Faster Polynomial Factorization over Finite Fields via Batched Modular Composition

Zhenyu Xiong[1,2] Mingsheng Wang[1,2]

[1]State Key Laboratory of Cyberspace Security Defense,
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
xiongzhenyu@iie.ac.cn, wangmingsheng@iie.ac.cn

Abstract

In the algebraic model, where one counts operations in $\mathbb{F}_q$ uniformly in the characteristic, the best published algorithm for factoring a polynomial of degree n over $\mathbb{F}_q$, due to Kaltofen and Shoup with the rectangular products of Huang and Pan, uses $n^{1.80535+o(1)} \log q$ operations, and its bottleneck is the modular compositions of the giant steps. A new algorithm is presented for these compositions. Like the algorithm of Neiger, Salvy, Schost and Villard, the new method is based on the relation matrices of the argument. Since all compositions at one level of the giant steps share one argument, the relation matrices and a table of remainders are computed only once, and the remaining work consists of rectangular matrix products. For the two-matrix evaluation, we prove that a random change of basis puts every Frobenius image in good position. This reduces the cost of factorization in this model

$$\text{from} \quad n^{1.80535+o(1)} \log q \quad \text{to} \quad n^{1.5532+o(1)} \log q,$$

and to $n^{3/2+o(1)} \log q$ if $\omega = 2$; with the compositions of Neiger, Salvy, Schost and Villard performed one at a time, the exponent would be 1.7147. In the bit model, and algebraically in small characteristic, Kedlaya and Umans obtained the exponent 3/2. For a single composition with an arbitrary argument modulo a separable polynomial, the content of one short relation separates large value fibers, which gives a Las Vegas algorithm for every input at the exponent $(\omega + 3)/4 < 1.3429$, previously known only for generic inputs, and a fourth evaluation level, with truncated powers obtained by transposition, lowers it to 1.3309. As applications, we obtain faster algorithms for normal bases, minimal polynomials and irreducibility testing, and, for large q, factorization in $O(n^{1.5983}) + n^{1+o(1)} \log q$ operations.

Keywords. Modular composition, polynomial factorization, finite fields, fast matrix multiplication, algebraic complexity, polynomial matrices.

MSC codes. 68W30, 12Y05, 11Y16, 68Q25.

Contents

1 Introduction **3**
- 1.1 Polynomial factorization and modular composition . . . 3
- 1.2 Our results . . . 4
- 1.3 Main ideas . . . 6
- 1.4 Related work . . . 8

2 Preliminaries **9**
- 2.1 Notation and model of computation . . . 9
- 2.2 Relation modules, Popov forms and remainders . . . 11
- 2.3 Modular composition after Neiger, Salvy, Schost and Villard . . . 12
- 2.4 Two relation matrices . . . 13
- 2.5 The Kaltofen–Shoup framework . . . 14

I Batched composition and polynomial factorization **14**

3 Batched composition with a fixed argument **14**
- 3.1 Relation matrices as presentations . . . 14
- 3.2 Shared precomputation . . . 15
- 3.3 Remainder doubling . . . 16
- 3.4 Batched evaluation . . . 16
- 3.5 The one-matrix theorem . . . 17

4 Generating arguments **18**
- 4.1 Good position . . . 19
- 4.2 The two-matrix batch . . . 20
- 4.3 Large batches . . . 21

5 Polynomial factorization **22**
- 5.1 Extension of scalars . . . 22
- 5.2 Distinct-degree factorization with batched giant steps . . . 23
- 5.3 Equal-degree factorization . . . 25
- 5.4 Proof of Theorem 1.1 . . . 25
- 5.5 The choice of the parameter . . . 26
- 5.6 Large fields . . . 27

II Composition with an arbitrary argument **28**

6 Every argument **28**
- 6.1 A faster change of basis and single composition . . . 29
- 6.2 Small batches . . . 30
- 6.3 A fourth level . . . 33
- 6.4 Bounded fibers . . . 34
- 6.5 The content separator . . . 36
- 6.6 Long outer polynomials and the general theorem . . . 38

7 Experiments **39**

8 Open problems **40**

Appendices **41**

A The rectangular exponent programs **41**

B Proofs for batched composition **43**

C Proofs for fast certified change of basis **47**

D Proofs for every argument **50**

E Further results **56**

References **61**

1 Introduction

1.1 Polynomial factorization and modular composition

Let $\mathbb{F}_q$ be a finite field with q elements. We consider the problem of factoring a polynomial $f \in \mathbb{F}_q[x]$ of degree n into irreducible polynomials, and we count operations in $\mathbb{F}_q$, as in the algebraic model of Kaltofen and Shoup [16]. All algorithms that we discuss proceed in three stages:

- the *square-free factorization*, which reduces the problem to a square-free f;
- the *distinct-degree factorization*, which splits a square-free f into the products $f^{[e]}$ of its irreducible factors of degree e, $1 \le e \le n$;
- the *equal-degree factorization*, which splits each $f^{[e]}$ into its irreducible factors.

The first stage costs $n^{1+o(1)} \log q$ operations, and the third one reduces to modular compositions and powerings. The main difficulty therefore lies in the distinct-degree stage.

Berlekamp gave a probabilistic polynomial-time algorithm [2], and Cantor and Zassenhaus introduced the equal-degree method [9]. Later, von zur Gathen and Shoup showed how fast computations with the Frobenius map lead to an algorithm with $n^{2+o(1)}+n^{1+o(1)} \log q$ operations [37]. Kaltofen and Shoup then obtained the first subquadratic algorithm by a baby-step/giant-step strategy [16], which we now recall. Their starting point is the fact that, for $i \ge j \ge 0$,

$$x^{q^i} - x^{q^j} = \left(x^{q^{i-j}} - x\right)^{q^j} \tag{1.1}$$

is divisible precisely by the irreducible polynomials whose degree divides $i-j$. Let $l = \lceil n^\beta \rceil$ for a parameter $\beta \in [0,1]$, and $m_D = \lceil n/(2l) \rceil$. Their Algorithm D computes, in five steps labelled (D1)–(D5) in [16],

$$h_i = x^{q^i} \operatorname{rem} f, \qquad G_j = x^{q^{lj}} \operatorname{rem} f, \qquad I_j = \prod_{0 \le i < l} (G_j - h_i) \operatorname{rem} f, \tag{1.2}$$

for $0 \le i \le l$ and $1 \le j \le m_D$, followed by gcds. More precisely, (D1) computes the *baby steps* h_i and (D2) the *giant steps* G_j (denoted H_j in [16]), and (D3) forms the interval polynomials I_j. By Eq. (1.1), I_j is divisible by the irreducible factors of f whose degree divides some integer in $((j-1)l, jl]$. Step (D4) therefore computes the gcds $F_j = \gcd(f^*, I_j)$ for $j = 1, 2, \ldots$, replacing f^* by f^*/F_j each time, and F_j is the product of the irreducible factors of degree in $((j-1)l, jl]$. Finally, (D5) splits each F_j by the gcds with $G_j - h_i$.

