Vincent Neiger ..... LIP6, Sorbonne Université, France
Bruno Salvy ..... Inria, ENS Lyon, France
Éric Schost ..... U. Waterloo, Canada
Gilles Villard ..... CNRS, ENS Lyon, France

faster modular composition, using two relation matrices

outline

- **context and result**
  - univariate polynomial computations
  - modular composition and minpoly
  - improved algebraic complexity bound
- **motivations and algorithms**
- **overview of the approach**

modular composition

composition: (coefficients in a field)

[in] $a(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$

[in] $p(y) = p_0 + p_1 y + p_2 y^2 + \cdots + p_{n-1} y^{n-1}$

[out] $p(a(x)) = p_0 + p_1 a(x) + p_2 a(x)^2 + \cdots + p_{n-1} a(x)^{n-1}$

size n^2 $\rightsquigarrow$ optimistic target complexity $= O(n^2)$

modular composition

composition: (coefficients in a field)

[in] $a(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$

[in] $p(y) = p_0 + p_1 y + p_2 y^2 + \cdots + p_{n-1} y^{n-1}$

[out] $p(a(x)) = p_0 + p_1 a(x) + p_2 a(x)^2 + \cdots + p_{n-1} a(x)^{n-1}$

size n^2 $\rightsquigarrow$ optimistic target complexity $= O(n^2)$

modular composition:

[in] $f(x) = f_0 + f_1 x + f_2 x^2 + \cdots + f_{n-1} x^{n-1} + x^n$

[out] $p(a) \bmod f = p_0 \alpha_0 + p_1 \alpha_1 + p_2 \alpha_2 + \cdots + p_{n-1} \alpha_{n-1}$

where $\alpha_k = a^k \bmod f \in \mathbb{K}[x]_{<n}$

size n $\rightsquigarrow$ optimistic target complexity $= O(n)$

modular composition

composition: (coefficients in a field)

[in] $a(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$

[in] $p(y) = p_0 + p_1 y + p_2 y^2 + \cdots + p_{n-1} y^{n-1}$

[out] $p(a(x)) = p_0 + p_1 a(x) + p_2 a(x)^2 + \cdots + p_{n-1} a(x)^{n-1}$

size n^2 $\rightsquigarrow$ optimistic target complexity $= O(n^2)$

modular composition:

[in] $f(x) = f_0 + f_1 x + f_2 x^2 + \cdots + f_{n-1} x^{n-1} + x^n$

[out] $p(a) \bmod f = p_0 \alpha_0 + p_1 \alpha_1 + p_2 \alpha_2 + \cdots + p_{n-1} \alpha_{n-1}$

where $\alpha_k = a^k \bmod f \in \mathbb{K}[x]_{<n}$

size n $\rightsquigarrow$ optimistic target complexity $= O(n)$

a fundamental problem:

[Kaltofen-Shoup '98+'99] [Abelard-Couvreur-Lecerf '20+'22] [Shoup '99] [Giesbrecht-Jamshidpey-Schost '21+'26]

- polynomial factorization
- Riemann-Roch space computations
- minimal polynomial modulo f
- computing normal bases

"fast": measuring efficiency

efficient algorithms for polynomials, matrices, power series, ...
with coefficients in some base field $\mathbb{K}$

- low **complexity bound**
- low **execution time**

..., low memory usage, low power consumption, ...

- prime field $\mathbb{K} = \mathbb{Z}/p\mathbb{Z}$
- rational numbers $\mathbb{K} = \mathbb{Q}$

- finite extensions $\mathbb{K}[x]/\langle f(x)\rangle$
- rational fractions $\mathbb{K}(x, y, z)$

"fast": measuring efficiency

efficient algorithms for polynomials, matrices, power series, ...
with coefficients in some base field $\mathbb{K}$

- low **complexity bound**
- low **execution time**

..., low memory usage, low power consumption, ...

- prime field $\mathbb{K} = \mathbb{Z}/p\mathbb{Z}$
- rational numbers $\mathbb{K} = \mathbb{Q}$

- finite extensions $\mathbb{K}[x]/\langle f(x)\rangle$
- rational fractions $\mathbb{K}(x, y, z)$

algebraic complexity bounds
$\rightsquigarrow$ count basic operations in $\mathbb{K}$

- operations $+, -, \times, \div, =$
- notation $O(\cdot)$ and $\tilde{O}(\cdot)$
- this talk: no parallelism, single thread

related?

for the few runtimes in this talk:

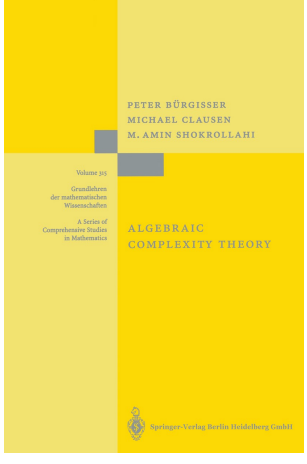
- my laptop with FLINT/NTL/PML
- prime field $\mathbb{K}$ of size $\approx 2^{60} \approx 10^{18}$

practical performance
$\rightsquigarrow$ measure software runtime

fundamental computations on polynomials

most problems: quasi-linear via reductions to multiplication

$$M(n) \in O(n \log(n) \log\log(n)) \subset \tilde{O}(n)$$



fundamental computations on polynomials

most problems: quasi-linear via reductions to multiplication

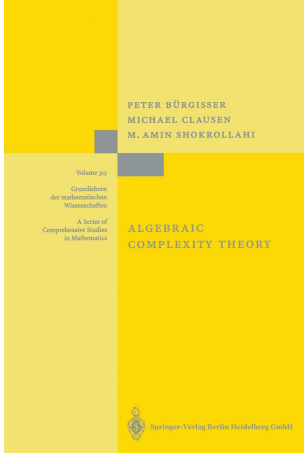
$$M(n) \in O(n \log(n) \log\log(n)) \subset \tilde{O}(n)$$

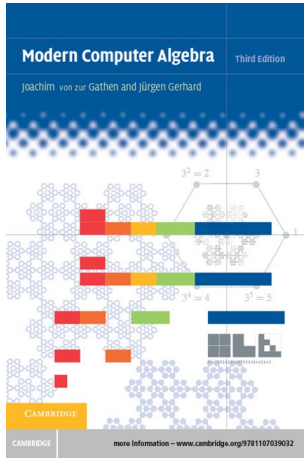
$O(M(n))$

- addition $a + b$, multiplication $a * b$
- division with remainder $a = qb + r$
- power series inverse $a^{-1} \bmod x^n$

