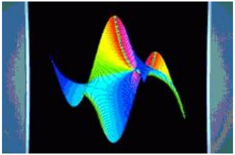
Fast Computation of Common Left Multiples of Linear Ordinary Differential Operators

Alin Bostan (INRIA, France)



joint work with

Frédéric Chyzak, Bruno Salvy and Ziming Li

ACA 2012, Sofia, Bulgaria, June 28, 2012

Main objects & Aim

- $\mathbb{K}$ = an effective field (e.g., $\mathbb{K} = \mathbb{Q}$, or $\mathbb{K} = \mathbb{F}_p$)
- $\mathbb{K}[x]\langle\partial\rangle$ = Weyl algebra of linear differential operators with polynomial coefficients in $\mathbb{K}[x]$; commutation rule $\partial x = x\partial + 1$

Main objects & Aim

- $\mathbb{K}$ = an effective field (e.g., $\mathbb{K} = \mathbb{Q}$, or $\mathbb{K} = \mathbb{F}_p$)
- $\mathbb{K}[x]\langle\partial\rangle$ = Weyl algebra of linear differential operators with polynomial coefficients in $\mathbb{K}[x]$; commutation rule $\partial x = x\partial + 1$

Algebraic formalization of the notion of linear differential equation

$$a_r(x)y^{(r)}(x) + \cdots + a_1(x)y'(x) + a_0(x)y(x) = 0$$

$$\Longleftrightarrow$$

$$L(y) = 0, \quad \text{where} \quad L = a_r(x)\partial^r + \cdots + a_1(x)\partial + a_0(x)$$

Commutation rule formalizes Leibniz's rule $(fg)' = f'g + fg'$.

Main objects & Aim

• $\mathbb{K}$ = an effective field (e.g., $\mathbb{K} = \mathbb{Q}$, or $\mathbb{K} = \mathbb{F}_p$)
• $\mathbb{K}[x]\langle\partial\rangle$ = Weyl algebra of linear differential operators with polynomial coefficients in $\mathbb{K}[x]$; commutation rule $\partial x = x\partial + 1$

Algebraic formalization of the notion of linear differential equation

$$a_r(x)y^{(r)}(x) + \cdots + a_1(x)y'(x) + a_0(x)y(x) = 0$$
$$\Longleftrightarrow$$
$$L(y) = 0, \quad \text{where} \quad L = a_r(x)\partial^r + \cdots + a_1(x)\partial + a_0(x)$$

Commutation rule formalizes Leibniz's rule $(fg)' = f'g + fg'$.

▷ General aim: understand complexity of operations in $\mathbb{K}[x]\langle\partial\rangle$
▷ Specific aim: tight bounds and fast algorithms for LCLM

Definitions

The *least common left multiple* (*LCLM*) of $L_1, \dots, L_k \in \mathbb{K}[x]\langle\partial\rangle$ is the least order operator $L \in \mathbb{K}[x]\langle\partial\rangle$, primitive w.r.t. x, such that

$$L = Q_1 L_1 = \cdots = Q_k L_k \quad \text{for some cofactors } Q_1, \dots, Q_k \in \mathbb{K}(x)\langle\partial\rangle$$

Definitions

The *least common left multiple* (*LCLM*) of $L_1, \dots, L_k \in \mathbb{K}[x]\langle\partial\rangle$ is the least order operator $L \in \mathbb{K}[x]\langle\partial\rangle$, primitive w.r.t. x, such that

$$L = Q_1 L_1 = \cdots = Q_k L_k \quad \text{for some cofactors } Q_1, \dots, Q_k \in \mathbb{K}(x)\langle\partial\rangle$$

▷ When $\mathbb{K} = \mathbb{C}$, this coincides with the least order differential operator whose solution space $V(L)$ equals $V(L_1) + \cdots + V(L_k)$

Definitions

The *least common left multiple* (*LCLM*) of $L_1, \ldots, L_k \in \mathbb{K}[x]\langle\partial\rangle$ is the least order operator $L \in \mathbb{K}[x]\langle\partial\rangle$, primitive w.r.t. x, such that

$$L = Q_1 L_1 = \cdots = Q_k L_k \quad \text{for some cofactors } Q_1, \ldots, Q_k \in \mathbb{K}(x)\langle\partial\rangle$$

