

Quasi-optimal computation of the p -curvature

Alin Bostan



joint work with

Xavier Caruso (Univ. Rennes 1) and Éric Schost (Univ. Waterloo)



Toronto, Canada

FIELDS

15th September 2015

Workshop on Symbolic Combinatorics and Computational Differential Algebra

Main objects and goal

- $k =$ a field of prime characteristic p , typically \mathbb{F}_p
- $k(x)\langle\partial\rangle =$ the non-commutative (right/left) Euclidean algebra of linear differential operators $L = a_0 + a_1\partial + \cdots + a_r\partial^r$, for $a_i \in k(x)$

Def: p -curvature $\mathbf{A}_p(L)$ of $L =$ the matrix in $\mathcal{M}_r(k(x))$ whose (i, j) entry is the coefficient of ∂^i in $\partial^{p+j} R \bmod L$ for $0 \leq i, j < r$

Main objects and goal

- $k =$ a field of prime characteristic p , typically \mathbb{F}_p
- $k(x)\langle\partial\rangle =$ the non-commutative (right/left) Euclidean algebra of linear differential operators $L = a_0 + a_1\partial + \cdots + a_r\partial^r$, for $a_i \in k(x)$

Def: p -curvature $\mathbf{A}_p(L)$ of $L =$ the matrix of ∂^p modulo L
= $\textit{“differential Frobenius map”}$

Main objects and goal

- $k =$ a field of prime characteristic p , typically \mathbb{F}_p
- $k(x)\langle\partial\rangle =$ the non-commutative (right/left) Euclidean algebra of linear differential operators $L = a_0 + a_1\partial + \cdots + a_r\partial^r$, for $a_i \in k(x)$

Def: p -curvature $\mathbf{A}_p(L)$ of $L =$ the matrix of ∂^p modulo L
 $=$ “*differential Frobenius map*”

Goal: design a quasi-optimal algorithm for $\mathbf{A}_p(L)$

Main objects and goal

- $k =$ a field of prime characteristic p , typically \mathbb{F}_p
- $k(x)\langle\partial\rangle =$ the non-commutative (right/left) Euclidean algebra of linear differential operators $L = a_0 + a_1\partial + \cdots + a_r\partial^r$, for $a_i \in k(x)$

Def: p -curvature $\mathbf{A}_p(L)$ of $L =$ the matrix of ∂^p modulo L
= “differential Frobenius map”

Goal: design a quasi-optimal algorithm for $\mathbf{A}_p(L)$

- ▷ Quasi-optimality = complexity almost linear in the size of $\mathbf{A}_p(L)$
- ▷ Complexity is measured in terms of arithmetic operations in k
- ▷ Caveat: to simplify matters, assume input L has size $\mathcal{O}(1)$

Example

$$L = (5x^2 + 4)\partial^2 + (4x^2 + 6x + 5)\partial + 2x + 2 \in \mathbb{F}_7[x]\langle\partial\rangle$$

Euclidean right division in $\mathbb{F}_7(x)\langle\partial\rangle$:

$$\partial^7 = (\dots)L + \frac{(x+1)(x^2+x-1)}{(x+3)(x-3)^2}\partial + \frac{4x(x-1)}{(x+3)(x-3)^2}$$

$$\partial^8 = (\dots)L + \frac{2(x+1)(x^2+x-1)}{(x+3)(x-3)^2}\partial + \frac{x(x-1)}{(x+3)(x-3)^2}$$

$$\implies \mathbf{A}_7(L) = \begin{bmatrix} \frac{4x(x-1)}{(x+3)(x-3)^2} & \frac{x(x-1)}{(x+3)(x-3)^2} \\ \frac{(x+1)(x^2+x-1)}{(x+3)(x-3)^2} & \frac{2(x+1)(x^2+x-1)}{(x+3)(x-3)^2} \end{bmatrix}$$