The baby steps cost $n^{1+\beta+o(1)} \log q$ operations by repeated q-th powering, and the steps (D3)–(D5) cost $n^{1+\beta+o(1)}+n^{2-\beta+o(1)}$ by fast multipoint evaluation and gcds. However, the giant steps cannot be computed by powering at a comparable cost. Instead, since the coefficients of G_i lie in $\mathbb{F}_q$ and are fixed by $u \mapsto u^q$, one has

$$G_{s+i} = G_i(G_s) \operatorname{rem} f \qquad (i, s \ge 1). \tag{1.3}$$

The giant steps are therefore *modular compositions*. Kaltofen and Shoup compute them by doubling: once $G_1, \ldots, G_s$ are known, the s compositions $G_1(G_s), \ldots, G_s(G_s)$ give $G_{s+1}, \ldots, G_{2s}$. With Brent–Kung blocking, this costs $n^{\omega-(\omega-1)\beta/2+o(1)}$, where ω is a feasible exponent of square matrix multiplication. Balancing the three costs,

$$n^{1+\beta+o(1)} \log q + n^{2-\beta+o(1)} + n^{\omega-(\omega-1)\beta/2+o(1)}, \qquad \beta = \frac{2(\omega-1)}{\omega+1}, \tag{1.4}$$

gives their bound

$$n^{(3\omega-1)/(\omega+1)+o(1)} \log q. \tag{1.5}$$

Huang and Pan replaced the square products by rectangular ones and obtained the exponent 1.80535 [14, Thm. 10.6]. This is the best published exponent in the algebraic model for fields of arbitrary characteristic. In the bit model, Kedlaya and Umans obtained almost-linear modular composition, and hence factorization in $(n^{3/2+o(1)} + n^{1+o(1)} \log q) \log^{1+o(1)} q$ bit operations, independently of ω [17]; their algorithm for general characteristic uses the representation of the coefficients and is not algebraic, while for characteristic $p = n^{o(1)}$ they also give an algebraic algorithm with exponent 3/2. The present paper works in the algebraic model, uniformly in the characteristic; its exponents are not compared with these bit-model or small-characteristic bounds, except in Remark 5.10.

Let us now recall the modular composition problem. Given a field $\mathbb{K}$, a monic $f \in \mathbb{K}[x]$ of degree n and $a, g \in \mathbb{K}[x]_{<n}$, compute

$$g(a) \operatorname{rem} f. \tag{1.6}$$

For instance, over $\mathbb{F}_q$ the argument $a = x^q \operatorname{rem} f$ represents the Frobenius map, and the giant steps above are compositions with arguments of this kind. Brent and Kung gave the classical algebraic algorithm [6]. Recently, Neiger, Salvy, Schost and Villard improved the bound for every input [21], and then for generic inputs [22]. The three costs are, respectively,

$$O(n^{(\omega+1)/2}), \qquad \tilde{O}(n^{\kappa}), \quad \kappa = 1 + \frac{1}{1/(\omega-1) + 2/(\omega_2 - 2)}, \qquad \tilde{O}(n^{(\omega+3)/4}), \tag{1.7}$$

where ω_2 is a feasible exponent for the product of an $n \times n$ by an $n \times n^2$ matrix. With the known bounds, $\kappa < 1.4294$ and $(\omega+3)/4 < 1.3429$. The second algorithm is Las Vegas for every input. In contrast, the third one assumes that (f, a) is generic, and a Frobenius image is not a random argument. In small characteristic, algebraic algorithms with an additional dependence on the characteristic are also known [26, 4].

The giant steps have a feature that a single composition does not have: at each level of the doubling, *one* argument G_s serves s compositions. Van der Hoeven and Lecerf suggested sharing precomputation among such compositions in the bit model, without a bound [32]. This leads to the problem of *batched modular composition with a fixed argument*: given f, a and $g_1, \ldots, g_k$, compute

$$g_1(a) \operatorname{rem} f, \ \ldots, \ g_k(a) \operatorname{rem} f.$$

In this paper, we solve this problem faster than k independent compositions, and we use the solution inside Algorithm D.

1.2 Our results

We retain the algorithm of Kaltofen and Shoup. The square-free factorization, the baby steps, the formation and separation of intervals (steps (D1) and (D3)–(D5)) and the equal-degree stage are used as they are. We replace only the computation of the giant steps (step (D2)) by a faster algorithm for batched composition, and we prove the following theorem.

Theorem 1.1 (factorization)**.** *There is a Las Vegas algorithm which, given a finite field $\mathbb{F}_q$ and $f \in \mathbb{F}_q[x]$ of degree n, returns the factorization of f into irreducible polynomials using an expected number of*

$$n^{(\omega+7)/6+o(1)} \log q$$

operations in $\mathbb{F}_q$, for every feasible $\omega \leq 3$. With rectangular matrix multiplication the exponent is

$$e_2(\omega, \omega_2) = \frac{5\omega_2 - 4\omega + 2}{3(\omega_2 - \omega + 1)}$$

for feasible $\omega \leq 3$ and ω_2, which gives $O(n^{1.5552} \log q)$. With the full table of known rectangular bounds, the expected cost is $O(n^{1.5532} \log q)$. If $\omega = 2$ the cost is $n^{3/2+o(1)} \log q$.

Table 1: Exponents e of algebraic factorization algorithms for fields of arbitrary characteristic, whose cost is $n^{e+o(1)} \log q$ operations in $\mathbb{F}_q$; the last block lists results in other models. For the comparison, $\log q = n^{o(1)}$. Decimal values are approximate and use the full table of rectangular bounds (Section 2.1); the closed forms use only ω and ω_2. The last column is the value of e for $\omega = 2$.

method	exponent	value	$\omega = 2$
Berlekamp [2]	ω	2.3713	2
von zur Gathen–Shoup [37]	2	2	2
Kaltofen–Shoup [16]	$\frac{3\omega-1}{\omega+1}$	1.815[†]	5/3
Kaltofen–Shoup, Huang–Pan [14]	rectangular	1.80535	5/3
Kaltofen–Shoup, current rectangular bounds[*]	rectangular	1.7813	5/3
Kaltofen–Shoup with [21][*]	$1 + \kappa/2$	1.7147	5/3
Ours, single compositions (Theorem 6.21)	$1 + \frac{\omega+3}{8}$	1.6714	13/8
Ours, single compositions, four levels (Theorem 6.12)	$1 + \frac{\kappa_4}{2}$	1.6655	13/8
Ours, one relation matrix (Corollary 5.9)	$\frac{3\omega}{\omega+2}$; rect.	1.6274; 1.6214	3/2
Ours, two relation matrices (Theorem 1.1)	$\frac{\omega+7}{6}$; e_2; full	1.5619; 1.5552; **1.5532**	3/2
Other models, for comparison only			
Kedlaya–Umans [17], bit model, independent of ω	3/2	1.5	3/2
Kedlaya–Umans [17], algebraic, characteristic $n^{o(1)}$	3/2	1.5	3/2

[*]Our derivation from the cited algorithms. [†]With the historical bound $\omega = 2.375477$; the current bound $\omega = 2.371339$ gives 1.8135. The costs of the first two rows are $n^{e+o(1)} + n^{1+o(1)} \log q$.

