A symbolic-numeric validation algorithm for linear ODEs with Newton-Picard method

Florent Bréhard[1], Nicolas Brisebarre[2], Mioara Joldes[3]

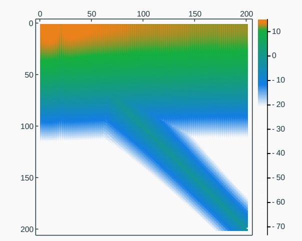
ACA 2023, Warsaw

July 20, 2023

[1] CRIStAL, CNRS, Université de Lille

[2] LIP, CNRS, École Normale Supérieure de Lyon

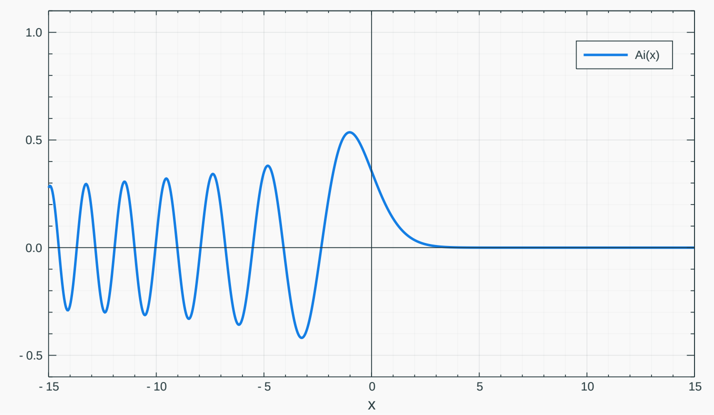
[3] LAAS-CNRS, Toulouse



Introduction

An Example: Airy Function

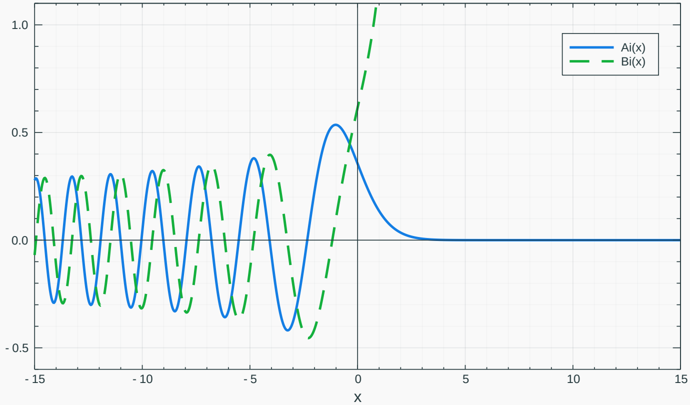
- $\text{Ai}(x) = \frac{1}{\pi} \int_{0}^{+\infty} \cos\left(\frac{t^3}{3} + xt\right) dt$
- Linear Initial Value Problem: $y''(x) - xy(x) = 0, \quad \begin{cases} y(0) = \dfrac{1}{3^{\frac{2}{3}} \Gamma\left(\frac{2}{3}\right)} \\ y'(0) = -\dfrac{1}{3^{\frac{1}{3}} \Gamma\left(\frac{1}{3}\right)} \end{cases}$



An Example: Airy Function

- $\operatorname{Ai}(x) = \dfrac{1}{\pi} \displaystyle\int_0^{+\infty} \cos\left(\frac{t^3}{3} + xt\right) dt$

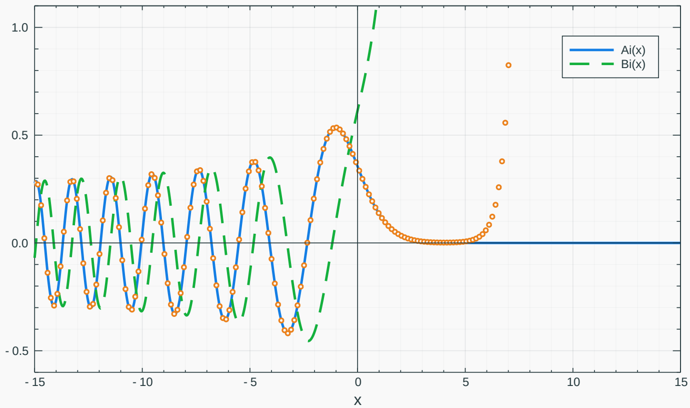
- Linear Initial Value Problem: $y''(x) - xy(x) = 0, \quad \begin{cases} y(0) = \dfrac{1}{3^{\frac{2}{3}}\Gamma\left(\frac{2}{3}\right)} \\ y'(0) = -\dfrac{1}{3^{\frac{1}{3}}\Gamma\left(\frac{1}{3}\right)} \end{cases}$



An Example: Airy Function

- $\mathrm{Ai}(x) = \dfrac{1}{\pi} \displaystyle\int_{0}^{+\infty} \cos\left(\frac{t^3}{3} + xt\right) dt$

- Linear Initial Value Problem: $\quad y''(x) - xy(x) = 0, \quad \begin{cases} y(0) = \dfrac{1}{3^{\frac{2}{3}}\,\Gamma\left(\frac{2}{3}\right)} \\ y'(0) = -\dfrac{1}{3^{\frac{1}{3}}\,\Gamma\left(\frac{1}{3}\right)} \end{cases}$



A Brief Review of Functional Analytic Validation Methods for ODEs

- Higher order methods proposed in the literature:
  - *Makino & Berz (1998–), CAPD, VNODE-LP:* Taylor methods
  - *Ultra-arithmetic (1982–), Benoit & Joldes & Mezzarobba (2017), Dzetkulič (2015) :* Picard iterations
  - *Zgliczynski (2002), Wilczak & Zgliczynski (2011):* Lohner methods
  - *Kedem (1981), Plum (1991):* Resolvent kernel/Green function based methods
  - *Mezzarobba (2011):* Majorant series (for D-finite functions)
  - *Yamamoto (1998), Lessard & Mireles James & van den Berg & al. (2014–), Bréhard & Brisebarre & Joldes (2018):* Newton-like a posteriori validated spectral methods
    $\rightarrow$ a.k.a. Newton-Galerkin

A Brief Review of Functional Analytic Validation Methods for ODEs

- Higher order methods proposed in the literature:
  - *Makino & Berz (1998–), CAPD, VNODE-LP:* Taylor methods
  - *Ultra-arithmetic (1982–), Benoit & Joldes & Mezzarobba (2017), Dzetkulič (2015) :* Picard iterations
  - *Zgliczynski (2002), Wilczak & Zgliczynski (2011):* Lohner methods
  - *Kedem (1981), Plum (1991):* Resolvent kernel/Green function based methods
  - *Mezzarobba (2011):* Majorant series (for D-finite functions)
  - *Yamamoto (1998), Lessard & Mireles James & van den Berg & al. (2014–), Bréhard & Brisebarre & Joldes (2018):* Newton-like a posteriori validated spectral methods
    $\rightarrow$ a.k.a. Newton-Galerkin

- Comparison of Newton-Picard and Newton-Galerkin:
  - Newton-Galerkin: truncate integral operators (Galerkin projections) to design Newton operators
  - Newton-Picard: inspired by Picard iterations to design Newton operators

A Posteriori Validation Paradigm

$Ly = 0,\ \{y^{(i)}(0) = v_i\}_{i=0}^{r-1}$ linear IVP

Step 1 — Numerical approximation method: Spectral method

- *Polynomial approximations in Chebyshev basis*
- *Floating-point operations*

$\widetilde{y}$

Step 2 — A posteriori validation: Newton-Galerkin vs Newton-Picard

- *Interval operations + RPAs*
- *Newton-like fixed-point operator*

ε

Rigorous Polynomial Approximation (RPA) for y
= pair $(\widetilde{y}, \varepsilon)$ s.t. $\|\widetilde{y} - y\| \leqslant \varepsilon$

Chebyshev Approximations using Spectral Methods

From Differential Equations to Integral Equations (1)

- $L\{y\} \quad = \quad y^{(r)} + a_{r-1}(x)y^{(r-1)} + \cdots + a_1(x)y' + a_0(x)y \quad = \quad h(x),$

  with initial conditions: $\quad y^{(i)}(0) = v_i \quad$ for $\quad 0 \leqslant i < r, \quad \deg a_i \leqslant s$