$O(M(n)\log(n))$

- eval./interp. $a \leftrightarrow a(x_1), \ldots, a(x_n)$
- extended GCD $au + bv = \gcd(a, b)$
- rational reconstruction $a = \frac{p}{q} \bmod M$





fundamental computations on polynomials

most problems: quasi-linear via reductions to multiplication

$$M(n) \in O(n \log(n) \log\log(n)) \subset \tilde{O}(n)$$

$O(M(n))$

- addition $a + b$, multiplication $a * b$
- division with remainder $a = qb + r$
- power series inverse $a^{-1} \bmod x^n$

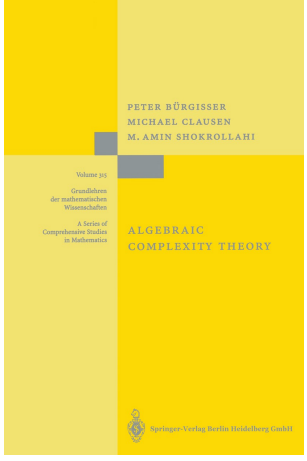
$O(M(n)\log(n))$

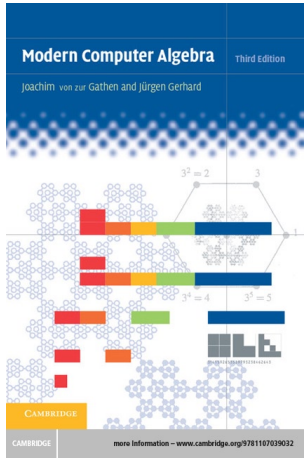
- eval./interp. $a \leftrightarrow a(x_1), \ldots, a(x_n)$
- extended GCD $au + bv = \gcd(a, b)$
- rational reconstruction $a = \frac{p}{q} \bmod M$

recent breakthrough: in $O(M(n)\log(n))$,

- series composition $p(a) \bmod x^n$
- series power projections $\ell(a^k \bmod x^n)$

[Kinoshita-Li 2024]
via [Bostan-Mori 2021]





fundamental computations on polynomials

most problems: quasi-linear via reductions to multiplication

$$M(n) \in O(n \log(n) \log\log(n)) \subset \tilde{O}(n)$$

$O(M(n))$

- addition $a + b$, multiplication $a * b$
- division with remainder $a = qb + r$
- power series inverse $a^{-1} \bmod x^n$

$O(M(n)\log(n))$

- eval./interp. $a \leftrightarrow a(x_1), \ldots, a(x_n)$
- extended GCD $au + bv = \gcd(a, b)$
- rational reconstruction $a = \frac{p}{q} \bmod M$

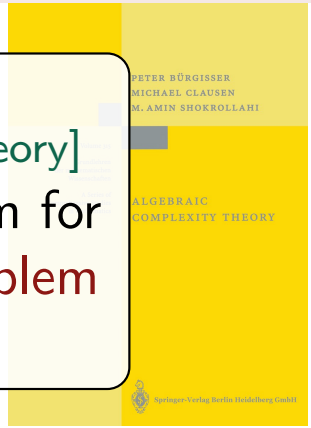
recent breakthrough: in $O(M(n)\log(n))$,

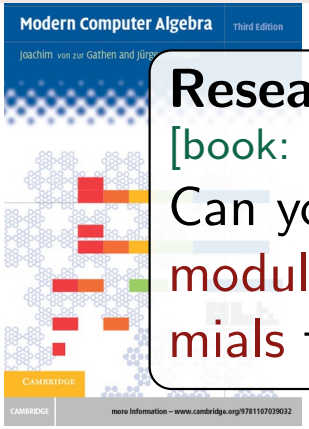
- series composition $p(a) \bmod x^n$
- series power projections $\ell(a^k \bmod x^n)$

[Kinoshita-Li 2024]
via [Bostan-Mori 2021]

most problems, except...

Open problem 2.4
[book: Algebraic Complexity Theory]
Can Brent&Kung's algorithm for the general composition problem be improved substantially?





Research problem 12.19
[book: Modern Computer Algebra]
Can you improve the cost for modular composition of polynomials to, say, $\tilde{O}(n^{1.5})$ or better?

target problems and complexity result

[in] polynomials $a, f \in \mathbb{K}[x]$ with $n = \deg(f) > \deg(a)$

modular composition

[in] polynomial $p \in \mathbb{K}[y]$

[out] $p(a) \bmod f$

$\xleftrightarrow{T}$

power projections

[in] linear form $\ell : \mathbb{K}[x]/\langle f \rangle \to \mathbb{K}$

[out] $[\ell(a^k \bmod f)]_{k=0,\ldots,n-1}$

minimal polynomial

[out] small-degree $p \in \mathbb{K}[y]$ s.t. $p(a) = 0 \bmod f$

target problems and complexity result

[in] polynomials $a, f \in \mathbb{K}[x]$ with $n = \deg(f) > \deg(a)$

modular composition

[in] polynomial $p \in \mathbb{K}[y]$

[out] $p(a) \bmod f$

$\longleftrightarrow$ T

power projections

[in] linear form $\ell : \mathbb{K}[x]/\langle f \rangle \to \mathbb{K}$

[out] $[\ell(a^k \bmod f)]_{k=0,\ldots,n-1}$

minimal polynomial

[out] small-degree $p \in \mathbb{K}[y]$ s.t. $p(a) = 0 \bmod f$

exponent	$\omega = 3$	$\omega = 2.81..$	$\omega = 2.37..$	$\omega = 2.0..$
$(\omega+1)/2$	2	1.9	1.69	1.5

previous work (deterministic)

- naive: $O(nM(n)) \subseteq \tilde{O}(n^2)$
- baby/giant steps: $O(n^{(\omega+1)/2})$