With the known bounds for rectangular products, the exponent 1.5532 improves the published exponent 1.80535; see Table 1, where all decimal values use the same table of rectangular bounds. The classical scheme with these bounds gives 1.7813. Part of the gain comes from the faster compositions of [21]: performed one at a time inside Algorithm D, they already give the exponent 1.7147, and our single compositions give 1.6655. The batching itself lowers the exponent from 1.7147 to 1.5532, and from 5/3 to 3/2 if $\omega = 2$. The bound holds over every finite field. Algorithms tailored to small characteristic compete only when the characteristic p is very small compared with n; for instance, our bound is better as soon as $p \geq n^{1/12}$ and $\log q = n^{o(1)}$ (Remark 5.10).

Theorem 1.1 follows from a bound for batched composition. Let f be separable and let a generate the algebra $\mathcal{A} = \mathbb{K}[x]/\langle f \rangle$, as every Frobenius image does when f is square-free. Then we compute k compositions with the argument a in

$$\tilde{O}(n^{(\omega+2)/3} + n^{2/3}\mathsf{MM}(n^{1/3}, n^{1/3}, k)) \tag{1.8}$$

operations, where $\mathsf{MM}(u, v, w)$ is the cost of multiplying a $u \times v$ by a $v \times w$ matrix. The first term is a precomputation that depends only on (f, a), and the second one is the cost of the evaluation, which is linear in the coefficients of the g_i. For example, with square blocks and $k \geq n^{1/3}$, the bound is $\tilde{O}(kn^{(\omega+1)/3})$, that is, $n^{1.1238}$ per composition, and $n^{1+o(1)}$ per composition if $\omega = 2$. In Algorithm D, the doubling levels use batches of size up to $k \approx n^{1-\beta}$, and balancing Eq. (1.8) against the baby steps gives Theorem 1.1.

The generator hypothesis can be removed: for every argument modulo a separable polynomial, the same batch bound holds, and a single composition costs

$$\tilde{O}(n^{(\omega+3)/4}), \tag{1.9}$$

which was previously known only for generic inputs [22]. Factorization needs only the case of generators, but the general case answers, for separable f, the Las Vegas question left open there. Between one and $n^{1/3}$ compositions, a third evaluation level and a dimension that grows with k give $\tilde{O}(n^{(\omega+3)/4}k^{(\omega-1)/4})$, which joins the two bounds, for generating arguments (Theorem 6.10) and, with the separator, for every argument modulo a separable f (Theorem 6.23).

A fourth evaluation level lowers the exponent of one composition further. The obstacle to it is the change of basis, whose truncated powers would exceed the length allowed by [22]; we compute them instead by transposing a four-level evaluation. This gives, for every argument modulo a separable polynomial,

$$n^{\kappa_4+o(1)} = O(n^{1.3309}), \qquad \kappa_4 \le \frac{\omega+3}{4},$$

with equality $\kappa_4 = 5/4$ if $\omega = 2$ (Theorems 6.12 and 6.23); the gain comes from rectangular matrix products. For large batches with a generating argument, the reshaping method of [20], with relations taken from either relation module according to their width, gives $\tilde{O}(n^{(\omega+1)/2}+nk)$ (Theorem 4.9), hence $\tilde{O}(n^2)$ for $k = n$. Finally, the bounds $n^{\kappa_4+o(1)}$ and $n^{\Lambda_4(t)+o(1)}$ extend to moduli with repeated factors whose separable power decomposition is known, in particular to every modulus over a finite field, with an additional term $n^{1+o(1)}\log q$ (Theorem E.11).

When $\log q$ is large, the baby steps can also be batched: with $h_{s+i} = h_i(h_s)$, they are computed by the same doubling as the giant steps, and only h_1 needs the powering. This gives $n^{H+o(1)} + n^{1+o(1)}\log q$ with $H \le \frac{2}{3} + \frac{1}{3}\omega(1,1,\frac{3}{2}) < 1.5983$ (Corollary 5.11). For $\log q = n^{\rho+o(1)}$, the exponent $\max(H, 1+\rho)$ is smaller than that of Theorem 1.1 with β chosen again for the given q when $0.0983 < \rho < \omega_2/3$, and equal to it for larger ρ.

Since the evaluation is a linear map in the coefficients of the g_i, it can be transposed, and we construct a normal basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$ in

$$O(n^{1.6251}) + n^{1+o(1)}\log q \tag{1.10}$$

operations in $\mathbb{F}_q$, and we obtain similar improvements for minimal polynomials and irreducibility testing (Appendix E).

1.3 Main ideas

Our algorithms build on the relation modules, the reconstruction and the certificates of [20, 21], and on the auxiliary relations of [22]. Our contribution consists of three technical points: an evaluation that is linear in the outer polynomials, through a table of remainders; good position for prescribed generators; and a separation of large value fibers for arbitrary arguments.

Relation matrices as presentations For an argument a and $m \ge 1$, the polynomials $r(x,y)$ with $\deg_x r < m$ and $r(x,a) \equiv 0 \bmod f$ form a free $\mathbb{K}[y]$-module, and a basis is given by a matrix $P \in \mathbb{K}[y]^{m\times m}$, the *relation matrix*. When a generates $\mathcal{A}$, the quotient $\mathbb{K}[y]^m/\mathcal{L}(P)$ by the column module of P is isomorphic to $\mathcal{A}$, and

$$g(a) \operatorname{rem} f = \iota(g(Y)[e_1]), \tag{1.11}$$

where Y is the multiplication by y on the quotient and ι maps a vector $(v_0,\dots,v_{m-1})$ to $\sum_i v_i(a)x^i \operatorname{rem} f$. Brent–Kung blocking applies in this presentation. However, instead of storing the powers of a in $\mathcal{A}$, which have n coefficients each, we store the remainders of vectors modulo P: with $g = \sum_{j<J} g^{(j)} y^{jD}$ and $\deg g^{(j)} < D$,

$$P_j = (y^{jD}e_1) \operatorname{Rem} P, \qquad g\,e_1 \equiv \sum_{j<J} g^{(j)} P_j \pmod{\mathcal{L}(P)}. \tag{1.12}$$

The table $(P_j)_{j<J}$ depends only on (f,a), and we compute it once by repeated squaring of Y^D. For a batch, the reduction of all $g_i e_1$ then becomes one product of the table by the matrix of the coefficient blocks $g_i^{(j)}$, followed by one division by P. Both are rectangular matrix products with entries of degree $O(n/m)$, and both are linear in the coefficients of the g_i, whereas the reduction of [21] computes a kernel basis that depends on g. This linearity is what allows a batch, and also the transposed applications.