From Differential Equations to Integral Equations (1)

- $L\{y\} \quad = \quad y^{(r)} + a_{r-1}(x)y^{(r-1)} + \cdots + a_1(x)y' + a_0(x)y \quad = \quad h(x),$

  with initial conditions: $\quad y^{(i)}(0) = v_i \quad$ for $\quad 0 \leqslant i < r, \quad \deg a_i \leqslant s$

  - $f = y^{(r)} \quad \Rightarrow \quad y^{(i)}(x) = \int_0^x \int_0^{t_1} \cdots \int_0^{t_{r-1-i}} f(t_{r-i}) dt_{r-i} \ldots t_1 \; + \sum_{j=i}^{r-1} \frac{x^{j-i}}{(j-i)!} v_j$

From Differential Equations to Integral Equations (1)

- $L\{y\} \quad = \quad y^{(r)} + a_{r-1}(x)y^{(r-1)} + \cdots + a_1(x)y' + a_0(x)y \quad = \quad h(x),$

  with initial conditions: $\quad y^{(i)}(0) = v_i \quad$ for $\quad 0 \leqslant i < r, \quad \deg a_i \leqslant s$

  - $f \;=\; y^{(r)} \quad \Rightarrow \quad y^{(i)}(x) = \int_0^x \frac{(x-t)^{r-1-i}}{(r-1-i)!} f(t)dt \;+\; \sum_{j=i}^{r-1} \frac{x^{j-i}}{(j-i)!} v_j$

From Differential Equations to Integral Equations (1)

- $\mathbf{L}\{y\} \quad = \quad y^{(r)} + a_{r-1}(x)y^{(r-1)} + \cdots + a_1(x)y' + a_0(x)y \quad = \quad h(x),$

  with initial conditions: $\quad y^{(i)}(0) = v_i \quad$ for $\quad 0 \leqslant i < r, \quad \deg a_i \leqslant s$

  - $f = y^{(r)} \quad \Rightarrow \quad y^{(i)}(x) = \int_0^x \frac{(x-t)^{r-1-i}}{(r-1-i)!} f(t)dt \; + \sum_{j=i}^{r-1} \frac{x^{j-i}}{(j-i)!} v_j$

$\Rightarrow$ Integral equation $\quad f + \mathbf{K}\{f\} = g, \quad \mathbf{K}\{f\}(x) = \int_0^x \mathfrak{K}(x,t) f(t)dt$

$$\mathfrak{K}(x,t) = \sum_{i=0}^{r-1} a_i(x) \frac{(x-t)^{r-1-i}}{(r-1-i)!} \qquad g(x) = h(x) - \sum_{i=0}^{r-1} a_i(x) \sum_{j=i}^{r-1} \frac{x^{j-i}}{(j-i)!} v_j$$

From Differential Equations to Integral Equations (1)

- $\mathbf{L}\{y\} = y^{(r)} + a_{r-1}(x)y^{(r-1)} + \cdots + a_1(x)y' + a_0(x)y = h(x),$

  with initial conditions: $y^{(i)}(0) = v_i$ for $0 \leqslant i < r,$ $\deg a_i \leqslant s$

  - $f = y^{(r)} \Rightarrow y^{(i)}(x) = \int_0^x \frac{(x-t)^{r-1-i}}{(r-1-i)!} f(t)dt + \sum_{j=i}^{r-1} \frac{x^{j-i}}{(j-i)!} v_j$

$\Rightarrow$ Integral equation $f + \mathbf{K}\{f\} = g$, $\mathbf{K}\{f\}(x) = \int_0^x \mathfrak{K}(x,t) f(t)dt$

$$\mathfrak{K}(x,t) = \sum_{i=0}^{r-1} a_i(x) \frac{(x-t)^{r-1-i}}{(r-1-i)!} \qquad g(x) = h(x) - \sum_{i=0}^{r-1} a_i(x) \sum_{j=i}^{r-1} \frac{x^{j-i}}{(j-i)!} v_j$$

Example (Airy)

$$y'' - xy = 0 \quad \Rightarrow \quad f + \int_0^x (xt - x^2) f(t)dt = v_0 x + v_1 x^2 \qquad f = \mathrm{Ai}''$$

From Differential Equations to Integral Equations (2)

- $L\{y\} = (-1)^r y^{(r)} + (-1)^{r-1}(b_{r-1}(x)y)^{(r-1)} + \cdots - (b_1(x)y)' + b_0(x)y = h(x)$,
  with initial conditions: $\quad y^{(i)}(0) = v_i \quad$ for $\quad 0 \leqslant i < r, \quad \deg b_i \leqslant s$

From Differential Equations to Integral Equations (2)

- $L\{y\} = (-1)^r y^{(r)} + (-1)^{r-1}(b_{r-1}(x)y)^{(r-1)} + \cdots - (b_1(x)y)' + b_0(x)y = h(x)$,

  with initial conditions: $\quad y^{(i)}(0) = v_i \quad$ for $\quad 0 \leqslant i < r, \quad \deg b_i \leqslant s$

  - $$\int_0^x \int_0^{t_1} \ldots \int_0^{t_{r-1}} (b_i y)^{(i)}(t_r) dt_r \ldots dt_1 = \int_0^x \frac{(x-t)^{r-1-i}}{(r-1-i)!} b_i(t) y(t) dt - \ldots$$

From Differential Equations to Integral Equations (2)

- $\mathbf{L}\{y\} = (-1)^r y^{(r)} + (-1)^{r-1}(b_{r-1}(x)y)^{(r-1)} + \cdots - (b_1(x)y)' + b_0(x)y = h(x),$

  with initial conditions: $y^{(i)}(0) = \nu_i \quad \text{for} \quad 0 \leqslant i < r, \quad \deg b_i \leqslant s$

- $$\int_0^x \int_0^{t_1} \cdots \int_0^{t_{r-1}} (b_i y)^{(i)}(t_r)dt_r \ldots dt_1 = \int_0^x \frac{(x-t)^{r-1-i}}{(r-1-i)!} b_i(t)y(t)dt - \ldots$$

$\Rightarrow$ Integral equation $\quad y + \mathbf{K}\{y\} = g, \quad \mathbf{K}\{y\}(x) = \int_0^x \mathfrak{K}(x,t)y(t)dt$

$$\mathfrak{K}(x,t) = \sum_{i=0}^{r-1} (-1)^{r-i} \frac{(x-t)^{r-1-i}}{(r-1-i)!} b_i(t) \qquad g(x) = (-1)^r \int_0^x \frac{(x-t)^{r-1}}{(r-1)!} h(t)dt + \ldots$$

From Differential Equations to Integral Equations (2)

- $\mathbf{L}\{y\} = (-1)^r y^{(r)} + (-1)^{r-1}(b_{r-1}(x)y)^{(r-1)} + \cdots - (b_1(x)y)' + b_0(x)y = h(x),$
  with initial conditions: $\quad y^{(i)}(0) = v_i \quad$ for $\quad 0 \leqslant i < r, \quad \deg b_i \leqslant s$
   - $$\int_0^x \int_0^{t_1} \cdots \int_0^{t_{r-1}} (b_i y)^{(i)}(t_r) dt_r \ldots dt_1 = \int_0^x \frac{(x-t)^{r-1-i}}{(r-1-i)!} b_i(t) y(t) dt - \ldots$$

$\Rightarrow$ Integral equation $\quad y + \mathbf{K}\{y\} = g, \quad \mathbf{K}\{y\}(x) = \int_0^x \mathfrak{K}(x,t) y(t) dt$

$$\mathfrak{K}(x,t) = \sum_{i=0}^{r-1} (-1)^{r-i} \frac{(x-t)^{r-1-i}}{(r-1-i)!} b_i(t) \qquad g(x) = (-1)^r \int_0^x \frac{(x-t)^{r-1}}{(r-1)!} h(t) dt + \ldots$$

Example (Airy)

$$y'' - xy = 0 \qquad \Rightarrow \qquad y + \int_0^x (t^2 - xt) y(t) dt = v_0 + v_1 x$$