▷ When $\mathbb{K} = \mathbb{C}$, this coincides with the least order differential operator whose solution space $V(L)$ equals $V(L_1) + \cdots + V(L_k)$

▷ Mathematically, the rational Weyl algebra $\mathbb{K}(x)\langle\partial\rangle$ is nicer

Definitions

The *least common left multiple* (*LCLM*) of $L_1, \dots, L_k \in \mathbb{K}[x]\langle\partial\rangle$ is the least order operator $L \in \mathbb{K}[x]\langle\partial\rangle$, primitive w.r.t. x, such that

$$L = Q_1 L_1 = \cdots = Q_k L_k \quad \text{for some cofactors } Q_1, \dots, Q_k \in \mathbb{K}(x)\langle\partial\rangle$$

▷ When $\mathbb{K} = \mathbb{C}$, this coincides with the least order differential operator whose solution space $V(L)$ equals $V(L_1) + \cdots + V(L_k)$

▷ Mathematically, the rational Weyl algebra $\mathbb{K}(x)\langle\partial\rangle$ is nicer
▷ Algorithmically, the polynomial Weyl algebra $\mathbb{K}[x]\langle\partial\rangle$ is nicer

Rational Weyl algebra is Euclidean

Theorem [Libri 1833, Brassinne 1864, Wedderburn 1932, Ore 1932]
$\mathbb{K}(x)\langle\partial\rangle$ is a non-commutative (left and right) Euclidean domain:
for $A, B \in \mathbb{K}(x)\langle\partial\rangle$, there exist unique $Q, R \in \mathbb{K}(x)\langle\partial\rangle$ such that

$$A = QB + R, \qquad \text{and} \qquad \text{ord}(R) < \text{ord}(B).$$

(This is called the Euclidean right division of A by B.)

As a consequence, any $A, B \in \mathbb{K}(x)\langle\partial\rangle$ admit a greatest common right divisor (GCRD) and a least common left multiple (LCLM).

Moreover, $\text{GCRD}(A, B)$ and $\text{LCLM}(A, B)$ can be computed by a non-commutative version of the extended Euclidean algorithm.

Motivations

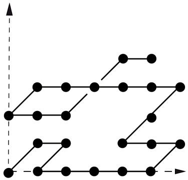
- LCLM and GCRD are the most basic operations after product
- Complexity of product is now well understood (cf. previous talk)
- Several higher level algorithms rely on the efficiency of LCLMs:
  - addition of D-finite functions
  - formal invariants of linear differential systems [Barkatou & Chyzak & Loday-Richaud 2003]
  - symbolic summation and integration [Le 2003]
  - exponential solutions of LODEs [Cluzeau & van Hoeij 2004]

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

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



Theorem [B. & Kauers 2010] $G(x,y,t)$ is an algebraic function.†

→ Effective, computer-driven discovery and proof

→ Proof involves a LCLM computation of two 11th order (guessed) differential operators for $G(x,0,t)$, and $G(0,y,t)$. The LCLM has order 20, tridegree (359,717,279) in (t,x,y), 1.5 billion coefficients

† The MinPoly $P(x,y,t,G(x,y,t)) = 0$ has $> 10^{11}$ monomials; ≈30Gb (!)

Previous results – prehistory

Libri 1833, Brassinne 1864: structural analogy between polynomials and LODEs; bases of a non-commutative elimination theory (GCRD, LCLM); Euclidean-type algorithm for GCRD

Von Escherich 1883: differential resultant of two linear differential operators; embryo of a linear algebra algorithm for LCLM

Heffter 1896, Poole 1936: first linear algebra algorithm for LCLM

Pierce 1903: roots of subresultant theory for LODEs

Blumberg 1912: first systematic algebraic account of LODEs

Ore 1932: Euclidean-type theory in algebraic framework of *skew polynomials*; formula for LCLM in terms of Euclidean remainders

Wedderburn 1932: LCLM computation by non-commutative version of the extended Euclidean algorithm

Previous results – modern times

Bronstein & Petkovšek 1996: algorithms for skew polynomials

Grigoriev 1990: extension of von Escherich's Sylvester-type matrix to several operators; fast GCRD algorithm for several LODEs

Giesbrecht 1992, 1998: Heffter-style LCLM algorithm $O(r^{\omega})$ ops. in $\mathbb{K}(x)$; Euclidean-style LCLM algorithm $\widetilde{O}(r^2)$ ops. in $\mathbb{K}(x)$

