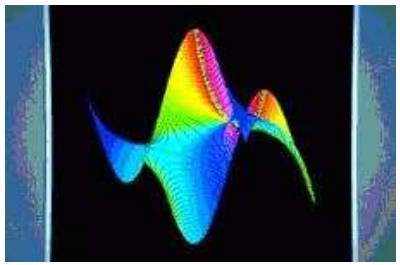
# Common Multiples of Linear Differential and Difference Operators

Frédéric Chyzak
(joint work in progress with A. Bostan, Z. Li & B. Salvy)

Project-Team Algorithms, INRIA




March 17, 2008

# Arithmetic Complexity of Polynomial Multiplication

For polynomial of degrees $\leq n$:

- naive algorithm in $O(n^2)$;
- Karatsuba in $O\left(n^{\log_2 3}\right)$ by evaluation-interpolation at 0, 1, $\infty$;
- Toom–Cook (lower exponents by more evaluation points);
- FFT in $O(n \log n \log\log n)$ by eval.-interp. at $n$th roots of 1;
- Fürer's 2007 algorithm in $O\left(n \log n\, 2^{O(\log^* n)}\right)$.

Using FFT is quasi-optimal in the output size: $\tilde{O}(n)$.

Throughout:

- only arithmetic complexity is considered;
- $\mathrm{M}(n)$ denotes complexity of polynomial multiplication.

# Polynomial Multiplication is the Commutative Yardstick

Polynomial multiplication permits $O(\mathrm{M}(n))$ for:

- Euclidean division, divisions of series, exp. and log. of series, series solutions of linear ODEs via Newton iteration;
- evaluation-interpolation at points in geometric progression.

Polynomial multiplication permits $O(\mathrm{M}(n)\log n)$ for:

- GCD, LCM, Bézout relations, Padé approx., resultants via "half GCD";
- evaluation-interpolation at general points, shifts of the indeterminate by divide-and-conquer approach;
- special changes of polynomial bases.

However, not: series composition, general changes of polynomial bases, factorisation.

# Bivariate Polynomials from $K[x, y]$

For polynomials of degree $\leq n$ in $x$ and $\leq r$ in $y$.

Multiplication in $O\big(\mathrm{M}(nr)\big)$ by $f(x, y) \leftrightarrow f(x, x^{2n+1})$.

LCM of elements viewed in $K(x)[y]$ in $O\big(\mathrm{M}(nr)\log r\big)$.

Using FFT is quasi-optimal in the output size: $\tilde{O}(nr)$.

# Matrix Multiplication as a Non-Commutative Yardstick?

Two skew-polynomial settings for differential operators:

- $\partial = d/dx$: $W = K\langle x, \partial\rangle$, where $\partial x = x\partial + 1$.
- $\theta = x\,d/dx$: $E = K\langle x, \theta\rangle$, where $\theta x = x(\theta + 1)$.

- van der Hoeven (2002): skew-polynomial multiplication in bidegree $(n, n)$ in either algebra reduces to matrix multiplication.
- Bostan, C., Le Roux (submitted):
  - $\mathrm{SkewM}(n, n) \leq 8\,\mathrm{MM}(n)$ in both algebras.
  - Computational equivalence: $O\big(\mathrm{MM}(n)\big) = O\big(\mathrm{SkewM}(n, n)\big)$.

Natural question: What about other operations in $W$ and $E$?

# Sketch and Improvement of van der Hoeven's Algorithm

$$\begin{array}{c} A \text{ and } B \\ \text{of bidegree } (n,n) \end{array} \;\to\; C = BA = \sum_{i=0}^{2n} x^i\, C_i(\theta),\ \deg C_i \le 2n.$$

$$\theta^j(x^k) = k^j x^k \;\to\; C(x^k) = \sum_{i=0}^{2n} C_i(k)\, x^{i+k}.$$

By Lagrange interpolation: $\big(C_i(k)\big)_{0\le i,k\le 2n} \to \big(C_i(\theta)\big)_{0\le i\le 2n}$.

$$K[x]_{\le 2n} \xrightarrow{A\cdot} K[x]_{\le 3n} \xrightarrow{B\cdot} K[x]_{\le 4n}.$$

$$\text{Complexity } \in \underbrace{O\big(\mathrm{MM}(n)\big)}_{\text{composition}} + \underbrace{O\big(n\,\mathrm{M}(n)\log n\big)}_{\text{eval.-interp.}} \subset O\big(\mathrm{MM}(n)\big).$$

Fast evaluation-interpolation was not available in van der Hoeven's youth: $48\,\mathrm{MM}(n) > 24\,\mathrm{MM}(n) > 8\,\mathrm{MM}(n)$.