Spectral Method for Airy Function

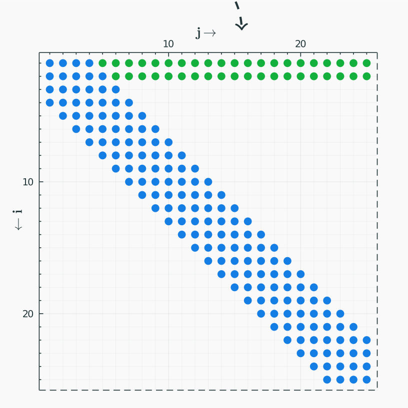
$$y(x) \quad + \quad \int_0^x (t^2 - xt) y(t) dt \quad = \quad g(x)$$

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ y_N \\ y_{N+1} \\ \vdots \end{pmatrix} \quad +$$

Spectral Method for Airy Function

$$y(x) \quad + \quad \int_0^x (t^2 - xt)y(t)dt \quad = \quad g(x)$$

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \\ y_{N+1} \\ \vdots \end{pmatrix} \quad + $$



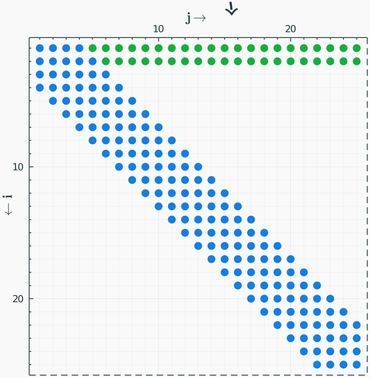
$$\bullet \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \\ y_{N+1} \\ \vdots \end{pmatrix}$$

$\mathbf{K}$: almost-banded, compact

Spectral Method for Airy Function

$$y(x) + \int_0^x (t^2 - xt)y(t)dt = g(x)$$

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \\ y_{N+1} \\ \vdots \end{pmatrix} + \quad$$



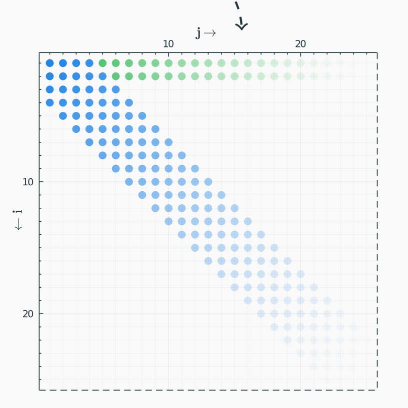
$$\bullet \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \\ y_{N+1} \\ \vdots \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_N \\ g_{N+1} \\ \vdots \end{pmatrix}$$

$\mathbf{K}$: almost-banded, compact

Spectral Method for Airy Function

$$y(x) \quad + \quad \int_0^x (t^2 - xt)y(t)dt \quad = \quad g(x)$$

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \\ y_{N+1} \\ \vdots \end{pmatrix} \quad +$$



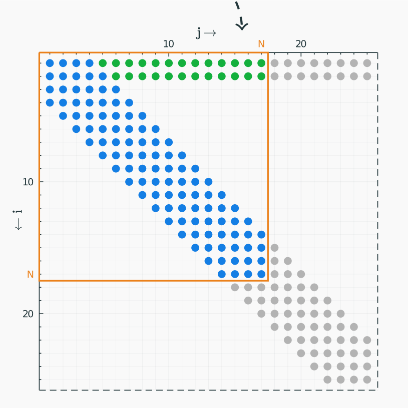
$$\bullet \quad \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \\ y_{N+1} \\ \vdots \end{pmatrix} \quad = \quad \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_N \\ g_{N+1} \\ \vdots \end{pmatrix}$$

K : almost-banded, compact

Spectral Method for Airy Function

$$y(x) \quad + \quad \int_0^x (t^2 - xt)y(t)dt \quad = \quad g(x)$$

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \\ \hline 0 \\ \vdots \end{pmatrix} +$$

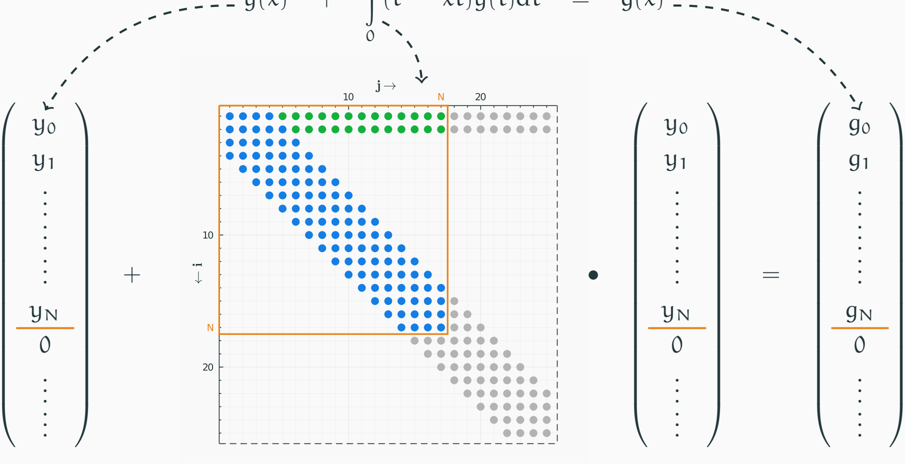


$$\bullet \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \\ \hline 0 \\ \vdots \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_N \\ \hline 0 \\ \vdots \end{pmatrix}$$

$K^{[N]}$: truncated operator

Spectral Method for Airy Function

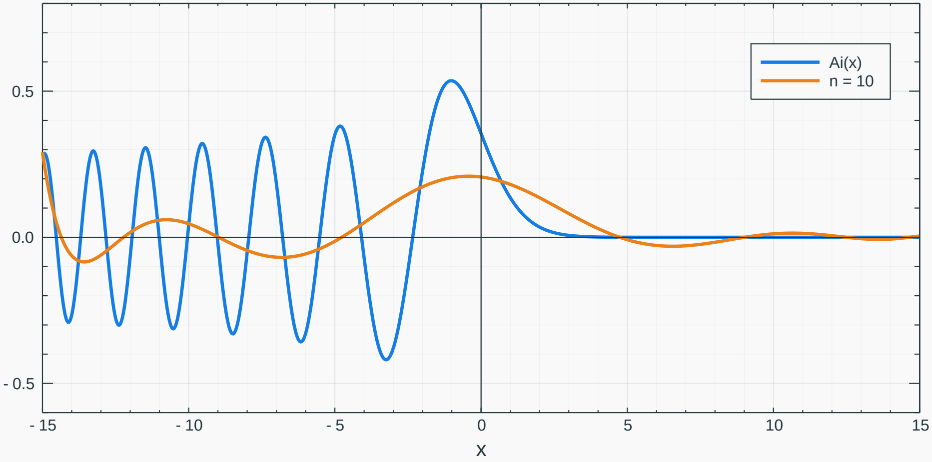
$$y(x) \quad + \quad \int_0^x (t^2 - xt)y(t)dt \quad = \quad g(x)$$



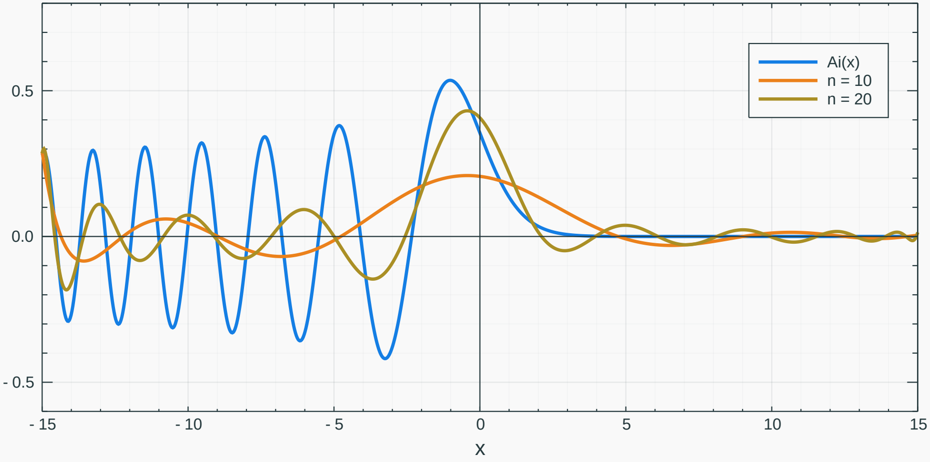
$\mathbf{K}^{[N]}$: truncated operator

- Solution in $\mathcal{O}((r+s)^2 N)$ arith. op. with Olver and Townsend's QR algorithm