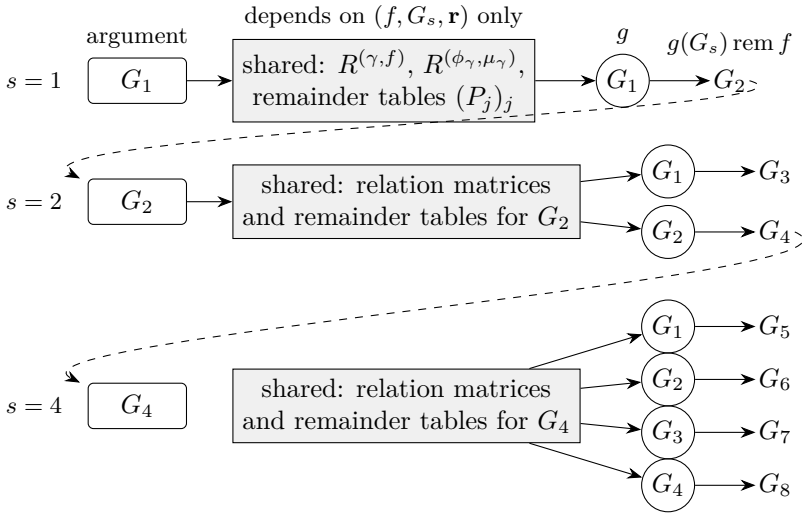

Figure 1: The giant steps $G_j = x^{q^{lj}} \operatorname{rem} f$ of [16, Algorithm D] by doubling, $G_{s+i} = G_i(G_s) \operatorname{rem} f$. At each level a single argument G_s is composed with the s polynomials $G_1, \ldots, G_s$. Everything that depends on the argument (relation matrices, the change of basis, the table of remainders $P_j = (y^{jd} e_1) \operatorname{Rem} P$) is computed once per level and shared by the s compositions; only the products with the coefficient blocks of the g's are done per polynomial, and they are done together as matrix products. The last output of a level is the argument of the next one (dashed).

Good position for prescribed generators The algorithm of [22] uses two relation matrices, one over $\mathbb{K}[y]$ and one over $\mathbb{K}[x]$, and requires both to have the expected degrees. For a generic argument this holds, but our arguments are prescribed. We first apply a random change of basis γ, as in [21], which reduces $g(a) \operatorname{rem} f$ to two compositions, with arguments γ modulo f and ϕ_γ modulo the minimal polynomial μ_γ of γ. When f is separable, the degree conditions become rank conditions on the matrices

$$[\zeta_\nu^i \lambda_\nu^j]_{\nu;\,(i,j)}, \tag{1.13}$$

where ζ_ν and λ_ν are, respectively, the coordinates of the roots and the values of the argument at them. A Vandermonde specialization shows that the relevant minors are nonzero polynomials in the coefficients of γ, for every generator a, in particular for every Frobenius image (Proposition 4.2).

Batching the giant steps At the level s of the doubling Eq. (1.3), all compositions share the argument G_s, which generates $\mathcal{A}$ (Lemma 5.2). The change of basis, both relation matrices and both remainder tables are therefore computed once per level, and only the evaluation is done for each of the s polynomials, by rectangular products; see Fig. 1. Since the levels double, the total cost of the giant steps is, up to a logarithmic factor, that of the last level, a batch of size at most $n^{1-\beta}$.

Separation of large value fibers An arbitrary argument may take the same value at many roots of f, and then the relation matrices cannot have the expected degrees. Small fibers are handled by certified submodules of relations instead of bases of the full module. For large fibers, we take one short relation and its content,

$$r(x, y) = \sum_i x^i c_i(y), \qquad P = \gcd_i c_i(y). \tag{1.14}$$

Table 2: Composition with one fixed argument: exponents of n for $k = n^t$ inputs, with square blocks. In the last column, "yes" means a randomized guarantee for every argument in the stated class, "no" means a hypothesis of random or generic input, and "sep." means that f is separable.

method	precomputation	total for k	argument; guarantee
Brent–Kung, batched [16]	—	$\frac{\omega+1}{2} + \frac{\omega-1}{2}t$	any; —
k calls to [21]	κ per input	$t + \kappa$	any, sep.; —
Neiger–Rosenkilde–Solomatov [20]	ω	$\max(\omega, 1+t)$	random, square-free; no
k calls to [22]	$\frac{\omega+3}{4}$ per input	$t + \frac{\omega+3}{4}$	generic (a, f); no
Ours, one relation matrix	$\frac{\omega+2}{3} + \frac{\omega-1}{3}t$	$\max(\frac{\omega+2}{3} + \frac{\omega-1}{3}t, t + \frac{\omega}{2})$	any, sep.; —
Ours, two relation matrices	$\frac{\omega+2}{3}$	$\max(\frac{\omega+2}{3}, t + \frac{\omega+1}{3})$	any, sep.; yes
Ours, three levels, $t \le \frac{1}{3}$	$\frac{\omega+3}{4} + \frac{\omega-1}{4}t$	$\frac{\omega+3}{4} + \frac{\omega-1}{4}t$	any, sep.; yes
Ours, reshaping	$\frac{\omega+1}{2}$	$\max(\frac{\omega+1}{2}, 1+t)$	generator, sep.; yes

The Brent–Kung entry holds for $0 \le t \le 1$. The entry for [20] takes outer degree $< n$. The last four rows are Theorem 3.6, Theorems 4.6 and 6.21, Theorems 6.10 and 6.23 and Theorem 4.9, respectively.

If a value c is taken at more than $\deg_x r$ roots, then $r(x, c) = 0$, so that $P(c) = 0$. Hence $\gcd(f, P(a))$ separates a factor of f on which a takes few values from a cofactor with small fibers, deterministically (Lemma 6.18).

The complete algorithm Combining these ingredients, the algorithm of Theorem 1.1 runs as follows on input f of degree n.

(1) Compute the square-free decomposition of f. Each square-free part is treated separately.

(2) Compute the baby steps $h_i = x^{q^i} \operatorname{rem} f$, $0 \le i \le l$, with $l = \lceil n^\beta \rceil$.

(3) Compute the giant steps $G_1 = h_l, \dots, G_{m_D}$ by doubling. At each level, with argument G_s:

- precompute the change of basis, the two relation matrices and the two remainder tables for G_s;
- evaluate $G_1(G_s), \dots, G_s(G_s)$ modulo f together, by rectangular matrix products.

(4) Form the interval polynomials I_j and split them by gcds, as in Algorithm D.

(5) Split each equal-degree part by the method of von zur Gathen and Shoup, with the single compositions of [21].

With $\beta = 0.5531$, the top level of step (3) has batch size about $n^{0.447}$. Steps (2) and (3) are then balanced at $n^{1.5532}$, up to the factor $\log q$ of the baby steps, while steps (4) and (5) are cheaper.

1.4 Related work

Neiger, Rosenkilde and Solomatov obtain quasi-linear composition after a precomputation on (f, a) [20]. However, their guarantee for the balancedness of the evaluation is proved for a random argument, and so it does not cover prescribed Frobenius arguments. Table 2 compares the costs of k compositions with one argument. Amortized multipoint evaluation [30, 29, 31, 33] fixes the evaluation points; applying it at the roots of f would require the factorization that we seek.