# Least Common Left Multiples

$A =$ algebra of skew polys. with polynomial coefficients ($W$, $E$, $R$).
$A(x) =$ algebra of skew polys. with rational coefficients:

- $W(x) = K(x)\langle\partial\rangle$, $\partial f = f\partial + f'$ (differential operators),
- $E(x) = K(x)\langle\theta\rangle$, $\theta f = f\theta + xf'$ (Eulerian operators).
- $R(x) = K(x)\langle\sigma\rangle$, $\sigma f = f(x+1)\sigma$ (shift operators).

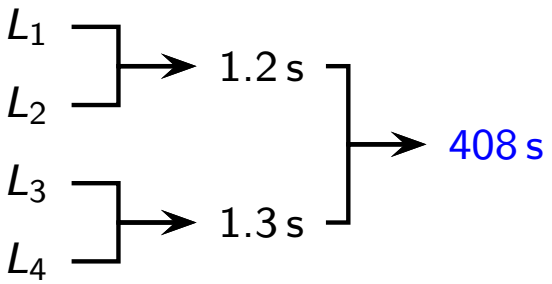
LCLM:

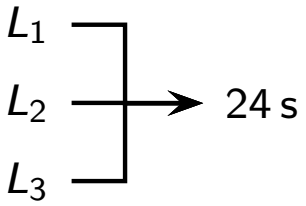
- common left multiple with least order in $A(x)$,
- normalized in $A$ with no content,
- for given $L_i \in A$, $L = \mathrm{LCLM}(L_1, \ldots, L_k) = C_i L_i$ with $C_i \in A(x)$.

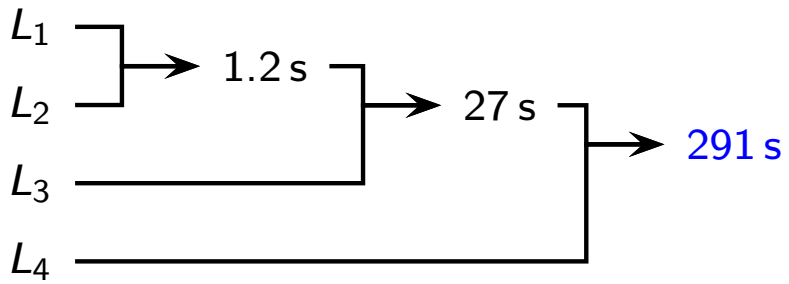
$$L\,(\mathrm{Sol}\,L_1 + \cdots + \mathrm{Sol}\,L_k) = 0.$$

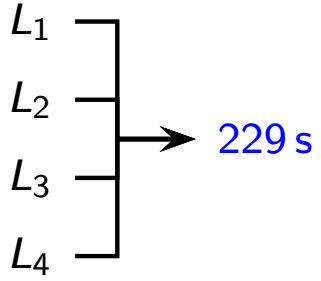
# Iterated LCLMs are bad!

Maple 11's `DEtools[LCLM]` by van Hoeij ($-100 < \text{coeffs.} < 100$):