Basics on differential equations in characteristic p

Main differences with the characteristic zero case

- *more constants*: the derivative of x^p is zero
- *simpler solutions* [Honda 1981]:

$$\dim_{k(x^p)} \text{Sol}(L, k(x)) = \dim_{k((x^p))} \text{Sol}(L, k((x)))$$

- *lack of Cauchy's theorem*: the common dimension $\dim\text{sol}(L)$ of these solution spaces is generally $< r = \text{ord}(L)$

E.g. $y' = y$ has no solution in $\mathbb{F}_p((x)) \setminus \{0\}$

Connection between solutions and p -curvature

- [Katz & Cartier 1970] $\text{rank}(\mathbf{A}_p(L)) = r - \dim\text{sol}(L)$

→ p -curvature measures to what extent $\dim\text{sol}(L)$ is close to r

Example (continued)

$$L = (5x^2 + 4)\partial^2 + (4x^2 + 6x + 5)\partial + 2x + 2 \in \mathbb{F}_7[x]\langle\partial\rangle$$

- 7-curvature of L

$$\mathbf{A}_7(L) = \begin{bmatrix} \frac{4x(x-1)}{(x+3)(x-3)^2} & \frac{x(x-1)}{(x+3)(x-3)^2} \\ \frac{(x+1)(x^2+x-1)}{(x+3)(x-3)^2} & \frac{2(x+1)(x^2+x-1)}{(x+3)(x-3)^2} \end{bmatrix}$$

- Katz-Cartier:

$$1 = \text{rank}(\mathbf{A}_7(L)) = 2 - \text{dimsol}(L) \implies \text{dimsol}(L) = 1$$

- In fact

$$\text{Basis}_{\mathbb{F}_7(x^7)}\text{Sol}(L, \mathbb{F}_7(x)) = \{1 - 2x^2 - x^3\}$$

A useful tool for theory

Grothendieck's conjecture ('70s) An operator $\Gamma \in \mathbb{Q}[x]\langle\partial\rangle$ has a basis of algebraic solutions over $\mathbb{Q}(x)$ if and only if $\mathbf{A}_p(\Gamma \bmod p)$ is zero for almost all primes p .

- ▷ Proved by [Katz 1982] for *Picard-Fuchs operators*, but still widely open in the general case (even for second-order operators)

The Chudnovsky² theorem (1985) The minimal-order operator $\Gamma \in \mathbb{Q}[x]\langle\partial\rangle$ annihilating a D-finite power series in $\mathbb{Z}[[x]]$ with positive convergence radius is globally nilpotent, i.e., the p -curvatures $\mathbf{A}_p(\Gamma \bmod p)$ are nilpotent for almost all primes p .

- ▷ Valid more generally for *G-functions*

A useful tool for algorithms

p -curvature used in computer algebra:

- [van der Put 1995] for **factoring operators** in $\mathbb{F}_p(x)\langle\partial\rangle$
- [Cluzeau 2003] for **decomposing differential systems** over $\mathbb{F}_p(x)$
- [Cluzeau & van Hoeij 2004] as **filter in modular algorithms** for operators in $\mathbb{Q}(x)\langle\partial\rangle$



Improving the complexity of the p -curvature computation is an interesting problem in its own right

A useful tool for applications

- in enumerative combinatorics (classification of lattice walks)
- in statistical physics (square lattice Ising model)

Typical task: given a power series $S \in \mathbb{Z}[[x]]$, decide if S is **D-finite**

Differential guessing: from the first $N \gg 0$ terms of S , compute a differential operator $\Gamma \in \mathbb{Q}[x]\langle \partial \rangle$ that annihilates $S \bmod x^N$

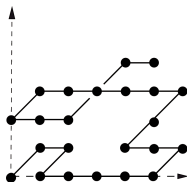
- ▷ One way to **empirically certify** the correctness of Γ is to *look at the p -curvature* $\mathbf{A}_p(\Gamma \bmod p)$ for a random (large) prime p
- if $\mathbf{A}_p(\Gamma \bmod p)$ is **nilpotent**, then S is **very probably D-finite**
 - if $\mathbf{A}_p(\Gamma \bmod p)$ is **zero**, then S is **very probably algebraic**