Other composition algorithms treat the bit model [17, 28, 34], the modulus $f = x^n$ [18], or decomposable moduli [24]. Multivariate multipoint evaluation is algebraic in small characteristic

Table 3: Notation.

symbol	definition	symbol	definition
$\mathcal{A}$	$\mathbb{K}[x]/\langle f\rangle$, $\deg f = n$, $a \in \mathbb{K}[x]_{<n}$	$\mathsf{MM}(u,v,w)$	cost of $(u\times v)\cdot(v\times w)$ over $\mathbb{K}$
$k,\ t$	$k = n^t$ outer polynomials $g_1,\dots,g_k$	$\omega,\ \omega_2$	$\omega(1,1,1)$, $\omega(1,1,2)$
$m,\ d$	$d = \lceil n/m\rceil$	α	$\sup\{u : \omega(1,1,u) = 2\}$
$\bar m,\ \bar d$	$\bar d = \lceil n/\bar m\rceil$	$U(u,v,w)$	bound Eq. (2.2) on $\omega(u,v,w)$
$\mathcal{M}_m^{(a,f)}$	$\{r : \deg_x r < m,\ r(x,a)\equiv 0 \bmod f\}$	κ	$1+\left(\frac{1}{\omega-1}+\frac{2}{\omega_2-2}\right)^{-1}$
$\mathcal{N}_{\bar m}^{(a,f)}$	$\{r : \deg_y r < \bar m,\ r(x,a)\equiv 0 \bmod f\}$	$c(n,m,d)$	$(m+n/d)\,d^{\omega_2/2}$
$\mu_\gamma,\ \phi_\gamma$	$\mu_\gamma(\gamma)\equiv 0$, $\phi_\gamma(\gamma)\equiv a \bmod f$	$C(n,k)$	cost of k compositions, Definition 2.1
$\mathbf{r},\ S$	$\mathbf{r}\in S^{n+\max(m,2)}$, $S\subseteq\mathbb{K}$ finite	$\Lambda(t)$	$C(n,\lceil n^t\rceil)\le n^{\Lambda(t)+o(1)}$
$v \operatorname{Rem} P$	$\tilde v\equiv v \bmod \mathcal{L}(P)$, $\deg\tilde v_i < r_i$	$l,\ m_D$	$l=\lceil n^\beta\rceil$, $m_D=\lceil n/(2l)\rceil$
$P_j,\ \Pi^{(\nu)}$	$(y^{jD}e_1)\operatorname{Rem} P$, $(y^{2^\nu D}I_N)\operatorname{Rem} P$	$h_i,\ G_j$	$x^{q^i}\operatorname{rem} f$, $x^{q^{lj}}\operatorname{rem} f$
$B,\ s$	$s=\lceil 2d/B\rceil$ (Nüsken–Ziegler)		

[26, 4] and has a non-algebraic solution over every finite field [3]. For factorization, Guo, Narayanan and Umans relate the exponent 3/2 to other algebraic problems [13], and Umans and Wang give a conditional exponent 4/3 with a different arrangement of baby and giant steps [27]. Berlekamp's algorithm computes the fixed space $\ker(Q-I)$ of the Frobenius matrix Q and then splits by gcds; its black-box variants [15] also rely on the action of the Frobenius map, and our batched compositions may also apply there.

2 Preliminaries

2.1 Notation and model of computation

Polynomials and matrices Let $\mathbb{K}$ be a field and $\mathbb{F}_q$ have $q = p^e$ elements, p prime. Write $R[x]_{<n}$ for degree-$<n$ polynomials over a ring R, and $\mathbb{K}[x,y]_{<(m,d)}$ for bidegree $<(m,d)$. As in [21, §2], identify $\sum_{i<m} v_i(y)x^i$ with the column $(v_0,\dots,v_{m-1})^{\mathsf T}\in\mathbb{K}[y]^m$. In particular, $g\in\mathbb{K}[y]$ corresponds to $g\,e_1$, where $e_1=(1,0,\dots,0)^{\mathsf T}$. For monic f, write $g\operatorname{rem} f$ for the Euclidean remainder and $\mathcal{A}=\mathbb{K}[x]/\langle f\rangle$. We take the polynomial multiplication time

$$\mathsf{M}(n) = O(n\log n\log\log n),$$

which holds over any commutative algebra [8]. The remaining notation is collected in Table 3.

Model of computation We use uniform computation trees over $\mathbb{K}$, in which the operations $+,-,\times,\div$ and zero tests cost one unit and the characteristic is known. Over $\mathbb{F}_q$, this is the model of [37, 16]. An algorithm in this model works uniformly in the characteristic and over every field; it cannot access the representation of field elements, so that it cannot, for instance, lift coefficients to the integers, reduce them modulo other primes and reconstruct them by Chinese remaindering, as the bit-model algorithm of [17] does, and the inverse Frobenius map is not a unit-cost operation. A bound in this model gives a bit-complexity bound by implementing each operation, but not conversely. Random elements are uniform and independent in a finite subset $S\subseteq\mathbb{K}$. If $\varphi\in\mathbb{F}_q[z]$ is irreducible of degree e, then one operation in $\mathbb{F}_{q^e}=\mathbb{F}_q[z]/\langle\varphi\rangle$ costs $\tilde O(e)$ operations in $\mathbb{F}_q$, and for $e=O(\log n)$ such a φ is found by a Las Vegas algorithm with $(\log n)^{O(1)}\log q$ operations in $\mathbb{F}_q$ [36, §14.9].

An algorithm is *Las Vegas with failure probability* ε if, for every input and every value of its random elements, it returns either a correct output or Fail, the latter with probability at most ε; for $\varepsilon\le 1/2$, repetition until success gives an always correct output at twice the expected cost. For algorithms whose cost is itself random, such as equal-degree factorization, "Las Vegas" means that the output is always correct and the cost bound holds in expectation.

Definition 2.1 (batched modular composition with a fixed argument)**.** Given $f \in \mathbb{K}[x]$ monic and separable of degree n, $a \in \mathbb{K}[x]_{<n}$ and $g_1, \ldots, g_k \in \mathbb{K}[y]$ of degree $O(n)$, the problem Batched Composition asks for $g_1(a) \operatorname{rem} f, \ldots, g_k(a) \operatorname{rem} f$. We write $C(n,k)$ for the worst-case number of operations in $\mathbb{K}$ of a Las Vegas algorithm with failure probability at most $1/2$ whose failure event depends only on f, a and the random elements; in particular, the algorithm is correct for all g_i, even if they are chosen after the random elements are revealed. For $t \geq 0$ we call *batch curve* any function Λ with $C(n, \lceil n^t \rceil) \leq n^{\Lambda(t)+o(1)}$.

Soft-O We write $c' = \tilde{O}(c)$ if $c' = O(c \log^j c)$ for some constant j. A bound $n^{e+o(1)}$ means $O(n^{e+\varepsilon})$ for every $\varepsilon > 0$. The quantity $\log q$ is never hidden in these notations.

Matrix multiplication $\mathsf{MM}(u,v,w)$ denotes the minimal number of operations in $\mathbb{K}$ of a bilinear algorithm that multiplies a $u \times v$ by a $v \times w$ matrix, augmented by $uv + vw + uw$ to account for the sizes of inputs and outputs; permuting (u,v,w) changes it by a constant factor at most. For reals $u, v, w \geq 0$, $\omega(u,v,w)$ is the infimum of the τ such that $\mathsf{MM}(n^u, n^v, n^w) = O(n^\tau)$, with $\omega = \omega(1,1,1)$, $\omega_2 = \omega(1,1,2)$, and $\alpha = \sup\{u : \omega(1,1,u) = 2\}$. In symbolic statements ω (resp. ω_2) denotes any *feasible* exponent, i.e., one for which an algorithm with cost $O(n^\omega)$ (resp. $O(n^{\omega_2})$) exists, with $2 \leq \omega \leq 3$; numerical exponents derived from infima are stated as $n^{e+o(1)}$. The following properties suffice [7, Ch. 15], [19]:

(i) $\omega(u,v,w)$ is invariant under permutations of (u,v,w).