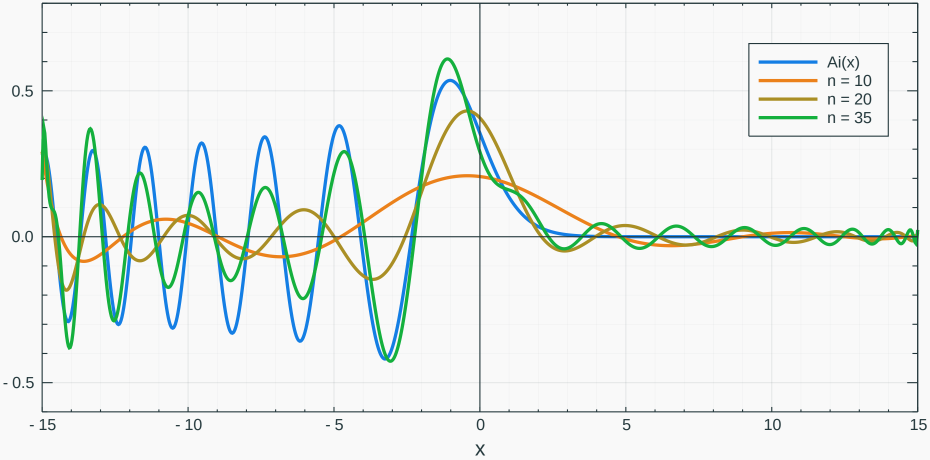
Spectral Method for Airy Function



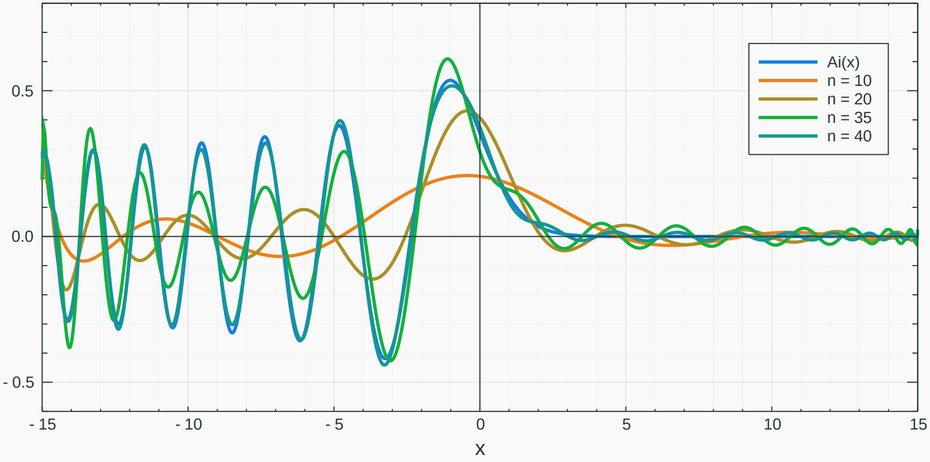
Spectral Method for Airy Function



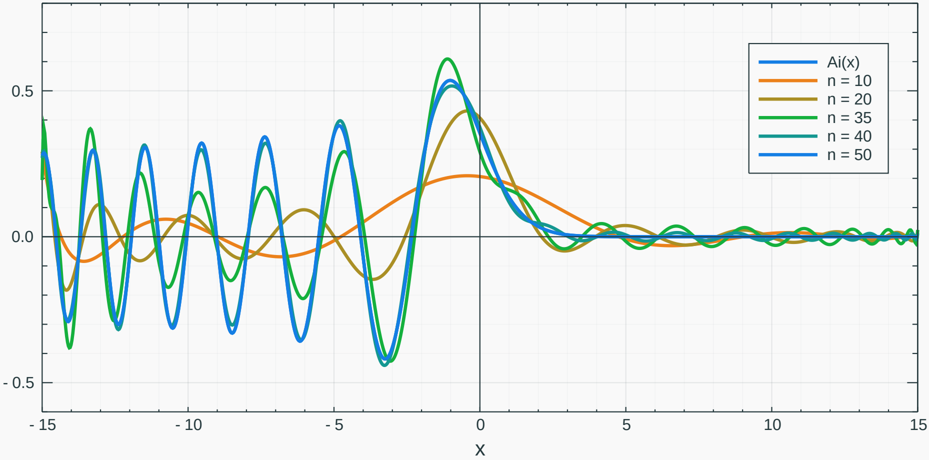
Spectral Method for Airy Function



Spectral Method for Airy Function

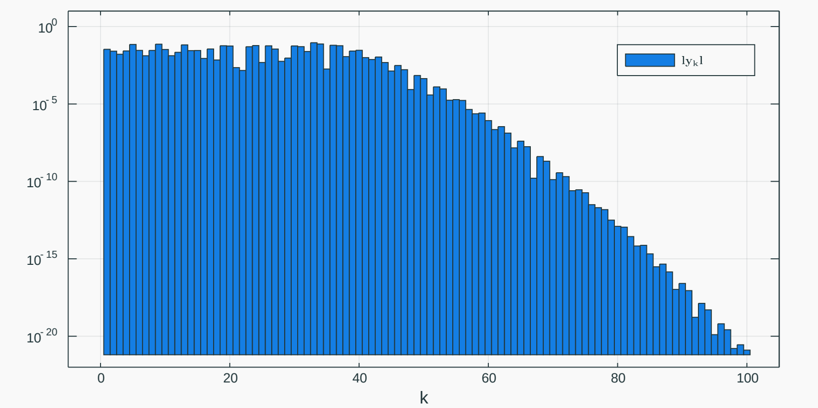


Spectral Method for Airy Function



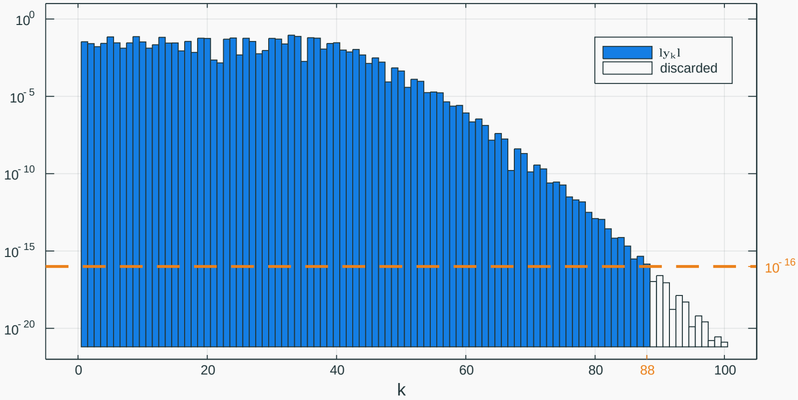
Chebyshev Coefficients of Airy Function

- Super-algebraic decay of Chebyshev coefficients for analytic functions



Chebyshev Coefficients of Airy Function

- Super-algebraic decay of Chebyshev coefficients for analytic functions



- Heuristic truncation methods in Chebfun, ...
- Need for validation algorithms

Fixed-Point Based Validation

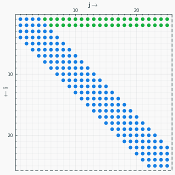
A Posteriori Validation Paradigm

Step 1 ∘ Numerical approximation method

- complexity
- numerical stability
- asymptotic convergence

$\widetilde{y}$

Step 2 ∘ A posteriori validation algorithm



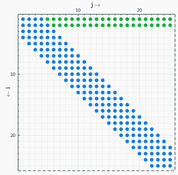
Rigorous Polynomial Approximation (RPA) for y

$= \text{ pair } (\widetilde{y}, \varepsilon) \text{ s.t. } \|\widetilde{y} - y\| \leqslant \varepsilon$

