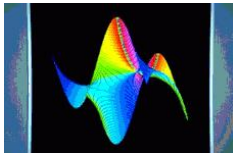


Polynomial and Rational Solutions of Linear Differential or Difference Equations

Bruno Salvy
Bruno.Salvy@inria.fr

Algorithms Project, Inria



Joint work with Alin Bostan and Thomas Cluzeau

Liouville (1833) did most of it!

$$a_0(x)y^{(d)}(x) + \cdots + a_d(x)y(x) = 0.$$

- Polynomial solutions:

Si l'on se bornait à demander les intégrales entières, le problème n'offrirait aucune difficulté.

- Algorithm for rational solutions, later improved by Abramov *et alii*.



Our result: **better complexity** (still exponential).

Timings

$$(2x + 1)^3 y'' + (2x + 1)(8x + 3)y' + 2N((4x + 2)N - 4x - 1)y = 0$$

N	LFS	BinSplit	BsGs
2^{10}	1.58	0.10	0.02
2^{12}	62.59	0.44	0.03
2^{14}	4597.2	2.40	0.07
2^{16}	> 4Gb	14.67	0.19
2^{18}		89.05	0.44
2^{20}		528.73	1.07
2^{22}		3060.1	2.54
2^{24}		> 1h	6.09

LFS: Liouville's method

(Maple) $\tilde{O}(N^2)$

BinSplit: Our deterministic

algo (Maple) $\tilde{O}(N)$

BsGs: Our probabilistic algo

(Magma) $\tilde{O}(\sqrt{N})$

Output different

Applications

- Indefinite hypergeometric summation [Gosper78]

$$\sum_{k=0}^n \frac{(3k)!}{k!(k+1)!(k+2)!27^k} = \frac{(81n^2 + 261n + 200)(3n+2)!}{40(n+2)!(n+1)!n!27^n} - \frac{9}{2}$$

- Definite summation and integration [Zeilberger90, Chyzak00]

$$\sum_{n=0}^{\infty} J_{2n+1/2}(x) = \int_0^x \frac{\cos t}{\sqrt{2\pi t}} dt$$

- Liouvillian solutions of LDEs [Marotte1898, Kovacic86, Singer81, . . .]
- Hypergeometric solutions of LREs [Petkovšek90]
- Desingularization of LDEs and LREs [ChDuLeMaMiSa05]

They all

- **need rational or polynomial solutions** of LDEs or LREs;
- **waste time** when none exists.

Singular Points & Indicial Polynomial

$$\mathcal{L}y(x) = a_0(x)y^{(d)}(x) + \cdots + a_d(x)y(x) = 0.$$

- Cauchy: singular points only at roots of a_0 .
- Indicial polynomial $P_\alpha(\lambda)$:

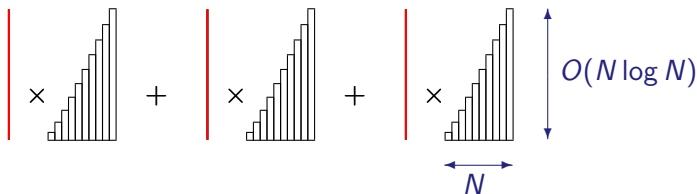
$$\mathcal{L}(x - \alpha)^\lambda(1 + c_1(x - \alpha) + \cdots) = P_\alpha(\lambda)(x - \alpha)^{\lambda+c}(1 + \cdots).$$

$$\underbrace{(n+3)(n+2)u_{n+3} + \cdots}_{P_0(n+3)} + \cdots + 8 \underbrace{(n-N)(n-N+1)u_n}_{P_\infty(n)} = 0$$

- Exponential bound on degree and orders of poles
- Rational solutions [Liouville1833]: patch up local polar behaves

$$y(x) := \tilde{y}(x) \prod_{\substack{a_0(\alpha)=0, \\ P_\alpha(N_\alpha)=0, \\ N_\alpha \in \mathbb{Z}^-}} (x - \alpha)^{N_\alpha}.$$

Shape of Polynomial Solutions



- Size of Solutions: $O(N^2 \log N)$ bits
- Compact representation **Compact representation**: $O(N \log N)$ bits

Theorem (BoCISa05)

Our algorithm BinSplitPolySols computes compact representation and degree in $\tilde{O}(N)$ bit operations. Knowing the degree D , the expanded polynomial is computed in $\tilde{O}(D^2)$ bit operations.

Quasi-optimal!

Fast Multiplication

- **Balanced** input: size $T \times$ size T
 - Naïve: $I(T) = O(T^2)$
 - Karatsuba (1963): $I(T) = O(T^{1.59})$
 - Fast Fourier Transform: $I(T) = O(T \log T \log \log T)$
[Schönhage-Strassen71]
- Same for polynomials ($M(T)$).
- Many applications via **Newton** iteration, including division.

