Interval summation of differentially finite series

Marc Mezzarobba

CNRS, Sorbonne Université

MAX seminar, April 28, 2020

The Problem

Compute an enclosure of $\sum_{n=0}^{N-1} u_n \zeta^n$ for a differentially finite $u(z)$.

diff. finite — $u(z) = \sum_{n=0}^{\infty} u_n z^n$ solution of a linear ODE

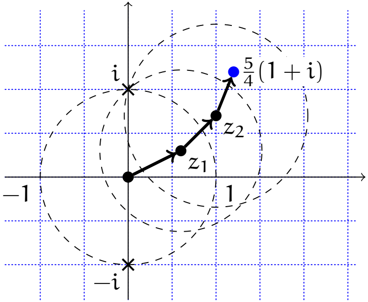
$$p_r(z)\, u^{(r)}(z) + \cdots + p_0(z)\, u(z) = 0, \quad p_k \in \mathbb{C}[z]$$

enclosure — return a interval containing the sum (rigorous bounds)

partial sum — truncation order given

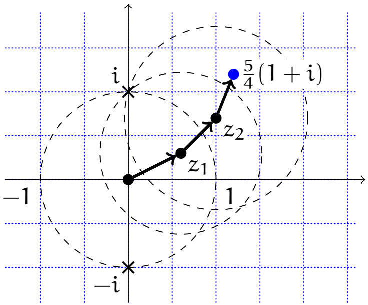
Basic brick of **Taylor methods** for ODEs with polynomial coefficients

Taylor Methods



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

Taylor Methods



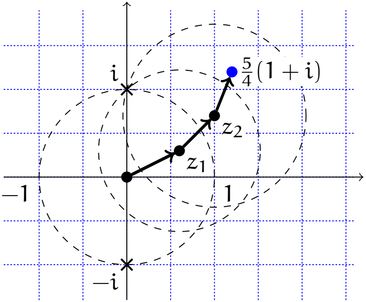
Too costly for classical scientific computing.

Better "from a computer algebra perspective":

- Arbitrary precision
- Rigorous error bounds
- Singular cases
- Complex "time" variables
- Value at a single point

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

Taylor Methods



Too costly for classical scientific computing.

Better "from a computer algebra perspective":

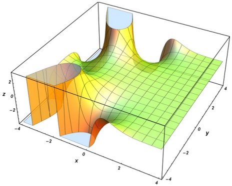
- Arbitrary precision
- Rigorous error bounds
- Singular cases
- Complex "time" variables
- Value at a single point

- Locally, the solutions are given by **convergent power series** (Cauchy)
- **Sum the series** numerically to get "initial values" at a new point
- Differentially finite case: **recurrences**

$$L(z, {}^{d}/_{dz}) \cdot u(z) = 0 \quad \Leftrightarrow \quad L(S^{-1}, S\,n) \cdot (u_n)_{n \in \mathbb{Z}} = 0$$

Applications

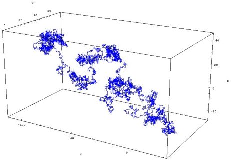
- **Special functions**



- **Combinatorics**

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

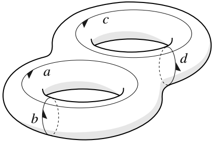
  random walks on lattices,
  asymptotics of P-recursive sequences…



- **Numerical (Real) Algebraic Geometry**

  via **Picard-Fuchs equations**

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



- **"Numerical differential algebra"**

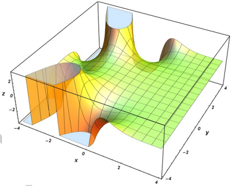
  via **connection / monodromy / Stokes matrices**

  operator factoring, heuristic diff. Galois groups
  [van der Hoeven 2007; Johansson-Kauers-M. 2013; …]

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

Applications

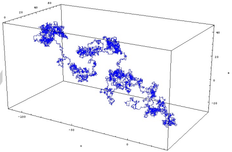
- **Special functions**



- **Combinatorics**

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

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

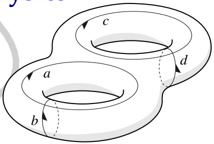


- **Numerical (Real) Algebraic Geometry**

  via **Picard-Fuchs equations**

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

Math. physics



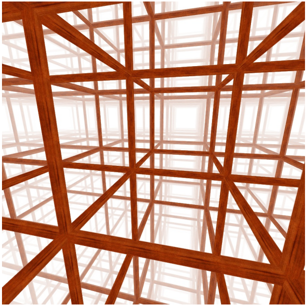
- **“Numerical differential algebra”**

  via **connection / monodromy / Stokes matrices**

  operator factoring, heuristic diff. Galois groups
  [van der Hoeven 2007; Johansson-Kauers-M. 2013; …]

