^n}{n!} = \underbrace{\left(\frac{1}{0!} + \frac{x}{1!} + \cdots + \frac{x^{\ell-1}}{(\ell-1)!}\right)}_{= \frac{T(0,\ell)}{Q(0,\ell)} \text{ (recurse)}} + \frac{x^\ell}{Q(0,\ell)} \underbrace{\left(1 + \frac{x}{\ell+1} + \cdots + \frac{x^{\ell-1}}{(\ell+1)\cdots(2\ell-1)!}\right)}_{= \frac{T(\ell,N)}{Q(\ell,N)} \text{ (recurse)}}$$

$$= \frac{Q(\ell,N)\,T(0,\ell) + x^\ell T(\ell,N)}{Q(0,\ell)\,Q(\ell,N)} = \frac{T(0,N)}{Q(0,N)}$$

Product version:

$$u_n := \frac{x^n}{n!} = \frac{x}{n} u_{n-1}$$

$$\begin{pmatrix} u_n \\ \Sigma_n \end{pmatrix} = \underbrace{\begin{pmatrix} x & 0 \\ N & N \end{pmatrix} \cdots \begin{pmatrix} x & 0 \\ \ell+1 & \ell+1 \end{pmatrix}}_{} \underbrace{\begin{pmatrix} x & 0 \\ \ell & \ell \end{pmatrix} \cdots \begin{pmatrix} x & 0 \\ 1 & 1 \end{pmatrix}}_{} \begin{pmatrix} u_0 \\ \Sigma_0 \end{pmatrix}$$

Exponential by binary splitting

[Schroeppel, ~1972; Brent, 1976]

Sum version:

$x \in \mathbb{Z} \qquad N = 2\ell$

$$\Sigma_N = \sum_{n=0}^{N-1} \frac{x^n}{n!} = \underbrace{\left(\frac{1}{0!}+\frac{x}{1!}+\cdots+\frac{x^{\ell-1}}{(\ell-1)!}\right)}_{=\frac{T(0,\ell)}{Q(0,\ell)}\text{ (recurse)}} + \frac{x^\ell}{Q(0,\ell)} \underbrace{\left(1+\frac{x}{\ell+1}+\cdots+\frac{x^{\ell-1}}{(\ell+1)\cdots(2\ell-1)!}\right)}_{=\frac{T(\ell,N)}{Q(\ell,N)}\text{ (recurse)}}$$

$$= \frac{Q(\ell,N)\,T(0,\ell)+x^\ell T(\ell,N)}{Q(0,\ell)\,Q(\ell,N)} = \frac{T(0,N)}{Q(0,N)}$$

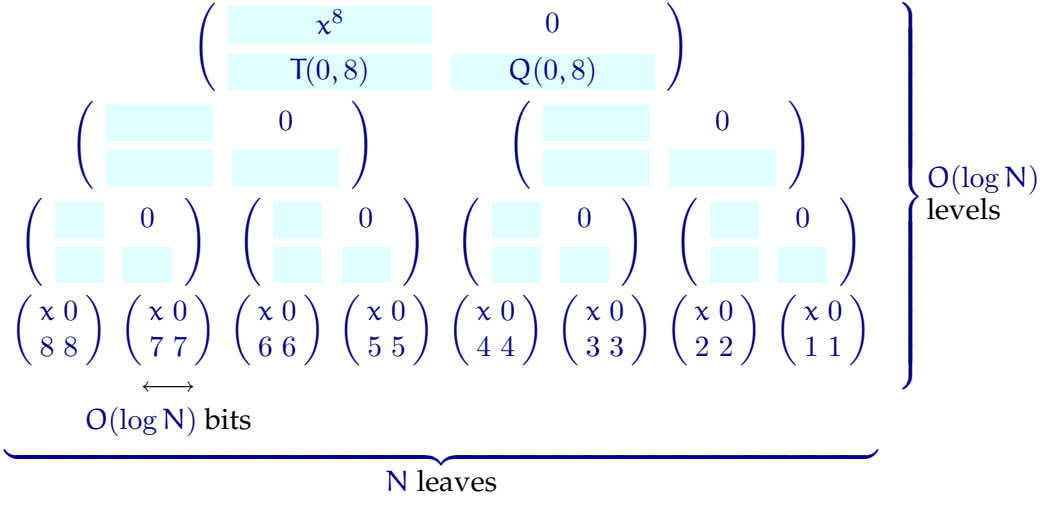
Product version:

$$u_n := \frac{x^n}{n!} = \frac{x}{n}u_{n-1}$$

$$\begin{pmatrix} u_n \\ \Sigma_n \end{pmatrix} = \underbrace{\begin{pmatrix} x & 0 \\ N & N \end{pmatrix}\cdots\begin{pmatrix} x & 0 \\ \ell+1 & \ell+1 \end{pmatrix}}_{\begin{pmatrix} x^\ell & 0 \\ T(\ell,N) & Q(\ell,N) \end{pmatrix}} \underbrace{\begin{pmatrix} x & 0 \\ \ell & \ell \end{pmatrix}\cdots\begin{pmatrix} x & 0 \\ 1 & 1 \end{pmatrix}}_{\begin{pmatrix} x^\ell & 0 \\ T(0,\ell) & Q(0,\ell) \end{pmatrix}} \begin{pmatrix} u_0 \\ \Sigma_0 \end{pmatrix}$$

Complexity

$\text{bit size}(x) = O(1)$



Row k from the bottom costs $O\big(2^k M(\,2^k \log N\,)\big) = O\big(M(N \log N)\big)$ bit ops

Whole product tree $O(M(N \log^2 N))$ bit ops (final division: $O(M(p))$

High-precision evaluation of classical constants

Theorem. [D. & G. Chudnovsky, 1987]

$$\frac{1}{\pi} = \frac{12}{c^{3/2}} \sum_{n=0}^{\infty} (-1)^n \frac{(6n)!}{(3n)!\, n!^3} \frac{(an+b)}{c^{3n}}, \qquad \begin{cases} a = 545140134 \\ b = 13591409 \\ c = 640320 \end{cases}$$

convergence: 14 decimal digits / term (!)

