Fast linear algebra for computing LDEs and recurrences

Louis Gaillard, ENS de Lyon, LIP, Lyon

joint work with Alin Bostan, Bruno Salvy, Gilles Villard

Functional Equations in Limoges, March 30, 2026

D-finite functions

$f(x) \in \mathsf{K}[[x]]$ solution of

$$a_r(x)f^{(r)}(x) + \cdots + a_1(x)f'(x) + a_0(x)f(x) = 0, \quad a_i(x) \in \mathsf{K}[x]$$

D-finite functions

$f(x) \in \mathrm{K}[[x]]$ solution of

$$a_r(x)f^{(r)}(x) + \cdots + a_1(x)f'(x) + a_0(x)f(x) = 0, \quad a_i(x) \in \mathrm{K}[x]$$

$L(f) = 0$ with $L = a_r\partial^r + \cdots + a_1\partial + a_0$

$r =$ order of L, $\qquad d = \max(\deg a_i) =$ degree of L

Equation as data structure

$L = a_r\partial^r + \cdots + a_1\partial + a_0$ is a good data structure for representing solutions

Equation as data structure

$L = a_r\partial^r + \cdots + a_1\partial + a_0$ is a good data structure for representing solutions

Useful for numerical evaluation, local and asymptotic analyisis, proofs of identities,...

Equation as data structure

$L = a_r\partial^r + \cdots + a_1\partial + a_0$ is a good data structure for representing solutions

Useful for numerical evaluation, local and asymptotic analyisis, proofs of identities,...

Examples:

$$\cos(x) \quad \{f'' + f = 0, f(0) = 1, f'(1) = 0\}$$

$$e^{\arctan(x)} \quad \{(1+x^2)f' - f = 0, f(0) = 1\}$$

$$A(x) = \sum_{n=0}^{\infty}\left(\sum_{k=0}^{n}\binom{n}{k}^2\binom{n+k}{k}^2\right)x^n$$

$$\{x^2(x^2-34x+1)f^{(3)}+x(6x^2-153x+3)f'' + (7x^2-112x+1)f' + (x-5)f = 0, f(0) = 1\}$$

Equation as data structure

$L = a_r\partial^r + \cdots + a_1\partial + a_0$ is a good data structure for representing solutions

Useful for numerical evaluation, local and asymptotic analyisis, proofs of identities,. . .

Examples:

$$\cos(x) \quad \{f'' + f = 0, f(0) = 1, f'(1) = 0\}$$

$$e^{\arctan(x)} \quad \{(1 + x^2)f' - f = 0, f(0) = 1\}$$

$$A(x) = \sum_{n=0}^{\infty}\left(\sum_{k=0}^{n}\binom{n}{k}^2\binom{n+k}{k}^2\right)x^n$$

$$\{x^2(x^2 - 34x + 1)f^{(3)} + x(6x^2 - 153x + 3)f'' + (7x^2 - 112x + 1)f' + (x - 5)f = 0, f(0) = 1\}$$

Need efficient algorithms for computing LDEs

Equation as data structure

$L = a_r\partial^r + \cdots + a_1\partial + a_0$ is a good data structure for representing solutions

Useful for numerical evaluation, local and asymptotic analyisis, proofs of identities,...

Examples:

$$\cos(x) \quad \{f'' + f = 0, f(0) = 1, f'(1) = 0\}$$

$$e^{\arctan(x)} \quad \{(1+x^2)f' - f = 0, f(0) = 1\}$$

$$A(x) = \sum_{n=0}^{\infty}\left(\sum_{k=0}^{n}\binom{n}{k}^2\binom{n+k}{k}^2\right)x^n$$

$$\{x^2(x^2-34x+1)f^{(3)}+x(6x^2-153x+3)f'' + (7x^2-112x+1)f' + (x-5)f = 0, f(0) = 1\}$$

Need efficient algorithms for computing LDEs

Similar notion for recurrences: P-recursive sequences

Typical algorithmic problems

- **Effective closure properties**
  Given L_1, L_2 s.t. $L_1(f) = L_2(g) = 0$
  Compute L s.t. $L(f \diamond g) = 0$
  where $\diamond \in \{+, \times\}$

$\cos(x) + e^{\arctan(x)}$ satisfies

$$a(x)y^{(3)} + b(x)y'' + a(x)y' + b(x)y = 0,$$
$$a(x) = (x^2+1)(x^4+2x^2-2x+2),$$
$$b(x) = -x(x^3+8x-6)$$

Typical algorithmic problems

- **Effective closure properties**
  Given L_1, L_2 s.t. $L_1(f) = L_2(g) = 0$
  Compute L s.t. $L(f \diamond g) = 0$
  where $\diamond \in \{+, \times\}$

$\cos(x) + e^{\arctan(x)}$ satisfies

$$a(x)y^{(3)} + b(x)y'' + a(x)y' + b(x)y = 0,$$
$$a(x) = (x^2 + 1)(x^4 + 2x^2 - 2x + 2),$$
$$b(x) = -x(x^3 + 8x - 6)$$

- **Polynomial Equation $\rightarrow$ LDE**
  Given $f(x)$ algebraic s.t. $P(x, f(x)) = 0$
  Compute L s.t $L(f) = 0$

Catalan generating series $C(x) = 1 + xC(x)^2$
$2C(x) + 2(-1 + 5x)C'(x) + x(4x - 1)C''(x) = 0$

Typical algorithmic problems

- **Effective closure properties**
  Given L_1, L_2 s.t. $L_1(f) = L_2(g) = 0$
  Compute L s.t. $L(f \diamond g) = 0$
  where $\diamond \in \{+, \times\}$

$\cos(x) + e^{\arctan(x)}$ satisfies

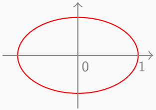
$$a(x)y^{(3)} + b(x)y'' + a(x)y' + b(x)y = 0,$$
$$a(x) = (x^2+1)(x^4+2x^2-2x+2),$$
$$b(x) = -x(x^3+8x-6)$$

- **Polynomial Equation $\to$ LDE**
  Given $f(x)$ algebraic s.t. $P(x, f(x)) = 0$
  Compute L s.t $L(f) = 0$

Catalan generating series $C(x) = 1 + xC(x)^2$
$2C(x) + 2(-1+5x)C'(x) + x(4x-1)C''(x) = 0$

- **Reduction-based Creative telescoping**
  Find an LDE satisfied by $I(t) = \int_\gamma f(x,t)dx$
  for f D-finite

Perimeter of an ellipse



$$p(e) = 2\int_{-1}^{1} \sqrt{\frac{1-e^2x^2}{1-x^2}}dx$$
$$(e-e^3)p''(e) + (e^2-1)p'(e) + ep(e) = 0$$

A unified algorithmic scheme

All previous problems can be seen as special instances of...

A unified algorithmic scheme

All previous problems can be seen as special instances of...

General Problem

Input: $T \in \mathsf{K}(x)^{n \times n}$, $a \in \mathsf{K}[x]^n$, let $\theta = \partial + T$

Output: $\eta = (\eta_0, \ldots, \eta_m) \in \mathsf{K}[x]^{m+1} \setminus \{0\}$ s.t.:

$$\eta_0 \cdot a + \eta_1 \cdot \theta a + \cdots + \eta_m \cdot \theta^m a = 0$$

A unified algorithmic scheme

All previous problems can be seen as special instances of...

General Problem

Input: $T \in \mathrm{K}(x)^{n\times n}$, $a \in \mathrm{K}[x]^n$, let $\theta = \partial + T$

Output: $\eta = (\eta_0, \dots, \eta_m) \in \mathrm{K}[x]^{m+1} \setminus \{0\}$ s.t.:

$$\eta_0 \cdot a + \eta_1 \cdot \theta a + \cdots + \eta_m \cdot \theta^m a = 0$$

$$\begin{bmatrix} a & & & \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \\ \\ \end{bmatrix} \stackrel{?}{=} 0$$

Pseudo-Krylov system

A unified algorithmic scheme

All previous problems can be seen as special instances of...