A Posteriori Validation Paradigm

Step 1 Numerical approximation method

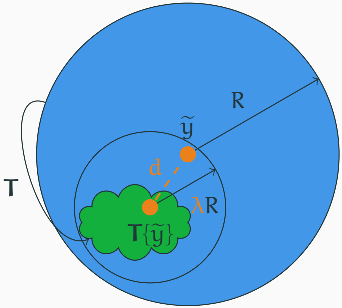
- complexity
- numerical stability
- asymptotic convergence



$\widetilde{y}$

Step 2 A posteriori validation algorithm

ε



Banach Fixed-Point Theorem

$$\frac{d}{1+\lambda} \leqslant \|\widetilde{y} - y\| \leqslant \frac{d}{1-\lambda}$$

$$d = \|T\{\widetilde{y}\} - \widetilde{y}\|$$

Rigorous Polynomial Approximation (RPA) for y

$$= \text{pair } (\widetilde{y}, \varepsilon) \text{ s.t. } \|\widetilde{y} - y\| \leqslant \varepsilon$$

Fixed-Point Based Validation

Newton-Galerkin Algorithm

Newton-Galerkin Validation Algorithm

- Newton: $y + \mathbf{K}\{y\} = g \qquad \Leftrightarrow \qquad \mathbf{T}\{y\} = y \quad \begin{cases} \mathbf{T}\{y\} := y - \mathbf{A}\{\, y + \mathbf{K}\{y\} - g \,\} \\ \mathbf{A} \approx (\mathbf{I} + \mathbf{K})^{-1} \end{cases}$

Newton-Galerkin Validation Algorithm

- Newton: $y + \mathbf{K}\{y\} = g \quad \Leftrightarrow \quad \mathsf{T}\{y\} = y \quad \begin{cases} \mathsf{T}\{y\} := y - \boldsymbol{A}\{\, y + \mathbf{K}\{y\} - g \,\} \\ \boldsymbol{A} \approx (\mathbf{I} + \mathbf{K})^{-1} \end{cases}$

Define approx inverse $\boldsymbol{A}$

- choose $N_{\mathcal{G}}$
- compute numerical
- $\boldsymbol{A} \approx (\mathbf{I} + \mathbf{K}^{[N_{\mathcal{G}}]})^{-1}$

Newton-Galerkin Validation Algorithm

- Newton: $y + \mathbf{K}\{y\} = g \quad \Leftrightarrow \quad \mathsf{T}\{y\} = y \quad \begin{cases} \mathsf{T}\{y\} := y - \mathbf{A}\{\, y + \mathbf{K}\{y\} - g \,\} \\ \mathbf{A} \approx (\mathbf{I} + \mathbf{K})^{-1} \end{cases}$

Define approx inverse $\mathbf{A}$

- choose $N_{\mathcal{G}}$
- compute numerical
  $\mathbf{A} \approx (\mathbf{I} + \mathbf{K}^{[N_{\mathcal{G}}]})^{-1}$

Bound the operator norm of T

$$\lambda := \|D\mathsf{T}\| = \|\mathbf{I} - \mathbf{A}\,(\mathbf{I} + \mathbf{K})\|$$
$$\leqslant \underbrace{\|\mathbf{I} - \mathbf{A}\,(\mathbf{I} + \mathbf{K}^{[N_{\mathcal{G}}]})\|}_{\text{approximation error}} + \underbrace{\|\mathbf{A}\,(\mathbf{K} - \mathbf{K}^{[N_{\mathcal{G}}]})\|}_{\text{truncation error}}$$

Newton-Galerkin Validation Algorithm

- Newton: $\quad y + \mathbf{K}\{y\} = g \quad \Leftrightarrow \quad \mathbf{T}\{y\} = y \quad \begin{cases} \mathbf{T}\{y\} := y - \boldsymbol{A}\{\, y + \mathbf{K}\{y\} - g \,\} \\ \boldsymbol{A} \approx (\mathbf{I} + \mathbf{K})^{-1} \end{cases}$

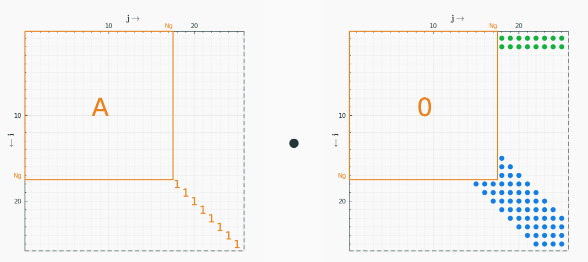
Define approx inverse $\boldsymbol{A}$

- choose $N_{\mathcal{G}}$
- compute numerical
- $\boldsymbol{A} \approx (\mathbf{I} + \mathbf{K}^{[N_{\mathcal{G}}]})^{-1}$

Bound the operator norm of $\mathbf{T}$

$$\lambda := \|D\mathbf{T}\| = \|\mathbf{I} - \boldsymbol{A}\,(\mathbf{I} + \mathbf{K})\|$$

$$\leqslant \underbrace{\|\mathbf{I} - \boldsymbol{A}\,(\mathbf{I} + \mathbf{K}^{[N_{\mathcal{G}}]})\|}_{\text{approximation error}} + \underbrace{\|\boldsymbol{A}\,(\mathbf{K} - \mathbf{K}^{[N_{\mathcal{G}}]})\|}_{\text{truncation error}}$$



Newton-Galerkin Validation Algorithm

- Newton: $y + \mathbf{K}\{y\} = g \quad \Leftrightarrow \quad \mathbf{T}\{y\} = y \quad \begin{cases} \mathbf{T}\{y\} := y - \mathbf{A}\{y + \mathbf{K}\{y\} - g\} \\ \mathbf{A} \approx (\mathbf{I} + \mathbf{K})^{-1} \end{cases}$

Define approx inverse $\mathbf{A}$

- choose $N_{\mathcal{G}}$
- compute numerical
- $\mathbf{A} \approx (\mathbf{I} + \mathbf{K}^{[N_{\mathcal{G}}]})^{-1}$

Bound the operator norm of $\mathbf{T}$

$$\lambda := \|D\mathbf{T}\| = \|\mathbf{I} - \mathbf{A}(\mathbf{I} + \mathbf{K})\| \leqslant \underbrace{\|\mathbf{I} - \mathbf{A}(\mathbf{I} + \mathbf{K}^{[N_{\mathcal{G}}]})\|}_{\text{approximation error}} + \underbrace{\|\mathbf{A}(\mathbf{K} - \mathbf{K}^{[N_{\mathcal{G}}]})\|}_{\text{truncation error}}$$

- $\|\tilde{y} - y\| \leqslant \|\mathbf{T}\{\tilde{y}\} - \tilde{y}\| / (1 - \lambda)$

Theorem (Bréhard, Brisebarre, Joldes) — ACM Trans. Math. Softw.
Validating a degree N approximation $\tilde{y}$ requires:

$$\mathcal{O}(\, N_{\mathcal{G}}^2 (r + s) + (N_{\mathcal{G}} + r + s) N \,) \qquad \text{arithmetic operations}$$

Newton-Galerkin Validation Algorithm

- Newton: $y + \mathbf{K}\{y\} = g \quad \Leftrightarrow \quad \mathbf{T}\{y\} = y \quad \begin{cases} \mathbf{T}\{y\} := y - \mathbf{A}\{ y + \mathbf{K}\{y\} - g \} \\ \mathbf{A} \approx (\mathbf{I} + \mathbf{K})^{-1} \end{cases}$

Define approx inverse $\boldsymbol{A}$

- choose $N_{\mathcal{G}}$
- compute numerical
  $\mathbf{A} \approx (\mathbf{I} + \mathbf{K}^{[N_{\mathcal{G}}]})^{-1}$

Bound the operator norm of $\mathbf{T}$

$$\lambda := \|D\mathbf{T}\| = \|\mathbf{I} - \mathbf{A}(\mathbf{I} + \mathbf{K})\|$$
$$\leqslant \underbrace{\|\mathbf{I} - \mathbf{A}(\mathbf{I} + \mathbf{K}^{[N_{\mathcal{G}}]})\|}_{\text{approximation error}} + \underbrace{\|\mathbf{A}(\mathbf{K} - \mathbf{K}^{[N_{\mathcal{G}}]})\|}_{\text{truncation error}}$$