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

Pólya Walks



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

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

where

#walks of length n
ending at origin

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

satisfies an LODE with polynomial coefficients

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

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

[thanks to B. Salvy]

First return after n steps:

$$f(z) = \sum_{n=0}^{\infty} f_n z^n$$

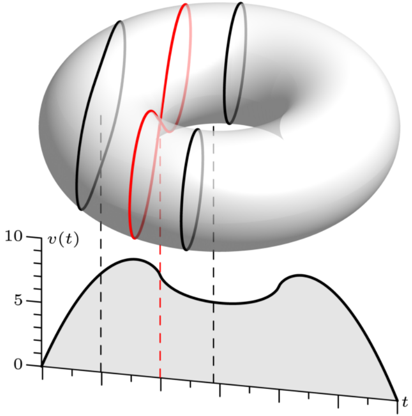
$$f\left(\frac{1}{2d}\right) = \sum_{n=0}^{\infty} \frac{f_n}{(2d)^n}$$

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

```
sage: from ore_algebra.examples import polya
sage: 1 - 1/polya.dop[10].numerical_solution([0]*9+[1], [0, 1/(2*10)], 1e-50).real()
[0.05619753597426778812097369256252412572131681661862 +/- 7.03e-51]
```

Volumes of Compact Semi-Algebraic Sets

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



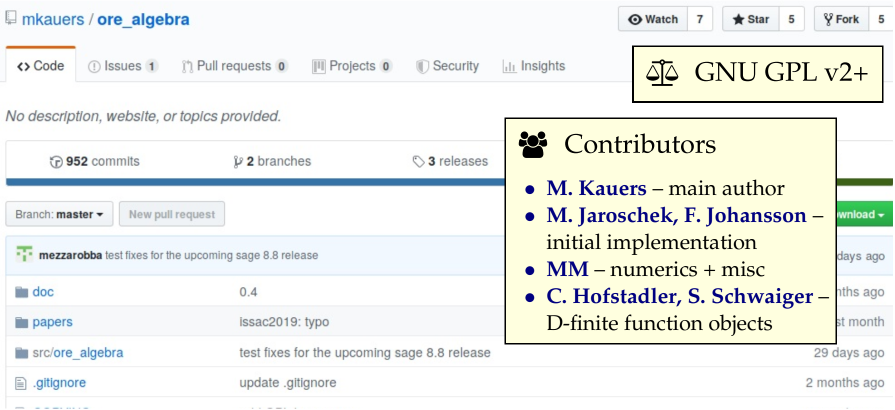
- The **“slice volume”** function satisfies a Picard-Fuchs eqn
- Except at critical values of the projection, it is analytic

➙ Compute initial values by recursive calls, integrate the equation

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

```
···· slice #2: ρ = 10866099/4849664
····· slice length = [3.95699242690042041342397892533404623584614411033674866606926914003 +/- 5.52e-66]
···· integrating PF equation over [1.010906176264399?, 2.989093823735602?]...
···· ...piece volume = [8.10844587166147220133178843300791539013253760904431939702317 34 +/- 8.50e-62]
··· slice volume = [24.85863912287043868696646961582254943981378134071631307423220 +/- 5.78e-60]
·· integrating PF equation over [-1, 1]...
·· ...piece volume = [39.4784176043574344753379639995046045412547976289631625 06 +/- 6.38e-55]
[39.4784176043574344753379639995046045412547976289631625 06 +/- 6.38e-55]
```

ore_algebra



⚖ GNU GPL v2+

Contributors

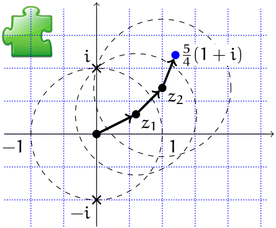
- **M. Kauers** – main author
- **M. Jaroschek, F. Johansson** – initial implementation
- **MM** – numerics + misc
- **C. Hofstadler, S. Schwaiger** – D-finite function objects

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

Try it online at

http://marc.mezzarobba.net/oaademo

Main Ingredients

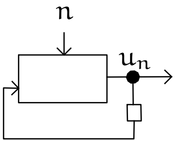


Taylor method

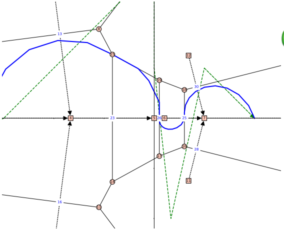
$$\sum_{\nu \in \lambda + \mathbb{Z}} \sum_{k=0}^{K} y_{\nu,k} \, z^{\nu} \frac{\log(z)^k}{k!}$$