General Problem

Input: $T \in \mathsf{K}(x)^{n\times n}$, $a \in \mathsf{K}[x]^n$, let $\theta = \partial + T$

Output: $\eta = (\eta_0, \dots, \eta_m) \in \mathsf{K}[x]^{m+1} \setminus \{0\}$ s.t.:

$$\eta_0 \cdot a + \eta_1 \cdot \theta a + \cdots + \eta_m \cdot \theta^m a = 0$$

$$\begin{bmatrix} a & \theta a & & & \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \\ \\ \end{bmatrix} \overset{?}{=} 0$$

Pseudo-Krylov system

A unified algorithmic scheme

All previous problems can be seen as special instances of...

General Problem

Input: $T \in \mathsf{K}(x)^{n\times n}$, $a \in \mathsf{K}[x]^n$, let $\theta = \partial + T$

Output: $\eta = (\eta_0, \dots, \eta_m) \in \mathsf{K}[x]^{m+1} \setminus \{0\}$ s.t.:

$$\eta_0 \cdot a + \eta_1 \cdot \theta a + \cdots + \eta_m \cdot \theta^m a = 0$$

$$\begin{bmatrix} a & \theta a & \cdots & \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \end{bmatrix} \stackrel{?}{=} 0$$

Pseudo-Krylov system

A unified algorithmic scheme

All previous problems can be seen as special instances of...

General Problem

Input: $T \in \mathrm{K}(x)^{n\times n}$, $a \in \mathrm{K}[x]^n$, let $\theta = \partial + T$

Output: $\eta = (\eta_0, \dots, \eta_m) \in \mathrm{K}[x]^{m+1} \setminus \{0\}$ s.t.:

$$\eta_0 \cdot a + \eta_1 \cdot \theta a + \dots + \eta_m \cdot \theta^m a = 0$$

$$\begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix} = 0$$

Pseudo-Krylov system

A unified algorithmic scheme

All previous problems can be seen as special instances of...

General Problem

Input: $T \in \mathbb{K}(x)^{n\times n}$, $a \in \mathbb{K}[x]^n$, let $\theta = \partial + T$

Output: $\eta = (\eta_0, \ldots, \eta_m) \in \mathbb{K}[x]^{m+1} \setminus \{0\}$ s.t.:

$$\eta_0 \cdot a + \eta_1 \cdot \theta a + \cdots + \eta_m \cdot \theta^m a = 0$$

$$\begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix} = 0$$

Pseudo-Krylov system

Key fact: [G. 25,26] In all specific problems,

a structure inherited from $\boxed{T = XM^{-1}Y}$

with X, M, Y polynomial matrices of *small* degrees

A unified algorithmic scheme

All previous problems can be seen as special instances of...

General Problem

Input: $T \in \mathbb{K}(x)^{n\times n}$, $a \in \mathbb{K}[x]^n$, let $\theta = \partial + T$

Output: $\eta = (\eta_0, \dots, \eta_m) \in \mathbb{K}[x]^{m+1} \setminus \{0\}$ s.t.:

$$\eta_0 \cdot a + \eta_1 \cdot \theta a + \cdots + \eta_m \cdot \theta^m a = 0$$

$$\begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix} = 0$$

Pseudo-Krylov system

Key fact: [G. 25,26] In all specific problems,

a structure inherited from $\boxed{T = XM^{-1}Y}$

with X, M, Y polynomial matrices of *small* degrees

Exploit the structure for

A unified algorithmic scheme

All previous problems can be seen as special instances of...

General Problem

Input: $T \in \mathbb{K}(x)^{n\times n}$, $a \in \mathbb{K}[x]^n$, let $\theta = \partial + T$

Output: $\eta = (\eta_0, \dots, \eta_m) \in \mathbb{K}[x]^{m+1} \setminus \{0\}$ s.t.:

$$\eta_0 \cdot a + \eta_1 \cdot \theta a + \cdots + \eta_m \cdot \theta^m a = 0$$

$$\begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix} = 0$$

Pseudo-Krylov system

Key fact: [G. 25,26] In all specific problems,

a structure inherited from $\boxed{T = XM^{-1}Y}$

with X, M, Y polynomial matrices of *small* degrees

Exploit the structure for

- Degree bounds [G. 25,26]

A unified algorithmic scheme

All previous problems can be seen as special instances of...

General Problem

Input: $T \in \mathbb{K}(x)^{n\times n}$, $a \in \mathbb{K}[x]^n$, let $\theta = \partial + T$

Output: $\eta = (\eta_0, \dots, \eta_m) \in \mathbb{K}[x]^{m+1} \setminus \{0\}$ s.t.:

$$\eta_0 \cdot a + \eta_1 \cdot \theta a + \cdots + \eta_m \cdot \theta^m a = 0$$

$$\begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix} = 0$$

Pseudo-Krylov system

Key fact: [G. 25,26] In all specific problems,

a structure inherited from $\boxed{T = XM^{-1}Y}$

with X, M, Y polynomial matrices of *small* degrees

Exploit the structure for

- Degree bounds [G. 25,26]
- Solving the system **Now**

Example: Closure by sum

$L_1 = (x^2+1)\ \partial^2\ -(x+2)\ \partial\ -3$ and $L_2 = x^2\ \partial^2\ -(x+3)\ \partial\ -2$

Compute $L = L_1 \oplus L_2$

an operator s.t. $L(\alpha) = L(\alpha_1 + \alpha_2) = 0$ for all α_1, α_2 s.t. $L_1(\alpha_1) = L_2(\alpha_2) = 0$

Example: Closure by sum

$L_1 = (x^2+1)\ \partial^2\ -(x+2)\ \partial\ -3$ and $L_2 = x^2\ \partial^2\ -(x+3)\ \partial\ -2$

Compute $L = L_1 \oplus L_2$

an operator s.t. $L(\alpha) = L(\alpha_1 + \alpha_2) = 0$ for all α_1, α_2 s.t. $L_1(\alpha_1) = L_2(\alpha_2) = 0$

Algorithm: Express $\alpha^{(\ell)}$, for $\ell = 0, 1, \ldots,$ on the generating set $A = (\alpha_1, \alpha_1', \alpha_2, \alpha_2')$ until a linear relation is found

Example: Closure by sum

$L_1 = (x^2+1)\ \partial^2\ -(x+2)\ \partial\ -3$ and $L_2 = x^2\ \partial^2\ -(x+3)\ \partial\ -2$

Compute $L = L_1 \oplus L_2$

an operator s.t. $L(\alpha) = L(\alpha_1 + \alpha_2) = 0$ for all α_1, α_2 s.t. $L_1(\alpha_1) = L_2(\alpha_2) = 0$

Algorithm: Express $\alpha^{(\ell)}$, for $\ell = 0, 1, \ldots,$ on the generating set $A = (\alpha_1, \alpha_1', \alpha_2, \alpha_2')$ until a linear relation is found

$\alpha = \alpha_1 + \alpha_2$

$$\begin{bmatrix} 1 & & \\ 0 & & \\ 1 & & \\ 0 & & \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \\ \\ \\ \end{bmatrix} \overset{?}{=} 0$$

Example: Closure by sum

$L_1 = (x^2+1)\ \partial^2\ -(x+2)\ \partial\ -3$ and $L_2 = x^2\ \partial^2\ -(x+3)\ \partial\ -2$