[Brent-Kung 1978] [Shoup 1994]

target problems and complexity result

[in] polynomials $a, f \in \mathbb{K}[x]$ with $n = \deg(f) > \deg(a)$

modular composition

[in] polynomial $p \in \mathbb{K}[y]$

[out] $p(a) \bmod f$

$\longleftrightarrow$ T

power projections

[in] linear form $\ell : \mathbb{K}[x]/\langle f \rangle \to \mathbb{K}$

[out] $[\ell(a^k \bmod f)]_{k=0,\ldots,n-1}$

minimal polynomial

[out] small-degree $p \in \mathbb{K}[y]$ s.t. $p(a) = 0 \bmod f$

exponent	$\omega = 3$	$\omega = 2.81..$	$\omega = 2.37..$	$\omega = 2.0..$
$(\omega+1)/2$	2	1.9	1.69	1.5
$(\omega+2)/3$	1.67	1.6	1.46	1.33

previous work (deterministic)

- naive: $O(nM(n)) \subseteq \tilde{O}(n^2)$
- baby/giant steps: $O(n^{(\omega+1)/2})$

[Brent-Kung 1978] [Shoup 1994]

previous work

- bit complexity: $O((n \log q)^{1+\varepsilon})$
- algebraic, Las Vegas: $\tilde{O}(n^{(\omega+2)/3})$

[Kedlaya-Umans '11] [N.-S.-S.-V.'24]

target problems and complexity result

[in] polynomials $a, f \in \mathbb{K}[x]$ with $n = \deg(f) > \deg(a)$

modular composition

[in] polynomial $p \in \mathbb{K}[y]$

[out] $p(a) \bmod f$

$\longleftrightarrow$ T

power projections

[in] linear form $\ell : \mathbb{K}[x]/\langle f \rangle \to \mathbb{K}$

[out] $[\ell(a^k \bmod f)]_{k=0,\ldots,n-1}$

minimal polynomial

[out] small-degree $p \in \mathbb{K}[y]$ s.t. $p(a) = 0 \bmod f$

result: complexity $\tilde{O}(n^{(\omega+3)/4})$, for generic $a(x)$

exponent	$\omega = 3$	$\omega = 2.81..$	$\omega = 2.37..$	$\omega = 2.0..$
$(\omega+1)/2$	2	1.9	1.69	1.5
$(\omega+2)/3$	1.67	1.6	1.46	1.33
$(\omega+3)/4$	1.5	1.45	1.34	1.25

previous work (deterministic)

- naive: $O(nM(n)) \subseteq \tilde{O}(n^2)$
- baby/giant steps: $O(n^{(\omega+1)/2})$

[Brent-Kung 1978] [Shoup 1994]

previous work

- bit complexity: $O((n \log q)^{1+\varepsilon})$
- algebraic, Las Vegas: $\tilde{O}(n^{(\omega+2)/3})$

[Kedlaya-Umans '11] [N.-S.-S.-V.'24]

outline

- **context and result**
  - univariate polynomial computations
  - modular composition and minpoly
  - improved algebraic complexity bound
- **motivations and algorithms**
  - complexity and software motivations
  - baby steps – giant steps
  - computing fewer powers $a^k \bmod f$
- **overview of the approach**

- breakthrough: bit cost $O((n \log(q))^{1+\varepsilon})$ for any $\varepsilon > 0$, over $\mathbb{F}_q$
[Kedlaya-Umans '08+'11] [Bhargava-Ghosh-Kumar-Mohapatra '23]

- improved algorithm and analysis: $n^{5/(\log_2 n)^{1/2}} \tilde{O}(n \log r)$ over $\mathbb{Z}/r\mathbb{Z}$
[van der Hoeven-Lecerf, J.Complexity'20 + preprint'26]
more precisely, $O(n \cdot n^{5/(\log_2 n)^{1/2}} \cdot (\log n)^{13.5} \cdot \log r \cdot (\log \log r)^3)$

almost linear bit complexity

complexity and software motivations

- breakthrough: bit cost $O((n \log(q))^{1+\varepsilon})$ for any $\varepsilon > 0$, over $\mathbb{F}_q$
[Kedlaya-Umans '08+'11] [Bhargava-Ghosh-Kumar-Mohapatra '23]

- improved algorithm and analysis: $n^{5/(\log_2 n)^{1/2}} \tilde{O}(n \log r)$ over $\mathbb{Z}/r\mathbb{Z}$
[van der Hoeven-Lecerf, J.Complexity'20 + preprint'26]
more precisely, $O(n \cdot n^{5/(\log_2 n)^{1/2}} \cdot (\log n)^{13.5} \cdot \log r \cdot (\log\log r)^3)$

almost linear bit complexity

complexity and software motivations

practical considerations

n	$n^{5/(\log_2 n)^{1/2}}$
10^4	$3.1 \cdot 10^5$
10^5	$1.4 \cdot 10^6$
10^6	$5.2 \cdot 10^6$
10^7	$1.8 \cdot 10^7$
10^8	$5.7 \cdot 10^7$
10^9	$1.7 \cdot 10^8$
10^{10}	$4.7 \cdot 10^8$
10^{11}	$1.3 \cdot 10^9$
10^{12}	$3.2 \cdot 10^9$
10^{13}	$7.8 \cdot 10^9$

$\longleftarrow = 32^{\sqrt{\log_2 n}}$

$n^{5/(\log_2 n)^{1/2}} \cdot (\log n)^{13.5} > n$ until $n \approx 10^{49}$

"*it remains a major challenge to make [Kedlaya and Umans'] algorithm efficient for practical input sizes*"

[van der Hoeven-Lecerf '20]

- breakthrough: bit cost $O((n \log(q))^{1+\varepsilon})$ for any $\varepsilon > 0$, over $\mathbb{F}_q$
[Kedlaya-Umans '08+'11] [Bhargava-Ghosh-Kumar-Mohapatra '23]