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

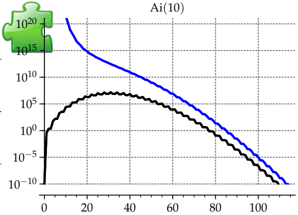
Logarithmic series



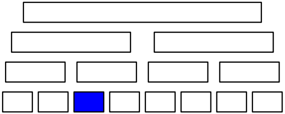
Recurrences



Path optimization



Error bounds



Binary splitting

Arbitrary Precision: Complexity vs Overhead

	***Approx.* cost**
Classical numerical analysis, *e.g.* RK4	$\mathrm{tiny} \cdot r\,s\,2^{\mathbf{p}/4}$
Taylor method, direct summation	$r\,s\,\mathbf{p}^2$
"Nonscalar" methods [Smith, Johansson…]	$r\,s^{\omega}\,(\mathbf{p}^{3/2} + \mathrm{tiny} \cdot \mathbf{p}^2)$
Fast multipoint evaluation	$r\,s^{\omega}\,\mathbf{p}^{3/2}$
Binary splitting [Schroeppel, Chudnovsky & Chudnovsky…]	$r^{\omega}\,s^{\omega}\,\mathbf{p}$

$r =$ diff. eq. order, $s =$ rec. order, $p =$ target accuracy in bits

target accuracy $p \Rightarrow$ #terms to sum $\approx p$

Arbitrary Precision: Complexity vs Overhead

	***Approx.* cost**
Classical numerical analysis, *e.g.* RK4	$\mathrm{tiny} \cdot r\, s\, 2^{\mathbf{p}/4}$
Taylor method, direct summation	$r\, s\, \mathbf{p}^2$
"Nonscalar" methods [Smith, Johansson…]	$r\, s^{\omega}\, (\mathbf{p}^{3/2} + \mathrm{tiny} \cdot \mathbf{p}^2)$
Fast multipoint evaluation	$r\, s^{\omega}\, \mathbf{p}^{3/2}$
Binary splitting [Schroeppel, Chudnovsky & Chudnovsky…]	$r^{\omega}\, s^{\omega}\, \mathbf{p}$

moderate prec wrt equation size

$r =$ diff. eq. order, $s =$ rec. order, $p =$ target accuracy in bits

target accuracy $p \Rightarrow$ #terms to sum $\approx p$

The Problem

Compute an enclosure of $\sum_{n=0}^{N-1} u_n\, \zeta^n$ for a differentially finite $u(z)$.

Input:

$$\left.\begin{array}{rll} L \in \mathbb{IC}[z]\langle {}^{d}/_{dz} \rangle & — & \text{differential operator} \\ u_{0:r-1} \in \mathbb{IC} & — & \text{initial values} \end{array}\right\} u(z) = \sum_{n=0}^{\infty} u_n\, z^n$$

$$\begin{array}{rll} \zeta \in \mathbb{IC} & — & \text{evaluation point} \\ N \in \mathbb{N} & — & \text{truncation order} \\ p \in \mathbb{N} & — & \text{target precision} \end{array}$$

Output: $y \in \mathbb{IC}$ — interval $\ni u_{:N}(\zeta)$ of width $\approx 2^{-p}$

Assumptions:

$$\begin{array}{rll} \text{ordinary point} & — & a_r(0) \neq 0 \\ \text{“obviously” cvgt} & — & |\zeta| < \min\{|\xi| : a_r(\xi) = 0\} \\ \text{geometric cvgce} & — & N = \Theta(p) \end{array}$$

$$L = a_r(z) \left(\frac{d}{dz}\right)^r + \cdots$$

Recurrences in Interval Arithmetic

Recurrence step: (naïve interval arithmetic, with $u_n \approx \Theta(1)$)

$$\mathbf{u_n} = \frac{-1}{b_s(n)} \left[b_{s-1}(n)\, \mathbf{u_{n-1}} + \cdots + b_1(n)\, \mathbf{u_{n-s+1}} + b_0(n)\, \mathbf{u_{n-s}} \right]$$

$$\operatorname{rad} \approx \rho$$

$$\operatorname{rad} \gtrsim \left(\sum_i \left| \frac{b_i(s)}{b_s(n)} \right| \right) \rho \gtrapprox s\, \rho$$

$\operatorname{rad}(\mathbf{u_n}) = 2^{\Theta(n)}$ $\qquad$ $\operatorname{rad}(\sum^N u_n\, \zeta^n) = 2^{\Theta(N)}$ (unless ζ small)

Accuracy target 2^{-p} $\quad \Rightarrow \quad$ Need $\mathbf{\Omega(p)}$ guard bits

(This is not a numerical stability issue.)