Compute $L = L_1 \oplus L_2$

an operator s.t. $L(\alpha) = L(\alpha_1 + \alpha_2) = 0$ for all α_1, α_2 s.t. $L_1(\alpha_1) = L_2(\alpha_2) = 0$

Algorithm: Express $\alpha^{(\ell)}$, for $\ell = 0, 1, \ldots,$ on the generating set $A = (\alpha_1, \alpha_1', \alpha_2, \alpha_2')$ until a linear relation is found

$\alpha = \alpha_1 + \alpha_2 \quad \alpha' = \alpha_1' + \alpha_2'$

$$\begin{bmatrix} 1 & 0 & & \\ 0 & 1 & & \\ 1 & 0 & & \\ 0 & 1 & & \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \\ \\ \end{bmatrix} \overset{?}{=} 0$$

Example: Closure by sum

$L_1 = (x^2+1)\ \partial^2\ -(x+2)\ \partial\ -3$ and $L_2 = x^2\ \partial^2\ -(x+3)\ \partial\ -2$

Compute $L = L_1 \oplus L_2$

an operator s.t. $L(\alpha) = L(\alpha_1+\alpha_2) = 0$ for all α_1, α_2 s.t. $L_1(\alpha_1) = L_2(\alpha_2) = 0$

Algorithm: Express $\alpha^{(\ell)}$, for $\ell = 0, 1, \dots,$ on the generating set $A = (\alpha_1, \alpha_1', \alpha_2, \alpha_2')$ until a linear relation is found

$$\alpha = \alpha_1 + \alpha_2 \quad \alpha' = \alpha_1' + \alpha_2'$$

$$\alpha'' = \frac{3}{x^2+1}\alpha_1 + \frac{x+2}{x^2+1}\alpha_1' + \frac{2}{x^2}\alpha_2 + \frac{x+3}{x^2}\alpha_2'$$

$$\begin{bmatrix} 1 & 0 & \frac{3}{x^2+1} & & \\ 0 & 1 & \frac{x+2}{x^2+1} & & \\ 1 & 0 & \frac{2}{x^2} & & \\ 0 & 1 & \frac{x+3}{x^2} & & \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \eta_2 \\ \\ \\ \end{bmatrix} \stackrel{?}{=} 0$$

Example: Closure by sum

$$L_1 = (x^2+1)\ \partial^2\ -(x+2)\ \partial\ -3 \text{ and } L_2 = x^2\ \partial^2\ -(x+3)\ \partial\ -2$$

Compute $L = L_1 \oplus L_2$

an operator s.t. $L(\alpha) = L(\alpha_1+\alpha_2) = 0$ for all α_1, α_2 s.t. $L_1(\alpha_1) = L_2(\alpha_2) = 0$

Algorithm: Express $\alpha^{(\ell)}$, for $\ell = 0, 1, \ldots,$ on the generating set $A = (\alpha_1, \alpha_1', \alpha_2, \alpha_2')$ until a linear relation is found

$$\alpha = \alpha_1 + \alpha_2 \quad \alpha' = \alpha_1' + \alpha_2'$$

$$\alpha'' = \frac{3}{x^2+1}\alpha_1 + \frac{x+2}{x^2+1}\alpha_1' + \frac{2}{x^2}\alpha_2 + \frac{x+3}{x^2}\alpha_2'$$

$$\alpha^{(3)} = \cdots$$

$$\alpha^{(4)} = \cdots$$

$$\begin{bmatrix} 1 & 0 & \frac{3}{x^2+1} & & \\ 0 & 1 & \frac{x+2}{x^2+1} & \alpha^{(3)} & \alpha^{(4)} \\ 1 & 0 & \frac{2}{x^2} & & \\ 0 & 1 & \frac{x+3}{x^2} & & \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \eta_2 \\ \eta_3 \\ \eta_4 \end{bmatrix} = 0$$

Example: Closure by sum

$L_1 = (x^2+1)\ \partial^2\ -(x+2)\ \partial\ -3$ and $L_2 = x^2\ \partial^2\ -(x+3)\ \partial\ -2$

Compute $L = L_1 \oplus L_2$

an operator s.t. $L(\alpha) = L(\alpha_1 + \alpha_2) = 0$ for all α_1, α_2 s.t. $L_1(\alpha_1) = L_2(\alpha_2) = 0$

Algorithm: Express $\alpha^{(\ell)}$, for $\ell = 0, 1, \ldots,$ on the generating set $A = (\alpha_1, \alpha_1', \alpha_2, \alpha_2')$ until a linear relation is found

$$\alpha = \alpha_1 + \alpha_2 \quad \alpha' = \alpha_1' + \alpha_2'$$

$$\alpha'' = \frac{3}{x^2+1}\alpha_1 + \frac{x+2}{x^2+1}\alpha_1' + \frac{2}{x^2}\alpha_2 + \frac{x+3}{x^2}\alpha_2'$$

$$\alpha^{(3)} = \cdots$$

$$\alpha^{(4)} = \cdots$$

$$\begin{bmatrix} 1 & 0 & \frac{3}{x^2+1} & & \\ 0 & 1 & \frac{x+2}{x^2+1} & \alpha^{(3)} & \alpha^{(4)} \\ 1 & 0 & \frac{2}{x^2} & & \\ 0 & 1 & \frac{x+3}{x^2} & & \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \eta_2 \\ \eta_3 \\ \eta_4 \end{bmatrix} = 0$$

Solve the linear system and get $L = \eta_4 \cdot \partial^4 + \eta_3 \cdot \partial^3 + \eta_2 \cdot \partial^2 + \eta_1 \cdot \partial + \eta_0$

Closure by sum $T = XM^{-1}Y$

$L_1 = (x^2 + 1)\ \partial^2\ -(x + 2)\ \partial\ -3$ and $L_2 = x^2\ \partial^2\ -(x + 3)\ \partial\ -2$

Closure by sum $T = XM^{-1}Y$

$L_1 = (x^2+1)\ \partial^2\ -(x+2)\ \partial\ -3$ and $L_2 = x^2\ \partial^2\ -(x+3)\ \partial\ -2$

$\alpha^{(\ell)} = c_{1,0}(x)\alpha_1 + c_{1,1}(x)\alpha_1' + c_{2,0}(x)\alpha_2 + c_{2,1}(x)\alpha_2'$ represented by vector $V_\ell \in \mathrm{K}(x)^4$

Closure by sum $T = XM^{-1}Y$

$L_1 = (x^2+1)\ \partial^2\ -(x+2)\ \partial\ -3$ and $L_2 = x^2\ \partial^2\ -(x+3)\ \partial\ -2$

$\alpha^{(\ell)} = c_{1,0}(x)\alpha_1 + c_{1,1}(x)\alpha_1' + c_{2,0}(x)\alpha_2 + c_{2,1}(x)\alpha_2'$ represented by vector $V_\ell \in \mathrm{K}(x)^4$

Leibniz rule: $\boxed{V_{\ell+1} = \partial_x V_\ell + T \cdot V_\ell = \theta V_\ell}$

Closure by sum $T = XM^{-1}Y$

$L_1 = (x^2+1)\,\partial^2 \; -(x+2)\,\partial \; -3$ and $L_2 = x^2\,\partial^2 \; -(x+3)\,\partial \; -2$

$\alpha^{(\ell)} = c_{1,0}(x)\alpha_1 + c_{1,1}(x)\alpha_1' + c_{2,0}(x)\alpha_2 + c_{2,1}(x)\alpha_2'$ represented by vector $V_\ell \in \mathrm{K}(x)^4$