1 hypergeometric series, 1 square root, 1 division

cost $O\big(M(p \log(p)^2)\big)$ for p digits

AGM: $O(M(p) \log(p))$ [Salamin 1976, Brent 1978]
but higher overhead

Binary splitting for ODEs

[D. & G. Chudnovsky, 1987, 1990]

The recurrence $q_0(n)\,y_n + \cdots + q_d(n)\,y_n = 0$ rewrites into

$$\begin{pmatrix} y_{n-d+1} \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} = \underbrace{\begin{pmatrix} & 1 & & \\ & & \ddots & \\ & & & 1 \\ -\frac{q_d(n)}{q_0(n)} & -\frac{q_{d-1}(n)}{q_0(n)} & & -\frac{q_1(n)}{q_0(n)} \end{pmatrix}}_{q_0(n)^{-1}B(n)} \begin{pmatrix} y_{n-d} \\ \vdots \\ y_{n-2} \\ y_{n-1} \end{pmatrix}$$

Binary splitting for ODEs

[D. & G. Chudnovsky, 1987, 1990]

The recurrence $q_0(n)\,y_n + \cdots + q_d(n)\,y_n = 0$ rewrites into

$$\begin{pmatrix} y_{n-d+1} \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} = \underbrace{\begin{pmatrix} & 1 & & \\ & & \ddots & \\ & & & 1 \\ -\frac{q_d(n)}{q_0(n)} & -\frac{q_{d-1}(n)}{q_0(n)} & & -\frac{q_1(n)}{q_0(n)} \end{pmatrix}}_{q_0(n)^{-1}B(n)} \begin{pmatrix} y_{n-d} \\ \vdots \\ y_{n-2} \\ y_{n-1} \end{pmatrix}$$

For the sums $\Sigma_n = \sum_{n=0}^{N-1} y_n x^n$ this implies

$$\begin{pmatrix} y_{n-d+1}\,x^n \\ \vdots \\ y_n\,x^n \\ \Sigma_n \end{pmatrix} = \frac{1}{q_0(n)} \left(\begin{array}{ccc|c} & B(n) & & 0 \\ & \left(\begin{array}{c} \\ \\ \end{array}\right) x & & \vdots \\ & & & 0 \\ \hline 0 & 0 \;\cdots & q_0(n) & q_0(n) \end{array}\right) \begin{pmatrix} y_{n-d}\,x^{n-1} \\ \vdots \\ y_{n-1}\,x^{n-1} \\ \Sigma_{n-1} \end{pmatrix}$$

Binary splitting for ODEs

$$\begin{pmatrix} y_{n-d+1}\,x^n \\ \vdots \\ y_n\,x^n \\ \Sigma_n \end{pmatrix} = \frac{1}{q_0(n)} \underbrace{\left(\begin{array}{ccc|c} & & & 0 \\ & \left(\begin{array}{ccc} & & \\ & & \\ \blacksquare & \cdots & \blacksquare \end{array}\right) x & & \vdots \\ & & & 0 \\ \hline 0 & \cdots & q_0(n) & q_0(n) \end{array}\right)}_{P(n)} \begin{pmatrix} y_{n-d}\,x^{n-1} \\ \vdots \\ y_{n-1}\,x^{n-1} \\ \Sigma_{n-1} \end{pmatrix}$$

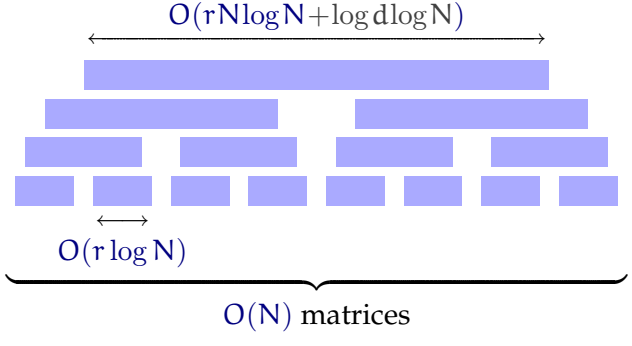
Algorithm.

1. Compute $P(N)\,P(N-1)\cdots P(r)$ as a product tree
2. Apply to $[0,\dots,0,y_0\,x^{r-1},\dots,y_{r-1}\,x^{r-1},\Sigma_{r-1}]^T$, take the last entry
3. Divide by $q_0(N)\cdots q_0(r)$ ($=$ bottom right entry of the matrix product)

r solutions for $\sim$ the price of one

Complexity

$\mathrm{size}(x) = O(1)$ (ok at intermediate points)

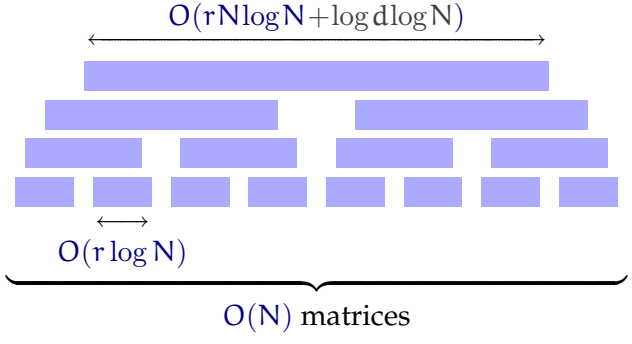


$$= \left(\begin{array}{c|c} \begin{pmatrix} & & \blacksquare \\ & \iddots & \\ \blacksquare & \cdots & \blacksquare \end{pmatrix} x & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ \hline 0 \;\; 0 \;\; \cdots \;\; q_0(n) & q_0(n) \end{array} \right)$$

$$\Pi = \left(\begin{array}{c|c} \cdot\, x^{N-r} & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ \hline \blacksquare \;\; \blacksquare \;\; \cdots \;\; \blacksquare & \blacksquare \end{array} \right)$$

Complexity

$\mathrm{size}(x) = O(1)$ (ok at intermediate points)



$$= \left(\begin{array}{c|c} \left(\begin{array}{ccc} & & \blacksquare \\ & \ddots & \\ \blacksquare & \cdots & \blacksquare \end{array} \right) x & \begin{array}{c} 0 \\ \vdots \\ 0 \end{array} \\ \hline 0 \;\; 0 \;\; \cdots \;\; q_0(n) & q_0(n) \end{array} \right)$$

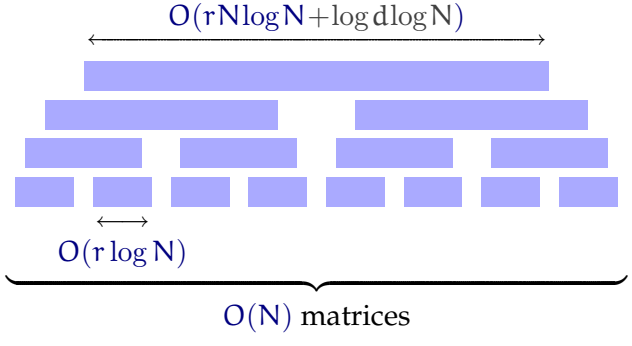
$$\Pi = \left(\begin{array}{c|c} \blacksquare \cdot x^{N-r} & \begin{array}{c} 0 \\ \vdots \\ 0 \end{array} \\ \hline \blacksquare \;\; \blacksquare \;\; \cdots \;\; \blacksquare & \blacksquare \end{array} \right)$$

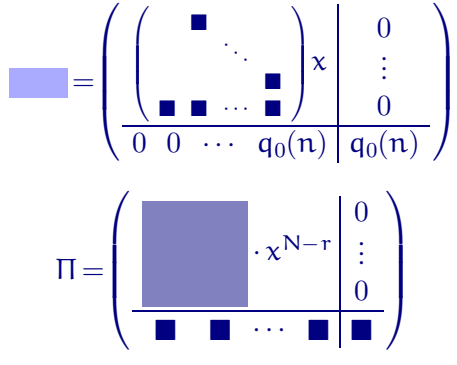
Cost for $r \leqslant d = O(N)$:

- Top node $O(d^\omega M(r N \log N))$
- Whole tree $O((\log N) \cdot [\text{top node}])$

Complexity

$\mathrm{size}(x) = O(1)$ (ok at intermediate points)



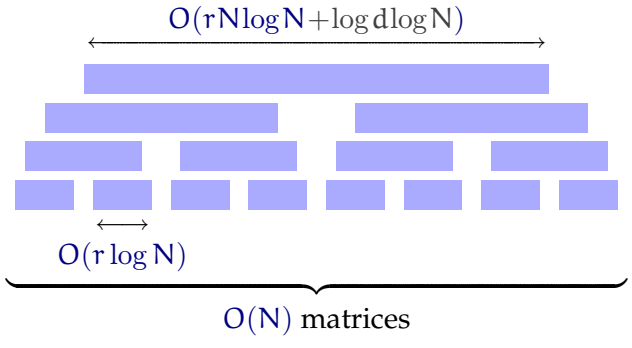


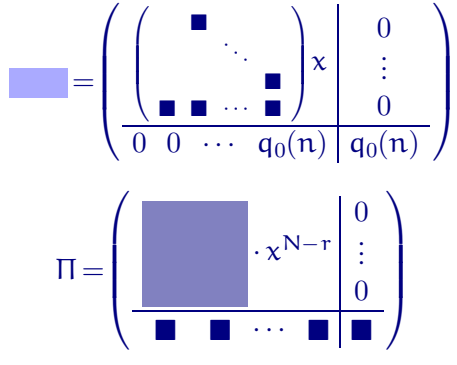
Cost for $r \leqslant d = O(N)$:

- Top node $O(d^\omega M(r N \log N))$
  $O(d^2 M(r N \log N) + d^\omega r N \log N)$ (FFT model)
  $O(d^2 M(r N \log N) + d^{\omega + o(1)} r N \log^{1+o(1)} N)$ [Harvey, van der Hoeven, 2018]
- Whole tree $O((\log N) \cdot [\text{top node}])$

Complexity

$\mathrm{size}(x) = O(1)$ (ok at intermediate points)





Cost for $r \leqslant d = O(N)$:

- Leaves $O(N\, d\, r\, M(r \log N))$ naïvely, actually negligible
- Top node $O(d^\omega\, M(r\, N \log N))$
  $O(d^2\, M(r\, N \log N) + d^\omega\, r\, N \log N)$ (FFT model)
  $O(d^2\, M(r\, N \log N) + d^{\omega+o(1)}\, r\, N \log^{1+o(1)} N)$ [Harvey, van der Hoeven, 2018]
- Whole tree $O((\log N) \cdot [\text{top node}])$

Remarks

- Using **unreduced fractions** is crucial
  (otherwise integer gcds would dominate the cost)
- But removing “easy” common factors helps

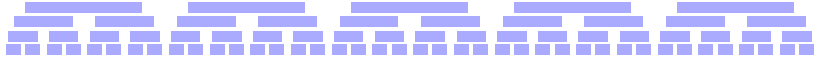
Remarks

- Using **unreduced fractions** is crucial

  (otherwise integer gcds would dominate the cost)
- But removing "easy" common factors helps

- In the top $\Theta(\log \log N)$ levels, entry sizes become $> p$

  Switch to iterative multiplication at precision $\sim p$



  (Also improves space complexity)

Remarks

- Using **unreduced fractions** is crucial

  (otherwise integer gcds would dominate the cost)
- But removing "easy" common factors helps

- In the top $\Theta(\log \log N)$ levels, entry sizes become $> p$

  Switch to iterative multiplication at precision $\sim p$

  (Also improves space complexity)

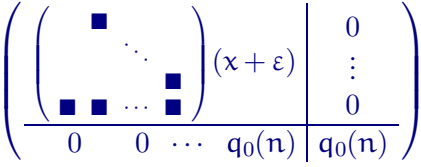
- More constant-factor savings in the FFT model

Derivatives

$$\Sigma_n(x) = \sum_{n=0}^{N-1} y_n x^n$$

Replace x by $x+\varepsilon$, work modulo ε^r:

$$\Sigma_n(x+\varepsilon) = \Sigma_n(x) + \Sigma_n'(x)\,\varepsilon + \cdots + \frac{1}{(r-1)!}\,\Sigma_n^{(r-1)}(x)\,\varepsilon^{r-1} + O(\varepsilon^r)$$



$$\Pi = \left(\begin{array}{c|c} \blacksquare \cdot (x+\varepsilon)^{N-r} & \begin{matrix}0\\ \vdots \\ 0\end{matrix} \\ \hline \blacksquare \quad \blacksquare \quad \cdots \quad \blacksquare & \blacksquare \end{array}\right)$$

$\blacksquare$ does not depend on ε (but $\blacksquare\ \blacksquare\ \cdots\ \blacksquare$ does) $\Rightarrow$ constant factor ($r \leqslant d$)

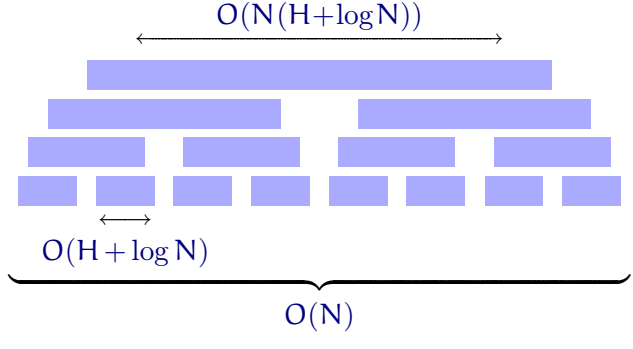
Regular singular points

Similarly, compute modulo $\mu(\lambda)$ and S_k^K to handle x^λ and logs

Dependency on the evaluation point

x of bit size H

$r, d, h = O(1)$



$$M\Big(\overbrace{N\underbrace{(H+\log N)}_{\text{size of each leaf}}}^{\text{size of each row}}\Big)\underbrace{\log N}_{\text{depth}} \quad = \quad \tilde{\Theta}(p^2) \quad \text{if } N, H = \Theta(p)$$

The bit-burst method for $\exp(x)$

[Brent, 1976]

Write
$$\begin{aligned} x &= 0.\underbrace{\xi_1}\underbrace{\xi_2\xi_3}\underbrace{\xi_4\xi_5\xi_6\xi_7}\underbrace{\xi_8\xi_9\xi_{10}\xi_{11}\xi_{12}\xi_{13}\xi_{14}\xi_{15}}\xi_{16}\xi_{17}\ldots \\ &= m_0 + m_1 + m_2 + \cdots + m_{K-1} \\ \exp(x) &= \exp(m_0)\exp(m_1)\cdots\exp(m_{K-1}) \end{aligned}$$

$$\begin{cases} m_k \leqslant 2^{-2^k+1} \\ m_k \text{ fits on } 2^k \text{ bits} \end{cases}$$