Algorithms	Ours	Before
Degrees (probabilistic)	$O(M(\sqrt{N})I(\log N))$	—
Compact and degree	$O(I(N \log N) \log N)$	$O(N^2 I(\log N))$
Expanded, degree D	$O(DI(D \log D))$	

Factorial

Problem (Fast computation of $N! = 1 \times \cdots \times N$)

Naïve way: complexity $O(N^2 I(\log N))$

Binary Splitting:

$$N! = \underbrace{(1 \times \cdots \times \lfloor N/2 \rfloor)}_{\text{size } \frac{1}{2} N \log N} \times \underbrace{(\lceil N/2 \rceil \times \cdots \times N)}_{\text{size } \frac{1}{2} N \log N}$$

and recurse.

Complexity $O(I(N \log N) \log N)$.

Numerous applications [Hakmem72, Brent75, Chudnovsky²⁸⁸]

Even faster way [Borwein85, Schönhage94] $O(I(N \log N))$.

Binary Splitting for Polynomial Solutions

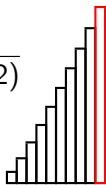
$$* + *x + \dots + *x^N + *x^{N+1} + *x^{N+2} + *x^{N+3} + O(x^{N+4})$$

$$(n+3)(n+2)u_{n+3} + \dots + 8(n-N)(n-N+1)u_n = 0$$

First order recurrence on vectors $U_n = {}^t(u_{n+3}, u_{n+2}, u_{n+1})$:

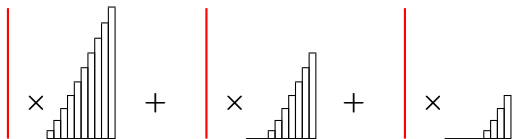
$$U_N = \underbrace{\begin{pmatrix} * & * & * \\ (N+3)(N+2) & 0 & 0 \\ 0 & (N+3)(N+2) & 0 \end{pmatrix}}_{A(N)} \frac{U_{N-1}}{(N+3)(N+2)}$$

$$= \underbrace{A(N) \cdots A(-1)}_{\text{matrix factorial}} \frac{U_{-2}}{(N+2)(N+1)!^2}$$



Complexity: like for $N!$.

Operations on the Compact Representation



Compact Representation: recurrence & generalized initial conditions.

- Division by a power of x : easy
 - Evaluation: linear recurrence for $v_k = \sum_{n \leq k} u_n x^n$
 $\rightarrow v_N$ by binary splitting (for x algebraic).
 Applications: $\exp(1), \gamma, \pi, \dots$ [Hakmem, Brent, ChCh]
 - Evaluation of a derivative: *idem*
- \rightarrow Compact representation at another point
- \rightarrow **Rational solutions.**

From Recurrence to Recurrence

$$a_0(n)u(n+d) + \cdots + a_d(n)u(n) = 0.$$

In general, coefficients of u **do not** satisfy a linear recurrence.

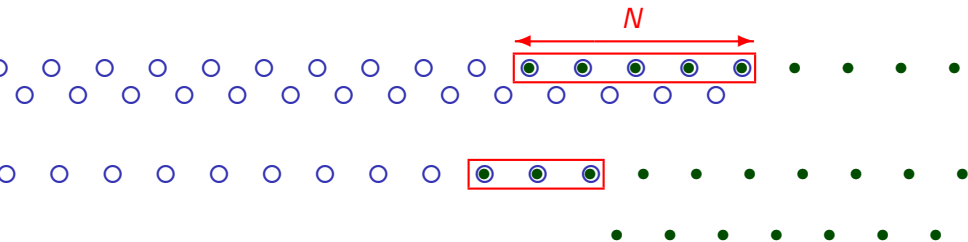
They do on a binomial basis [Abramov-Bronstein-Petkovšek 95]:

$$u(n) = \sum_{k=0}^N c_k \binom{n-a}{k}.$$

→ **same algorithm** for the compact representation

Poles of Rational Solutions

$$a_0(n)a_0(n)u(n+d) + \cdots + a_d(n)a_d(n)u(n) = 0.$$



Rational solutions

Abramov 89: bound the poles and look for polynomials.

Probabilistic Algorithm

Theorem (BoCISa05)

Given $c \geq 0$, our algorithm BsGsPolySols computes the degrees of polynomial solutions with $\tilde{O}(\sqrt{N})$ bit operations. The result is correct with probability $\geq 1 - 1/(2 \log^c N)$.