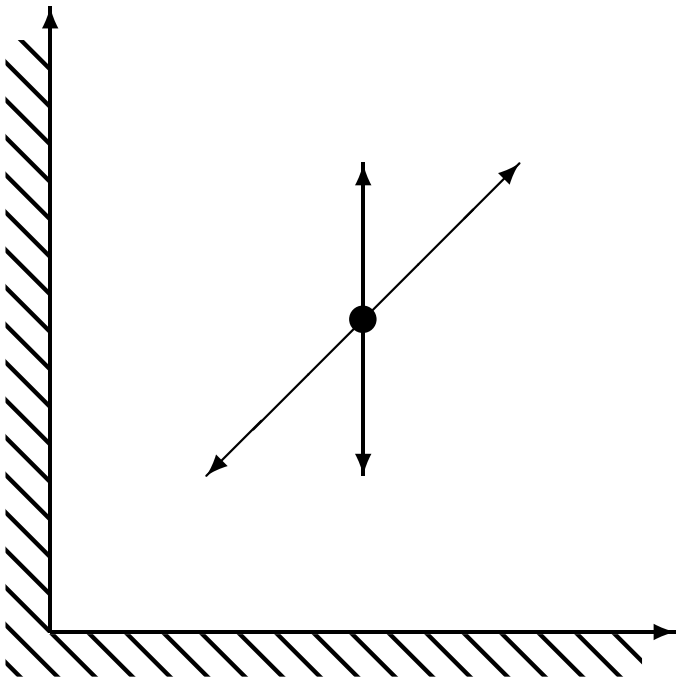




$$\deg_\partial L_i = r_i, \quad \deg_x L_i = d_i$$
$$\downarrow$$
$$\deg_\partial L \le R, \quad \deg_x L \le B$$

| $k$ | $R$ | $B$ |
|---|---|---|
| 1 | 3 | 9 |
| 2 | 6 | 72 |
| 3 | 9 | 189 |
| 4 | 12 | 360 |

# Proving Gessel's Conjecture...



Generating series of $n$-step walks from $(0,0)$ to $(i,j)$:

$$Q(t;x,y) := \sum_{n,i,j\leq 0} f(n,i,j)\, x^i y^j t^n.$$

Conjecture (Gessel, about excursions): $Q(t;0,0)$ is D-finite.

Conjecture (Gessel?, Bostan–Kauers?): $Q(t;x,y)$ is D-finite in $t$.

$$Q(t;x,y) = \frac{1 - \frac{t}{xy}\big(Q(t;x,0) - Q(t;0,0)\big) - \frac{t}{x}\left(1+\frac{1}{y}\right) Q(t;0,y)}{1 - \left(x + \frac{1}{x} + xy + \frac{1}{xy}\right) t}.$$

B & K's Idea: an annihilator of $Q(t;x,y)$ is an LCLM related to annihilators of $Q(t;x,0)$ and $Q(t;0,y)$.

# . . . A Matter of Time?

$$L_1(x,t,\partial_t)\big(Q(t;x,0)\big) = O\left(t^{997}\right)$$
$$L_2(y,t,\partial_t)\big(Q(t;0,y)\big) = O\left(t^{1001}\right)$$

with $\deg_{\partial_t} L_1 = 11$ $\deg_{\partial_t} L_2 = 11$, $\deg_t L_1 \leq 96$ $\deg_t L_2 \leq 68$, $\deg_x L_1 \leq 78$ $\deg_x L_1 \leq 28$

LCLM $L$ has $\deg_{\partial_t} L \leq 22$ $\deg_t L \leq 1968$, $\deg_x L \leq 936$ $\deg_y L \leq 336$, thus $14 \cdot 10^9$ coeffs.!

Direct calculation too large $\longrightarrow \approx 3 \cdot 10^5$ evaluations of $(x, y)$

Each evaluated LCLM calculation takes 3 min over $\mathrm{GF}_{9001}$ in usual Magma, 1 min with new algorithm, should not take more than 2 s over $\mathbb{Q}$ to hope for $L$ in less than a week!

Interpolation would then prove $L(x,y,t,\partial_t)\big(Q(t;x,y)\big) = O(t^{\approx 1000})$.

Another of Zeilberger's techniques should conclude $\cdots = O(t^{\infty})$.