$K = O(\log p)$

The bit-burst method for $\exp(x)$

[Brent, 1976]

Write
$$\begin{aligned} x &= 0.\underbrace{\xi_1}\underbrace{\xi_2\xi_3}\underbrace{\xi_4\xi_5\xi_6\xi_7}\underbrace{\xi_8\xi_9\xi_{10}\xi_{11}\xi_{12}\xi_{13}\xi_{14}\xi_{15}}\xi_{16}\xi_{17}\ldots \\ &= m_0 + m_1 + m_2 + \cdots + m_{K-1} \\ \exp(x) &= \exp(m_0)\exp(m_1)\cdots\exp(m_{K-1}) \end{aligned}$$

$$\begin{cases} m_k \leqslant 2^{-2^k+1} \\ m_k \text{ fits on } 2^k \text{ bits} \end{cases}$$

$K = O(\log p)$

For $\exp(m_k)$:

$H \leqslant 2^k$

$\downarrow$

$M\big(N\,(H + \log N)\big)\log N$

$\uparrow$

because $m_k \lesssim 2^{-2^k}$:
$N = O(2^{-k}p)$

The bit-burst method for $\exp(x)$

[Brent, 1976]

Write
$$\begin{aligned} x &= 0.\underbrace{\xi_1}\underbrace{\xi_2\xi_3}\underbrace{\xi_4\xi_5\xi_6\xi_7}\underbrace{\xi_8\xi_9\xi_{10}\xi_{11}\xi_{12}\xi_{13}\xi_{14}\xi_{15}}\xi_{16}\xi_{17}\ldots \\ &= m_0 + m_1 + m_2 + \cdots + m_{K-1} \\ \exp(x) &= \exp(m_0)\exp(m_1)\cdots\exp(m_{K-1}) \end{aligned}$$

$\begin{cases} m_k \leqslant 2^{-2^k+1} \\ m_k \text{ fits on } 2^k \text{ bits} \end{cases}$

$K = O(\log p)$

For $\exp(m_k)$:

$H \leqslant 2^k$ (↓)

$$M\big(N\,(H+\log N)\big)\log N \;=\; O\big(M(p+2^{-k}p\log p)\log p\big)$$

(↑) because $m_k \lesssim 2^{-2^k}$: $N = O(2^{-k}p)$

The bit-burst method for $\exp(x)$

[Brent, 1976]

Write $x = 0.\underbrace{\xi_1}\underbrace{\xi_2\xi_3}\underbrace{\xi_4\xi_5\xi_6\xi_7}\underbrace{\xi_8\xi_9\xi_{10}\xi_{11}\xi_{12}\xi_{13}\xi_{14}\xi_{15}}\xi_{16}\xi_{17}\ldots$

$= m_0 + m_1 + m_2 + \cdots + m_{K-1}$

$\exp(x) = \exp(m_0)\exp(m_1)\cdots\exp(m_{K-1})$

$\begin{cases} m_k \leqslant 2^{-2^k+1} \\ m_k \text{ fits on } 2^k \text{ bits} \end{cases}$

$K = O(\log p)$

For $\exp(m_k)$:

$H \leqslant 2^k$ (↓)

$M\big(N\,(H + \log N)\big)\log N = O\big(M(p + 2^{-k}p\log p)\log p\big)$

$= O\big(M(p\log p + 2^{-k}p\log^2 p)\big)$

(↑) because $m_k \lesssim 2^{-2^k}$: $N = O(2^{-k}p)$

Total: $\displaystyle\sum_{k=0}^{K-1} \cdots$

The bit-burst method for $\exp(x)$

[Brent, 1976]

Write
$$\begin{aligned} x &= 0.\underbrace{\xi_1\xi_2\xi_3}\underbrace{\xi_4\xi_5\xi_6\xi_7}\underbrace{\xi_8\xi_9\xi_{10}\xi_{11}\xi_{12}\xi_{13}\xi_{14}\xi_{15}}\xi_{16}\xi_{17}\ldots \\ &= m_0 + m_1 + m_2 + \cdots + m_{K-1} \\ \exp(x) &= \exp(m_0)\exp(m_1)\cdots\exp(m_{K-1}) \end{aligned}$$

$$\begin{cases} m_k \leqslant 2^{-2^k+1} \\ m_k \text{ fits on } 2^k \text{ bits} \end{cases}$$

$K = O(\log p)$

For $\exp(m_k)$:

$H \leqslant 2^k$ ↓

$$M\big(N\,(H+\log N)\big)\log N = O\big(M(p + 2^{-k}p\log p)\log p\big) = O\big(M(p\log p + 2^{-k}p\log^2 p)\big)$$

↑ because $m_k \lesssim 2^{-2^k}$: $N = O(2^{-k}p)$

Total:
$$\sum_{k=0}^{K-1} \cdots = O\left(M\left(K\,p\log p + \sum_{k=0}^{K-1} 2^{-k}p\log^2 p\right)\right)$$

The bit-burst method for $\exp(x)$

[Brent, 1976]

Write
$$\begin{aligned} x &= 0.\underbrace{\xi_1}\underbrace{\xi_2\xi_3}\underbrace{\xi_4\xi_5\xi_6\xi_7}\underbrace{\xi_8\xi_9\xi_{10}\xi_{11}\xi_{12}\xi_{13}\xi_{14}\xi_{15}}\xi_{16}\xi_{17}\ldots \\ &= m_0 + m_1 + m_2 + \cdots + m_{K-1} \\ \exp(x) &= \exp(m_0)\exp(m_1)\cdots\exp(m_{K-1}) \end{aligned}$$

$$\begin{cases} m_k \leqslant 2^{-2^k+1} \\ m_k \text{ fits on } 2^k \text{ bits} \end{cases}$$

$K = O(\log p)$

For $\exp(m_k)$:

$H \leqslant 2^k$ ↓

$$M\big(N\,(H+\log N)\big)\log N = O\big(M(p + 2^{-k}p\log p)\log p\big) = O\big(M(p\log p + 2^{-k}p\log^2 p)\big)$$

↑ because $m_k \lesssim 2^{-2^k}$: $N = O(2^{-k}p)$

Total:
$$\sum_{k=0}^{K-1} \cdots = O\left(M\left(K\,p\log p + \sum_{k=0}^{K-1} 2^{-k}p\log^2 p\right)\right) = O(M(p\log^2 p))$$

The bit-burst method for $\exp(x)$

[Brent, 1976]

Write $x = 0.\underbrace{\xi_1}\underbrace{\xi_2\xi_3}\underbrace{\xi_4\xi_5\xi_6\xi_7}\underbrace{\xi_8\xi_9\xi_{10}\xi_{11}\xi_{12}\xi_{13}\xi_{14}\xi_{15}}\xi_{16}\xi_{17}\ldots$

$= m_0 + m_1 + m_2 + \cdots + m_{K-1}$

$\exp(x) = \exp(m_0)\exp(m_1)\cdots\exp(m_{K-1})$

$\begin{cases} m_k \leqslant 2^{-2^k+1} \\ m_k \text{ fits on } 2^k \text{ bits} \end{cases}$