Leibniz rule: $\boxed{V_{\ell+1} = \partial_x V_\ell + T \cdot V_\ell = \theta V_\ell}$ with

$$T = \begin{bmatrix} 0 & \frac{3}{x^2+1} & & \\ 1 & \frac{x+2}{x^2+1} & & \\ & & 0 & \frac{2}{x^2} \\ & & 1 & \frac{x+3}{x^2} \end{bmatrix} = \begin{bmatrix} 0 & 3 & & \\ 1 & x+2 & & \\ & & 0 & 2 \\ & & 1 & x+3 \end{bmatrix} \cdot \begin{bmatrix} 1 & & & \\ & x^2+1 & & \\ & & 1 & \\ & & & x^2 \end{bmatrix}^{-1}$$

Solving pseudo-Krylov systems

Assume $T = XM^{-1}Y \in \mathsf{K}(x)^{n\times n}$ with $\Delta = \det M$ and degree $\delta = \deg \Delta$

Solving pseudo-Krylov systems

Assume $T = XM^{-1}Y \in \mathsf{K}(x)^{n\times n}$ with $\Delta = \det M$ and degree $\delta = \deg \Delta$

$$\begin{bmatrix} \vdots & \vdots & & \vdots \\ a & \theta a & \cdots & \theta^m a \\ \vdots & \vdots & & \vdots \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix} = 0$$

Assume $T = XM^{-1}Y \in \mathsf{K}(x)^{n\times n}$ with $\Delta = \det M$ and degree $\delta = \deg \Delta$

Degree of the column denominator:

$$\begin{bmatrix} 0 & & & \\ \vdots & \vdots & & \vdots \\ a & \theta a & \cdots & \theta^m a \\ \vdots & \vdots & & \vdots \\ & & & \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix} = 0$$

Solving pseudo-Krylov systems

Assume $T = XM^{-1}Y \in \mathbb{K}(x)^{n\times n}$ with $\Delta = \det M$ and degree $\delta = \deg \Delta$

Degree of the column denominator:

$$\begin{bmatrix} 0 & \delta & & \\ \vdots & \vdots & & \vdots \\ a & \theta a & \cdots & \theta^m a \\ \vdots & \vdots & & \vdots \\ & & & \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix} = 0$$

Solving pseudo-Krylov systems

Assume $T = XM^{-1}Y \in \mathsf{K}(x)^{n\times n}$ with $\Delta = \det M$ and degree $\delta = \deg \Delta$

$$\begin{array}{c c}
\begin{array}{l}\text{Degree of the}\\ \text{column denominator:}\end{array} & \begin{array}{cccc} 0 & \delta & \cdots & m\delta \end{array}\\
 & \begin{bmatrix} \vdots & \vdots & & \vdots \\ a & \theta a & \cdots & \theta^m a \\ \vdots & \vdots & & \vdots \\ & & & \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix} = 0
\end{array}$$

Solving pseudo-Krylov systems

Assume $T = XM^{-1}Y \in \mathsf{K}(x)^{n\times n}$ with $\Delta = \det M$ and degree $\delta = \deg \Delta$

$$
\begin{array}{c}
\text{Degree of the column denominator:} \\
\\
\\
\\
\end{array}
\begin{bmatrix}
0 & \delta & \cdots & m\delta \\
\vdots & \vdots & & \vdots \\
a & \theta a & \cdots & \theta^m a \\
\vdots & \vdots & & \vdots \\
& & &
\end{bmatrix}
\cdot
\begin{bmatrix}
\eta_0 \\ \eta_1 \\ \vdots \\ \eta_m
\end{bmatrix}
= 0
$$

$m = O(n)$
$\omega =$ exponent for matrix multiplication

Reduces to **Polynomial linear algebra** in degree $O(n\delta)$: $\widetilde{O}(n^{\omega+1}\delta)$ [Zhou Labahn Storjohann 12]

Best known **complexity** for all specific problems except closure properties

Solving pseudo-Krylov systems

Assume $T = XM^{-1}Y \in \mathsf{K}(x)^{n\times n}$ with $\Delta = \det M$ and degree $\delta = \deg \Delta$

Degree of the column denominator:

$$\begin{bmatrix} 0 & \delta & \cdots & m\delta \\ \vdots & \vdots & & \vdots \\ a & \theta a & \cdots & \theta^m a \\ \vdots & \vdots & & \vdots \\ & & & \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix} = 0$$

$m = O(n)$
$\omega =$ exponent for matrix multiplication

Reduces to **Polynomial linear algebra** in degree $O(n\delta)$: $\widetilde{O}(n^{\omega+1}\delta)$ [Zhou Labahn Storjohann 12]

Best known **complexity** for all specific problems except closure properties

⚠ **Size of output** in $O(n^2\delta)$ [G. 25 & 26]

Solving pseudo-Krylov systems

Assume $T = XM^{-1}Y \in \mathsf{K}(x)^{n\times n}$ with $\Delta = \det M$ and degree $\delta = \deg \Delta$

Degree of the column denominator:

$$\begin{array}{c} \begin{array}{cccc} 0 & \delta & \cdots & m\delta \end{array} \\ \left[\begin{array}{cccc} \vdots & \vdots & & \vdots \\ a & \theta a & \cdots & \theta^m a \\ \vdots & \vdots & & \vdots \\ & & & \end{array}\right] \end{array} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix} = 0$$

$m = O(n)$
$\omega =$ exponent for matrix multiplication

Reduces to **Polynomial linear algebra** in degree $O(n\delta)$: $\widetilde{O}(n^{\omega+1}\delta)$ [Zhou Labahn Storjohann 12]

Best known **complexity** for all specific problems except closure properties

⚠ **Size of output** in $O(n^2\delta)$ [G. 25 & 26]

Target complexity: $\widetilde{O}(n^{\omega}\delta)$

Width of T

Smith normal form:

$$VMU = \begin{bmatrix} \gamma_1 & & \\ & \ddots & \\ & & \gamma_h \end{bmatrix} \qquad V, U \in \mathsf{K}[x]^{h\times h} \text{ unimodular, } \gamma_i \mid \gamma_{i+1}$$

$$\gamma_i = \text{gcd of all minors of dimension } i \text{ of } M$$

Generically, $\gamma_i = 1$ for $i < h$ and $\gamma_h = \det M = \Delta$.

Width of T

Smith normal form:

$$VMU = \begin{bmatrix} \gamma_1 & & \\ & \ddots & \\ & & \gamma_h \end{bmatrix} \quad V, U \in \mathsf{K}[x]^{h\times h} \text{ unimodular, } \gamma_i \mid \gamma_{i+1}$$

$$\gamma_i = \gcd \text{ of all minors of dimension } i \text{ of } M$$

Generically, $\gamma_i = 1$ for $i < h$ and $\gamma_h = \det M = \Delta$. In that case,

$$\Delta M^{-1} = U \begin{bmatrix} \Delta & & & \\ & \ddots & & \\ & & \Delta & \\ & & & 1 \end{bmatrix} V, \text{ and } \Delta T = \Delta X M^{-1} Y = uv^t \mod \Delta$$

T has width 1

Exploit the structure for T of width 1

$\Delta = \det M, \delta = \deg \Delta$

T admits a rank 1 decomposition

$$T = XM^{-1}Y = \frac{uv^t}{\Delta} + W, \quad u, v \in \mathsf{K}[x]^n_{<\delta}, W \in \mathsf{K}[x]^{n\times n}_{<\delta}$$

Exploit the structure for T of width 1

$\Delta = \det M, \delta = \deg \Delta$

T admits a rank 1 decomposition

$$T = XM^{-1}Y = \frac{uv^t}{\Delta} + W, \quad u, v \in \mathrm{K}[x]^n_{<\delta}, W \in \mathrm{K}[x]^{n\times n}_{<\delta}$$