A Toy Example

[Boldo 2009]

$$c_{n+1} = 2\,c_n - c_{n-1} \qquad (c_0 = \diamond(1/3),\, c_{-1} = 0)$$

$n =$	Interval	Floating-Point
0	$[0.33333333333333333 \pm 1.49e-17]$	0.333333333333333
5	$[2.00000000000000 \pm 3.78e-15]$	2.00000000000000
10	$[3.666666666667 \pm 5.74e-13]$	3.66666666666667
15	$[5.3333333333 \pm 5.29e-11]$	5.33333333333334
20	$[7.00000000 \pm 1.60e-9]$	7.00000000000001
25	$[8.666667 \pm 4.65e-7]$	8.66666666666668
30	$[10.3333 \pm 4.41e-5]$	10.3333333333333
35	$[12.000 \pm 8.82e-4]$	12.0000000000000
40	$[1.4e+1 \pm 0.406]$	13.6666666666667
45	$[\pm 21.3]$	15.3333333333334
50	$[\pm 5.04e+2]$	17.000000000000

Naïve Error Analysis

(absolute error / fixed-point arithmetic)

$$
\begin{aligned}
c_{n+1} &= 2\,c_n - c_{n-1} \\
\tilde{c}_{n+1} &= \diamond(2\,\tilde{c}_n - \tilde{c}_{n-1}) \\
&= 2\,\tilde{c}_n - \tilde{c}_{n-1} + \varepsilon_n \qquad |\varepsilon_n| \leqslant \mathbf{u}
\end{aligned}
$$

Naïve Error Analysis

(absolute error / fixed-point arithmetic)

$$\begin{aligned} c_{n+1} &= 2\,c_n - c_{n-1} \\ \tilde{c}_{n+1} &= \diamond(2\,\tilde{c}_n - \tilde{c}_{n-1}) \\ &= 2\,\tilde{c}_n - \tilde{c}_{n-1} + \varepsilon_n \qquad |\varepsilon_n| \leqslant \mathbf{u} \end{aligned}$$

$$|\tilde{c}_{n+1} - c_{n+1}| \leqslant 2\,|\tilde{c}_n - c_n| + |\tilde{c}_{n-1} - c_{n-1}| + \mathbf{u}$$

Naïve Error Analysis

(absolute error / fixed-point arithmetic)

$$\begin{aligned} c_{n+1} &= 2\,c_n - c_{n-1} \\ \tilde{c}_{n+1} &= \diamond(2\,\tilde{c}_n - \tilde{c}_{n-1}) \\ &= 2\,\tilde{c}_n - \tilde{c}_{n-1} + \varepsilon_n \qquad |\varepsilon_n| \leqslant \mathbf{u} \end{aligned}$$

$$|\tilde{c}_{n+1} - c_{n+1}| \leqslant 2\,|\tilde{c}_n - c_n| + |\tilde{c}_{n-1} - c_{n-1}| + \mathbf{u}$$

- induction : $|\tilde{c}_n - c_n| \leqslant 3^n\,\mathbf{u}$

Naïve Error Analysis

(absolute error / fixed-point arithmetic)

$$\begin{aligned} c_{n+1} &= 2\,c_n - c_{n-1} \\ \tilde{c}_{n+1} &= \diamond(2\,\tilde{c}_n - \tilde{c}_{n-1}) \\ &= 2\,\tilde{c}_n - \tilde{c}_{n-1} + \varepsilon_n \qquad |\varepsilon_n| \leqslant \mathbf{u} \end{aligned}$$

$$|\tilde{c}_{n+1} - c_{n+1}| \leqslant 2\,|\tilde{c}_n - c_n| + |\tilde{c}_{n-1} - c_{n-1}| + \mathbf{u}$$

- induction : $|\tilde{c}_n - c_n| \leqslant 3^n\,\mathbf{u}$ 

Naïve Error Analysis

(absolute error / fixed-point arithmetic)

$$\begin{aligned} c_{n+1} &= 2\,c_n - c_{n-1} \\ \tilde{c}_{n+1} &= \diamond(2\,\tilde{c}_n - \tilde{c}_{n-1}) \\ &= 2\,\tilde{c}_n - \tilde{c}_{n-1} + \varepsilon_n \qquad |\varepsilon_n| \leqslant \mathbf{u} \end{aligned}$$

$$|\tilde{c}_{n+1} - c_{n+1}| \leqslant 2\,|\tilde{c}_n - c_n| + |\tilde{c}_{n-1} - c_{n-1}| + \mathbf{u}$$

- induction : $|\tilde{c}_n - c_n| \leqslant 3^n\,\mathbf{u}$
- slightly sharper estimate :

$$|\tilde{c}_n - c_n| \leqslant \frac{(1+\sqrt{2})^n + (1-\sqrt{2})^n - 2}{4}\,\mathbf{u} \approx 2.4^n\,\mathbf{u}$$