$K = O(\log p)$

For $\exp(m_k)$:

$H \leqslant 2^k$ (↓)

$$M\big(N\,(H + \log N)\big)\log N = O\big(M(p + 2^{-k}p\log p)\log p\big) = O\big(M(p\log p + 2^{-k}p\log^2 p)\big)$$

(↑) because $m_k \lesssim 2^{-2^k}$: $N = O(2^{-k}p)$

Total:

$$\sum_{k=0}^{K-1} \cdots = O\left(M\left(K p\log p + \sum_{k=0}^{K-1} 2^{-k}p\log^2 p\right)\right) = O(M(p\log^2 p))$$

FFT model: $O\left(\frac{M(p\log^2 p)}{\log\log p}\right)$ [van der Hoeven, Johansson, 2024]; AGM: $O(M(p\log p))$ [Salamin 1976, Brent 1978]

The bit-burst method for $\exp(x)$

[Brent, 1976]

Write $x = 0.\underbrace{\xi_1}\underbrace{\xi_2\xi_3}\underbrace{\xi_4\xi_5\xi_6\xi_7}\underbrace{\xi_8\xi_9\xi_{10}\xi_{11}\xi_{12}\xi_{13}\xi_{14}\xi_{15}}\xi_{16}\xi_{17}\ldots$

$= m_0 + m_1 + m_2 + \cdots + m_{K-1}$

$\exp(x) = \exp(m_0)\exp(m_1)\cdots\exp(m_{K-1})$

$\begin{cases} m_k \leqslant 2^{-2^k+1} \\ m_k \text{ fits on } 2^k \text{ bits} \end{cases}$

$K = O(\log p)$

For $\exp(m_k)$:

$H \leqslant 2^k$ (↓)

$$M\big(N\,(H + \log N)\big)\log N = O\big(M(p + 2^{-k}p\log p)\log p\big) = O\big(M(p\log p + 2^{-k}p\log^2 p)\big)$$

(↑) because $m_k \lesssim 2^{-2^k}$: $N = O(2^{-k}p)$

Total:

$$\sum_{k=0}^{K-1} \cdots = O\left(M\left(K p\log p + \sum_{k=0}^{K-1} 2^{-k}p\log^2 p\right)\right) = O(M(p\log^2 p))$$

FFT model: $O\left(\frac{M(p\log^2 p)}{\log\log p}\right)$ [van der Hoeven, Johansson, 2024]; AGM: $O(M(p\log p))$ [Salamin 1976, Brent 1978]

The bit-burst method for ODEs

[D. & G. Chudnovsky, 1987, 1990]

Write

$$x = 0.\underbrace{\xi_1}\underbrace{\xi_2\xi_3}\underbrace{\xi_4\xi_5\xi_6\xi_7}\underbrace{\xi_8\xi_9\xi_{10}\xi_{11}\xi_{12}\xi_{13}\xi_{14}\xi_{15}}\xi_{16}\xi_{17}\dots$$

Compute by binary splitting the **connection matrices** along the path

$$\begin{aligned} 0.\xi_1 &\to 0.\xi_1\xi_2\xi_3 \\ &\to 0.\xi_1\xi_2\xi_3\xi_4\xi_5\xi_6\xi_7 \\ &\to 0.\xi_1\xi_2\xi_3\xi_4\xi_5\xi_6\xi_7\xi_8\xi_9\xi_{10}\xi_{11}\xi_{12}\xi_{13}\xi_{14}\xi_{15} \\ &\to \cdots \\ &\to x. \end{aligned}$$

At the kth step: the 'local' differential operator involves coeffs of $h = \Theta(2^k)$ bits
the evaluation point has size $H = \Theta(2^k)$
only $O(2^{-k}p)$ terms are needed.

Cost $\approx d^\omega M(p \log^2 p)$ for p bits.

Summary

Theorem. [D. & G. Chudnovsky, 1990; van der Hoeven, 1999, 2001; M., 2010]

For a fixed differential equation,
a fixed path in $\mathbb{C}\setminus\{\text{singular points}\}$, possibly joining regular singular points,
one can compute the connection matrix with an absolute error bounded by 2^{-p} in

$$O(M(p \log^2 p)) \text{ bit operations}$$

- Cost for a complete fundamental matrix $\approx$ cost for a single series
- Good in practice for small equations
- For $d > r$, the hidden dependency on d (about d^2 to d^ω) quickly becomes problematic.

6 Error Bounds

Error sources

Truncation errors

$$\sum_{n=0}^{\infty} y_n x^n \approx \sum_{n=0}^{N-1} y_n x^n$$

Error propagation (mesoscopic)

E.g., $\Delta_\ell \cdots \Delta_1 \Delta_0$ from approximate Δ_i,

$f_0(x) + \cdots + f_{K-1}(x) \log^{K-1}(x)$ from approximate $f_k(x)$ and $\log(x)$, …

Rounding errors (basic arithmetic)

In algorithms that work on approximate complex numbers:

$$y_n \approx -\big[\, q_1(n) \mathbin{\tilde{\times}} y_{n-1} \mathbin{\tilde{+}} \cdots \big] \mathbin{\tilde{/}} q_0(n)$$

Interval (ball) arithmetic:

$$\begin{aligned} [m_1 \pm \rho_1] + [m_2 \pm \rho_2] &\ni x_1 + x_2 \\ [m_1 \pm \rho_1] \times [m_2 \pm \rho_2] &\ni x_1 \times x_2 \\ &\cdots \end{aligned} \qquad \text{for} \quad \begin{aligned} m_1 - \rho_1 \leqslant x_1 \leqslant m_1 + \rho_1 \\ m_2 - \rho_2 \leqslant x_2 \leqslant m_2 + \rho_2 \end{aligned}$$

Truncation errors

Problem. Bound the tail

$$y(\xi) = \underbrace{\sum_{n=0}^{N-1} y_n \xi^n}_{\text{known}} + \underbrace{\sum_{n=N}^{\infty} y_n \xi^n}_{|\cdot| \leqslant ?}$$

given the "last few" coefficients $y_{N-1}, y_{N-2}, \dots$ of a truncated solution.

For us: 0 ordinary point (everything generalizes to regular singular points).

Truncation errors

Problem. Bound the tail

$$y(\xi) = \underbrace{\sum_{n=0}^{N-1} y_n \xi^n}_{\text{known}} + \underbrace{\sum_{n=N}^{\infty} y_n \xi^n}_{|\cdot| \leqslant ?}$$

given the "last few" coefficients $y_{N-1}, y_{N-2}, \dots$ of a truncated solution.

Example. When $y(x) = e^x$:

"worst case" function (indep. of N) ↓ first neglected term ↓

$$\sum_{n=N}^{\infty} \frac{\xi^n}{n!} \leqslant \frac{|\xi|^N}{N!} \sum_{n=0}^{\infty} \frac{N!}{(N+n)!} |\xi^n| \leqslant e^{|\xi|} \frac{|\xi|^N}{N!}$$

For us: 0 ordinary point (everything generalizes to regular singular points).