- $\|\tilde{y} - y\| \leqslant \|\mathbf{T}\{\tilde{y}\} - \tilde{y}\| / (1 - \lambda)$

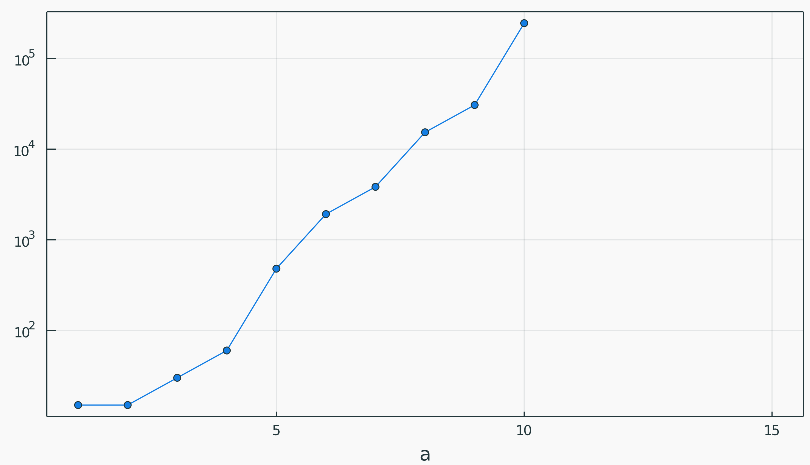
Theorem (Bréhard, Brisebarre, Joldes) — ACM Trans. Math. Softw.
Validating a degree N approximation $\tilde{y}$ requires:

$$\mathcal{O}(\ N_{\mathcal{G}}^2 (r + s) + (N_{\mathcal{G}} + r + s) N\ ) \qquad \text{arithmetic operations}$$

…but $N_{\mathcal{G}}$ may be exponential w.r.t. the $\|a_i\|$!

Newton-Galerkin Validation on Airy Example

- Newton-Galerkin method for Airy function over $[-a, a]$

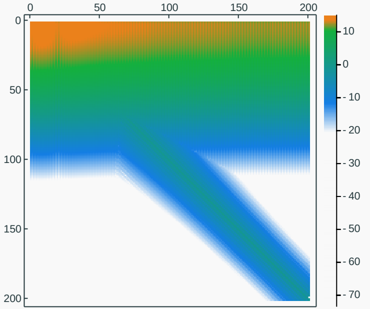


$\Rightarrow$ Truncation index $N_{\mathcal{G}}$ grows exponentially fast!

Fixed-Point Based Validation

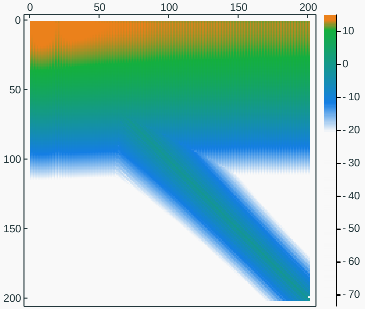
Newton-Picard Algorithm

Almost Banded Approximate Inverses



- $(\mathbf{I}+\mathbf{K})^{-1}$ is "asymptotically almost-banded"

Almost Banded Approximate Inverses

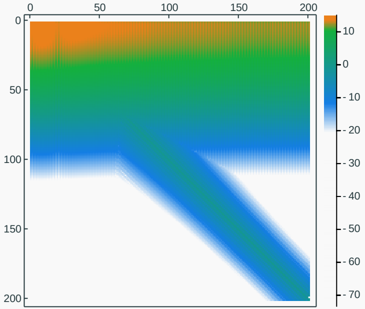


- $(\mathbf{I}+\mathbf{K})^{-1}$ is "asymptotically almost-banded"

… because of Picard iterations:

$$(\mathbf{I}+\mathbf{K})^{-1} = \mathbf{I} - \mathbf{K} + \mathbf{K}^2 - \cdots + (-1)^n \mathbf{K}^n + \ldots$$

Almost Banded Approximate Inverses



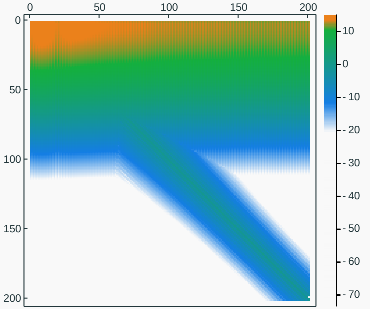
- $(\mathbf{I}+\mathbf{K})^{-1}$ is "asymptotically almost-banded"

  … because of Picard iterations:

$$(\mathbf{I}+\mathbf{K})^{-1} = \mathbf{I} - \mathbf{K} + \mathbf{K}^2 - \cdots + (-1)^n \mathbf{K}^n + \dots$$

$$\left.\begin{aligned} \mathbf{K} &\rightsquigarrow \mathfrak{K}(x,t) \\ \mathbf{L} &\rightsquigarrow \mathfrak{L}(x,t) \end{aligned}\right\} \quad \Rightarrow \quad \mathbf{K}\,\mathbf{L} \rightsquigarrow \mathfrak{K} * \mathfrak{L} \quad \text{where} \quad (\mathfrak{K} * \mathfrak{L})(x,t) := \int_t^x \mathfrak{K}(x,s)\mathfrak{L}(s,t)ds$$

Almost Banded Approximate Inverses



- $(\mathbf{I}+\mathbf{K})^{-1}$ is "asymptotically almost-banded"

  … because of Picard iterations:

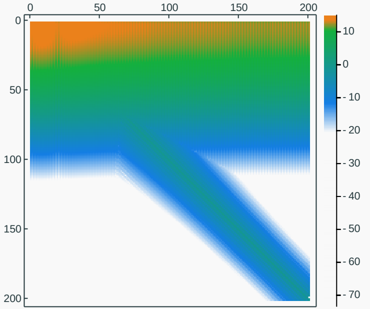
$$(\mathbf{I}+\mathbf{K})^{-1} = \mathbf{I} - \mathbf{K} + \mathbf{K}^2 - \cdots + (-1)^n \mathbf{K}^n + \dots$$

$$\left.\begin{aligned} \mathbf{K} &\rightsquigarrow \mathfrak{K}(x,t) \\ \mathbf{L} &\rightsquigarrow \mathfrak{L}(x,t) \end{aligned}\right\} \quad \Rightarrow \quad \mathbf{K}\,\mathbf{L} \rightsquigarrow \mathfrak{K} * \mathfrak{L} \quad \text{where} \quad (\mathfrak{K} * \mathfrak{L})(x,t) := \int_t^x \mathfrak{K}(x,s)\mathfrak{L}(s,t)ds$$

Iterated Kernels and Resolvent Kernel

$$\left.\begin{aligned} |\mathfrak{K}^{*n}(x,t)| &\leqslant \frac{(x-t)^{n-1}}{(n-1)!} M^n \\ M &:= \sup_{0 \leqslant t \leqslant x \leqslant T} |\mathfrak{K}(x,t)| \end{aligned}\right\}$$

Almost Banded Approximate Inverses



- $(\mathbf{I}+\mathbf{K})^{-1}$ is "asymptotically almost-banded"

  ... because of Picard iterations:

$$(\mathbf{I}+\mathbf{K})^{-1} = \mathbf{I} - \mathbf{K} + \mathbf{K}^2 - \cdots + (-1)^n \mathbf{K}^n + \dots$$
$$= \mathbf{I} + \mathbf{R} \quad \text{where} \quad \mathbf{R} \rightsquigarrow \mathfrak{R}(x,t)$$

$$\left.\begin{aligned} \mathbf{K} &\rightsquigarrow \mathfrak{K}(x,t) \\ \mathbf{L} &\rightsquigarrow \mathfrak{L}(x,t) \end{aligned}\right\} \quad \Rightarrow \quad \mathbf{K}\,\mathbf{L} \rightsquigarrow \mathfrak{K} * \mathfrak{L} \quad \text{where} \quad (\mathfrak{K} * \mathfrak{L})(x,t) := \int_t^x \mathfrak{K}(x,s)\mathfrak{L}(s,t)ds$$