Reduction: $\forall f \in \mathrm{K}[x]^n_{<\delta}$, $\boxed{\theta f = \beta \cdot u/\Delta + r}$ with $\beta \in \mathrm{K}[x]_{<\delta}, r \in \mathrm{K}[x]^n_{<\delta}$

Exploit the structure for T of width 1

$\Delta = \det M, \delta = \deg \Delta$

T admits a rank 1 decomposition

$$T = XM^{-1}Y = \frac{uv^t}{\Delta} + W, \quad u, v \in \mathrm{K}[x]^n_{<\delta}, W \in \mathrm{K}[x]^{n\times n}_{<\delta}$$

Reduction: $\forall f \in \mathrm{K}[x]^n_{<\delta}$, $\boxed{\theta f = \beta \cdot u/\Delta + r}$ with $\beta \in \mathrm{K}[x]_{<\delta}, r \in \mathrm{K}[x]^n_{<\delta}$

allows to compute a **realisation**

$K = \begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} = F + N \cdot D^{-1} \cdot A$ with F, N, D, A polynomial matrices of degree $\leq \delta$

Exploit the structure for T of width 1

$\Delta = \det M, \delta = \deg \Delta$

T admits a rank 1 decomposition

$$T = XM^{-1}Y = \frac{uv^t}{\Delta} + W, \quad u, v \in \mathrm{K}[x]^n_{<\delta}, W \in \mathrm{K}[x]^{n\times n}_{<\delta}$$

Reduction: $\forall f \in \mathrm{K}[x]^n_{<\delta}$, $\boxed{\theta f = \beta \cdot u/\Delta + r}$ with $\beta \in \mathrm{K}[x]_{<\delta}, r \in \mathrm{K}[x]^n_{<\delta}$

allows to compute a **realisation**

$$K = \begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} = F + N \cdot D^{-1} \cdot A$$ with F, N, D, A polynomial matrices of degree $\leq \delta$

Schur complement trick: $K \cdot \eta = 0 \iff \exists z, \begin{bmatrix} D & A \\ -N & F \end{bmatrix} \cdot \begin{bmatrix} z \\ \eta \end{bmatrix} = 0$

Polynomial linear algebra in degree δ!

New algorithm

A **unified** and **fast** algorithm solving pseudo-Krylov systems for T of width 1

On input matrix $T = XM^{-1}Y \in \mathsf{K}(x)^{n\times n}$, $a \in \mathsf{K}[x]^n$, and $m \geq 1$

1. Compute vectors u, v
   in a rank 1 decomposition of T
2. Build the polynomial matrices F, N, D, A
3. Compute $\begin{bmatrix} Z \\ H \end{bmatrix}$ a minimal basis of $\ker \begin{bmatrix} D & A \\ -N & F \end{bmatrix}$
4. Return H

$$K = F + ND^{-1}A \in \mathsf{K}(x)^{n\times(m+1)}$$

New algorithm

A **unified** and **fast** algorithm solving pseudo-Krylov systems for T of width 1

On input matrix $T = XM^{-1}Y \in \mathsf{K}(x)^{n\times n}$, $a \in \mathsf{K}[x]^n$, and $m \geq 1$

1. Compute vectors u, v in a rank 1 decomposition of T
2. Build the polynomial matrices F, N, D, A
3. Compute $\begin{bmatrix} Z \\ H \end{bmatrix}$ a minimal basis of $\ker \begin{bmatrix} D & A \\ -N & F \end{bmatrix}$
4. Return H

$K = F + ND^{-1}A \in \mathsf{K}(x)^{n\times(m+1)}$

1.,2. $\rightarrow \widetilde{O}(n^2\delta)$ in applications

New algorithm

A **unified** and **fast** algorithm solving pseudo-Krylov systems for T of width 1

On input matrix $T = XM^{-1}Y \in \mathsf{K}(x)^{n\times n}$, $a \in \mathsf{K}[x]^n$, and $m \geq 1$

1. Compute vectors u, v in a rank 1 decomposition of T
2. Build the polynomial matrices F, N, D, A
3. Compute $\begin{bmatrix} Z \\ H \end{bmatrix}$ a minimal basis of $\ker \begin{bmatrix} D & A \\ -N & F \end{bmatrix}$
4. Return H

$K = F + ND^{-1}A \in \mathsf{K}(x)^{n\times(m+1)}$

1.,2. $\to \widetilde{O}(n^2\delta)$ in applications

$\to \widetilde{O}\left(n^\omega \delta\right)$ [Zhou Labahn Storjohann 12]

Choice of m

$$K \cdot \eta = \begin{bmatrix} a & \theta a & \cdots & \theta^{m} a \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_{m} \end{bmatrix} = 0$$

Choice of m

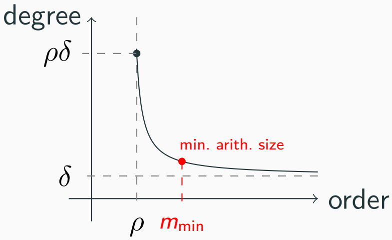
$$K \cdot \eta = \begin{bmatrix} a & \theta a & \cdots & \theta^{m} a \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix} = 0$$

- Minimal order LDE $m = \rho$: $\dim \ker K = 1$
  Ex.: **Least** Common Left Multiple, **Minimal** order telescoper

$$\mathrm{ord}(L) = \rho \leq n \quad \deg_x(L) = O(\rho\delta)$$

Choice of m

$$K \cdot \eta = \begin{bmatrix} a & \theta a & \cdots & \theta^{m} a \end{bmatrix} \cdot \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix} = 0$$



[Chen Jaroschek Kauers Singer 13] [G. 26]

- Minimal order LDE $m = \rho$: $\dim\ker K = 1$

  Ex.: **Least** Common Left Multiple, **Minimal** order telescoper

$$\mathrm{ord}(L) = \rho \leq n \quad \deg_x(L) = O(\rho\delta)$$

- Higher order LDE $m = \rho + \sigma$: $\dim\ker K = \sigma + 1$, there exists L s.t.

$$\mathrm{ord}(L) = m \quad \deg_x(L) = O(\frac{m}{\sigma+1}\delta)$$

Resulting algorithms for our instances

For pseudo-Krylov systems coming from applications,
it frequently results in a **fast** algorithm for generic inputs:

Resulting algorithms for our instances

For pseudo-Krylov systems coming from applications,
it frequently results in a **fast** algorithm for generic inputs:

- Improve best known complexity bound by one order of magnitude $\widetilde{O}(n^{\omega}\delta)$ vs. $\widetilde{O}(n^{\omega+1}\delta)$
  - **Algeqtodiffeq** [Bostan Chyzak Salvy Lecerf Schost 07] and extension
  - **Creative telescoping** for simple integration of D-finite functions of order ≤ 1

    [Bostan Chen Chyzak Li 10, Bostan Chen Chyzak Li Xin 13, Chen Huang Kauers Li 15, Bostan Dumont Salvy 16]

Resulting algorithms for our instances

For pseudo-Krylov systems coming from applications,
it frequently results in a **fast** algorithm for generic inputs:

- Improve best known complexity bound by one order of magnitude $\widetilde{O}(n^\omega \delta)$ vs. $\widetilde{O}(n^{\omega+1}\delta)$
  - **Algeqtodiffeq** [Bostan Chyzak Salvy Lecerf Schost 07] and extension
  - **Creative telescoping** for simple integration of D-finite functions of order ≤ 1

    [Bostan Chen Chyzak Li 10, Bostan Chen Chyzak Li Xin 13, Chen Huang Kauers Li 15, Bostan Dumont Salvy 16]
- Retrieve best known complexity bound
  - **Closure by sum** [Bostan Chyzak Salvy Li 12] [van der Hoeven 16]