(ii) It is positively homogeneous: $\omega(cu, cv, cw) = c\,\omega(u,v,w)$ for $c \geq 0$.

(iii) It is subadditive: $\omega(u+u', v+v', w+w') \leq \omega(u,v,w) + \omega(u',v',w')$ (tensor product of algorithms).

(iv) It is nondecreasing in each argument (padding with zeros).

(v) $\omega(1,0,0) = 1$ and $\omega(1,1,0) = 2$.

Properties (ii) and (iii) imply that $\omega(\cdot,\cdot,\cdot)$ is convex on $\mathbb{R}^3_{\geq 0}$. In particular, for $1 \leq s \leq 2$, writing $(1,1,s) = (2-s)(1,1,1) + (s-1)(1,1,2)$ gives

$$\omega(1,1,s) \leq (2-s)\,\omega + (s-1)\,\omega_2, \tag{2.1}$$

and $\omega_2 \leq \omega + 1$, since $(1,1,2) = (1,1,1) + (0,0,1)$. More generally, if (g_j, w_j) is a finite list of points with $\omega(g_j) \leq w_j$, then for every $v \in \mathbb{R}^3_{\geq 0}$

$$\omega(v) \leq U(v) := \min\Big\{ \sum_j \lambda_j w_j : \ \lambda \geq 0, \ \sum_j \lambda_j g_j \geq v \Big\}. \tag{2.2}$$

The numerical values used in this paper are $\omega < 2.371339$ and $\omega_2 < 3.250035$ [1, Table 1] and $\alpha > 0.321334$ [35], which we call the *basic bounds*, together with the other entries $\omega(1,1,k)$, $k \in \{0.33, 0.34, 0.35, 0.4, 0.5, 0.5275, 0.6, 0.7, 0.8, 0.9, 1.5\}$, of [1, Table 1]; with them we speak of the *full table*. Closed forms such as $(\omega+7)/6$ and $e_2(\omega,\omega_2)$ below use only ω and ω_2; unless stated otherwise, decimal values are computed with the full table. Exponents in complexity bounds of the form $O(n^e)$ are rounded up; other decimals, such as values of programs and optimal parameters, are approximations. A bilinear algorithm over $\mathbb{K}$ works over every commutative $\mathbb{K}$-algebra [7, Ch. 15], so that the product of a $u \times v$ by a $v \times w$ matrix with entries in $\mathbb{K}[x]_{<D}$ costs $O(\mathsf{MM}(u,v,w)\mathsf{M}(D)) = \tilde{O}(D\,\mathsf{MM}(u,v,w))$, and by Kronecker substitution the same holds for entries in $\mathbb{K}[x,y]_{<(D_1,D_2)}$ with $D = D_1 D_2$; with one operand fixed, such a product is linear in the other. The phrase "if $\omega = 2$" allows a factor $n^{o(1)}$, by taking every feasible exponent $2+\varepsilon$. Finally, since the output of k compositions has kn coefficients, every batch curve satisfies $\Lambda(t) \geq 1 + t$.

2.2 Relation modules, Popov forms and remainders

For $m \geq 1$, the *module of relations*

$$\mathcal{M}_m^{(a,f)} = \{r \in \mathbb{K}[x,y]_{<(m,\cdot)} : \ r(x, a(x)) \equiv 0 \bmod f\}$$

is a free $\mathbb{K}[y]$-submodule of $\mathbb{K}[y]^m$ of rank m. A *matrix of relations* is a nonsingular $R \in \mathbb{K}[y]^{m\times m}$ whose columns lie in $\mathcal{M}_m^{(a,f)}$, and we write $\mathcal{L}(R)$ for the $\mathbb{K}[y]$-module spanned by its columns. We follow the conventions of [21, §§4.1, 5.2, 6.1]. More precisely, for a nonsingular $P \in \mathbb{K}[y]^{N\times N}$, the row degrees are $r_i = \max_j \deg P_{ij}$, and P is *row reduced* if its row leading matrix is invertible. Moreover, P is in *Popov form* if it is in column weak Popov form with diagonal pivots and its row leading matrix is the identity. Every rank-N submodule of $\mathbb{K}[y]^N$ has a unique basis in Popov form. If P is row reduced with row degrees (r_i), then for every $v \in \mathbb{K}[y]^N$ there is a unique $\tilde{v}$ with $v - \tilde{v} \in \mathcal{L}(P)$ and $\deg \tilde{v}_i < r_i$ for all i. We write $\tilde{v} = v \operatorname{Rem} P$ and extend this notation columnwise to matrices. In the row convention of [23], $v \operatorname{Rem} P$ is the transpose of the remainder of v^{T} modulo the column-reduced matrix P^{T} [23, Thm. 2.1].

Lemma 2.2 (division with remainder)**.** *Let $P \in \mathbb{K}[y]^{N\times N}$ be row reduced, with row degrees $(r_1, \ldots, r_N)$ and entries of degree at most Δ_0, and let $\delta \geq 1$.*

(a) After a precomputation that depends on (P, δ) only and uses $\tilde{O}(N^\omega(\Delta_0 + \delta))$ operations, for every $V \in \mathbb{K}[y]^{N\times k}$ with $\deg V_{ij} < r_i + \delta$, the remainder $V \operatorname{Rem} P$ can be computed with $\tilde{O}((\Delta_0 + \delta)\,\mathsf{MM}(N, N, k))$ operations.

(b) The computation of $V \operatorname{Rem} P$ in (a) is a linear straight-line program in the coefficients of V, whose constants are computed in the precomputation.

(c) Without precomputation, $V \operatorname{Rem} P$ costs $\tilde{O}((k + N)N^{\omega-1}(\Delta_0 + \delta))$ operations.

Proof. Transpose to the row convention: $M = P^{\mathsf{T}}$ is column reduced with column degrees (r_i), and $F = V^{\mathsf{T}}$ satisfies $\operatorname{cdeg} F < \operatorname{cdeg} M + (\delta, \ldots, \delta)$. By [23, Thm. 2.1, Lemma 3.1] there are unique Q, R with $F = QM + R$ and $\operatorname{cdeg} R < \operatorname{cdeg} M$, and moreover $\deg Q < \delta$ and $R^{\mathsf{T}} = V \operatorname{Rem} P$. Let $M_{\mathrm{rev}} = M(y^{-1}) \operatorname{diag}(y^{r_i})$, $F_{\mathrm{rev}} = F(y^{-1}) \operatorname{diag}(y^{r_i+\delta-1})$ and $Q_{\mathrm{rev}} = y^{\delta-1}Q(y^{-1})$. The constant coefficient $M_{\mathrm{rev}}(0)$ is the column leading matrix of M, which is invertible, and reversing $F = QM + R$ shows that the reversed remainder contributes only terms of order at least δ; hence

$$Q_{\mathrm{rev}} = F_{\mathrm{rev}} M_{\mathrm{rev}}^{-1} \bmod y^\delta,$$

for every number k of rows. This is the identity used by PM-QuoRem [23, Algorithm 1], which returns $(Q, F - QM)$.