Setting

Rewrite the equation $a_r(x)\,y^{(r)}(x) + \cdots + a_0(x)\,y(x) = 0$ as

$$x \begin{bmatrix} y'(x) \\ y''(x) \\ \vdots \\ y^{(r)}(x) \end{bmatrix} = x \underbrace{\frac{1}{a_r(x)} \begin{bmatrix} & & a_r(x) & \\ & & & \ddots \\ & & & a_r(x) \\ -a_0(x) & -a_1(x) & \cdots & -a_{r-1}(x) \end{bmatrix}}_{A(x) = \frac{P(x)}{a_r(x)}} \underbrace{\begin{bmatrix} y(x) \\ y'(x) \\ \vdots \\ y^{(r-1)}(x) \end{bmatrix}}_{Y(x)}$$

W.l.o.g., assume $a_r(0) = 1$.

Setting

Rewrite the equation $a_r(x)\,y^{(r)}(x) + \cdots + a_0(x)\,y(x) = 0$ as

$$x \begin{bmatrix} y'(x) \\ y''(x) \\ \vdots \\ y^{(r)}(x) \end{bmatrix} = \underbrace{x \frac{1}{a_r(x)} \begin{bmatrix} & & a_r(x) & \\ & \iddots & & \\ a_r(x) & & & \\ -a_0(x) & -a_1(x) & \cdots & -a_{r-1}(x) \end{bmatrix}}_{A(x) = \frac{P(x)}{a_r(x)}} \underbrace{\begin{bmatrix} y(x) \\ y'(x) \\ \vdots \\ y^{(r-1)}(x) \end{bmatrix}}_{Y(x)}$$

Denote

$$Y(x) = \underbrace{\sum_{n=0}^{N-1} Y_n x^n}_{\tilde{Y}(x) \text{ (known)}} + \underbrace{\sum_{n=0}^{\infty} Y_n x^n}_{R(x)} \qquad (N \geqslant 1)$$

W.l.o.g., assume $a_r(0) = 1$.

A residual

$Y' - A\,Y = 0 \qquad Y = \tilde{Y} + R$
$A = a_r^{-1}\,P$

Using the known $\tilde{Y}$, compute

$$a_r\,x\,\tilde{Y}' - x\,P\,\tilde{Y} \;=\; Q$$

Orders of magnitude: $Q \approx$ first neglected terms $\approx$ tail of $y(x)$

A residual

$Y' - A\,Y = 0 \qquad Y = \tilde{Y} + R$
$A = a_r^{-1}\,P$

Using the known $\tilde{Y}$, compute

$$\begin{aligned} a_r\,x\,\tilde{Y}' - x\,P\,\tilde{Y} &= Q = \blacksquare\,x^N + \cdots + \blacksquare\,x^{N+d} \\ &= -(a_r\,x\,R' - x\,P\,R) \end{aligned}$$

Orders of magnitude: $Q \approx$ first neglected terms $\approx$ tail of $y(x)$

A residual

$$Y' - A\,Y = 0 \qquad Y = \tilde{Y} + R$$
$$A = a_r^{-1}\,P$$

Using the known $\tilde{Y}$, compute

$$\begin{aligned} a_r\,x\,\tilde{Y}' - x\,P\,\tilde{Y} &= Q = \blacksquare\,x^N + \cdots + \blacksquare\,x^{N+d} \\ &= -(a_r\,x\,R' - x\,P\,R) \end{aligned}$$

to get an explicit equation

$$x\,R' - x\,A\,R = \Phi \qquad \text{where} \qquad \Phi = -\frac{Q}{a_r} = x^N\,\frac{\blacksquare + \cdots + \blacksquare\,x^d}{a_r(x)}.$$

Orders of magnitude: $Q \approx$ first neglected terms $\approx$ tail of $y(x)$

Majorants

$$Y' - A\,Y = 0$$

$$x\,R'(x) - x\,A(x)\,R(x) = \Phi(x)$$

tail of $Y(x)$ $\downarrow$

$x\,A(x)$: $\uparrow$ $\frac{P}{a_r}$

$\Phi(x)$: $\uparrow$ $\frac{Q}{a_r}$

Majorants

$$Y' - A\,Y = 0$$

$$x\,R'(x) - x\,A(x)\,R(x) = \Phi(x)$$

tail of $Y(x)$ ↓

$$\frac{P}{a_r} \qquad \frac{Q}{a_r}$$

Notation. Write $u \ll \hat{u}$ when $\forall n, \|u_n\| \leqslant \hat{u}_n$.

$$u(x) = \sum_n u_n x^n$$
$$\hat{u}(x) = \sum_n \hat{u}_n x^n$$

Lemma. If $u \ll \hat{u}$ and $v \ll \hat{v}$ for some norm(s) s.t. $\|u_i v_j\| \leqslant \|u_i\|\,\|v_j\|$, then $u\,v \ll \hat{u}\,\hat{v}$.

Majorants

$Y' - A\,Y = 0$

$$x\,R'(x) - x\,A(x)\,R(x) = \Phi(x)$$

tail of $Y(x)$ ↓

$x\,A(x)\,R(x)$: $\uparrow$ $\frac{P}{a_r}$

$\Phi(x)$: $\uparrow$ $\frac{Q}{a_r}$

Algorithm.

- Write $a_r(x) = \prod_i \left(1 - \frac{x}{\xi_i}\right)$, compute $\check{\xi}_i \approx |\xi_i|$ with $0 < \check{\xi}_i < |\xi_i|$.

Notation. Write $u \ll \hat{u}$ when $\forall n, \|u_n\| \leqslant \hat{u}_n$.

$u(x) = \sum_n u_n x^n$
$\hat{u}(x) = \sum_n \hat{u}_n x^n$

Lemma. If $u \ll \hat{u}$ and $v \ll \hat{v}$ for some norm(s) s.t. $\|u_i v_j\| \leqslant \|u_i\|\,\|v_j\|$, then $u\,v \ll \hat{u}\,\hat{v}$.

Majorants

$$Y' - A\,Y = 0$$

$$x\,R'(x) - x\,A(x)\,R(x) = \Phi(x)$$

$x\,R'(x)$: tail of $Y(x)$ ↓

$x\,A(x)\,R(x)$: $\frac{P}{a_r}$ ↑

$\Phi(x)$: $\frac{Q}{a_r}$ ↑

Algorithm.

- Write $a_r(x) = \prod_i \left(1 - \frac{x}{\xi_i}\right)$, compute $\check{\xi}_i \approx |\xi_i|$ with $0 < \check{\xi}_i < |\xi_i|$.

  Then $\frac{1}{a_r} \ll \frac{1}{\check{a}_r}$ where $\check{a}_r(x) = \prod_i \left(1 - \frac{x}{\check{\xi}_i}\right)$.

Notation. Write $u \ll \hat{u}$ when $\forall n, \|u_n\| \leqslant \hat{u}_n$.