Resulting algorithms for our instances

For pseudo-Krylov systems coming from applications,
it frequently results in a **fast** algorithm for generic inputs:

- Improve best known complexity bound by one order of magnitude $\widetilde{O}(n^\omega \delta)$ vs. $\widetilde{O}(n^{\omega+1}\delta)$
  - **Algeqtodiffeq** [Bostan Chyzak Salvy Lecerf Schost 07] and extension
  - **Creative telescoping** for simple integration of D-finite functions of order ≤ 1

    [Bostan Chen Chyzak Li 10, Bostan Chen Chyzak Li Xin 13, Chen Huang Kauers Li 15, Bostan Dumont Salvy 16]
- Retrieve best known complexity bound
  - **Closure by sum** [Bostan Chyzak Salvy Li 12] [van der Hoeven 16]

Yet for **Closure by product** and its extensions, *To be continued*
the matrices T involved do not admit a rank 1 decomposition

Computation of the realisation

$\Delta = \det M, \delta = \deg \Delta$

Wanted: $K = \begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} = F + N \cdot D^{-1} \cdot A$

Reduction: $\forall f \in \mathbb{K}[x]^n_{<\delta}$, $\boxed{\theta f = \beta \cdot u / \Delta + r}$ with $\beta \in \mathbb{K}[x]_{<\delta}, r \in \mathbb{K}[x]^n_{<\delta}$

Computation of the realisation

$\Delta = \det M, \delta = \deg \Delta$

Wanted: $K = \begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} = F + N \cdot D^{-1} \cdot A$

Reduction: $\forall f \in \mathbb{K}[x]^n_{<\delta}$, $\boxed{\theta f = \beta \cdot u/\Delta + r}$ with $\beta \in \mathbb{K}[x]_{<\delta}, r \in \mathbb{K}[x]^n_{<\delta}$

$K_i = \theta^i a,\ Q_i = \theta^{i-1}(u/\Delta)$

Computation of the realisation

$\Delta = \det M, \delta = \deg \Delta$

Wanted: $K = \begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} = F + N \cdot D^{-1} \cdot A$

Reduction: $\forall f \in \mathbb{K}[x]^n_{<\delta}$, $\boxed{\theta f = \beta \cdot u/\Delta + r}$ with $\beta \in \mathbb{K}[x]_{<\delta}, r \in \mathbb{K}[x]^n_{<\delta}$

$K_i = \theta^i a,\ Q_i = \theta^{i-1}(u/\Delta)$

- $K_1 = \theta a =$ $\beta_1 Q_1 + f_1$

Computation of the realisation

$\Delta = \det M, \delta = \deg \Delta$

Wanted: $K = \begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} = F + N \cdot D^{-1} \cdot A$

Reduction: $\forall f \in \mathbb{K}[x]^n_{<\delta}$, $\boxed{\theta f = \beta \cdot u/\Delta + r}$ with $\beta \in \mathbb{K}[x]_{<\delta}, r \in \mathbb{K}[x]^n_{<\delta}$

$K_i = \theta^i a,\ Q_i = \theta^{i-1}(u/\Delta)$

- $K_1 = \theta a =$ $\alpha_{1,1} Q_1 + f_1$

Computation of the realisation

$\Delta = \det M, \delta = \deg \Delta$

Wanted: $K = \begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} = F + N \cdot D^{-1} \cdot A$

Reduction: $\forall f \in \mathbb{K}[x]^n_{<\delta}$, $\boxed{\theta f = \beta \cdot u/\Delta + r}$ with $\beta \in \mathbb{K}[x]_{<\delta}, r \in \mathbb{K}[x]^n_{<\delta}$

$K_i = \theta^i a,\ Q_i = \theta^{i-1}(u/\Delta)$

- $K_1 = \theta a = \alpha_{1,1} Q_1 + f_1$
- $K_2 = \alpha_{1,1} Q_2 + (\alpha'_{1,1} + \beta_2) Q_1 + f_2$

Computation of the realisation

$\Delta = \det M, \delta = \deg \Delta$

Wanted: $K = \begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} = F + N \cdot D^{-1} \cdot A$

Reduction: $\forall f \in \mathbb{K}[x]^n_{<\delta}$, $\boxed{\theta f = \beta \cdot u/\Delta + r}$ with $\beta \in \mathbb{K}[x]_{<\delta}, r \in \mathbb{K}[x]^n_{<\delta}$

$K_i = \theta^i a,\ Q_i = \theta^{i-1}(u/\Delta)$

- $K_1 = \theta a = \alpha_{1,1} Q_1 + f_1$
- $K_2 = \alpha_{2,2} Q_2 + \alpha_{1,2} Q_1 + f_2$

Computation of the realisation

$\Delta = \det M, \delta = \deg \Delta$

Wanted: $K = \begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} = F + N \cdot D^{-1} \cdot A$

Reduction: $\forall f \in \mathbb{K}[x]^n_{<\delta}$, $\boxed{\theta f = \beta \cdot u/\Delta + r}$ with $\beta \in \mathbb{K}[x]_{<\delta}, r \in \mathbb{K}[x]^n_{<\delta}$

$K_i = \theta^i a,\ Q_i = \theta^{i-1}(u/\Delta)$

- $K_1 = \theta a = \alpha_{1,1} Q_1 + f_1$
- $K_2 = \alpha_{2,2} Q_2 + \alpha_{1,2} Q_1 + f_2$
- $K_3 = \alpha_{2,2} Q_3 + (\alpha'_{2,2} + \alpha_{1,2}) Q_2 + (\alpha'_{1,2} + \beta_3) Q_1 + f_3$

Computation of the realisation

$\Delta = \det M, \delta = \deg \Delta$

Wanted: $K = \begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} = F + N \cdot D^{-1} \cdot A$

Reduction: $\forall f \in \mathbb{K}[x]^n_{<\delta}$, $\boxed{\theta f = \beta \cdot u/\Delta + r}$ with $\beta \in \mathbb{K}[x]_{<\delta}, r \in \mathbb{K}[x]^n_{<\delta}$

$K_i = \theta^i a,\ Q_i = \theta^{i-1}(u/\Delta)$

- $K_1 = \theta a = \alpha_{1,1} Q_1 + f_1$
- $K_2 = \alpha_{2,2} Q_2 + \alpha_{1,2} Q_1 + f_2$
- $K_3 = \alpha_{3,3} Q_3 + \alpha_{2,3} Q_2 + \alpha_{1,3} Q_1 + f_3$

θ

Computation of the realisation

$\Delta = \det M, \delta = \deg \Delta$

Wanted: $K = \begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} = F + N \cdot D^{-1} \cdot A$

Reduction: $\forall f \in \mathbb{K}[x]^n_{<\delta}$, $\boxed{\theta f = \beta \cdot u/\Delta + r}$ with $\beta \in \mathbb{K}[x]_{<\delta}, r \in \mathbb{K}[x]^n_{<\delta}$

$K_i = \theta^i a,\ Q_i = \theta^{i-1}(u/\Delta)$

- $K_1 = \theta a = \alpha_{1,1} Q_1 + f_1$
- $K_2 = \alpha_{2,2} Q_2 + \alpha_{1,2} Q_1 + f_2$
- $K_3 = \alpha_{3,3} Q_3 + \alpha_{2,3} Q_2 + \alpha_{1,3} Q_1 + f_3$
- $\vdots$

Computation of the realisation

$\Delta = \det M, \delta = \deg \Delta$

Wanted: $K = \begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} = F + N \cdot D^{-1} \cdot A$