Idea:

- 1 Compute the matrix factorial
 - with only $O(\sqrt{N})$ operations
 - modulo a prime of bit size only $O(\log N)$
- 2 Bound the probability that the rank drops.

Baby-Steps/Giant-Steps

Problem ($N! \bmod p$ in less than N operations)

Naïve: N arithmetic operations

Strassen 76: $\tilde{O}(\sqrt{N})$.

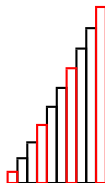
- 1 Compute $Q(x) = (x + 1) \cdots (x + k)$ $O(M(k) \log k)$;
- 2 Evaluate $Q(0), Q(k), \dots, Q(N - k)$ $O(NM(k) \log(k)/k^2)$;
- 3 Compute their product $O(N/k)$.

Recurrences via matrix factorials [Chudnovsky²88]:

$$O(m^\omega M(\sqrt{dN}) + m^2 M(\sqrt{dN}) \log(dN))$$

We improve this to

$$O(m^\omega \sqrt{dN} + m^2 M(\sqrt{dN}))$$



Multipoint Evaluation

Problem (Evaluate P at x_1, \dots, x_d , $\deg P = d$.)

Naïve: $O(d^2)$ operations.

Borodin & Moenck 74: Recursive computation

$$Q_i^j := (x - x_i) \cdots (x - x_j)$$

$$P(x_i) = \begin{cases} (P \bmod Q_1^{\lfloor d/2 \rfloor})(x_i), & i \leq d/2 \\ (P \bmod Q_{\lfloor d/2 \rfloor}^d)(x_i), & i > d/2. \end{cases}$$

$$\begin{aligned} \text{Complexity: } C(d) &= \underbrace{2C(d/2)}_{\text{recursion}} + \underbrace{2M(d/4)}_Q + \underbrace{\frac{7}{2}M(d)}_{\text{divisions}} \\ &= O(M(d) \log d). \end{aligned}$$

Better constant in [Bostan-Lecerf-Schost 03] (Tellegen).

Polynomial Extrapolation

Problem $((P(0), \dots, P(d)) \mapsto (P(a), P(a+1), \dots, P(a+d)), \deg P = d)$

Interpolation / evaluation: $O(M(d) \log d)$.

Bostan-Gaudry-Schost 03:

$$\begin{aligned}
 P(a+k) &= \sum_{i=0}^d P(i) \frac{\prod_{j \neq i} (a+k-j)}{\prod_{j \neq i} (j-i)}, \quad (\text{Lagrange}) \\
 &= \prod_j (a+k-j) \underbrace{\sum_{i=0}^d \frac{P(i)}{\prod_{j \neq i} (j-i)} \frac{1}{a+k-i}}_{[x^k] \sum_{i=0}^d \frac{P(i)}{\prod_{j \neq i} (j-i)} x^i \sum_{i=0}^{2d} \frac{1}{a+i-d} x^i}.
 \end{aligned}$$

Complexity: $O(M(d))$.

Faster $N! \bmod p$

Bostan-Gaudry-Schost 03:

- Recall Strassen:
 - ① Compute $Q(x) = (x + 1) \cdots (x + k)$ $O(M(k) \log k)$;
 - ② Evaluate $Q(0), Q(k), \dots, Q(N - k)$ $O(NM(k) \log(k)/k^2)$;
- Recursion:
 - from $(x + 1) \cdots (x + k/2)$ evaluated at $(0, \dots, (N - k)/2)$,
 - 3 extrapolations.**
- Complexity ($k = \sqrt{N}$): $O(M(k))$
- Generalizes to matrix factorials in arbitrary degree.

Probability Estimate

Kernel of a matrix factorial

Lemma (Zippel)

A, B positive integers. If p prime is chosen uniformly at random in $\mathcal{S} = \{B, \dots, 2B\}$ then $\Pr(p|A) \leq 2 \log(A)/B$.

Proof:

- Number of distinct divisors of A in \mathcal{S} : $< \log(A)/\log(B)$
- Size of \mathcal{S} : $\geq B/2 \log(B)$.

Matrix factorials:

Size of the entries $O(N \log N)$, size of minors *idem*.

Precise estimates for B available

(and required for an implementation).

Conclusion

- Compact representation & fast operations
→ quasi-optimal algorithms for rational functions.
- Also, a criterion in polynomial complexity for a large class.
- Extensions
 - Non-homogeneous case: $\mathcal{L}y = r$
 - Parameterized non-homogeneous: $\mathcal{L}y = \lambda_1 r_1 + \dots + \lambda_k r_k$
 - Faster Zeilberger, Chyzak.
 - Possible extension to linear differential systems.