A combinatorial application: Gessel's conjecture

- **Gessel walks**: walks in \mathbb{N}^2 using only steps in $\mathcal{S} = \{\nearrow, \swarrow, \leftarrow, \rightarrow\}$
- $g(i, j, n)$ = number of **walks** from $(0, 0)$ to (i, j) with n steps in \mathcal{S}

Question: Nature of the generating function

$$G(u, v, x) = \sum_{i, j, n=0}^{\infty} g(i, j, n) u^i v^j x^n \in \mathbb{Q}[[u, v, x]]$$



Theorem [B. & Kauers 2010] $G(u, v, x)$ is an algebraic function.¹

→ Effective, computer-driven discovery and proof

→ Key step in discovery: **p -curvature computation** of two 11th order (guessed) differential operators for $G(u, 0, x)$, and $G(0, v, x)$

¹Minimal polynomial $P(u, v, x, G(u, v, x)) = 0$ has $> 10^{11}$ terms; $\approx 30\text{Gb}$ (!)

Previous work: first order

- [van der Put 1995] The p -curvature of $L = \partial - u(x)$ is

$$\mathbf{A}_p(L) = u^{(p-1)} + u^p$$

- ▷ No such closed form expression for order > 1 operators!

- [B. & Schost 2009] $\mathbf{A}_p(L)$ can be computed in time $\mathcal{O}(\log(p))$.

Proof: If $u = \sum_{i \geq 0} a_i x^i$, then $u^{(p-1)} = - \sum_{j \geq 1} a_{jp-1} x^{p(j-1)}$ (Wilson)

→ it is sufficient to compute by **binary powering** the $\mathcal{O}(1)$ terms $a_{p-1}, a_{2p-1}, \dots, a_{\deg(u)p-1}$ of the recurrent sequence $(a_n)_{n \geq 0}$.

Previous work: arbitrary order

① p -curvature $\mathbf{A}_p(L)$

size $\mathcal{O}(p)$



Binary powering can not be used to compute ∂^p in $\frac{k(x)\langle\partial\rangle}{k(x)\langle\partial\rangle L}$

- [Katz 1982]: algorithm of cost $\mathcal{O}(p^2)$, based on recurrence

$$\mathbf{A}_1 = \text{CompanionMatrix}(L), \quad \mathbf{A}_{k+1} = \mathbf{A}'_k + \mathbf{A}_1 \cdot \mathbf{A}_k$$

- [B. & Schost 2009]: first subquadratic algorithm $\mathcal{O}(p^{1.79})$
- [B. & Schost 2009]: for certain second-order operators $\tilde{\mathcal{O}}(p)$

Previous work: arbitrary order

① p -curvature $\mathbf{A}_p(L)$

size $\mathcal{O}(p)$



Binary powering can not be used to compute ∂^p in $\frac{k(x)\langle\partial\rangle}{k(x)\langle\partial\rangle L}$

- [Katz 1982]: algorithm of cost $\mathcal{O}(p^2)$, based on recurrence

$$\mathbf{A}_1 = \text{CompanionMatrix}(L), \quad \mathbf{A}_{k+1} = \mathbf{A}'_k + \mathbf{A}_1 \cdot \mathbf{A}_k$$

- [B. & Schost 2009]: first subquadratic algorithm $\mathcal{O}(p^{1.79})$
- [B. & Schost 2009]: for certain second-order operators $\tilde{\mathcal{O}}(p)$

② characteristic polynomial of $\mathbf{A}_p(L)$

size $\mathcal{O}(1)$

- [B., Caruso & Schost 2014]: $\tilde{\mathcal{O}}(\sqrt{p})$

③ polynomial solutions of L

size $\mathcal{O}(p)$

- [B. & Schost 2009]: quasi-optimal algorithm $\tilde{\mathcal{O}}(p)$

New result [B.-Caruso-Schost 2015]

Computation of $\mathbf{A}_p(L)$

for an arbitrary operator L

in quasi-linear time $\tilde{O}(p)$.