Reduction: $\forall f \in \mathbb{K}[x]^n_{<\delta}$, $\boxed{\theta f = \beta \cdot u/\Delta + r}$ with $\beta \in \mathbb{K}[x]_{<\delta}, r \in \mathbb{K}[x]^n_{<\delta}$

$K_i = \theta^i a, \; Q_i = \theta^{i-1}(u/\Delta)$

- $K_1 = \theta a = \alpha_{1,1} Q_1 + f_1$
- $K_2 = \alpha_{2,2} Q_2 + \alpha_{1,2} Q_1 + f_2$
- $K_3 = \alpha_{3,3} Q_3 + \alpha_{2,3} Q_2 + \alpha_{1,3} Q_1 + f_3$
- $\vdots$
- $K_m = \alpha_{m,m} Q_m + \cdots + \alpha_{2,m} Q_2 + \alpha_{1,m} Q_1 + f_m$

Computation of the realisation

$\Delta = \det M, \delta = \deg \Delta$

Wanted: $K = \begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} = F + N \cdot D^{-1} \cdot A$

Reduction: $\forall f \in \mathbb{K}[x]^n_{<\delta}$, $\boxed{\theta f = \beta \cdot u/\Delta + r}$ with $\beta \in \mathbb{K}[x]_{<\delta}, r \in \mathbb{K}[x]^n_{<\delta}$

$K_i = \theta^i a,\ Q_i = \theta^{i-1}(u/\Delta)$

- $K_1 = \theta a = \alpha_{1,1} Q_1 + f_1$
- $K_2 = \alpha_{2,2} Q_2 + \alpha_{1,2} Q_1 + f_2$
- $K_3 = \alpha_{3,3} Q_3 + \alpha_{2,3} Q_2 + \alpha_{1,3} Q_1 + f_3$
- $\vdots$
- $K_m = \alpha_{m,m} Q_m + \cdots + \alpha_{2,m} Q_2 + \alpha_{1,m} Q_1 + f_m$

$$K = \begin{bmatrix} a & \overline{F} \end{bmatrix} + \begin{bmatrix} 0 & \\ & \overline{Q} \end{bmatrix} \cdot \begin{bmatrix} 0 & \\ & \overline{A} \end{bmatrix} = F + Q \cdot A$$

$$\overline{F} = \begin{bmatrix} f_1 & \cdots & f_m \end{bmatrix}, \quad \overline{Q} = \begin{bmatrix} Q_1 & \cdots & Q_m \end{bmatrix}$$

$$\overline{A} = \begin{bmatrix} \alpha_{1,1} & \cdots & \alpha_{1,m} \\ & \ddots & \vdots \\ & & \alpha_{m,m} \end{bmatrix}$$

Differential Toeplitz structure

Computation of the realisation

$\Delta = \det M, \delta = \deg \Delta$

Wanted: $K = \begin{bmatrix} a & \theta a & \cdots & \theta^m a \end{bmatrix} = F + N \cdot D^{-1} \cdot A$

Reduction: $\forall f \in \mathbb{K}[x]^n_{<\delta}$, $\boxed{\theta f = \beta \cdot u/\Delta + r}$ with $\beta \in \mathbb{K}[x]_{<\delta}, r \in \mathbb{K}[x]^n_{<\delta}$

$K_i = \theta^i a,\ Q_i = \theta^{i-1}(u/\Delta)$

Similar reasoning on the Q_i's leads to

$$\overline{Q} = \overline{N} \cdot \overline{D}^{-1}$$

Thus,

$$N = \begin{bmatrix} 0 & \overline{N} \end{bmatrix}, \quad D = \begin{bmatrix} 1 & \\ & \overline{D} \end{bmatrix}^{-1}$$

$$K = \begin{bmatrix} a & \overline{F} \end{bmatrix} + \begin{bmatrix} 0 & \\ & \overline{Q} \end{bmatrix} \cdot \begin{bmatrix} 0 & \\ & \overline{A} \end{bmatrix} = F + Q \cdot A$$

$$\overline{F} = \begin{bmatrix} f_1 & \cdots & f_m \end{bmatrix}, \quad \overline{Q} = \begin{bmatrix} Q_1 & \cdots & Q_m \end{bmatrix}$$

$$\overline{A} = \begin{bmatrix} \alpha_{1,1} & \cdots & \alpha_{1,m} \\ & \ddots & \vdots \\ & & \alpha_{m,m} \end{bmatrix}$$

Differential Toeplitz structure

Conclusion

- **Unified** & **fast** algorithm for pseudo-Krylov systems for generic $T = XM^{-1}Y$
- Frequently yields a fast(er) algorithm for generic inputs in applications
- Convenient to implement using existing softwares
- Direct analogue for recurrences

Perspectives

- What can be done when no rank 1 decomposition of T exists?
- Handle more general classes of functions for Creative telescoping

  [Bostan Chyzak Lairez Salvy 2018, Brochet Salvy 2024]
- Efficient implementation of our approach
  tested on families of examples coming from applications

Thank you for your attention!

Example 2: Algeqtodiffeq

Series $C(x)$ root of polynomial $P(x, y) = p_e(x)y^e + \cdots + p_1(x)y + p_0(x)$

Example 2: Algeqtodiffeq

Series $C(x)$ root of polynomial $P(x,y) = p_e(x)y^e + \cdots + p_1(x)y + p_0(x)$

Cockle's algorithm: Compute $V_\ell \in \mathrm{K}(x)[y]/(P)$ such that $C^{(\ell)}(x) = V_\ell(x, C(x))$

[Bostan Chyzak Salvy Lecerf Schost 07] $V_\ell(x,y) = v_{0,\ell}(x) + \cdots + v_{e-1,\ell}(x) \cdot y^{d-1}$

LDE obtained as a linear relation between the V_ℓ's

Example 2: Algeqtodiffeq

Series $C(x)$ root of polynomial $P(x,y) = p_e(x)y^e + \cdots + p_1(x)y + p_0(x)$

Cockle's algorithm: Compute $V_\ell \in \mathrm{K}(x)[y]/(P)$ such that $C^{(\ell)}(x) = V_\ell(x, C(x))$

[Bostan Chyzak Salvy Lecerf Schost 07] $V_\ell(x,y) = v_{0,\ell}(x) + \cdots + v_{e-1,\ell}(x) \cdot y^{d-1}$

LDE obtained as a linear relation between the V_ℓ's

$$C^{(\ell+1)}(x) = \left(C^{(\ell)}(x)\right)' = (V_\ell(x, C(x)))'$$

Example 2: Algeqtodiffeq

Series $C(x)$ root of polynomial $P(x,y) = p_e(x)y^e + \cdots + p_1(x)y + p_0(x)$

Cockle's algorithm: Compute $V_\ell \in \mathrm{K}(x)[y]/(P)$ such that $C^{(\ell)}(x) = V_\ell(x, C(x))$

[Bostan Chyzak Salvy Lecerf Schost 07] $V_\ell(x,y) = v_{0,\ell}(x) + \cdots + v_{e-1,\ell}(x) \cdot y^{d-1}$

LDE obtained as a linear relation between the V_ℓ's

$$C^{(\ell+1)}(x) = \left(C^{(\ell)}(x)\right)' = (V_\ell(x, C(x)))'$$
$$= \frac{\partial V_\ell}{\partial x}\bigg|_{y=C(x)} + C'(x)\frac{\partial V_\ell}{\partial y}\bigg|_{y=C(x)}$$

Example 2: Algeqtodiffeq

Series $C(x)$ root of polynomial $P(x, y) = p_e(x)y^e + \cdots + p_1(x)y + p_0(x)$

Cockle's algorithm: Compute $V_\ell \in \mathrm{K}(x)[y]/(P)$ such that $C^{(\ell)}(x) = V_\ell(x, C(x))$