- improved algorithm and analysis: $n^{5/(\log_2 n)^{1/2}} \tilde{O}(n \log r)$ over $\mathbb{Z}/r\mathbb{Z}$
[van der Hoeven-Lecerf, J.Complexity'20 + preprint'26]
more precisely, $O(n \cdot n^{5/(\log_2 n)^{1/2}} \cdot (\log n)^{13.5} \cdot \log r \cdot (\log\log r)^3)$

almost linear bit complexity

complexity and software motivations

quasi-linear algebraic algorithm for series

- algorithm is valid over any field $\mathbb{K}$ (in fact, commutative ring)
- complexity bound is quasi-linear with small extra factors

n	[Brent-Kung 1978]		[Kinoshita-Li 2024]		speedup	
$\mathbb{K}$	$\mathbb{Z}/r\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}/r\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}/r\mathbb{Z}$	$\mathbb{Z}$
100	0.0000779	0.000298	0.000162	0.000443	0.481	0.673
200	0.000285	0.00243	0.000347	0.00284	0.821	0.856
400	0.000869	0.0153	0.000839	0.0144	1.036	1.062
800	0.00254	0.108	0.00193	0.0683	1.316	1.581
1600	0.00763	0.682	0.00434	0.339	1.758	2.012
3200	0.0238	4.651	0.00974	1.716	2.444	2.710
6400	0.0776	33.263	0.0218	8.462	3.560	3.931
12800	0.257	266.59	0.0487	43.331	5.277	6.152

baby steps – giant steps

[in] $f(x)$ of degree n, $a(x)$ of degree $< n$, $p(y)$ of degree $< n$

[out] $p(a(x)) \bmod f(x) = p_0\alpha_0(x) + p_1\alpha_1(x) + \cdots + p_{n-1}\alpha_{n-1}(x)$
where $\alpha_k = a^k \bmod f \in \mathbb{K}[x]_{<n}$

naive cost: $\tilde{O}(n^2)$

- dominant step: compute n powers of a mod f
- how to require fewer powers?

$$\sum_{k<n} p_k\alpha_k = \begin{bmatrix} p_0 & p_1 & \cdots & p_{n-1} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-1} \end{bmatrix}$$

baby steps – giant steps

[in] $f(x)$ of degree n, $a(x)$ of degree $< n$, $p(y)$ of degree $< n$

[out] $p(a(x)) \bmod f(x) = p_0\alpha_0(x) + p_1\alpha_1(x) + \cdots + p_{n-1}\alpha_{n-1}(x)$
where $\alpha_k = a^k \bmod f \in \mathbb{K}[x]_{<n}$

naive cost: $\tilde{O}(n^2)$

- dominant step: compute n powers of a mod f
- how to require fewer powers?

$$\sum_{k<n} p_k\alpha_k = \begin{bmatrix} p_0 & p_1 & \cdots & p_{n-1} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-1} \end{bmatrix}$$

split exponents as $k = ir + j$ with $r = \sqrt{n}$ and $i, j < r$
$\rightsquigarrow \alpha_k = a^{ir} \cdot a^j \bmod f = \alpha_{ir}\alpha_j \bmod f$

$$\sum_{i,j<r} p_{ir+j}\alpha_{ir}\alpha_j = \begin{bmatrix} \alpha_0 & \alpha_r & \cdots & \alpha_{(r-1)r} \end{bmatrix} \begin{bmatrix} p_0 & p_1 & \cdots & p_{r-1} \\ p_r & p_{r+1} & \cdots & p_{2r-1} \\ & & \vdots & \\ p_{n-r} & p_{n-r+1} & \cdots & p_{n-1} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{r-1} \end{bmatrix}$$

baby steps – giant steps

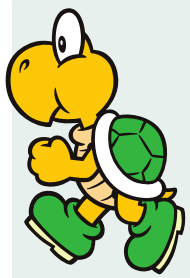
[in] $f(x)$ of degree n, $a(x)$ of degree $< n$, $p(y)$ of degree $< n$

[out] $p(a) \bmod f = \sum_{k<n} p_k \alpha_k$ where $\alpha_k = a^k \bmod f \in \mathbb{K}[x]_{<n}$

$2r$ powers of a mod f by splitting exponents
$k = ir + j$ with $r = \sqrt{n}$ and $i, j < r$

[Paterson-Stockmeyer 1971]
[Brent-Kung 1978]

$$p(a) \bmod f = \begin{bmatrix} \alpha_0 & \alpha_r & \cdots & \alpha_{(r-1)r} \end{bmatrix} \begin{bmatrix} p_0 & p_1 & \cdots & p_{r-1} \\ p_r & p_{r+1} & \cdots & p_{2r-1} \\ & & \vdots & \\ p_{n-r} & p_{n-r+1} & \cdots & p_{n-1} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{r-1} \end{bmatrix}$$



baby steps – giant steps

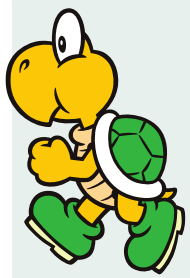
[in] $f(x)$ of degree n, $a(x)$ of degree $< n$, $p(y)$ of degree $< n$

[out] $p(a) \bmod f = \sum_{k<n} p_k \alpha_k$ where $\alpha_k = a^k \bmod f \in \mathbb{K}[x]_{<n}$

$2r$ powers of a mod f by splitting exponents
$k = ir + j$ with $r = \sqrt{n}$ and $i, j < r$

[Paterson-Stockmeyer 1971]
[Brent-Kung 1978]

$$p(a) \bmod f = \begin{bmatrix} \alpha_0 & \alpha_r & \cdots & \alpha_{(r-1)r} \end{bmatrix} \begin{bmatrix} p_0 & p_1 & \cdots & p_{r-1} \\ p_r & p_{r+1} & \cdots & p_{2r-1} \\ & & \vdots & \\ p_{n-r} & p_{n-r+1} & \cdots & p_{n-1} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{r-1} \end{bmatrix}$$



vector in $\mathbb{K}[x]_{<n}^{1\times\sqrt{n}}$ — square matrix in $\mathbb{K}^{\sqrt{n}\times\sqrt{n}}$ — vector in $\mathbb{K}[x]_{<n}^{\sqrt{n}\times 1}$

cost bound: $\tilde{O}(n^{3/2})$ for $2\sqrt{n}$ powers of a mod f
$+ O(n^{(\omega+1)/2})$ for $\mathbb{K}$-matrix multiplication $n \times \sqrt{n} \times n$

runtimes:
- much faster than naive approach
- $\tilde{O}(n^{3/2})$ regime lasts until largish n

baby steps – giant steps

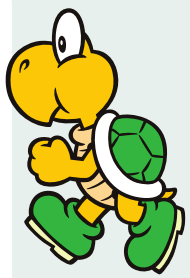
[in] $f(x)$ of degree n, $a(x)$ of degree $< n$, $p(y)$ of degree $< n$

[out] $p(a) \bmod f = \sum_{k<n} p_k \alpha_k$ where $\alpha_k = a^k \bmod f \in \mathbb{K}[x]_{<n}$