(a) Compute $X = M_{\mathrm{rev}}^{-1} \bmod y^\delta$ by Newton iteration, $X \leftarrow X + X(I - M_{\mathrm{rev}}X) \bmod y^{2^i}$, starting from $X = M_{\mathrm{rev}}(0)^{-1}$; this uses $O(\log \delta)$ products of $N \times N$ matrices of degree less than 2δ, hence $\tilde{O}(N^\omega \delta)$ operations. Then $Q_{\mathrm{rev}} = F_{\mathrm{rev}}X \bmod y^\delta$ is a product of a $k \times N$ by an $N \times N$ matrix of polynomials of degree less than δ, and QM is a product of a $k \times N$ matrix of degree less than δ by an $N \times N$ matrix of degree at most Δ_0; both cost $\tilde{O}((\Delta_0 + \delta)\mathsf{MM}(k, N, N))$, and $\mathsf{MM}(k, N, N) = O(\mathsf{MM}(N, N, k))$.

(b) Reversal and truncation are index manipulations, and both products multiply the variable matrix by fixed matrices; a bilinear algorithm with one operand fixed is linear in the other once its linear forms are precomputed.

(c) Combine (a) with square blocking: $N^\omega(\Delta_0 + \delta) + (\Delta_0 + \delta)\mathsf{MM}(N, N, k)$ is $O((N + k)N^{\omega-1}(\Delta_0 + \delta))$, since $\mathsf{MM}(N, N, k) = O(\lceil k/N \rceil N^\omega)$. □

Lemma 2.2 makes no assumption on $\deg \det P$, which may be as small as the dimension for the matrix of relations of an arbitrary argument.[1]

[1] The sharper bound $\tilde{O}(\lceil k/N \rceil N^{\omega-1} \deg \det P)$ of [23, Prop. 3.4] assumes $k\delta = O(\deg \det P)$ and $N = O(\deg \det P)$; see also [20, Prop. 7.4] for a single vector. Since all our matrices have degree $O(d)$, the uniform bound suffices.

Table 4: The stages of [21, Algorithm 8.1] on input $(f, a, g, \mathbf{r})$, with $n \geq 2$ and $d = \lceil n/m \rceil$. The change of basis and the second relation matrix are Algorithms 6.1 and 5.2 there, and the compositions use Algorithm 4.1 there. The first three stages depend only on $(f, a, \mathbf{r})$, and only the last three depend on g.

stage	computation	returns Fail if
normalization	$\hat{a} = (a + r_1)(x + r_2), \quad \hat{f} = f(x + r_2)$	$\gcd(\hat{a}, \hat{f}) \neq 1$ or $\hat{f}(0) = 0$
change of basis	$(R^{(\gamma,\hat{f})}, \mu_\gamma, \phi_\gamma) \leftarrow \textsc{ChangeOfBasis}(\hat{f}, \gamma, \hat{a}, m, d)$	it fails, or $\mu_\gamma(0) = 0$
second relation matrix	$R^{(\phi_\gamma,\mu_\gamma)} \leftarrow \textsc{MatrixOfRelations}(\mu_\gamma, \phi_\gamma, m, d, \mathbf{r}')$	it fails
first composition	$b = \hat{g}(\phi_\gamma) \operatorname{rem} \mu_\gamma$, $\hat{g} = g(y - r_1)$, with $R^{(\phi_\gamma,\mu_\gamma)}$	never
second composition	$b' = b(\gamma) \operatorname{rem} \hat{f}$ with $R^{(\gamma,\hat{f})}$	never
final shift	$g(a) \operatorname{rem} f = b'(x - r_2)$	never

Lemma 2.3 (Popov form)**.** *Let $R' \in \mathbb{K}[y]^{N\times N}$ be nonsingular with entries of degree at most Δ. The Popov basis P of $\mathcal{L}(R')$ can be computed deterministically with $\tilde{O}(N^\omega \max(1, \Delta))$ operations; P is row reduced, $\mathcal{L}(P) = \mathcal{L}(R')$, and the entries of P have degree at most Δ.*

Proof. Apply [23, Thm. 1.3] with shift 0 to R'^{T}, whose row module is our column module. The cost is $\tilde{O}(N^\omega \lceil D_{\mathrm{sum}}/N \rceil)$ with $D_{\mathrm{sum}} \leq N\Delta$ the sum of the column degrees. Transposing the 0-Popov form of [23] gives the Popov form of [21, §6.1], which is row reduced. Since $R' = PU$ with U unimodular and P is column reduced, $\deg P \leq \deg R' \leq \Delta$ by [21, Lemma 5.5]. □

2.3 Modular composition after Neiger, Salvy, Schost and Villard

We now recall the algorithm $\textsc{ModularCompositionBaseCase}(f, a, g, \mathbf{r})$ of Neiger, Salvy, Schost and Villard [21, Algorithm 8.1]. Its idea is to replace the argument a, which may be in bad position, by a random element γ of $\mathcal{A}$, and to express a as a polynomial in γ. Let $\mathbf{r} = (r_1, \ldots, r_{n+\max(m,2)})$ be the random tuple,[2] and put

$$\hat{g} = g(y - r_1), \qquad \hat{a} = (a + r_1)(x + r_2), \qquad \hat{f} = f(x + r_2),$$
$$\gamma = r_3 + r_4 x + \cdots + r_{n+2} x^{n-1}.$$

Let μ_γ be the minimal polynomial of γ mod $\hat{f}$, and suppose that it has degree n, so that there is a unique $\phi_\gamma \in \mathbb{K}[y]_{<n}$ with $\phi_\gamma(\gamma) \equiv \hat{a} \bmod \hat{f}$. Then $\hat{g}(\hat{a}) \equiv \hat{g}(\phi_\gamma)(\gamma)$ modulo $\hat{f}$, and since $\mu_\gamma(\gamma) \equiv 0$,

$$g(a) \operatorname{rem} f = b'(x - r_2), \qquad \text{where} \quad b = \hat{g}(\phi_\gamma) \operatorname{rem} \mu_\gamma, \quad b' = b(\gamma) \operatorname{rem} \hat{f}. \tag{2.3}$$

In this way, one composition with the argument a becomes two compositions, with the arguments ϕ_γ modulo μ_γ and γ modulo $\hat{f}$. Each of them is a bivariate composition with a matrix of relations, and the randomization puts both pairs in good position with high probability. The stages of the algorithm are listed in Table 4; for $n = 1$ it simply returns $g(a)$. In what follows, we write f and a for the shifted polynomials $\hat{f}$ and $\hat{a}$ when no confusion can arise.