Iterated Kernels and Resolvent Kernel

$$\left.\begin{aligned} |\mathfrak{K}^{*n}(x,t)| &\leqslant \frac{(x-t)^{n-1}}{(n-1)!} M^n \\ M &:= \sup_{0 \leqslant t \leqslant x \leqslant T} |\mathfrak{K}(x,t)| \end{aligned}\right\} \quad \Rightarrow \quad \mathfrak{R}(x,t) := \sum_{n=1}^{\infty} (-1)^n \mathfrak{K}^{*n}(x,t)$$

Newton-Picard Validation Algorithm

Validating a candidate approximation $\widetilde{y}$:

1. Compute $\mathbf{I} + \widetilde{\mathbf{R}} \approx (\mathbf{I} + \mathbf{K})^{-1}$ with $\widetilde{\mathbf{R}} \rightsquigarrow \widetilde{\mathfrak{R}}(x, t) \approx \mathfrak{R}(x, t)$ polynomial

   Newton operator: $\quad \mathbf{T}\{y\} := y - (\mathbf{I} + \widetilde{\mathbf{R}})\{ y + \mathbf{K}\{y\} - g \}$

Newton-Picard Validation Algorithm

Validating a candidate approximation $\widetilde{y}$:

1. Compute $\mathbf{I} + \widetilde{\mathbf{R}} \approx (\mathbf{I} + \mathbf{K})^{-1}$ with $\widetilde{\mathbf{R}} \rightsquigarrow \widetilde{\mathfrak{R}}(x,t) \approx \mathfrak{R}(x,t)$ polynomial

   Newton operator: $\quad \mathbf{T}\{y\} \;:=\; y - (\mathbf{I} + \widetilde{\mathbf{R}})\{\, y + \mathbf{K}\{y\} - g \,\}$

2. Bound the operator norm:

   $$-D\mathbf{T} \;=\; (\mathbf{I} + \widetilde{\mathbf{R}})(\mathbf{I} + \mathbf{K}) - \mathbf{I} \;=\; \mathbf{K} + \widetilde{\mathbf{R}} + \widetilde{\mathbf{R}}\mathbf{K} \quad \rightsquigarrow \quad \mathfrak{K} + \widetilde{\mathfrak{R}} + \widetilde{\mathfrak{R}} * \mathfrak{K} \;=:\; \mathfrak{E}$$

   with $\quad \lambda \;\geqslant\; T \sup_{0 \leqslant t \leqslant x \leqslant T} |\mathfrak{E}(x,t)|$

Newton-Picard Validation Algorithm

Validating a candidate approximation $\widetilde{y}$:

1. Compute $\mathbf{I} + \widetilde{\mathbf{R}} \approx (\mathbf{I} + \mathbf{K})^{-1}$ with $\widetilde{\mathbf{R}} \rightsquigarrow \widetilde{\mathfrak{R}}(x,t) \approx \mathfrak{R}(x,t)$ polynomial

   Newton operator: $\mathbf{T}\{y\} := y - (\mathbf{I} + \widetilde{\mathbf{R}})\{\, y + \mathbf{K}\{y\} - g \,\}$

2. Bound the operator norm:

   $$-D\mathbf{T} = (\mathbf{I} + \widetilde{\mathbf{R}})(\mathbf{I} + \mathbf{K}) - \mathbf{I} = \mathbf{K} + \widetilde{\mathbf{R}} + \widetilde{\mathbf{R}}\mathbf{K} \quad \rightsquigarrow \quad \mathfrak{K} + \widetilde{\mathfrak{R}} + \widetilde{\mathfrak{R}} * \mathfrak{K} =: \mathfrak{E}$$

   with $\lambda \geqslant T \sup_{0 \leqslant t \leqslant x \leqslant T} |\mathfrak{E}(x,t)|$

3. Compute $d = \|\mathbf{T}\{\widetilde{y}\} - \widetilde{y}\| = \|(\mathbf{I} + \widetilde{\mathbf{R}})\{\, \widetilde{y} + \mathbf{K}\{\widetilde{y}\} - g \,\}\|$

   and apply the Banach fixed-point theorem:

   $$\frac{d}{1+\lambda} \leqslant \|\widetilde{y} - y\| \leqslant \frac{d}{1-\lambda}$$

Newton-Picard Validation Algorithm

Validating a candidate approximation $\tilde{y}$:

1. Compute $\mathbf{I} + \widetilde{\mathbf{R}} \approx (\mathbf{I} + \mathbf{K})^{-1}$ with $\widetilde{\mathbf{R}} \rightsquigarrow \widetilde{\mathfrak{R}}(x,t) \approx \mathfrak{R}(x,t)$ polynomial

   Newton operator: $\quad \mathbf{T}\{y\} := y - (\mathbf{I} + \widetilde{\mathbf{R}})\{\, y + \mathbf{K}\{y\} - g \,\}$

2. Bound the operator norm:

   $$-\mathrm{D}\mathbf{T} = (\mathbf{I} + \widetilde{\mathbf{R}})(\mathbf{I} + \mathbf{K}) - \mathbf{I} = \mathbf{K} + \widetilde{\mathbf{R}} + \widetilde{\mathbf{R}}\mathbf{K} \quad \rightsquigarrow \quad \mathfrak{K} + \widetilde{\mathfrak{R}} + \widetilde{\mathfrak{R}} * \mathfrak{K} =: \mathfrak{E}$$

   with $\quad \lambda \geqslant T \sup_{0 \leqslant t \leqslant x \leqslant T} |\mathfrak{E}(x,t)|$

3. Compute $d = \|\mathbf{T}\{\tilde{y}\} - \tilde{y}\| = \|(\mathbf{I} + \widetilde{\mathbf{R}})\{\, \tilde{y} + \mathbf{K}\{\tilde{y}\} - g \,\}\|$

   and apply the Banach fixed-point theorem:

   $$\frac{d}{1+\lambda} \leqslant \|\tilde{y} - y\| \leqslant \frac{d}{1-\lambda}$$

> How to efficiently approximate $\mathfrak{R}(x,t)$?

Formulas for Resolvent Kernel Approximation

- $\mathbf{L} = \partial^r + a_{r-1}(x)\partial^{r-1} + \cdots + a_0(x) \quad \rightsquigarrow \quad$ $\mathbf{K}$ acting on $f = y^{(r)}$

$$\mathfrak{R}(x,t) = \sum_{i=0}^{r-1} \varphi_i^{(r)}(x)\psi_i(t)$$

Formulas for Resolvent Kernel Approximation

- $\mathbf{L} = \partial^r + a_{r-1}(x)\partial^{r-1} + \cdots + a_0(x) \quad \rightsquigarrow \quad$ $\mathbf{K}$ acting on $f = y^{(r)}$

$$\mathfrak{R}(x,t) = \sum_{i=0}^{r-1} \varphi_i^{(r)}(x)\psi_i(t)$$

- solve $\quad f + \mathbf{K}\{f\} = h_i, \quad f = \varphi_i^{(r)}$

  $\{\varphi_0, \ldots, \varphi_{r-1}\}$ basis of $\mathbf{L}\{\varphi\} = 0$

Formulas for Resolvent Kernel Approximation

- $\mathbf{L} = \partial^r + a_{r-1}(x)\partial^{r-1} + \cdots + a_0(x) \quad \rightsquigarrow \quad \mathbf{K}$ acting on $f = y^{(r)}$

$$\mathfrak{R}(x,t) = \sum_{i=0}^{r-1} \varphi_i^{(r)}(x)\psi_i(t)$$

- solve $\quad f + \mathbf{K}\{f\} = h_i, \quad f = \varphi_i^{(r)}$

  $\{\varphi_0, \ldots, \varphi_{r-1}\}$ basis of $\mathbf{L}\{\varphi\} = 0$

- solve $\quad f + \mathbf{K}^*\{f\} = h_i^*, \quad f = \psi_i$

  $\{\psi_0, \ldots, \psi_{r-1}\}$ basis of $\mathbf{L}^*\{\psi\} = 0$

Formulas for Resolvent Kernel Approximation

- $\mathbf{L} = \partial^r + a_{r-1}(x)\partial^{r-1} + \cdots + a_0(x) \quad \rightsquigarrow \quad \mathbf{K}$ acting on $f = y^{(r)}$