$2r$ powers of a mod f by splitting exponents
$k = ir + j$ with $r = \sqrt{n}$ and $i, j < r$

[Paterson-Stockmeyer 1971]
[Brent-Kung 1978]

$$p(a) \bmod f = \begin{bmatrix} \alpha_0 & \alpha_r & \cdots & \alpha_{(r-1)r} \end{bmatrix} \begin{bmatrix} p_0 & p_1 & \cdots & p_{r-1} \\ p_r & p_{r+1} & \cdots & p_{2r-1} \\ & & \vdots & \\ p_{n-r} & p_{n-r+1} & \cdots & p_{n-1} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{r-1} \end{bmatrix}$$



vector in $\mathbb{K}[x]_{<n}^{1\times\sqrt{n}}$ — square matrix in $\mathbb{K}^{\sqrt{n}\times\sqrt{n}}$ — vector in $\mathbb{K}[x]_{<n}^{\sqrt{n}\times 1}$

cost bound: $\tilde{O}(n^{3/2})$ for $2\sqrt{n}$ powers of a mod f
$+ O(n^{(\omega+1)/2})$ for $\mathbb{K}$-matrix multiplication $n \times \sqrt{n} \times n$

runtimes:
- much faster than naive approach
- $\tilde{O}(n^{3/2})$ regime lasts until largish n

how to require even fewer powers?

fewer powers, episode 1: bivariate input $p(y) \to P(x,y)$

[in] $f(x)$ and $a(x)$ as before,

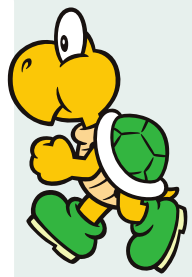
[in] $P(x,y) = \sum_{k<s^2} P_k(x) y^k$ of degrees $< (s, s^2)$ for $s = n^{1/3}$

[out] $P(x,a) \bmod f = \sum_{k<s^2} P_k(x) \alpha_k$ where $\alpha_k = a^k \bmod f$

$2s$ powers of a mod f by splitting exponents
$k = is + j$ with $s = n^{1/3}$ and $i, j < s$

[Paterson-Stockmeyer 1971]
[Brent-Kung 1978]
[Nüsken-Ziegler 2004]

$$P(x,a) \bmod f = \begin{bmatrix} \alpha_0 & \alpha_s & \cdots & \alpha_{(s-1)s} \end{bmatrix} \begin{bmatrix} P_0 & P_1 & \cdots & P_{s-1} \\ P_s & P_{s+1} & \cdots & P_{2s-1} \\ & & \vdots & \\ P_{s^2-s} & P_{s^2-s+1} & \cdots & P_{s^2-1} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{s-1} \end{bmatrix}$$



fewer powers, episode 1: bivariate input $p(y) \to P(x,y)$

[in] $f(x)$ and $a(x)$ as before,

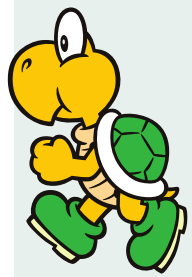
[in] $P(x,y) = \sum_{k<s^2} P_k(x)y^k$ of degrees $< (s, s^2)$ for $s = n^{1/3}$

[out] $P(x,a) \bmod f = \sum_{k<s^2} P_k(x)\alpha_k$ where $\alpha_k = a^k \bmod f$

$2s$ powers of a mod f by splitting exponents
$k = is + j$ with $s = n^{1/3}$ and $i, j < s$

[Paterson-Stockmeyer 1971]
[Brent-Kung 1978]
[Nüsken-Ziegler 2004]

$$P(x,a) \bmod f = \begin{bmatrix} \alpha_0 & \alpha_s & \cdots & \alpha_{(s-1)s} \end{bmatrix} \begin{bmatrix} P_0 & P_1 & \cdots & P_{s-1} \\ P_s & P_{s+1} & \cdots & P_{2s-1} \\ & & \vdots & \\ P_{s^2-s} & P_{s^2-s+1} & \cdots & P_{s^2-1} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{s-1} \end{bmatrix}$$



vector in $\mathbb{K}[x]^{1\times s}_{<n}$ — square matrix in $\mathbb{K}[x]^{s\times s}_{<s}$ — vector in $\mathbb{K}[x]^{s\times 1}_{<n}$

cost bound: $\tilde{O}(n^{4/3})$ for $2s$ powers of a mod f
$+ \tilde{O}(n^{(\omega+2)/3})$ for matrix multiplication in $\mathbb{K}[x]^{s\times s}_{<s^2}$

fewer powers, episode 1: bivariate input $p(y) \to P(x,y)$

[in] $f(x)$ and $a(x)$ as before,

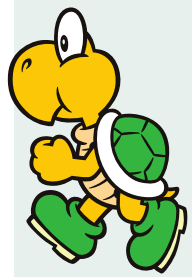
[in] $P(x,y) = \sum_{k<s^2} P_k(x)y^k$ of degrees $< (s, s^2)$ for $s = n^{1/3}$

[out] $P(x,a) \bmod f = \sum_{k<s^2} P_k(x)\alpha_k$ where $\alpha_k = a^k \bmod f$

$2s$ powers of a mod f by splitting exponents
$k = is + j$ with $s = n^{1/3}$ and $i, j < s$

[Paterson-Stockmeyer 1971]
[Brent-Kung 1978]
[Nüsken-Ziegler 2004]

$$P(x,a) \bmod f = \begin{bmatrix} \alpha_0 & \alpha_s & \cdots & \alpha_{(s-1)s} \end{bmatrix} \begin{bmatrix} P_0 & P_1 & \cdots & P_{s-1} \\ P_s & P_{s+1} & \cdots & P_{2s-1} \\ & & \vdots & \\ P_{s^2-s} & P_{s^2-s+1} & \cdots & P_{s^2-1} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{s-1} \end{bmatrix}$$



vector in $\mathbb{K}[x]^{1\times s}_{<n}$ — square matrix in $\mathbb{K}[x]^{s\times s}_{<s}$ — vector in $\mathbb{K}[x]^{s\times 1}_{<n}$

cost bound: $\tilde{O}(n^{4/3})$ for $2s$ powers of a mod f
$+ \tilde{O}(n^{(\omega+2)/3})$ for matrix multiplication in $\mathbb{K}[x]^{s\times s}_{<s^2}$

could we transform $p(y) \to P(x,y)$? s.t. $p(a) = P(x,a) \bmod f$

how to require even fewer powers?