▷ Precise complexity result for L of bidegree (d, r) in (x, ∂) :

$$\tilde{O}(p d r^\omega)$$

where ω is the exponent of matrix multiplication.

▷ Quasi-optimality: for $r > 1$, generic size of $\mathbf{A}_p(L)$ is $\Theta(p d r^2)$

▷ Extension to systems: same results for p -curvature of $Y' = AY$

The starting idea

- **Question:** Given L in $k(x)\langle\partial\rangle$, compute R in $k(x)\langle\partial\rangle$ such that

$$\partial^p = QL + R, \quad \text{ord}(R) < \text{ord}(L) = r$$

- **Idea:** *evaluation-interpolation*; on “points” = solutions of L
- ▷ **Fruitful strategy in related contexts:** product, lcm, gcd (**char. 0**)
[van der Hoeven '02, '12; B. '03; Benoit, B. & van der Hoeven '12]

The starting idea

- **Question:** Given L in $k(x)\langle\partial\rangle$, compute R in $k(x)\langle\partial\rangle$ such that

$$\partial^p = QL + R, \quad \text{ord}(R) < \text{ord}(L) = r$$

- **Idea:** *evaluation-interpolation*; on “points” = solutions of L
- ▷ **Fruitful strategy in related contexts:** product, lclm, gcd (char. 0)
[van der Hoeven '02, '12; B. '03; Benoit, B. & van der Hoeven '12]



If L had a full basis of power series solutions $\{y_1, \dots, y_r\}$, then $R = \sum_{j=0}^{r-1} a_j(x)\partial^j$ could be determined by solving a linear system

$$(\partial^p(y_1), \dots, \partial^p(y_r)) = (a_0, \dots, a_{r-1}) \cdot \text{Wronskian}(y_1, \dots, y_r)$$

with coefficients in $k[[x]]$ truncated modulo $x^{O(p)}$ $\rightsquigarrow \tilde{O}(p)$

The starting idea

- **Question:** Given L in $k(x)\langle\partial\rangle$, compute R in $k(x)\langle\partial\rangle$ such that

$$\partial^p = QL + R, \quad \text{ord}(R) < \text{ord}(L) = r$$

- **Idea:** *evaluation-interpolation*; on “points” = solutions of L
▷ **Fruitful strategy in related contexts:** product, lclm, gcd (char. 0)
[van der Hoeven '02, '12; B. '03; Benoit, B. & van der Hoeven '12]



If L had a full basis of power series solutions $\{y_1, \dots, y_r\}$, then $R = \sum_{j=0}^{r-1} a_j(x)\partial^j$ could be determined by solving a linear system

$$(\partial^p(y_1), \dots, \partial^p(y_r)) = (a_0, \dots, a_{r-1}) \cdot \text{Wronskian}(y_1, \dots, y_r)$$

with coefficients in $k[[x]]$ truncated modulo $x^{\mathcal{O}(p)}$ $\rightsquigarrow \tilde{\mathcal{O}}(p)$



Obstruction: Cauchy's theorem *does not hold in char. $p > 0$*

The starting idea

- **Question:** Given L in $k(x)\langle\partial\rangle$, compute R in $k(x)\langle\partial\rangle$ such that

$$\partial^p = QL + R, \quad \text{ord}(R) < \text{ord}(L) = r$$

- **Idea:** *evaluation-interpolation*; on “points” = solutions of L
▷ **Fruitful strategy in related contexts:** product, lclm, gcd (**char. 0**)
[van der Hoeven '02, '12; B. '03; Benoit, B. & van der Hoeven '12]