Naïve Error Analysis

(absolute error / fixed-point arithmetic)

$$
\begin{aligned}
c_{n+1} &= 2\,c_n - c_{n-1} \\
\tilde{c}_{n+1} &= \diamond(2\,\tilde{c}_n - \tilde{c}_{n-1}) \\
&= 2\,\tilde{c}_n - \tilde{c}_{n-1} + \varepsilon_n \qquad |\varepsilon_n| \leqslant \mathbf{u}
\end{aligned}
$$

$$|\tilde{c}_{n+1} - c_{n+1}| \leqslant 2\,|\tilde{c}_n - c_n| + |\tilde{c}_{n-1} - c_{n-1}| + \mathbf{u}$$

- induction : $|\tilde{c}_n - c_n| \leqslant 3^n\,\mathbf{u}$
- slightly sharper estimate :

$$|\tilde{c}_n - c_n| \leqslant \frac{(1+\sqrt{2})^n + (1-\sqrt{2})^n - 2}{4}\,\mathbf{u} \approx 2.4^n\,\mathbf{u}$$

(This is essentially what an interval evaluation does.)

The Same Analysis Done Right

(fixed-point)

$$
\begin{aligned}
c_{n+1} &= 2\,c_n - c_{n-1} \\
\tilde{c}_{n+1} &= \diamond(2\,\tilde{c}_n - \tilde{c}_{n-1}) \\
&= 2\,\tilde{c}_n - \tilde{c}_{n-1} + \varepsilon_n \qquad |\varepsilon_n| \leqslant \mathbf{u}
\end{aligned}
$$

$$\delta_n = \tilde{c}_n - c_n$$

$$\underbrace{\delta_{n+1}}_{\textit{global error}} = 2\,\delta_n - \delta_{n-1} + \underbrace{\varepsilon_n}_{\textit{local error}} \qquad (\delta_0 = \delta_1 = 0)$$

The Same Analysis Done Right

(fixed-point)

$$
\begin{aligned}
c_{n+1} &= 2\,c_n - c_{n-1} \\
\tilde{c}_{n+1} &= \diamond(2\,\tilde{c}_n - \tilde{c}_{n-1}) \\
&= 2\,\tilde{c}_n - \tilde{c}_{n-1} + \varepsilon_n \qquad |\varepsilon_n| \leqslant \mathbf{u}
\end{aligned}
$$

$$\delta_n = \tilde{c}_n - c_n$$

$$\delta_{n+1} = 2\,\delta_n - \delta_{n-1} + \varepsilon_n \qquad (\delta_0 = \delta_1 = 0)$$

global error *local error*

- $\delta_n = \sum_{k=1}^{n-1} k\,\varepsilon_{n-k}$

  $|\delta_n| \leqslant \dfrac{n\,(n-1)}{2}\,\mathbf{u}$

The Same Analysis Done Right

(fixed-point)

$$
\begin{aligned}
c_{n+1} &= 2\,c_n - c_{n-1} \\
\tilde{c}_{n+1} &= \diamond(2\,\tilde{c}_n - \tilde{c}_{n-1}) \\
&= 2\,\tilde{c}_n - \tilde{c}_{n-1} + \varepsilon_n \qquad |\varepsilon_n| \leqslant \mathbf{u}
\end{aligned}
$$

$$\delta_n = \tilde{c}_n - c_n$$

$$\delta_{n+1} = 2\,\delta_n - \delta_{n-1} + \varepsilon_n \qquad (\delta_0 = \delta_1 = 0)$$

global error *local error*

- $\displaystyle \delta_n = \sum_{k=1}^{n-1} k\,\varepsilon_{n-k}$

  $\displaystyle |\delta_n| \leqslant \frac{n\,(n-1)}{2}\,\mathbf{u}$

General Case

$$
\begin{aligned}
\mathbf{u_n} &= \frac{-1}{b_s(n)}\left[b_{s-1}(n)\,\mathbf{u_{n-1}} + \cdots + b_1(n)\,\mathbf{u_{n-s+1}} + b_0(n)\,\mathbf{u_{n-s}}\right] \\
\tilde{\mathbf{u}}_{\mathbf{n}} &= \frac{-1}{b_s(n)}\left[b_{s-1}(n)\,\tilde{\mathbf{u}}_{\mathbf{n-1}} + \cdots + b_1(n)\,\tilde{\mathbf{u}}_{\mathbf{n-s+1}} + b_0(n)\,\tilde{\mathbf{u}}_{\mathbf{n-s}}\right] + \varepsilon_n
\end{aligned}
$$

$\tilde{u}_n =$ computed sequence (e.g. floating-point)

local error,
known bound $|\varepsilon_n| \leqslant \hat{\varepsilon}_n$

The global error $\delta_n = \tilde{u}_n - u_n$ satisfies

$$
b_s(n)\,\delta_n + b_{s-1}(n)\,\delta_{n-1} + \cdots + b_0(n)\,\delta_{n-s} = b_s(n)\,\varepsilon_n
$$