$u(x) = \sum_n u_n x^n$
$\hat{u}(x) = \sum_n \hat{u}_n x^n$

Lemma. If $u \ll \hat{u}$ and $v \ll \hat{v}$ for some norm(s) s.t. $\|u_i v_j\| \leqslant \|u_i\|\,\|v_j\|$, then $u\,v \ll \hat{u}\,\hat{v}$.

Majorants

$$Y' - A\,Y = 0$$

$$\underset{\uparrow\ \frac{P}{a_r}}{x\,R'(x) - x\,A(x)\,R(x)} = \underset{\uparrow\ \frac{Q}{a_r}}{\Phi(x)}$$

(tail of $Y(x)$: $\downarrow$)

Algorithm.

- Write $a_r(x) = \prod_i \left(1 - \frac{x}{\xi_i}\right)$, compute $\check{\xi}_i \approx |\xi_i|$ with $0 < \check{\xi}_i < |\xi_i|$.

  Then $\frac{1}{a_r} \ll \frac{1}{\check{a}_r}$ where $\check{a}_r(x) = \prod_i \left(1 - \frac{x}{\check{\xi}_i}\right)$.

- Compute $\hat{p} \gg P$ using the known coefficients of P.

Notation. Write $u \ll \hat{u}$ when $\forall n, \|u_n\| \leqslant \hat{u}_n$.

$u(x) = \sum_n u_n x^n$
$\hat{u}(x) = \sum_n \hat{u}_n x^n$

Lemma. If $u \ll \hat{u}$ and $v \ll \hat{v}$ for some norm(s) s.t. $\|u_i v_j\| \leqslant \|u_i\|\,\|v_j\|$, then $u\,v \ll \hat{u}\,\hat{v}$.

Majorants

$$Y' - A\,Y = 0$$

$$\underset{\text{tail of } Y(x)}{\downarrow}$$

$$x\,R'(x) - x\,A(x)\,R(x) = \Phi(x)$$

$$\frac{\hat{p}}{\check{a}_r} \gg \frac{P}{a_r} \qquad \frac{Q}{a_r}$$

Algorithm.

- Write $a_r(x) = \prod_i \left(1 - \frac{x}{\xi_i}\right)$, compute $\check{\xi}_i \approx |\xi_i|$ with $0 < \check{\xi}_i < |\xi_i|$.

  Then $\frac{1}{a_r} \ll \frac{1}{\check{a}_r}$ where $\check{a}_r(x) = \prod_i \left(1 - \frac{x}{\check{\xi}_i}\right)$.

- Compute $\hat{p} \gg P$ using the known coefficients of P.

Notation. Write $u \ll \hat{u}$ when $\forall n, \|u_n\| \leqslant \hat{u}_n$.

$u(x) = \sum_n u_n x^n$
$\hat{u}(x) = \sum_n \hat{u}_n x^n$

Lemma. If $u \ll \hat{u}$ and $v \ll \hat{v}$ for some norm(s) s.t. $\|u_i v_j\| \leqslant \|u_i\|\,\|v_j\|$, then $u\,v \ll \hat{u}\,\hat{v}$.

Majorants

$$Y' - A\,Y = 0$$

$$x\,R'(x) - x\,A(x)\,R(x) = \Phi(x)$$

(tail of $Y(x)$ ↓; A ↑; Φ ↑)

$$\hat{A} := \frac{\hat{p}}{\check{a}_r} \gg \frac{P}{a_r} \qquad \frac{Q}{a_r}$$

Algorithm.

- Write $a_r(x) = \prod_i \left(1 - \frac{x}{\xi_i}\right)$, compute $\check{\xi}_i \approx |\xi_i|$ with $0 < \check{\xi}_i < |\xi_i|$.

  Then $\frac{1}{a_r} \ll \frac{1}{\check{a}_r}$ where $\check{a}_r(x) = \prod_i \left(1 - \frac{x}{\check{\xi}_i}\right)$.

- Compute $\hat{p} \gg P$ using the known coefficients of P.

Notation. Write $u \ll \hat{u}$ when $\forall n, \|u_n\| \leqslant \hat{u}_n$.

$u(x) = \sum_n u_n x^n$
$\hat{u}(x) = \sum_n \hat{u}_n x^n$

Lemma. If $u \ll \hat{u}$ and $v \ll \hat{v}$ for some norm(s) s.t. $\|u_i v_j\| \leqslant \|u_i\|\,\|v_j\|$, then $u\,v \ll \hat{u}\,\hat{v}$.

Majorants

$$Y' - A\,Y = 0$$

$$\underset{\text{tail of } Y(x)}{\downarrow}$$

$$x\,R'(x) - x\,A(x)\,R(x) = \Phi(x)$$

$$\hat{A} := \frac{\hat{p}}{\check{a}_r} \gg \frac{P}{a_r} \qquad \frac{Q}{a_r} \ll \frac{\hat{q}}{\check{a}_r} =: \hat{\Phi}$$

Algorithm.

- Write $a_r(x) = \prod_i \left(1 - \frac{x}{\xi_i}\right)$, compute $\check{\xi}_i \approx |\xi_i|$ with $0 < \check{\xi}_i < |\xi_i|$.

  Then $\frac{1}{a_r} \ll \frac{1}{\check{a}_r}$ where $\check{a}_r(x) = \prod_i \left(1 - \frac{x}{\check{\xi}_i}\right)$.

- Compute $\hat{p} \gg P$ using the known coefficients of P. Same for Q.

Notation. Write $u \ll \hat{u}$ when $\forall n, \|u_n\| \leqslant \hat{u}_n$.

$u(x) = \sum_n u_n x^n$
$\hat{u}(x) = \sum_n \hat{u}_n x^n$

Lemma. If $u \ll \hat{u}$ and $v \ll \hat{v}$ for some norm(s) s.t. $\|u_i v_j\| \leqslant \|u_i\|\,\|v_j\|$, then $u\,v \ll \hat{u}\,\hat{v}$.

The method of majorants

[Cauchy, 1842]

We have

$$x\,R'(x) = x\,A(x)\,R(x) + \Phi(x).$$

Expanding in power series on both sides and identifying leads to

$$\begin{cases} Y_N = \dfrac{1}{N}\,\Phi_N \\ Y_n = \dfrac{1}{n}\left(\Phi_n + \displaystyle\sum_{i=0}^{n-1-N} A_i\,Y_{n-1-i}\right), \; n > N. \end{cases}$$

The method of majorants

[Cauchy, 1842]

We have

$$x\,R'(x) = x\,A(x)\,R(x) + \Phi(x).$$

Expanding in power series on both sides and identifying leads to

$$\begin{cases} Y_N = \dfrac{1}{N}\,\Phi_N \\ Y_n = \dfrac{1}{n}\left(\Phi_n + \displaystyle\sum_{i=0}^{n-1-N} A_i\,Y_{n-1-i}\right), \; n > N. \end{cases}$$

Consider the scalar first-order equation

$$x\,\hat{\psi}'(x) = x\,\hat{A}(x)\,\hat{\psi}(x) + \hat{\Phi}(x).$$

The method of majorants

[Cauchy, 1842]