If L had a full basis of power series solutions $\{y_1, \dots, y_r\}$, then $R = \sum_{j=0}^{r-1} a_j(x)\partial^j$ could be determined by solving a linear system

$$(\partial^p(y_1), \dots, \partial^p(y_r)) = (a_0, \dots, a_{r-1}) \cdot \text{Wronskian}(y_1, \dots, y_r)$$

with coefficients in $k[[x]]$ truncated modulo $x^{\mathcal{O}(p)}$ $\rightsquigarrow \tilde{\mathcal{O}}(p)$



Idea: use “points” from a *different solution space*

The key : Hurwitz series

- ℓ = a ring in which p vanishes
- $\ell[[t]]^H$ = differential ring of *Hurwitz series* $\gamma_i(t) \approx \frac{t^i}{i!}$

$$f = a_0\gamma_0(t) + a_1\gamma_1(t) + a_2\gamma_2(t) + \cdots + a_i\gamma_i(t) + \cdots$$

where $a_i \in \ell$ and $\gamma_i(t) \cdot \gamma_j(t) = \binom{i+j}{i} \gamma_{i+j}(t)$, and $\gamma_{i+1}(t)' = \gamma_i(t)$.

Theorem [Cauchy's theorem for Hurwitz series]

For any $r \times r$ matrix A with coefficients in $\ell[[t]]^H$, and for any initial data $v \in \ell^r$, the Cauchy problem

$$\begin{cases} Y' = A \cdot Y \\ Y(0) = v \end{cases}$$

admits a (unique) solution Y with entries in $\ell[[t]]^H$.

Example (continued)

$$L = (5x^2 + 4)\partial^2 + (4x^2 + 6x + 5)\partial + 2x + 2 \in \mathbb{F}_7[x]\langle\partial\rangle$$

- Evaluation step*: basis of Hurwitz series solutions of L in $\mathbb{F}_7[[t]]^H$:

$$y_1 = \gamma_0 + 3\gamma_2 + \gamma_3$$

$$y_2 = \gamma_0 + 4\gamma_2 + \gamma_4 + 2\gamma_5 + 4\gamma_6 + \gamma_7 + 2\gamma_8 + 4\gamma_9 + \gamma_{10} + 2\gamma_{11} + 4\gamma_{12} + \gamma_{13} + \dots$$

- Interpolation step*: $\partial^7 = QL + R$, with $R = a_0 + a_1\partial$, implies

$$\begin{bmatrix} a_0 & a_1 \end{bmatrix} \cdot \begin{bmatrix} y_1 & y_2 \\ y_1' & y_2' \end{bmatrix} = \begin{bmatrix} y_1^{(7)} & y_2^{(7)} \end{bmatrix}$$

with solution

$$a_0 = 4x + 2x^2 + 5x^3 + 2x^4 + 4x^5 + 5x^6 + \dots = \frac{4x(x-1)}{(x+3)(x-3)^2}$$

$$a_1 = 1 + 5x + 6x^2 + x^3 + 6x^4 + 5x^5 + x^6 + \dots = \frac{(x+1)(x^2+x-1)}{(x+3)(x-3)^2}$$

The (long and tortuous) road from the starting idea to a quasi-optimal algorithm

Input: L in $k(x)\langle\partial\rangle$ of bidegree (d, r) in (x, ∂)

Output: its p -curvature $\mathbf{A}_p(L)$



Evaluation-interpolation on Hurwitz series at precision $2p$
gives access to the p -curvature of L modulo x^p



This strategy *is not sufficient* to compute the full p -curvature,
since $\mathbf{A}_p(L)$ has entries of degree $\approx pd$.



Turning it into a *complete* and *efficient* algorithm requires

- ① efficient computation with Hurwitz series
- ② fast evaluation/interpolation
- ③ gluing local information into global information

The (long and tortuous) road from the starting idea to a quasi-optimal algorithm

Input: L in $k(x)\langle\partial\rangle$ of bidegree (d, r) in (x, ∂)

Output: its p -curvature $\mathbf{A}_p(L)$



Evaluation-interpolation on Hurwitz series at precision $2p$
gives access to the p -curvature of L modulo x^p



This strategy *is not sufficient* to compute the full p -curvature,
since $\mathbf{A}_p(L)$ has entries of degree $\approx pd$.