General Case

$$\mathbf{u_n} = \frac{-1}{b_s(n)}\left[b_{s-1}(n)\,\mathbf{u_{n-1}} + \cdots + b_1(n)\,\mathbf{u_{n-s+1}} + b_0(n)\,\mathbf{u_{n-s}}\right]$$

$$\tilde{\mathbf{u}}_{\mathbf{n}} = \frac{-1}{b_s(n)}\left[b_{s-1}(n)\,\tilde{\mathbf{u}}_{\mathbf{n-1}} + \cdots + b_1(n)\,\tilde{\mathbf{u}}_{\mathbf{n-s+1}} + b_0(n)\,\tilde{\mathbf{u}}_{\mathbf{n-s}}\right] + \varepsilon_n$$

$\tilde{u}_n =$ computed sequence (e.g. floating-point)

local error,
known bound $|\varepsilon_n| \leqslant \hat{\varepsilon}_n$

The global error $\delta_n = \tilde{u}_n - u_n$ satisfies

$$b_s(n)\,\delta_n + b_{s-1}(n)\,\delta_{n-1} + \cdots + b_0(n)\,\delta_{n-s} = b_s(n)\,\varepsilon_n$$

Therefore:

$$a_r(z)\,\delta^{(r)}(z) + \cdots + a_1(z)\,\delta'(z) + a_0(z)\,\delta(z) = Q(z\,{}^{d}\!/_{dz})\,\varepsilon(z)$$

$$\delta(z) = \sum_n \delta_n z^n, \quad \varepsilon(z) = \sum_n \varepsilon_n z^n$$

Compute a **bound** on δ_n given one on ε_n?

The Majorant Method

[Cauchy 1842]

“Bound” an implicit equation whose series solutions can be determined iteratively by a simpler “model equation”

$$Y'(z) = A(z)\,Y(z) + B(z)$$

$$A(z), B(z) \in \mathbb{C}[[z]]^{r\times r}$$

The Majorant Method

[Cauchy 1842]

"Bound" an implicit equation whose series solutions can be determined iteratively by a simpler "model equation"

$$Y'(z) = A(z)\,Y(z) + B(z)$$

$$A(z), B(z) \in \mathbb{C}[[z]]^{r\times r}$$

$$(n+1)\,Y_n = \sum_{i+j=n} A_i\,Y_j + B_n$$

The Majorant Method

[Cauchy 1842]

“Bound” an implicit equation whose series solutions can be determined iteratively by a simpler “model equation”

$$Y'(z) = A(z)\,Y(z) + B(z)$$

$$A(z), B(z) \in \mathbb{C}[[z]]^{r\times r}$$

$$(n+1)\,Y_n = \sum_{i+j=n} A_i\,Y_j + B_n$$

$$\|A_i\| \leqslant \hat{a}_i \qquad \|B_n\| \leqslant \hat{b}_n$$

The Majorant Method

[Cauchy 1842]

"Bound" an implicit equation whose series solutions can be determined iteratively by a simpler "model equation"

$$Y'(z) = A(z)\,Y(z) + B(z)$$

$$A(z), B(z) \in \mathbb{C}[[z]]^{r\times r}$$

$$(n+1)\,Y_n = \sum_{i+j=n} A_i\,Y_j + B_n \qquad\qquad (n+1)\,\hat{y}_n = \sum_{i+j=n} \hat{a}_i\,\hat{y}_j + \hat{b}_n$$

$$\|A_i\| \leqslant \hat{a}_i \qquad \|B_n\| \leqslant \hat{b}_n$$

The Majorant Method

[Cauchy 1842]

"Bound" an implicit equation whose series solutions can be determined iteratively by a simpler "model equation"

$$Y'(z) = A(z)\,Y(z) + B(z)$$

$$A(z), B(z) \in \mathbb{C}[[z]]^{r\times r}$$

$$\hat{y}'(z) = \hat{a}(z)\,\hat{y}(z) + \hat{b}(z)$$

(1st order scalar eq.!)

$$(n+1)\,Y_n = \sum_{i+j=n} A_i\,Y_j + B_n$$

$$\|A_i\| \leqslant \hat{a}_i \qquad \|B_n\| \leqslant \hat{b}_n$$

$$(n+1)\,\hat{y}_n = \sum_{i+j=n} \hat{a}_i\,\hat{y}_j + \hat{b}_n$$

The Majorant Method

[Cauchy 1842]

"Bound" an implicit equation whose series solutions can be determined iteratively by a simpler "model equation"

$$Y'(z) = A(z)\,Y(z) + B(z) \qquad A(z), B(z) \in \mathbb{C}[[z]]^{r\times r}$$

$$\hat{y}'(z) = \hat{a}(z)\,\hat{y}(z) + \hat{b}(z)$$

(1st order scalar eq.!)