$$\mathfrak{R}(x,t) = \sum_{i=0}^{r-1} \varphi_i^{(r)}(x)\psi_i(t)$$

- solve $\quad f + \mathbf{K}\{f\} = h_i, \quad f = \varphi_i^{(r)}$

  $\{\varphi_0, \ldots, \varphi_{r-1}\}$ basis of $\mathbf{L}\{\varphi\} = 0$

- solve $\quad f + \mathbf{K}^*\{f\} = h_i^*, \quad f = \psi_i$

  $\{\psi_0, \ldots, \psi_{r-1}\}$ basis of $\mathbf{L}^*\{\psi\} = 0$

- Approximations using spectral methods to some degree $N_{\mathfrak{R}}$

Formulas for Resolvent Kernel Approximation

- $\mathbf{L} = \partial^r + a_{r-1}(x)\partial^{r-1} + \cdots + a_0(x) \quad \rightsquigarrow \quad \mathbf{K}$ acting on $f = y^{(r)}$

$$\mathfrak{R}(x,t) = \sum_{i=0}^{r-1} \varphi_i^{(r)}(x)\psi_i(t)$$

- solve $\quad f + \mathbf{K}\{f\} = h_i, \quad f = \varphi_i^{(r)}$

  $\{\varphi_0, \ldots, \varphi_{r-1}\}$ basis of $\mathbf{L}\{\varphi\} = 0$

- solve $\quad f + \mathbf{K}^*\{f\} = h_i^*, \quad f = \psi_i$

  $\{\psi_0, \ldots, \psi_{r-1}\}$ basis of $\mathbf{L}^*\{\psi\} = 0$

- Approximations using spectral methods to some degree $N_{\mathfrak{R}}$
- The idea is very close to the fast algorithm for power series solutions to ODEs, but in a symbolic-numeric setting
  (*Bostan, Chyzak, Ollivier, Salvy, Schost, Sedoglavic – 2006*)

Complexity Analysis of Newton-Picard Validation Algorithm

Validating a candidate approximation $\widetilde{y}$ of degree N:

1. Choose a validation degree $N_{\mathfrak{R}}$ and

   compute $\quad \widetilde{\mathfrak{R}}(x,t) := \sum_{i=0}^{r-1} \widetilde{\varphi}_i^{(r)}(x)\widetilde{\psi}_i(t) \approx \sum_{i=0}^{r-1} \varphi_i^{(r)}(x)\psi_i(t) =: \mathfrak{R}(x,t)$

2. Bound the norm of $\quad \mathfrak{E} := \mathfrak{K} + \widetilde{\mathfrak{R}} + +\widetilde{\mathfrak{R}} * \mathfrak{K}$

3. Compute defect $\quad \|\mathbf{T}\{\widetilde{y}\} - \widetilde{y}\| = \|(\mathbf{I} + \widetilde{\mathbf{R}})\{\,\widetilde{y} + \mathbf{K}\{\widetilde{y}\} - g\,\}\|$

Complexity Analysis of Newton-Picard Validation Algorithm

Validating a candidate approximation $\tilde{y}$ of degree N:

1. Choose a validation degree $N_{\mathfrak{R}}$ and

   compute $\quad \widetilde{\mathfrak{R}}(x,t) := \sum_{i=0}^{r-1} \widetilde{\varphi}_i^{(r)}(x)\widetilde{\psi}_i(t) \approx \sum_{i=0}^{r-1} \varphi_i^{(r)}(x)\psi_i(t) =: \mathfrak{R}(x,t)$

   complexity: $\quad \mathcal{O}((r+s)^2 N_{\mathfrak{R}})$

   $$\begin{cases} r = \text{order of } \mathbf{L} \\ s = \max \deg a_i \end{cases}$$

2. Bound the norm of $\quad \mathfrak{E} := \mathfrak{K} + \widetilde{\mathfrak{R}} + +\widetilde{\mathfrak{R}} * \mathfrak{K}$

   complexity: $\quad \mathcal{O}(r^2 N_{\mathfrak{R}}^2)$

3. Compute defect $\quad \|\mathbf{T}\{\tilde{y}\} - \tilde{y}\| = \|(\mathbf{I} + \widetilde{\mathbf{R}})\{\, \tilde{y} + \mathbf{K}\{\tilde{y}\} - g \,\}\|$

   complexity: $\quad \mathcal{O}(r N_{\mathfrak{R}} N)$

Complexity Analysis of Newton-Picard Validation Algorithm

Validating a candidate approximation $\tilde{y}$ of degree N:

1. Choose a validation degree $N_{\mathfrak{R}}$ and

   compute $\quad \widetilde{\mathfrak{R}}(x,t) := \sum_{i=0}^{r-1} \tilde{\varphi}_i^{(r)}(x)\tilde{\psi}_i(t) \approx \sum_{i=0}^{r-1} \varphi_i^{(r)}(x)\psi_i(t) =: \mathfrak{R}(x,t)$

   complexity: $\quad \mathcal{O}((r+s)^2 N_{\mathfrak{R}})$ $\quad \begin{cases} r = \text{order of } \mathbf{L} \\ s = \max\deg a_i \end{cases}$

2. Bound the norm of $\quad \mathfrak{E} := \mathfrak{K} + \widetilde{\mathfrak{R}} + +\widetilde{\mathfrak{R}} * \mathfrak{K}$

   complexity: $\quad \mathcal{O}(r^2 N_{\mathfrak{R}}^2)$

3. Compute defect $\quad \|\mathbf{T}\{\tilde{y}\} - \tilde{y}\| = \|(\mathbf{I} + \widetilde{\mathbf{R}})\{\,\tilde{y} + \mathbf{K}\{\tilde{y}\} - g\,\}\|$

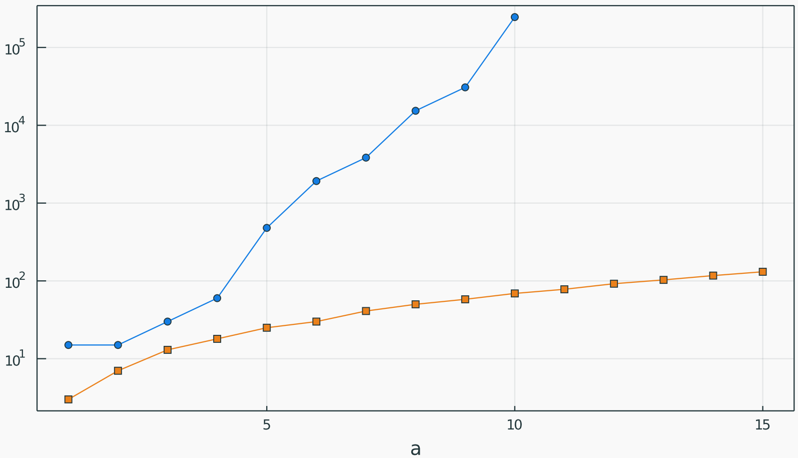
   complexity: $\quad \mathcal{O}(r N_{\mathfrak{R}} N)$

Newton-Galerkin's $N_{\mathcal{G}}$ *vs* Newton-Picard's $N_{\mathfrak{R}}$?

- $N_{\mathcal{G}} = \mathcal{O}\left(\sum_i \|\varphi_i^{(r)}\|\|\psi_i\|\right) \quad \rightsquigarrow \quad$ exponential in the $\|a_i\|$
- $N_{\mathfrak{R}} = \mathcal{O}\left(\log\left(\sum_i \|\varphi_i^{(r)}\|\|\psi_i\|\right)\right) \rightsquigarrow$ polynomial in the $\|a_i\|$

Newton-Galerkin vs Newton-Picard on Airy Example

- Validation for Airy function over $[-a, a]$



- $N_{\mathcal{G}}$ $\rightsquigarrow$ exponential in the $\|a_i\|$
- $N_{\mathfrak{R}}$ $\rightsquigarrow$ polynomial in the $\|a_i\|$