Turning it into a *complete* and *efficient* algorithm requires

- ① efficient computation with Hurwitz series FFT-like
- ② fast evaluation/interpolation Newton iteration
- ③ gluing local information into global information CRT

Efficient computation with Hurwitz series

Task: fast arithmetic in $\ell[[t]]^H$ modulo $\ell[[t]]_{\geq N}^H = \{ \sum_{i \geq N} a_i \gamma_i(t) \}$

Theorem For $N = mp^s$, with $s \geq 0$ and $m \in \{1, \dots, p\}$, there is an isomorphism of ℓ -algebras:

$$\ell[[t]]_{<N}^H := \ell[[t]]^H / \ell[[t]]_{\geq N}^H \simeq \ell[t_0, \dots, t_s] / (t_0^p, \dots, t_{s-1}^p, t_s^m).$$

Proof: Send $\gamma_{p^i}(t)$ to t_i and use Lucas' theorem. If $n = \sum_{i=0}^s n_i p^i$,

$$\gamma_n(t) = \gamma_{n_0}(t) \cdot \gamma_{n_1 p}(t) \cdots \gamma_{n_s p^s}(t) \quad \mapsto \quad \frac{t_0^{n_0}}{n_0!} \cdot \frac{t_1^{n_1}}{n_1!} \cdots \frac{t_s^{n_s}}{n_s!}$$

Theorem The product in $\ell[[t]]^H$ at precision $N = p^{\mathcal{O}(1)}$ can be performed with $\tilde{\mathcal{O}}(N)$ operations in k .

Proof: Use Kronecker's substitution + univariate FFT.

Fast differential system solving in Hurwitz series

- Newton iteration [B.-Chyzak-Ollivier-Salvy-Schost-Sedoglavic'07]

Input: an $r \times r$ system $Y' = AY$ with $A \in \mathcal{M}_r(\ell[[t]]^H)$, and $N \in \mathbb{N}$

Output: the fundamental matrix of solutions in $\ell[[t]]_{<N}^H$

- ① $Y = I_r + t A(0); Z = I_r; m = 2$
- ② **while** $m \leq N/2$:
- ③ $Z = Z + [Z(I_r - YZ)]^m$
- ④ $Y = Y - [Y(\int Z \cdot (Y' - [A]^{2m-1}Y))]^{2m}$
- ⑤ $m = 2m$
- ⑥ **return** Y

Theorem [Effective Cauchy's theorem] One can solve $Y' = AY$ in $\ell[[t]]^H$ at precision $N = p^{\mathcal{O}(1)}$ using $\tilde{\mathcal{O}}(Nr^\omega)$ operations in ℓ .

The algorithm in a nutshell

Input: a differential system $Y' = AY$, with $A \in \mathcal{M}_r(k[x][\frac{1}{f}])$

Output: its p -curvature \mathbf{A}_p [def: $\mathbf{A}_{k+1} = \mathbf{A}'_k + \mathbf{A}_1 \cdot \mathbf{A}_k$, $\mathbf{A}_1 = -A$]

- ① Choose $S \in k[x]$ separable and coprime with f
Let $\ell = k[x]/S$ and $\varphi_S : k[x][\frac{1}{f}]/S^p \xrightarrow{\sim} \ell[[t]]_{<p}^H$, $x \mapsto t + x$
- ② *Evaluation step (Newton iteration)* $\tilde{O}(pr^\omega)$ ops. in ℓ
Compute a fund. matrix Y_S in $\ell[[t]]_{<2p}^H$ of $Y' = \varphi_S(A)Y$
- ③ *Interpolation step* $\tilde{O}(pr^\omega)$ ops. in ℓ
Deduce $\mathbf{A}_p \bmod S^p$ as $\varphi_S^{-1} \left(-Y_S^{(p)} \cdot Y_S^{-1} \right)$
- ④ Use several S 's and CRT to get \mathbf{A}_p from the various $\mathbf{A}_p \bmod S^p$