Li 1998: coefficient growth matters; subresultant theory for Ore polynomials, determinantal formulas for GCRD and LCLM

Li 2002, Giesbrecht & Zhang 2003: $O(r^5 d^2)$ algorithm for LCLM of two operators of bidegree (d, r) in (x, ∂)

B. 2003: sketch of a Hermite-Padé evaluation-interpolation strategy for LCLMs; states $O(n^{\omega+2})$ for LCLM in bidegree (n, n)

van der Hoeven 2011: reduction of LCLM and GCRD to product; proves $O(n^{\omega+2})$ for LCLM in bidegree (n, n)

Existing strategies for LCLM of several operators

To compute $L = \mathrm{LCLM}(L_1, \ldots, L_k)$, with $\mathrm{bideg}(L_j) = (d, r)$:

1. iterative schemes + any algorithm for two operators:

$$L = \mathrm{LCLM}\left(L_1, \mathrm{LCLM}(L_2, \ldots, \mathrm{LCLM}(L_{k-1}, L_k)\right)$$

2. van Hoeij's algorithm (implemented in `DEtools`):

Idea:

$$\mathrm{rem}(L, L_1) = \ldots = \mathrm{rem}(L, L_k) = 0$$

amounts to a linear system in the unknown coefficients of L

▷ matrix has size $\leq kr$ and entries of degree $\leq krd$.

3. Abramov & Le & Li 2005: improvement of van Hoeij's algorithm

▷ predict $\mathrm{ord}(L)$ by a rank computation, then compute 1-dim kernel

Contributions

1. size bounds (order and degree) on LCLM and on cofactors
2. complexity analyses, and improvements, of previous algorithms
3. new, quasi-optimal algorithm for LCLM and cofactors
4. existence of common left multiples of smaller total size
5. quasi-optimal heuristic for the LCLM (without cofactors)
6. implementation (in Magma)

New matrix formulation for LCLMs

- Sylvester-type matrix for $P \in \mathbb{K}[x]\langle\partial\rangle$ and $n \geq \mathrm{ord}(P)$:

$$S_n(P) := \left(\partial^{n-\mathrm{ord}(P)}P,\ \cdots,\ \partial P,\ P\right)^T$$

▷ $S_n(P)$ has $n - \mathrm{ord}(P) + 1$ rows and $n + 1$ columns

- For $n \geq \max_i \mathrm{ord}(L_i)$, define the block-Sylvester matrix

$$M_n(L_1, \ldots, L_k) := \begin{pmatrix} S_n(L_1) & & & \\ & S_n(L_2) & & \\ & & \ddots & \\ & & & S_n(L_k) \\ S_n(-1) & S_n(-1) & \cdots & S_n(-1) \end{pmatrix}.$$

▷ M_n has $(k+1)(n+1) - \sum_{i=1}^{k} \mathrm{ord}(L_i)$ rows and $k(n+1)$ columns

New matrix formulation for LCLMs

Theorem (new) Assume $n \geq \ell := \mathrm{ord}\,(\mathrm{LCLM}(L_1, \dots, \; L_k))$. Then:

(i) L is a common left multiple of $L_1, \dots L_k$ such that

$$\mathrm{ord}(L) \leq n \quad \text{and} \quad L = Q_1 L_1 = \cdots = Q_k L_k$$

if and only if $(Q_1, \dots, Q_k, L)$ belongs to the left kernel of M_n.

(ii) $\ell = \mathrm{rank}(M_n) + \sum_{i=1}^{k} \mathrm{ord}(L_i) - k(n+1).$

(iii) $\dim \ker(M_\ell) = 1.$

New algorithm

1. Compute M_s, for $s = \mathrm{ord}(L_1) + \cdots + \mathrm{ord}(L_k)$
2. Determine its rank ρ; set $\ell := \rho + s - k(s+1)$
3. Extract submatrix M_ℓ of M_s, and find its 1-dim kernel $\mathcal{K}$
4. Construct the LCLM from the last $\ell + 1$ coordinates of $\mathcal{K}$

▷ Steps 2. and 3. have dominant cost.

▷ They amount to *rank* and *kernel* computation of a polynomial matrix of size $\leq k^2 r + k$ and degree $\leq d$.