fewer powers, new episode: bivariate powers $a(x)^k \to A_k(x,y)$

[in] $f(x)$ and $a(x)$ as before, back to $p(y)$ univariate of degree $< n$

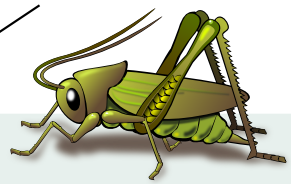
write $p(y) = \sum_{k<s^2} P_k(y) y^{ks}$ with $\deg(P_k) < s$, for $s = n^{1/3}$

[out] $p(a) \bmod f = \sum_{k<s^2} P_k(a)\alpha_{ks}$ where $\alpha_{ks} = a^{ks} \bmod f$

$2s$ powers of a mod f by splitting exponents

$ks = is^2 + js$ with $s = n^{1/3}$ and $i, j < s$

$$p(a) \bmod f = \begin{bmatrix} \alpha_0 & \alpha_{s^2} & \cdots & \alpha_{(s-1)s^2} \end{bmatrix} \begin{bmatrix} P_0 & P_1 & \cdots & P_{s-1} \\ P_s & P_{s+1} & \cdots & P_{2s-1} \\ & & \vdots & \\ P_{s^2-s} & P_{s^2-s+1} & \cdots & P_{s^2-1} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_s \\ \vdots \\ \alpha_{(s-1)s} \end{bmatrix}$$



vector in $\mathbb{K}[x]^{1\times s}_{<n}$ square matrix in $\mathbb{K}[y]^{s\times s}_{<s}$ vector in $\mathbb{K}[x]^{s\times 1}_{<n}$

fewer powers, new episode: bivariate powers $a(x)^k \to A_k(x,y)$

[in] $f(x)$ and $a(x)$ as before, back to $p(y)$ univariate of degree $< n$

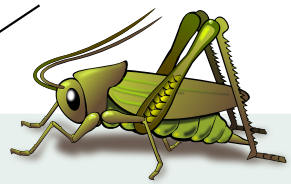
write $p(y) = \sum_{k<s^2} P_k(y) y^{ks}$ with $\deg(P_k) < s$, for $s = n^{1/3}$

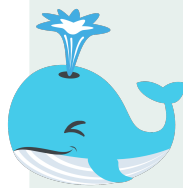
[out] $p(a) \bmod f = \sum_{k<s^2} P_k(a)\alpha_{ks}$ where $\alpha_{ks} = a^{ks} \bmod f$

$2s$ powers of a mod f by splitting exponents
$ks = is^2 + js$ with $s = n^{1/3}$ and $i, j < s$

$A_{ks}(x,y) \in \mathbb{K}[x,y]_{<(s^2,s)}$
$A_{ks}(x,a) = a^{ks} \bmod f$

$$p(a) \bmod f = \begin{bmatrix} A_0 & A_{s^2} & \cdots & A_{(s-1)s^2} \end{bmatrix} \begin{bmatrix} P_0 & P_1 & \cdots & P_{s-1} \\ P_s & P_{s+1} & \cdots & P_{2s-1} \\ & & \vdots & \\ P_{s^2-s} & P_{s^2-s+1} & \cdots & P_{s^2-1} \end{bmatrix} \begin{bmatrix} A_0 \\ A_s \\ \vdots \\ A_{(s-1)s} \end{bmatrix}$$





vector in $\mathbb{K}[x,y]^{1\times s}_{<(s^2,s)}$ square matrix in $\mathbb{K}[y]^{s\times s}_{<s}$ vector in $\mathbb{K}[x,y]^{s\times 1}_{<(s^2,s)}$

fewer powers, new episode: bivariate powers $a(x)^k \to A_k(x,y)$

[in] $f(x)$ and $a(x)$ as before, back to $p(y)$ univariate of degree $< n$

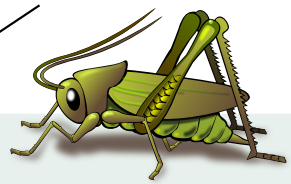
write $p(y) = \sum_{k<s^2} P_k(y) y^{ks}$ with $\deg(P_k) < s$, for $s = n^{1/3}$

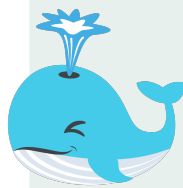
[out] $p(a) \bmod f = \sum_{k<s^2} P_k(a)\alpha_{ks}$ where $\alpha_{ks} = a^{ks} \bmod f$

$2s$ powers of a mod f by splitting exponents
$ks = is^2 + js$ with $s = n^{1/3}$ and $i, j < s$

$A_{ks}(x,y) \in \mathbb{K}[x,y]_{<(s^2,s)}$
$A_{ks}(x,a) = a^{ks} \bmod f$

$$p(a) \bmod f = \begin{bmatrix} A_0 & A_{s^2} & \cdots & A_{(s-1)s^2} \end{bmatrix} \begin{bmatrix} P_0 & P_1 & \cdots & P_{s-1} \\ P_s & P_{s+1} & \cdots & P_{2s-1} \\ & & \vdots & \\ P_{s^2-s} & P_{s^2-s+1} & \cdots & P_{s^2-1} \end{bmatrix} \begin{bmatrix} A_0 \\ A_s \\ \vdots \\ A_{(s-1)s} \end{bmatrix}$$





vector in $\mathbb{K}[x,y]^{1\times s}_{<(s^2,s)}$ — square matrix in $\mathbb{K}[y]^{s\times s}_{<s}$ — vector in $\mathbb{K}[x,y]^{s\times 1}_{<(s^2,s)}$

cost bound: $\tilde{O}(n^{4/3})$ for $2s$ powers of a mod f
$+ \tilde{O}(n^{(\omega+2)/3})$ for matrix multiplication in $\mathbb{K}[x,y]^{s\times s}_{<(s,s)}$

how to transform $a(x)^k \to A_k(x,y)$? s.t. $a^k = A_k(x,a) \bmod f$

how to require even fewer powers?