# Related Works for LCLMs in Specific Classes

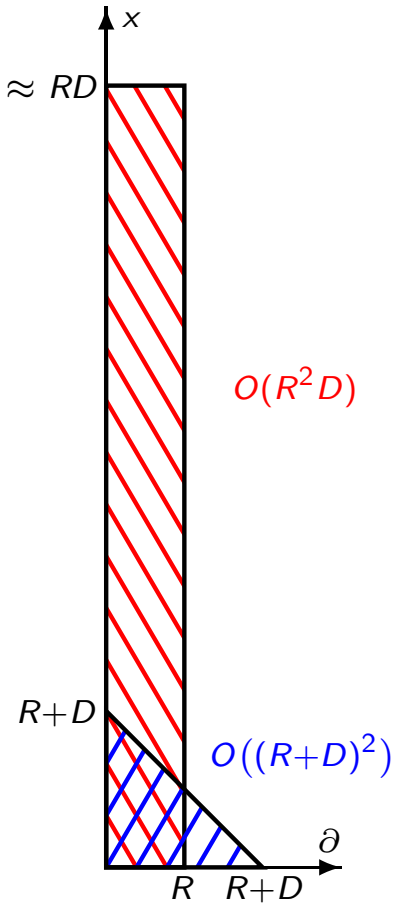
LCLMs of first-order operators in van Hoeij's `DEtools[DFactor]` (1997).

LCLMs of two operators by *A subresultant theory for linear ordinary differential polynomials*, Z. Li (1998).

LCLMs of the form $\mathrm{LCLM}\big(\{L(\beta x, \beta^{-1}\partial); \beta^k = 1\}\big)$ appear for the extraction of $k$-sections in multisummable D-finite series in *Remarques algorithmiques liées au rang d'un opérateur différentiel linéaire*, Barkatou, C., Loday-Richaud (2003).

LCLMs of telescopers are used for the summation of rational functions, after a partial fraction decomposition, in *A direct algorithm to construct the minimal Z-pairs for rational functions*, H. Q. Le (2003).

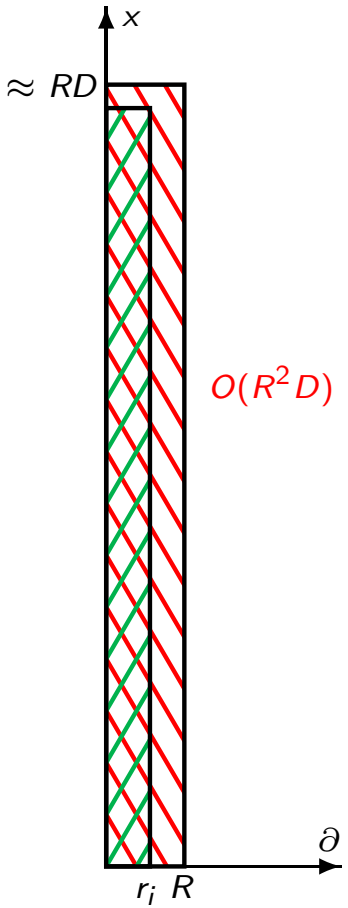
# Contributions




- Analysis of worst-case arithmetic complexity of existing algorithms for computing LCLMs.
- A tight bound $B \approx k^2 rd$ on the degree of $L$ in $x$.
- A first algorithm, better for large $r$.
- A bound $B' \approx 2kr$ on the total degree in $(x, \partial)$ in which common left multiples exist.
- A faster algorithm which computes the LCLM as the GCRD of two common multiples of small size. (Heuristic.)

Inspired by former work on algebraic series
(ISSAC'07, B., C., Lecerf, S., Schost).

# Poole's Method ($k = 2$)

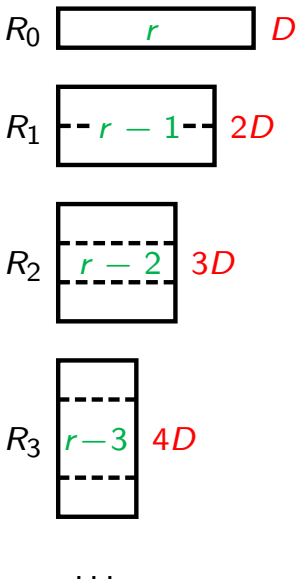



1. $R := r_1 + r_2$, $D := \max\{d_1, d_2\}$.
2. $M_{i,j,1} := x^i \partial^j L_1$ for $0 \leq i \leq (R+2)D$, $0 \leq j \leq r_2$.
3. $M_{i,j,2} := x^i \partial^j L_2$ for $0 \leq i \leq (R+2)D$, $0 \leq j \leq r_1$.
4. Find a $K$-linear dependency of the $M_{i,j,k}$.