▷ Compare with van Hoeij's matrix of size $\leq kr$ and degree $\leq krd$.

Polynomial linear algebra

Theorem [Storjohann & Villard 2005]
Let M be an $m \times n$ matrix with entries in $\mathbb{K}[x]_{<d}$. One can compute:

1. the rank ρ of M
2. a polynomial basis of the left kernel of M

in $\widetilde{O}(mn\,\rho^{\omega-2}\,d)$ operations in $\mathbb{K}$.

- the best known complexity result on polynomial linear algebra
- very beautiful, but complex, algorithm; not easy to implement
- when $\dim\ker M = 1$, algorithm in [Storjohann 2003] suffices; complexity $\mathcal{O}(\mathrm{MM}(n,d)\log d)$, where $\mathrm{MM}(n,d) = \widetilde{O}(n^\omega d)$

New algorithm

1. Compute M_s, for $s = \mathrm{ord}(L_1) + \cdots + \mathrm{ord}(L_k)$
2. Determine its rank ρ; set $\ell := \rho + s - k(s+1)$
3. Extract submatrix M_ℓ of M_s, and find its 1-dim kernel $\mathcal{K}$
4. Construct the LCLM from the last $\ell + 1$ coordinates of $\mathcal{K}$

▷ Steps 2. and 3. have dominant cost.

▷ They amount to *rank* and *kernel* computation of a polynomial matrix of size $\leq k^2 r + k$ and degree $\leq d$.

▷ Complexity (using Storjohann–Villard's algorithms): $\widetilde{O}((k^2 r)^\omega d)$

▷ Complexity of van Hoeij's algorithm (using S.–V.): $\widetilde{O}((kr)^\omega krd)$

A toy example

Let us compute the LCLM of $L_1 = \partial + x$ and of $L_2 = x\partial + 1$.

$$M_2(L_1, L_2) = \begin{bmatrix} 1 & x & 1 & & & \\ 0 & 1 & x & & & \\ & & & x & 2 & 0 \\ & & & 0 & x & 1 \\ -1 & & & -1 & & \\ & -1 & & & -1 & \\ & & -1 & & & -1 \end{bmatrix}$$

- ▷ $\operatorname{rank}(M_2) = 6$, $\operatorname{ord}(\operatorname{LCLM}(L_1, L_2)) = 6 + (1+1) - 2(2+1) = 2$
- ▷ Basis of the left kernel of M_2

$$(x^3 - x,\ -2,\ x^2 - 1,\ x^3 - 3x,\ x^3 - x,\ x^4 - x^2 - 2,\ x^3 - 3x)$$

Thus $\operatorname{LCLM}(L_1, L_2) = (x^3 - x)\partial^2 + (x^4 - x^2 - 2)\partial + x^3 - 3x$,
$Q_1 = (x^3 - x)\partial - 2$ and $Q_2 = (x^2 - 1)\partial + x^3 - 3x$

Main results (I)

Theorem Let $L_1, \dots, L_k \in \mathbb{K}[x]\langle\partial\rangle$ have bidegrees (d, r) in (x, ∂). Then $\mathrm{LCLM}(L_1, \dots, L_k)$

- has order at most kr,
- has degrees in x at most $dk(rk - r + 1)$,
- can be computed, together with the cofactors, in $\widetilde{O}(k^{2\omega} r^{\omega} d)$ arithmetic operations in $\mathbb{K}$.

- Size bounds are sharp (generically reached)
- Arithmetic size of the output is $k^4 r^2 d$
- New algorithm has quasi-optimal complexity (w.r.t. output size)

Main results (I)

Theorem Let $L_1, \dots, L_k \in \mathbb{K}[x]\langle\partial\rangle$ have bidegrees (d, r) in (x, ∂). Then $\mathrm{LCLM}(L_1, \dots, L_k)$

- has order at most kr,
- has degrees in x at most $dk(rk - r + 1)$,
- can be computed, together with the cofactors, in $\widetilde{O}(k^{2\omega} r^{\omega} d)$ arithmetic operations in $\mathbb{K}$.