outline

context and result

- univariate polynomial computations
- modular composition and minpoly
- improved algebraic complexity bound

motivations and algorithms

- complexity and software motivations
- baby steps – giant steps
- computing fewer powers a^k mod f

overview of the approach

- reductions modulo a bivariate ideal
- exploiting small $\mathbb{K}[x]$- and $\mathbb{K}[y]$-relations
- conclusion and perspectives

balancing degrees: from univariate to bivariate

choose parameters $m \ll d$ with $md = \Theta(n)$, and **combine:**

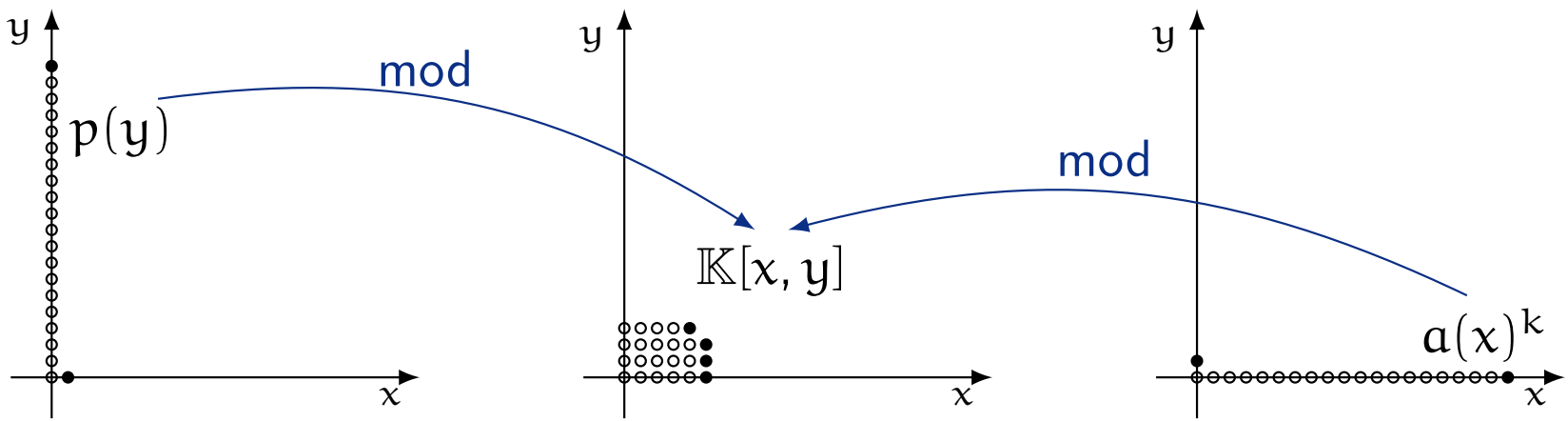
transform $p(y)$ into $P(x, y)$ such that $\begin{cases} \deg_{x,y}(P) < (m, d) \\ p(a) = P(x, a) \bmod f \end{cases}$

transform $\alpha_k(x)$ into $A_k(x, y)$ such that $\begin{cases} \deg_{x,y}(A_k) < (d, m) \\ a^k = A_k(x, a) \bmod f \end{cases}$

computations modulo $\mathcal{I} = \langle f(x), y - a(x) \rangle$

$p(y) = P(x, y) \bmod \mathcal{I}$

$y^k = a(x)^k = A_k(x, y) \bmod \mathcal{I}$



balancing degrees: from univariate to bivariate

choose parameters $m \ll d$ with $md = \Theta(n)$, and **combine:**

transform $p(y)$ into $P(x,y)$ such that $\begin{cases} \deg_{x,y}(P) < (m,d) \\ p(a) = P(x,a) \bmod f \end{cases}$

transform $\alpha_k(x)$ into $A_k(x,y)$ such that $\begin{cases} \deg_{x,y}(A_k) < (d,m) \\ a^k = A_k(x,a) \bmod f \end{cases}$

computations modulo $\mathcal{I} = \langle f(x), y - a(x) \rangle$

$p(y) = P(x,y) \bmod \mathcal{I}$

$y^k = a(x)^k = A_k(x,y) \bmod \mathcal{I}$

$\mathbb{K}[y]$-relation matrix: in $\mathbb{K}[y]^{m \times m}$
- "good" basis of $\mathcal{I} \cap \mathbb{K}[x,y]_{<(m,\cdot)}$
- generically small degrees $\sim n/m$
- fast computation: truncated powers and matrix Padé reconstruction

[Villard '18] [N.-S.-S.-V. '24]

$\mathbb{K}[x]$-relation matrix: in $\mathbb{K}[x]^{m \times m}$
- "good" basis of $\mathcal{I} \cap \mathbb{K}[x,y]_{<(\cdot,m)}$
- generically small degrees $\sim n/m$
- fast computation: shifted Popov basis of relations $\sum_{k<m} q_k a^k = 0 \bmod f$

[Storjohann '06] [Zhou-Labahn '12] [N. '16]

balancing degrees: from univariate to bivariate

choose parameters $m \ll d$ with $md = \Theta(n)$, and **combine:**

transform $p(y)$ into $P(x,y)$ such that $\begin{cases} \deg_{x,y}(P) < (m,d) \\ p(a) = P(x,a) \bmod f \end{cases}$

transform $\alpha_k(x)$ into $A_k(x,y)$ such that $\begin{cases} \deg_{x,y}(A_k) < (d,m) \\ a^k = A_k(x,a) \bmod f \end{cases}$

computations modulo $\mathcal{I} = \langle f(x), y - a(x) \rangle$

$p(y) = P(x,y) \bmod \mathcal{I}$

$y^k = a(x)^k = A_k(x,y) \bmod \mathcal{I}$

$\mathbb{K}[y]$**-relation matrix:** in $\mathbb{K}[y]^{m \times m}$
[Villard '18] [N.-S.-S.-V. '24]

$\mathbb{K}[x]$**-relation matrix:** in $\mathbb{K}[x]^{m \times m}$
[Storjohann '06] [Zhou-Labahn '12] [N. '16]

cost bound: $\tilde{O}(n^{(\omega+3)/4})$, taking $m = n^{1/4}$

$\tilde{O}(n^{5/4})$ for powers
$+ \tilde{O}(m^\omega \frac{n}{m})$ for polynomial matrix operations

conclusion and perspectives

faster algorithms & complexity results

- $P(x, y)$ in $\mathbb{K}[x, y]$ evaluated at points $(\alpha_i, \beta_i)_{1 \leqslant i \leqslant n}$
- $P(x, y)$ in $\mathbb{K}[x, y]$ composed with $y = a \bmod f$
- new techniques for computations modulo ideals $\langle f(x), y - a(x) \rangle$

baby steps–giant steps, bivariate matrices, univariate algebraic relations

- related: truncated powers, power projections, inverse composition, ...
- equivalences: bivariate composition – characteristic polynomial

perspectives & open questions

- careful implementations of the new algorithms

(the previous $(\omega + 2)/3$ algorithm already brings a speedup for large degrees)

- further consequences on related problems

(Guruswami-Sudan decoding, bivariate resultants, algebraic approximants, ...)