[Bostan Chyzak Salvy Lecerf Schost 07] $V_\ell(x, y) = v_{0,\ell}(x) + \cdots + v_{e-1,\ell}(x) \cdot y^{d-1}$

LDE obtained as a linear relation between the V_ℓ's

$$C^{(\ell+1)}(x) = \left(C^{(\ell)}(x)\right)' = (V_\ell(x, C(x)))'$$

$$= \frac{\partial V_\ell}{\partial x}\Big|_{y=C(x)} + C'(x)\frac{\partial V_\ell}{\partial y}\Big|_{y=C(x)}$$

$$= (\partial_x V_\ell + V_1 \cdot \partial_y V_\ell)(x, C(x))$$

Example 2: Algeqtodiffeq

Series $C(x)$ root of polynomial $P(x,y) = p_e(x)y^e + \cdots + p_1(x)y + p_0(x)$

Cockle's algorithm: Compute $V_\ell \in \mathrm{K}(x)[y]/(P)$ such that $C^{(\ell)}(x) = V_\ell(x, C(x))$

[Bostan Chyzak Salvy Lecerf Schost 07] $V_\ell(x,y) = v_{0,\ell}(x) + \cdots + v_{e-1,\ell}(x) \cdot y^{d-1}$

LDE obtained as a linear relation between the V_ℓ's

$$\begin{aligned}
C^{(\ell+1)}(x) &= \left(C^{(\ell)}(x)\right)' = (V_\ell(x, C(x)))' \\
&= \frac{\partial V_\ell}{\partial x}\Big|_{y=C(x)} + C'(x)\frac{\partial V_\ell}{\partial y}\Big|_{y=C(x)} \\
&= (\partial_x V_\ell + V_1 \cdot \partial_y V_\ell)\,(x, C(x))
\end{aligned}$$

$$\begin{aligned}
0 &= P(x, C(x)) \\
0 &= P_x(x, C(x)) + C'(x) P_y(x, C(x)) = 0 \\
C'(x) &= -\frac{P_x(x, C(x))}{P_y(x, C(x))} \text{ and } V_1 = -\frac{P_x}{P_y} \mod P
\end{aligned}$$

Example 2: Algeqtodiffeq

Series $C(x)$ root of polynomial $P(x,y) = p_e(x)y^e + \cdots + p_1(x)y + p_0(x)$

Cockle's algorithm: Compute $V_\ell \in \mathrm{K}(x)[y]/(P)$ such that $C^{(\ell)}(x) = V_\ell(x, C(x))$

[Bostan Chyzak Salvy Lecerf Schost 07] $V_\ell(x,y) = v_{0,\ell}(x) + \cdots + v_{e-1,\ell}(x) \cdot y^{d-1}$

LDE obtained as a linear relation between the V_ℓ's

$$\begin{aligned} C^{(\ell+1)}(x) &= \left(C^{(\ell)}(x)\right)' = (V_\ell(x, C(x)))' \\ &= \frac{\partial V_\ell}{\partial x}\bigg|_{y=C(x)} + C'(x)\frac{\partial V_\ell}{\partial y}\bigg|_{y=C(x)} \\ &= (\partial_x V_\ell + V_1 \cdot \partial_y V_\ell)(x, C(x)) \end{aligned}$$

$$\begin{aligned} 0 &= P(x, C(x)) \\ 0 &= P_x(x, C(x)) + C'(x)P_y(x, C(x)) = 0 \\ C'(x) &= -\frac{P_x(x, C(x))}{P_y(x, C(x))} \text{ and } V_1 = -\frac{P_x}{P_y} \mod P \end{aligned}$$

$$\begin{cases} V_{\ell+1} & = (\partial_x + T) \cdot V_\ell, \quad V_0 = y \\ T\colon v & \mapsto -\frac{P_x}{P_y} \cdot \partial_y(v) \mod P \end{cases}$$

Algeqtodiffeq $T = XM^{-1}Y$

$e = \deg_y(P) \quad \mathcal{A}_s = \mathsf{K}(x)[y]_{<s}$

$$\begin{aligned} T\colon \mathcal{A}_e &\to \mathcal{A}_e \\ v &\mapsto -\frac{P_x}{P_y} \cdot \partial_y(v) \mod P \end{aligned}$$

Algeqtodiffeq $T = XM^{-1}Y$

$e = \deg_y(P) \quad \mathcal{A}_s = \mathsf{K}(x)[y]_{<s}$

$$\begin{aligned} T\colon \mathcal{A}_e &\to \mathcal{A}_e \\ v &\mapsto -\frac{P_x}{P_y} \cdot \partial_y(v) \mod P \end{aligned}$$

$$\begin{tikzcd}
v \in \mathcal{A}_e \arrow[rr, bend left, "T"] \arrow[d, "Y"] & & Tv = U \in \mathcal{A}_e \\
w = -P_x \partial_y(v) \in \mathcal{A}_{2e-1} \arrow[r, bend right, "M^{-1} = \mathrm{Syl}^{-1}(P, P_y)"'] & \begin{array}{c}(U, V) \in \mathcal{A}_e \times \mathcal{A}_{e-1} \\ \text{s.t. } w = UP_y + VP\end{array} \arrow[ur, "X"] &
\end{tikzcd}$$

Computation of a rank one decomposition

Assumption: T admits a rank one decomposition $T = uv^t/\Delta + W$

How to compute u, v in $\widetilde{O}(n^2\delta)$?

Computation of a rank one decomposition

Assumption: T admits a rank one decomposition $T = uv^t/\Delta + W$

How to compute u, v in $\widetilde{O}(n^2\delta)$?

- Smith form of $B = \Delta T \implies \exists w, z$ s.t. $\gcd(w^t Bz, \Delta) = 1$

$$\mathbf{u} = Bz, \quad \mathbf{v}^t = w^t B, \quad e = w^t Bz \mod \Delta$$

Then, $\boxed{\Delta T = \mathbf{u}\mathbf{v}^t/e \mod \Delta}$

Computation of a rank one decomposition

Assumption: T admits a rank one decomposition $T = uv^t/\Delta + W$

How to compute u, v in $\widetilde{O}(n^2\delta)$?

- Smith form of $B = \Delta T \implies \exists w, z$ s.t. $\gcd(w^t B z, \Delta) = 1$

$$\mathbf{u} = Bz, \quad \mathbf{v}^t = w^t B, \quad e = w^t B z \mod \Delta$$

  Then, $\boxed{\Delta T = \mathbf{u}\mathbf{v}^t/e \mod \Delta}$

- Compute $z \in \mathrm{K}[x]_{<\delta}$ s.t. $\gcd(Bz, \Delta) = 1$ in $\widetilde{O}(n^2\delta)$ arith. ops. by adapting [Mulders Storjohann 1998]

Computation of a rank one decomposition

Assumption: T admits a rank one decomposition $T = uv^t/\Delta + W$

How to compute u, v in $\widetilde{O}(n^2\delta)$?

- Smith form of $B = \Delta T \implies \exists w, z$ s.t. $\gcd(w^t Bz, \Delta) = 1$

$$\mathbf{u} = Bz, \quad \mathbf{v}^t = w^t B, \quad e = w^t Bz \mod \Delta$$

  Then, $\boxed{\Delta T = \mathbf{uv}^t/e \mod \Delta}$

- Compute $z \in \mathrm{K}[x]_{<\delta}$ s.t. $\gcd(Bz, \Delta) = 1$ in $\widetilde{O}(n^2\delta)$ arith. ops.
  by adapting [Mulders Storjohann 1998]

- Compute $w \in \mathrm{K}[x]_{<\delta}$ s.t. $\gcd(w^t Bz, \Delta) = 1$ using $\leq n$ extended gcd in degree $\leq \delta$