$$(n+1)\,Y_n = \sum_{i+j=n} A_i\,Y_j + B_n \qquad (n+1)\,\hat{y}_n = \sum_{i+j=n} \hat{a}_i\,\hat{y}_j + \hat{b}_n$$

$$\|A_i\| \leqslant \hat{a}_i \qquad \|B_n\| \leqslant \hat{b}_n$$

$"f \ll \hat{f}" \mathrel{\hat{=}}$ coeffwise $\|f_n\| \leqslant \hat{f}_n$

$$A(z) \ll \hat{a}(z), \quad B(z) \ll \hat{b}(z), \quad \|Y_0\| \leqslant \hat{y}_0 \quad \Rightarrow \quad Y(z) \ll \hat{y}(z)$$

- $\hat{a}(z)$ easily computable if $A(z) \in \mathbb{C}(z)^{r\times r}$

Global Error

$$a_r(z)\,\delta^{(r)}(z) + \cdots + a_0(z)\,\delta(z) = Q(z\,{}^{d}\!/_{dz})\,\varepsilon(z) \quad \xrightarrow{\text{maj.}} \quad \hat{\delta}'(z) = \hat{a}(z)\,\hat{\delta}(z) + \hat{\varepsilon}(z)$$

$$|\delta_0|, \ldots, |\delta_{r-1}| \leqslant \hat{\delta}_0 \quad \Rightarrow \quad \delta(z) \ll \hat{\delta}(z)$$

Global Error

$$a_r(z)\,\delta^{(r)}(z)+\cdots+a_0(z)\,\delta(z)=Q(z\,{}^{d}\!/_{dz})\,\varepsilon(z)\quad\xrightarrow{\text{maj.}}\quad\hat\delta'(z)=\hat a(z)\,\hat\delta(z)+\hat\varepsilon(z)$$

$$|\delta_0|,\ldots,|\delta_{r-1}|\leqslant\hat\delta_0\quad\Rightarrow\quad\delta(z)\ll\hat\delta(z)$$

$$\hat\delta(z)=\hat h(z)\left(\mathrm{cst}+\int_0^z\frac{\hat\varepsilon(w)}{\hat h(w)}\,dw\right),\qquad\hat h(z)=\exp\int_0^z\hat a(z)\,dz$$

Global Error

$$a_r(z)\,\delta^{(r)}(z)+\cdots+a_0(z)\,\delta(z)=Q(z\,{}^{d}\!/\!_{dz})\,\varepsilon(z) \quad \xrightarrow{\text{maj.}} \quad \hat\delta'(z)=\hat a(z)\,\hat\delta(z)+\hat\varepsilon(z)$$

$$|\delta_0|,\ldots,|\delta_{r-1}|\leqslant\hat\delta_0 \quad\Rightarrow\quad \delta(z)\ll\hat\delta(z)$$

$$\hat\delta(z)=\hat h(z)\left(\cancel{\mathrm{cst}}+\int_0^z\frac{\hat\varepsilon(w)}{\hat h(w)}\,dw\right), \qquad \hat h(z)=\exp\int_0^z\hat a(z)\,dz$$

Global Error

$$a_r(z)\,\delta^{(r)}(z)+\cdots+a_0(z)\,\delta(z)=Q(z\,{}^{d}\!/_{dz})\,\varepsilon(z)\quad\xrightarrow{\text{maj.}}\quad\hat{\delta}'(z)=\hat{a}(z)\,\hat{\delta}(z)+\hat{\varepsilon}(z)$$

$$|\delta_0|,\ldots,|\delta_{r-1}|\leqslant\hat{\delta}_0\quad\Rightarrow\quad\delta(z)\ll\hat{\delta}(z)$$

$$\hat{\delta}(z)=\hat{h}(z)\left(\cancel{\mathrm{cst}}+\int_0^z\frac{\hat{\varepsilon}(w)}{\hat{h}(w)}\,dw\right),\qquad\hat{h}(z)=\exp\int_0^z\hat{a}(z)\,dz$$

take $\hat{\varepsilon}_n=\bar{\varepsilon}\,\hat{h}_n$ $\qquad$ $\bar{\varepsilon}\lesssim n^r\,\mathbf{u}$ since $|u_n|\lesssim\hat{h}_n$