We have

$$x\,R'(x) = x\,A(x)\,R(x) + \Phi(x).$$

Expanding in power series on both sides and identifying leads to

$$\begin{cases} Y_N = \dfrac{1}{N}\,\Phi_N \\ Y_n = \dfrac{1}{n}\left(\Phi_n + \displaystyle\sum_{i=0}^{n-1-N} A_i\,Y_{n-1-i}\right), \; n > N. \end{cases}$$

Consider the scalar first-order equation

$$x\,\hat{\psi}'(x) = x\,\hat{A}(x)\,\hat{\psi}(x) + \hat{\Phi}(x).$$

Expanding in power series on both sides and identifying leads to

$$\begin{cases} \hat{\psi}_N = \hat{\Phi}_N \\ \hat{\psi}_n = \left(\hat{\Phi}_n + \displaystyle\sum_{i=0}^{n-1-N} \hat{A}_i\,\hat{\psi}_{n-1-i}\right), \; n > N. \end{cases}$$

The method of majorants

[Cauchy, 1842]

We have

$$x\,R'(x) = x\,A(x)\,R(x) + \Phi(x).$$

Expanding in power series on both sides and identifying leads to

$$\begin{cases} Y_N = \dfrac{1}{N}\,\Phi_N \\ Y_n = \dfrac{1}{n}\left(\Phi_n + \displaystyle\sum_{i=0}^{n-1-N} A_i\,Y_{n-1-i}\right), \; n > N. \end{cases}$$

Consider the scalar first-order equation

$$x\,\hat{\psi}'(x) = x\,\hat{A}(x)\,\hat{\psi}(x) + \hat{\Phi}(x).$$

Expanding in power series on both sides and identifying leads to

$$\begin{cases} \hat{\psi}_N = \hat{\Phi}_N \\ \hat{\psi}_n = \left(\hat{\Phi}_n + \displaystyle\sum_{i=0}^{n-1-N} \hat{A}_i\,\hat{\psi}_{n-1-i}\right), \; n > N. \end{cases}$$

By induction, $\|Y_n\| \leqslant \hat{\psi}_n$ for all $n \geqslant N$.

I.e.: $R(x) \ll \hat{\psi}(x)$ where $\hat{\psi}(x)$ is the solution of $x\,\hat{\psi}'(x) = x\,\hat{A}(x)\,\hat{\psi}(x) + \hat{\Phi}(x)$ with $\hat{\psi}(0) = 0$.

Solving the majorant equation

$$\hat{A} = \frac{\hat{p}}{\check{a}} \qquad \hat{\Phi} = \frac{\hat{q}}{\check{a}}$$

The equation

$$x\,\hat{\psi}'(x) = x\,\hat{A}(x)\,\hat{\psi}(x) + \hat{\Phi}(x), \quad \hat{\psi}(0) = 0$$

admits a unique solution

$$\hat{\psi}(x) \;=\; h(x)\int_0^x \frac{t^{-1}\,\hat{\Phi}(t)}{h(t)}\,dt, \qquad h(x) \;=\; \exp\int_0^x \hat{A}(t)\,dt$$

Solving the majorant equation

$$\hat{A} = \frac{\hat{p}}{\check{a}} \qquad \hat{\Phi} = \frac{\hat{q}}{\check{a}}$$

The equation

$$x\,\hat{\psi}'(x) = x\,\hat{A}(x)\,\hat{\psi}(x) + \hat{\Phi}(x), \quad \hat{\psi}(0) = 0$$

admits a unique solution

$$\begin{aligned} \hat{\psi}(x) &= h(x) \int_0^x \frac{t^{-1}\,\hat{\Phi}(t)}{h(t)}\,dt, \qquad h(x) = \exp \int_0^x \hat{A}(t)\,dt \\ &= h(x) \int_0^x \frac{\hat{q}(t)}{\check{a}(t)\,h(t)}\,dt \qquad\qquad = \exp \int_0^x \frac{\hat{p}(t)}{\check{a}(t)}\,dt \end{aligned}$$

Solving the majorant equation

$$\hat{A} = \frac{\hat{p}}{\check{a}} \qquad \hat{\Phi} = \frac{\hat{q}}{\check{a}}$$

The equation

$$x\,\hat{\psi}'(x) = x\,\hat{A}(x)\,\hat{\psi}(x) + \hat{\Phi}(x), \quad \hat{\psi}(0) = 0$$

admits a unique solution

$$\begin{aligned} \hat{\psi}(x) &= h(x) \int_0^x \frac{t^{-1}\,\hat{\Phi}(t)}{h(t)}\,dt, & h(x) &= \exp \int_0^x \hat{A}(t)\,dt \\ &= h(x) \int_0^x \frac{\hat{q}(t)}{\check{a}(t)\,h(t)}\,dt & &= \exp \int_0^x \frac{\hat{p}(t)}{\check{a}(t)}\,dt \end{aligned}$$

This implies

$$\begin{aligned} \|Y_n\,\xi^n + Y_{n+1}\,\xi^{n+1} + \cdots\| &\leqslant \hat{\psi}(|\xi|) \\ &= \underbrace{\left(\exp \int_0^{|\xi|} \frac{\hat{p}(t)}{\check{a}(t)}\,dt\right)}_{\text{indep}(N)} \underbrace{\int_0^{|\xi|} \frac{\blacksquare\, t^{N-1} + \cdots + \blacksquare\, t^{N+d-1}}{\check{a}(t)\,h(t)}\,dt}_{\approx\text{first neglected term}} \end{aligned}$$

Remarks

Homogeneous version.

The same method works directly on the equation $Y' = A\,Y$ (no truncation, no Φ), yields a bound $Y(x) \ll \hat{Y}(x)$.

Typically much worse numerically.

Analyticity.

We have just proven the convergence of power series solutions!

(And thus the Cauchy existence theorem.)

Propagation of rounding errors.

To control the impact of rounding errors in the iterative computation of $(y_n)_n$, use a similar method with a series encoding the "local" errors on the rhs.

Conclusion

Summary

"Low" precision

Computing the sum $\approx$ computing the terms $\tilde{O}(r\,p^2)$ /solution
alt. $\tilde{O}(d\,p^2)$

"High" precision

Binary splitting + bit burst $\tilde{O}(d^\omega\,p)$ /fundamental matrix

Error bounds

Explicit residual + 1st order model (majorant) equation $\Rightarrow$ tail bound

Interval arithmetic does the rest

Some open questions

Algorithms and complexity

N terms in $N \log^{O(1)}(r) \log^{O(1)}(d)$ ops? — Eval in $d^{<2} p^{<3/2}$ bit ops?

Full complexity analysis as a function of the equation and evaluation point?

Fast implementations

Middle product — Practical FFT model over $\mathbb{Z}$ — Fast arithmetic at low precision

Alternative integration methods Rigorous and/or high-precision versions

Runge–Kutta methods — Multistep methods — Ehle's method — …

Improvements for big equations

Apparent singularities — Practical bounds for rounding errors — …