- Size bounds are sharp (generically reached)
- Arithmetic size of the output is $k^4 r^2 d$
- New algorithm has quasi-optimal complexity (w.r.t. output size)

Complexity of (improved versions of) existing algorithms:

Heffter's*	Li's*	van der Hoeven's	van Hoeij's
$\widetilde{O}\left(k^5 r^4 d\right)$	$\widetilde{O}\left(k^{\omega+3} r^{\omega+2} d\right)$	$\widetilde{O}\left(k^5 r^4 d\right)$	$\widetilde{O}\left(k^{\omega+1} r^{\omega+1} d\right)$

Algorithms marked by $(^\star)$ also compute cofactors for the same cost.

Main results (II)

Theorem Assume an algorithm $\mathcal{A}$ computes $\mathrm{LCLM}(L_1, L_2)$ in $\widetilde{O}(R^\alpha D^\beta)$ for any $L_1, L_2 \in \mathbb{K}[x]\langle\partial\rangle$ of bidegree (D, R) in (x, ∂).
Then, one can compute the LCLM of $L_1, \dots, L_k \in \mathbb{K}[x]\langle\partial\rangle$ of bidegrees (d, r) in (x, ∂) in $\widetilde{O}(k^{\alpha+2\beta} r^{\alpha+\beta} d^\beta)$ operations in $\mathbb{K}$.

Idea: Divide-and-conquer! (better than iterative schemes)

- partition the family $(L_1, \dots, L_k)$ into pairs,
- compute the LCLM of each pair using algorithm $\mathcal{A}$,
- remove polynomial content,
- compute LCLMs of new pairs, and so on.

Total cost:
$$\sum_{j=0}^{\lfloor \log(k) \rfloor} \frac{k}{2^{j+1}} \cdot (2^j r)^\alpha \cdot (4^j dr)^\beta = O(k^{\alpha+2\beta} r^{\alpha+\beta} d^\beta).$$

Main results (III)

Theorem Let $L_1, \dots, L_k \in \mathbb{K}[x]\langle\partial\rangle$ have bidegrees (n, n) in (x, ∂). There exist nonzero common left multiples of total degree $\leq 4kn$ in (x, ∂), and total arithmetic size $O\big(k^2 n^2\big)$.

- Recall: LCLM has order at most kn and degrees at most $k^2 n^2$
- CLM algorithm using Hermite-Padé approximants $\widetilde{O}(k^{\omega+1} n^{\omega+1})$

- Fast heuristic for computing the LCLM (without cofactors):

(*i*) compute $O(1)$ CLMs of order and degree at most $4kn$;

(*ii*) take two random linear combinations with coefficients in $\mathbb{K}[x]$;

(*iii*) return their GCRD.

$\longrightarrow$ Total cost: $\widetilde{O}(k^{\omega+1} n^{\omega+1})$, quasi-optimal w.r.t. the LCLM size

Experimental results†

n	Magma's LCLM	New	New+S	(D, N)	MM(N, D)	output size
2	0.01	0.00	0.01	(2,10)	0.01	65
3	0.01	0.01	0.03	(3,14)	0.01	175
4	0.02	0.01	0.07	(4,18)	0.03	369
6	0.10	0.06	0.17	(6,26)	0.06	1105
8	0.49	0.19	0.54	(8,34)	0.15	2465
12	6.84	0.91	1.37	(12,50)	0.41	7825
16	49.24	3.48	4.93	(16,66)	0.91	17985
23	718.02	20.51	11.09	(23,94)	2.60	51935
32	9355.47	115.53	40.83	(32,130)	6.73	137345
46	168434.66	791.01	130.40	(46,186)	21.51	402225

Timings (in sec.) for LCLMs of $k = 2$ random operators in $\mathbb{F}_{9001}[x]\langle\partial\rangle$ of bidegrees (n, n) in (x, ∂).

▷ complexity analyses and size bounds are confirmed in practice

† All computer calculations were performed on a Quad-Core Intel Xeon X5160 processor at 3GHz, with 8GB of RAM.

Conclusion, future work

Creeds:

- complexity analysis = tool for algorithmic design
- polynomial linear algebra = non-commutative complexity yardstick

Main result: quasi-optimal algorithm for LCLM + cofactors

To do:

- turn heuristic into a quasi-optimal algorithm for LCLM alone
- extension to the Ore framework, and to more operations (right division, symmetric product, exterior power, ...)
- bit complexity
- Lehmer-Knuth-Schönhage half-LCLM/GCRD algorithms?

Thanks for your attention!