Global Error

$$a_r(z)\,\delta^{(r)}(z)+\cdots+a_0(z)\,\delta(z)=Q(z\,{}^{\mathrm{d}}\!/_{\mathrm{d}z})\,\varepsilon(z) \quad \xrightarrow{\text{maj.}} \quad \hat{\delta}'(z)=\hat{a}(z)\,\hat{\delta}(z)+\hat{\varepsilon}(z)$$

$$|\delta_0|,\ldots,|\delta_{r-1}|\leqslant\hat{\delta}_0 \quad \Rightarrow \quad \delta(z)\ll\hat{\delta}(z)$$

$$\hat{\delta}(z)=\hat{h}(z)\left(\cancel{\mathrm{cst}}+\int_0^z\frac{\hat{\varepsilon}(w)}{\hat{h}(w)}\,\mathrm{d}w\right), \qquad \hat{h}(z)=\exp\int_0^z\hat{a}(z)\,\mathrm{d}z$$

take $\hat{\varepsilon}_n=\bar{\varepsilon}\,\hat{h}_n$ $\qquad$ $\bar{\varepsilon}\lesssim n^r\,\mathbf{u}$ since $|u_n|\lesssim\hat{h}_n$

$$|\tilde{u}(\zeta)-u(\zeta)|\leqslant\hat{\delta}(|\zeta|)=O(\bar{\varepsilon})$$

$$\#\text{guard bits}=o(p) \quad \text{for fixed L, ini, }\zeta$$

Global Error

$$a_r(z)\,\delta^{(r)}(z)+\cdots+a_0(z)\,\delta(z)=Q(z\,{}^{d}\!/_{dz})\,\varepsilon(z) \quad \xrightarrow{\text{maj.}} \quad \hat\delta'(z)=\hat a(z)\,\hat\delta(z)+\hat\varepsilon(z)$$

$$|\delta_0|,\ldots,|\delta_{r-1}|\leqslant\hat\delta_0 \quad\Rightarrow\quad \delta(z)\ll\hat\delta(z)$$

$$\hat\delta(z)=\hat h(z)\left(\mathrm{cst}+\int_0^z\frac{\hat\varepsilon(w)}{\hat h(w)}\,dw\right),\qquad \hat h(z)=\exp\int_0^z\hat a(z)\,dz$$

take $\hat\varepsilon_n=\bar\varepsilon\,\hat h_n$ $\qquad$ $\bar\varepsilon\lesssim n^r\,\mathbf{u}$ since $|u_n|\lesssim\hat h_n$

$$|\tilde u(\zeta)-u(\zeta)|\leqslant\hat\delta(|\zeta|)=O(\bar\varepsilon)$$

$\#$guard bits $=o(p)$ $\quad$ for fixed L, ini, ζ

The same computation yields a bound on the **truncation** error!

(Replace $\varepsilon(z)$ by a residual accounting for the negelected tail.)

Practical Issues

- The Cauchy majorants are *far* too coarse
  - Use sharper variants [M. 2019]

- Simple majorants cannot be sharp for small n
  - Switch from interval summation to running error analysis

- Need to choose the working precision ($\leftrightarrow$ cutoff point) in advance
  - Heuristics based on asymptotics…

- Good choice of $\hat{\varepsilon}(z)$ (tight & easily computable) not clear

Current status: works well for (some) large equations met in practice, but sometimes slower than naïve interval summation.

Closed-form Bounds by the same technique

Legendre Polynomials

- $\mathbf{P_{n+1}(x)} = \frac{1}{n+1}\left[(2n+1)\,x\,\mathbf{P_n(x)} - n\,\mathbf{P_{n-1}(x)}\right]$
- In fixed-point arithmetic:

$$|\tilde{p}_n - P_n(x)| \leqslant \frac{3}{4}(n+1)(n+2)\,\mathbf{u} \qquad (-1 \leqslant x \leqslant 1)$$

Relevant for the fast computation of Gauss-Legendre quadrature rules [Johansson-M. 2018]

Bernoulli Numbers

- $\mathbf{b_k} = \frac{1}{(2k)!\,4^k} - \sum_{j=0}^{k-1} \frac{\mathbf{b_j}}{(2k+1-2j)!\,4^{k-j}}, \qquad b_k = \frac{B_{2k}}{(2k)!}$
- In binary floating-point arithmetic: $\tilde{b}_k = b_k\,(1+\eta_k)$ where

$$|\eta_k| \leqslant c_1\,k\,(1+c_2\,\mathbf{u})^k = \text{“}O(k\,\mathbf{u})\text{”}$$

Answers a question of P. Zimmermann based on work of Brent and Harvey

Generating series + Cauchy majorants

$\Rightarrow$ Simple & general automatic running error analysis
of the summation of D-finite series

Same technique yields tight closed-form error bounds for related problems

General context: arbitrary-precision integration of linear ODEs with poly. coeff.

Has anyone seen this technique in the literature?

- Make it (more) practical!
- Regular singular case
- Error analysis based on generating series:
  Backward recurrences? RK methods? Beyond recurrences?

Code available at

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

Image Credits

- « Work in progress » icon: https://commons.wikimedia.org/wiki/File:Work_in_progress_icon.svg

-