COMPLEXITY: $\tilde{O}(pr^\omega d)$ operations in k , where $d = \deg(A)$

Experimental results

- For **random** operators in $\mathbb{F}_p[x]\langle\partial\rangle$ of bidegree (d, r) in (x, ∂)

(d, r)	p							
	157	281	521	983	1 811	3 433	6 421	12 007
(5, 5)	0.39 s	0.71 s	1.22 s	2.34 s	4.41 s	8.93 s	18.0 s	36.1 s
	0.26 s	0.76 s	2.69 s	9.05 s	32.6 s	145 s	593 s	2 132 s
(5, 11)	1.09 s	2.05 s	3.65 s	7.05 s	12.6 s	26.7 s	53.3 s	109 s
	1.25 s	3.70 s	12.8 s	45.5 s	163 s	725 s	2 942 s	—
(5, 20)	2.93 s	5.25 s	9.52 s	17.7 s	32.5 s	68.1 s	139 s	288 s
	4.29 s	12.4 s	42.5 s	153 s	548 s	2 460 s	—	—
(11, 20)	6.89 s	13.3 s	22.6 s	45.0 s	80.4 s	167 s	342 s	711 s
	11.6 s	34.7 s	121 s	486 s	1 943 s	—	—	—
(20, 20)	14.0 s	25.1 s	49.9 s	94.0 s	176 s	357 s	733 s	1 472 s
	27.0 s	84.5 s	314 s	1 283 s	—	—	—	—

Running times of the **new** algorithm, vs. **Katz's** algorithm

- For operators with **physical relevance**: e.g., $\phi_H^{(5)}$ in $(\mathbb{Z}/27449\mathbb{Z})[x]\langle\partial\rangle$, with $(d, r) = (108, 28)$ [Maillard et al. 2007]

→ (first column of) $\mathbf{A}_p(\phi_H^{(5)})$ computed in ≈ 19 hours, size ≈ 1 GB

Conclusion

This work:

- Computation of p -curvature $\mathbf{A}_p(L)$ in quasi-optimal time $\tilde{O}(p)$
- Basic tool: *fast evaluation/interpolation* on *Hurwitz series*

Next challenges:

- Compute invariant factors of $\mathbf{A}_p(L)$ in time $\tilde{O}(\sqrt{p})$
- Factor L in time $\tilde{O}(p)$

Thanks for your attention!

The algorithm in a nutshell – optimized version

Input: a differential system $Y' = AY$, with $A \in \mathcal{M}_r(k[x][\frac{1}{f}])$

Output: its p -curvature \mathbf{A}_p [def: $\mathbf{A}_{k+1} = \mathbf{A}'_k + \mathbf{A}_1 \cdot \mathbf{A}_k$, $\mathbf{A}_1 = -A$]

① Choose $S \in k[x]$ separable and coprime with f

Let $\ell = k[x]/S$ and $\varphi_S : k[x][\frac{1}{f}]/S^p \xrightarrow{\sim} \ell[[t]]_{<p}^H$, $x \mapsto t + x$

② *Evaluation step (Newton iteration)* $\tilde{O}(pr^\omega)$ ops. in ℓ

Compute a fund. matrix Y_S in $\ell[[t]]_{<p}^H$ of $Y' = \varphi_S(A)Y$

③ *Interpolation step* $\tilde{O}(pr^\omega)$ ops. in ℓ

Deduce $\mathbf{A}_p \bmod S^p$ as $\varphi_S^{-1}(Y_S \cdot \text{coeff}(AY_S, p-1) \cdot Y_S^{-1})$

④ Use several S 's and CRT to get \mathbf{A}_p from the various $\mathbf{A}_p \bmod S^p$

COMPLEXITY: $\tilde{O}(pr^\omega d)$ operations in k , where $d = \deg(A)$



uses only usual power series arithmetic, since $\ell[[t]]_{<p}^H \simeq \ell[t]/t^p$