$$\#\text{unkowns} = (R+2)\big((R+2)D+1\big),$$
$$\#\text{equations} = (R+1)\big((R+3)D+1\big),$$
$$\dim \text{kernel} \geq D+1.$$

Complexity at least $O(R^{2\omega} D^{\omega})$ for $k = 2$.
Iterative LCLMs in $O(kR^{3\omega} D^{\omega})$; by balancing inputs in $O(R^{3\omega} D^{\omega})$.

# Ore's Method: Extended GCRD Algorithm ($k = 2$)




1. $(R_0, U_0, V_0) := (L_1, 1, 0)$, $(R_1, U_1, V_1) := (L_2, 0, 1)$.
2. For $i = 1, 2, \ldots,$ until $R_{i+1} = 0$:
   - Perform right Euclidean div. $R_{i-1} = Q_i R_i + R_{i+1}$.
   - Maintain invariant $R_i = U_i L_1 + V_i L_2$:
     $U_{i+1} := Q_i U_i - U_{i-1}$, $V_{i+1} := Q_i V_i - V_{i-1}$.
3. For $m$ such that GCRD $= R_m \neq 0$ and $R_{m+1} = 0$, output $U_{m+1} L_1 = -V_{m+1} L_2$.

Generically: $\deg_\partial Q_i = 1$, $\deg_x Q_i = \deg_x R_i = (i+1)D$.
Euclidean div. costs $\tilde{O}\big((r-i)iD\big)$, if $r = \max\{r_1, r_2\}$.

Total complexity is $\tilde{O}(R^3 D)$ for $k = 2$.
Iterative LCLMs in $\tilde{O}(kR^4 D)$; by balancing inputs in $\tilde{O}(R^4 D)$.

# Van Hoeij's Method ($k \geq 2$): I. The Ideas

Generic solutions: $L_j(f_j) = 0, \ \ 1 \leq j \leq k. \qquad c_j = \mathrm{lc}(L_j).$

- If

$$\begin{pmatrix} \ell_0 & \dots & \ell_R \end{pmatrix} \begin{pmatrix} f_1 & \dots & f_k \\ \partial(f_1) & \dots & \partial(f_k) \\ \vdots & & \vdots \\ \partial^R(f_1) & \dots & \partial^R(f_k) \end{pmatrix} = 0,$$

  then $L = \ell_0 + \cdots + \ell_R \partial^R$ is a left common multiple of the $L_j$.

- For computations in $W(x)^k$:

$$\partial^i(f_j) = \frac{n_{i,j,0}}{c_j^i} f_j + \cdots + \frac{n_{i,j,r_j-1}}{c_j^i} \partial^{r_j-1}(f_j) = \frac{1}{c_j^i} n_{i,j}(f_j).$$

# Rank and Kernels: Storjohann and Villard's 2005 Algorithm

**Theorem** (Storjohann and Villard, 2005): Let $M$ be an $m \times n$ matrix of polynomials of degree $d$.

- The rank $\rho$ of $M$ can be computed together with $m - \rho$ linearly independent polynomial elements of the kernel in

$$\tilde{O}\left(\frac{mn}{\rho^2}\mathrm{MM}(\rho)\mathrm{M}(d)\right) \text{ arithmetic operations}$$

  by a (certified) randomized Las Vegas algorithm.
- The sum of the degrees of the basis elements is bounded by

$$\left(m - 2r + r\lceil \log_2 r \rceil\right) d.$$

For $\mathrm{MM}(s) = s^\omega$ and using FFT, this becomes $\tilde{O}(mn\rho^{\omega-2}d)$; if additionally $m = n$, this is $\tilde{O}(n^\omega d)$.