- design a Las Vegas randomization approach to avoid genericity
- very open: any idea towards quasi-linear, or maybe $(\omega + 4)/5$?

bi-level structures: an open problem

structured relations

$\rightsquigarrow$ P-recursive recurrences, algebraic/D-finite series, modular composition, bivariate interpolation, ...

$$p_1 f_1 + p_2 f_2 + \cdots + p_m f_m = 0 \bmod M$$

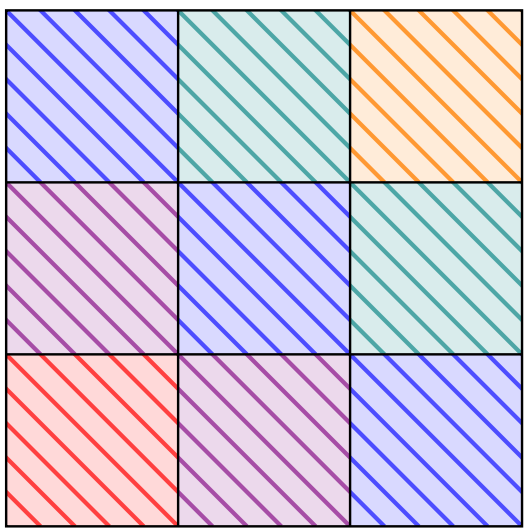
$\downarrow$ structured f_i's

$$p_1 1 + p_2 f + \cdots + p_m f^{m-1} = 0 \bmod M$$

$$p_1 f + p_2 f' + \cdots + p_m f^{(m-1)} = 0 \bmod M$$

- here, input/output size is $O(d)$
- most algorithms ignore this structure
- some recent progress [Villard 2018]

how to leverage this structure?



BTTB: block Toeplitz with Toeplitz blocks

$$\left[\begin{array}{cccc} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_8 \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_8^2 \\ \alpha_1^3 & \alpha_2^3 & \cdots & \alpha_8^3 \\ \alpha_1^4 & \alpha_2^4 & \cdots & \alpha_8^4 \\ \hdashline \beta_1 & \beta_2 & \cdots & \beta_8 \\ \alpha_1\beta_1 & \alpha_2\beta_2 & \cdots & \alpha_8\beta_8 \\ \alpha_1^2\beta_1 & \alpha_2^2\beta_2 & \cdots & \alpha_8^2\beta_8 \\ \hdashline \beta_1^2 & \beta_2^2 & \cdots & \beta_8^2 \end{array}\right]$$

bi-level Vandermonde

more complete timings of series composition

$\mathfrak{n}$	[Brent-Kung 1978]		[Kinoshita-Li 2024]		speedup	
$\mathfrak{n}$	$\mathbb{Z}/\mathrm{r}\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}/\mathrm{r}\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}/\mathrm{r}\mathbb{Z}$	$\mathbb{Z}$
100	0.0000779	0.000298	0.000162	0.000443	0.481	0.673
200	0.000285	0.00243	0.000347	0.00284	0.821	0.856
400	0.000869	0.0153	0.000839	0.0144	1.036	1.062
800	0.00254	0.108	0.00193	0.0683	1.316	1.581
1600	0.00763	0.682	0.00434	0.339	1.758	2.012
3200	0.0238	4.651	0.00974	1.716	2.444	2.710
6400	0.0776	33.263	0.0218	8.462	3.560	3.931
12800	0.257	266.59	0.0487	43.331	5.277	6.152
25600	0.882		0.115		7.670	
51200	3.036		0.266		11.414	
102400	10.82		0.646		16.749	
204800	37.76		1.556		24.267	
409600	140.573		3.736		37.627	
819200	497.694		10.274		48.442	

some timings of modular composition on NTL

complexity: $\tilde{O}(n^{3/2})$ for $O(\sqrt{n})$ multiplications by a and α modulo g
$+ O(n^{(\omega+1)/2})$ for matrix multiplication

in practice:
- much faster than naive approach
- $\tilde{O}(n^{3/2})$ regime lasts until largish n

```
// Horner evaluation h(a), modulo g :
zz_pX b;
b = coeff(h, n-1);
for (long k = n-2; k >= 0; --k)
{
    b = (a * b) % g;
    b = b + coeff(h, k);
}
```

n	Horner	Horner with precomputations	NTL built-in Brent-Kung
100	0.00229	0.00227	0.000441
200	0.0162	0.00691	0.00110
400	0.117	0.0278	0.00312
800	0.637	0.116	0.00944
1600	2.52	0.515	0.0281
3200	10.4	2.23	0.0884
6400	45.8	9.61	0.273

field $\mathbb{K} = \mathbb{F}_p$, prime p with 60 bits
NTL 11.4.3 on Intel Core i7-7600U @ 2.80GHz

software improvements

efficient implementation for the minimal polynomial
for large degrees, outperforms the state of the art

algorithm of [Neiger-Salvy-Schost-Villard '24]

field $\mathbb{K} = \mathbb{F}_p$, prime p with 60 bits
Intel Core i7-7600U @ 2.80GHz

random input polynomials $\Rightarrow$ "generic"

	general prime		FFT prime	
n	NTL	new	NTL	new
5k	0.349	0.496	0.130	0.208
20k	3.13	3.19	1.21	1.39
80k	31.5	23.6	13.9	10.7
320k	311	178	158	91.0

uses many types of computations on matrices over $\mathbb{K}[x]$
$\rightsquigarrow$ relies on the Polynomial Matrix Library

- multiplication for various parameters
- matrix-Padé approximation
- matrix division with remainder
- determinant
- system solving
- kernel

`https://github.com/vneiger/pml`