# Van Hoeij's Method ($k \geq 2$): II. Revisiting the Algorithm

1. For $0 \leq i \leq R = kr$, $1 \leq j \leq k$:
$$c_j^{-i} n_{i,j} := \mathrm{rem}(\partial v_{i-1,j}, L_j) = \mathrm{rem}(\partial^i, L_j).$$
2. View each $n_{i,j}$ as a row in $K[x]^{r_j}$ and determine the rank $\rho$ of $N = (n_{i,j})$ (Storjohann–Villard).
3. Randomly select $\rho$ columns from the first $\rho + 1$ rows of $N$ to get a matrix $M$ of rank $\rho$.
4. Determine the polynomial kernel of $M$ (Storjohann–Villard).
5. Modify the (single) basis element so as to take the denominators into account.

Complexity: $\tilde{O}(R^3 D)$ for filling in $V$; $\tilde{O}(R^{\omega+1} D)$ for computing the rank and the kernel; $\tilde{O}(R^3 D)$ for the last modification step.

# Our Linear-Algebraic Approach to Computing LCLMs

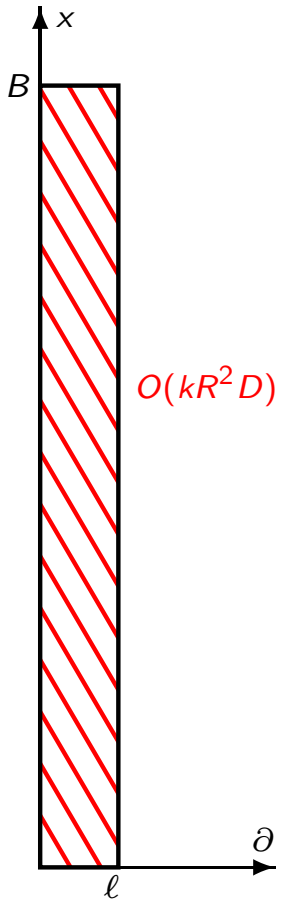
Setting: $\partial a = \sigma(a)\partial + \delta(a)$ where $\sigma$ and $\delta$ do not increase the degrees in $x$ and can be computed in linearly-many operations.

$$S_N(P) = \begin{pmatrix} \partial^{N-\deg_\partial P} P \\ \vdots \\ \partial P \\ P \end{pmatrix}, \quad M_N = \begin{pmatrix} S_N(L_1) & & \\ & \ddots & \\ & & S_N(L_k) \\ -S_{N+1}(1) & \dots & -S_{N+1}(1) \end{pmatrix}.$$

Left kernel of $M_N$ $\longleftrightarrow$ Common multiples and their cofactors.

Fast polynomial-kernel algorithm for solving (Storjohann–Villard).
Tight degree bound for the analysis (from Hadamard's bound).

# Degree Bound for Common Multiples of Minimal Order




$L_i \leftrightarrow (r_i, d_i)$, $R = r_1 + \cdots + r_k$, $D = \max_{1 \le i \le k} d_i$.
$L = \mathrm{LCLM}(L_1, \ldots, L_k) \to$ order $\ell \le R$.

Matrix $M_N$ for $N \le R$:

1. size $m_N \times n_N$ with $m_N = O(kR)$, $n_N = O(kR)$;
2. non-trivial left kernel for $N = R$: $m_R - n_R = 1$;
3. rank is $\rho_N = k(N+1) - R + \ell$, because

$$\ker M_N = \mathrm{span}\{L, \partial L, \ldots, \partial^{N-\ell} L\} = \big(A(x)\, L\big)_{\le N};$$

4. coefficients have degree $\le D$.

Hadamard's bound on $\det M_N$ for $N = \ell = \rho_R + R - k(R+1)$:

$$B = (k\ell + k - R)D = O(kRD).$$

# Algorithm for Common Multiples of Minimal Order

1. Compute $M_R$ and determine its rank $\rho$ (Storjohann–Villard).
2. $\ell := \rho + R - k(R+1)$.
3. Extract $M_\ell$ from $M_R$ by suppressing rows and columns.
4. Compute the kernel of $M_\ell$ (Storjohann–Villard).
5. Return the (single) polynomial solution $(C_1, \ldots, C_k, L)$ found.

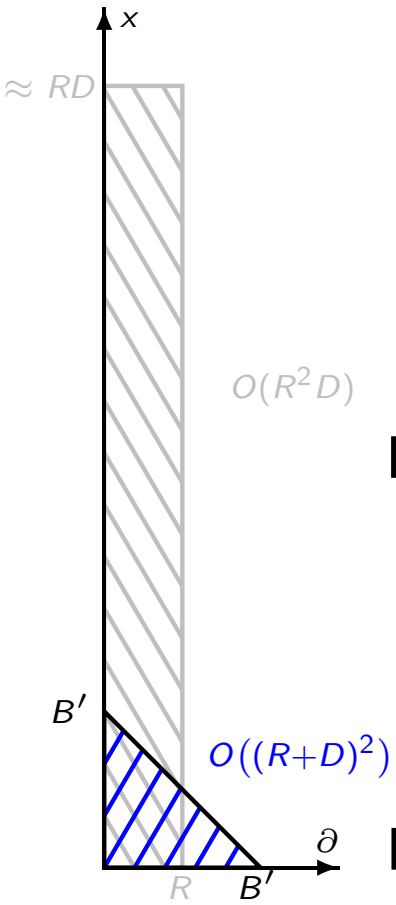
Complexity:

$$\tilde{O}(k^2 R^2 \rho^{\omega-2} D) = \tilde{O}(k^\omega R^\omega D) = \tilde{O}\big(\mathsf{MM}(\text{size})\,\mathsf{M}(\text{degree})\big).$$

# Towards Common Multiples of Small Total Degree

In the spirit of Poole's method: $S'_N(P) = \left(x^i \partial^j P\right)_{i+j=0}^{N-\deg_{x,\partial} P}$,

$$M'_N = \begin{pmatrix} S'_N(L_1) & & \\ & \ddots & \\ & & S'_N(L_k) \\ -S'_{N+1}(1) & \cdots & -S'_{N+1}(1) \end{pmatrix}.$$




For $\deg_{x,\partial} L_i \leq \delta$, a left kernel is ensured by the bound

$$N \geq B' = \left\lceil k\delta + \frac{\left(4k(k-1)\delta^2+1\right)^{1/2} - 3}{2} \right\rceil \approx 2k\delta.$$

Matrix of size $O(k^3\delta^2)$ $\longrightarrow$ linear algebra in $O(k^{3\omega}\delta^{2\omega})$?

# Final Algorithm for Common Multiples of Minimal Order

Series of size $O(B'^2) \subset O(k^2\delta^2)$ instead of operators of size $O(k^3r^2d) \subset O(k^3\delta^3)$!

1. Compute truncated series solutions of the $L_i$ at order $B'^2 + B'$. $O(kB'^2) \subset O(k^3\delta^2)$
2. Take a random linear combination of them. (same)
3. Derive its first $B'$ derivatives. $O(k^3\delta^3)$
4. Compute a Hermite–Padé approximant. $O\big(B'^\omega \mathrm{M}(B')\big) \subset \tilde{O}(k^{\omega+1}\delta^{\omega+1})$
5. Take any two CLMs and compute their GCRD. $O\big(r^\omega \mathrm{M}(d)\big) \subset \tilde{O}(\delta^{\omega+1})$

This algo. = series variant of van Hoeij's algo. + bounds.

# Conclusions

| Algorithm | Complexity | Best for |
|---|---|---|
| Poole's | $\tilde{O}(k^{3\omega} r^{3\omega} D)$ | |
| Ore's | $\tilde{O}(k^4 r^4 D)$ | |
| van Hoeij's | $\tilde{O}(k^{\omega+1} r^{\omega+1} D)$ | large $k, D$; fixed $r$ |
| Ours by S.–V. | $\tilde{O}(k^{2\omega} r^{\omega} D)$ | large $r$; fixed $k, D$ |
| Ours by P.–H. | $\tilde{O}(k^{\omega+1} \delta^{\omega+1})$ | large $k, r, D$ when $\delta \approx r \approx D$ |

$r, D \leq \delta \leq r + D$

When $k = 2$:

1. Li's subresultants give LCLM in $O(r^{\omega} D + r\delta^{\omega})$.
2. Do those algorithms extend to eigenring calculations?