On success, $R^{(\gamma,\hat{f})} \in \mathbb{K}[y]^{m\times m}$ is the Popov basis of $\mathcal{M}_m^{(\gamma,\hat{f})}$, of degree at most $2d$, and μ_γ has degree n [21, Prop. 6.1], while $R^{(\phi_\gamma,\mu_\gamma)}$ is a matrix of relations for $\mathcal{M}_{m'}^{(\phi_\gamma,\mu_\gamma)}$ of degree at most $2d$, with $m' \leq \max(1, 2(m-1))$ [21, Prop. 5.8]. Both routines are specified for every $m \leq n$ and $d \geq 1$, with cost $\tilde{O}(m^\omega d + c(n, m, d))$. In each composition, the vector $\hat{g}\, e_1$ (respectively $b\, e_1$) is reduced modulo the column module of the relation matrix, and the result is evaluated by the

[2] The change of basis reads r_{n+2}, which requires one extra element when $m = 1$, and the tuple $\mathbf{r}' = (r_{n+i})_{3\leq i\leq m}$ used by the second relation matrix is empty when $m \leq 2$.

method of Nüsken and Ziegler. Since these two compositions never fail, the event Fail depends only on $(f, a, \mathbf{r})$. In [21] the dimension is $m = \lceil n^\eta \rceil$ for an exponent η given there; we take m as a parameter.

Proposition 2.4 ([21, Prop. 8.1, 8.3], for an arbitrary m)**.** *Let $f \in \mathbb{K}[x]$ be monic of degree $n \geq 2$, $a \in \mathbb{K}[x]_{<n}$, $g \in \mathbb{K}[y]$ of degree $O(n)$, $m \in \{1, \ldots, n\}$, $d = \lceil n/m \rceil$ and $\mathbf{r} \in \mathbb{K}^{n+\max(m,2)}$. Algorithm 8.1 of [21], with the dimension m given as a parameter, returns $g(a) \operatorname{rem} f$ or Fail using $\tilde{O}(m^\omega d + c(n, m, d))$ operations in $\mathbb{K}$. The event Fail is a function of $(f, a, \mathbf{r})$ only. If f is separable and the entries of $\mathbf{r}$ are uniform and independent in a finite $S \subseteq \mathbb{K}$, then $\Pr[\mathsf{Fail}] \leq 6n^2/\operatorname{card}(S)$.*

Proof. Correctness for every m follows from Eq. (2.3) and from Propositions 4.4, 5.8 and 6.1 of [21], which are stated for arbitrary dimensions and degrees. For a returned relation matrix of dimension $m' \leq 2m$, the bivariate composition costs $\tilde{O}(m'^\omega(2d + n/m') + c(n, m', 2d)) = \tilde{O}(m^\omega d + c(n, m, d))$, because $m'^{\omega-1} n = O(m^\omega d)$ and $(m' + n/(2d))(2d)^{\omega_2/2} = O(md^{\omega_2/2})$ as $n/d \leq m$.

For the failure probability, each shift test fails only at the roots of a nonzero polynomial of degree at most n in the shift, namely $\operatorname{Res}_x(a + r_1, f)$ and $f(r_2)$. Let $\xi_\nu = \gamma(\zeta_\nu)$ at the roots ζ_ν of $\hat{f}$; since evaluation at the roots is an invertible linear substitution, $\prod_\nu \xi_\nu \prod_{\nu<\nu'}(\xi_\nu - \xi_{\nu'})$ is a nonzero polynomial of degree $n + n(n-1)/2$ in the coefficients of γ, whose nonvanishing makes γ a unit and a generator. The two Hankel conditions fail on the zero sets of nonzero polynomials of degrees at most $2n^2/m$ and $2n^2$ [21, Props. 7.6 and 7.8], and the certificate of the second relation matrix fails with probability at most $(m-1)/\operatorname{card}(S)$ [21, Prop. 5.8]. The total numerator, divided by n^2, is $\frac{5}{2} + \frac{5}{2n} + \frac{2}{m} + \frac{m-1}{n^2}$; this is convex in m, hence maximal on $[1, n]$ at an endpoint, where it is at most $23/4 < 6$ for $n \geq 2$. Finally, every failure test uses only $(f, a, \mathbf{r})$, and the translation of g involves no test. □

2.4 Two relation matrices

Following [22, §2], for $\bar{m} \geq 1$ we also use the $\mathbb{K}[x]$-module

$$\mathcal{N}_{\bar{m}}^{(a,f)} = \{r \in \mathbb{K}[x, y]_{<(\cdot,\bar{m})} : \; r(x, a) \equiv 0 \bmod f\},$$

free of rank $\bar{m}$, whose bases are represented by nonsingular matrices in $\mathbb{K}[x]^{\bar{m}\times\bar{m}}$, and all of them have determinant f up to a constant [22, §2.1].

Proposition 2.5 ([22, Prop. 2.3, Cor. 2.4])**.** *Let $n \geq \bar{m} > 0$, $f \in \mathbb{K}[x]$ of degree n and $a \in \mathbb{K}[x]_{<n}$. The Popov basis of $\mathcal{N}_{\bar{m}}^{(a,f)}$ can be computed deterministically with $\tilde{O}(\bar{m}^{\omega-1} n)$ operations; its degree $\bar{d}$ satisfies $\lceil n/\bar{m} \rceil \leq \bar{d} \leq n$. Within the same cost one can compute $A_0, \ldots, A_{\bar{m}-1} \in \mathbb{K}[x, y]_{<(\bar{d},\bar{m})}$ with $A_j(x, a) \equiv a^{j\bar{m}} \bmod f$.*

The construction of [22, Cor. 2.4] also returns polynomials B_j for the powers $a^{j\bar{m}^2}$; the two-level batched algorithms do not need them.

Lemma 2.6 (Krylov criterion [22, Lemma A.1])**.** *The degree of a minimal basis of $\mathcal{N}_{\bar{m}}^{(a,f)}$ is the least integer D such that the $n \times D\bar{m}$ matrix whose columns are the coefficient vectors of $x^i a^j \operatorname{rem} f$, $i < D$, $j < \bar{m}$, has rank n.*

For the module $\mathcal{M}$, let M_a be the matrix of multiplication by a in the monomial basis of $\mathbb{K}[x]/\langle f \rangle$ and $X = [I_m \; 0]^\mathsf{T}$. Consider the $md \times md$ block Hankel matrix $\mathrm{Hk}_{m,d}^{(a,f)} = [X^\mathsf{T} M_a^{i+j} X]_{0 \leq i,j < d}$ of [21, §7] and the algorithm CandidateBasis [21, Algorithm 5.1]. When a is a unit, the reconstruction in that algorithm uses the corresponding expansion at zero in inverse powers of a.

Proposition 2.7 ([21, Prop. 5.6, Prop. 7.1]). *Let $f \in \mathbb{K}[x]$ have degree n with $f(0) \neq 0$, $a \in \mathbb{K}[x]_{<n}$ with $\gcd(a, f) = 1$, $m \leq n$ and $d \geq 1$. CandidateBasis(f, a, m, d) uses $\tilde{O}(m^\omega d + c(n, m, d))$ operations and returns a weak Popov matrix of degree at most $2d$ and a flag in $\{\mathrm{Cert}, \mathrm{NoCert}\}$, with the following two properties.*

(i) (Soundness.) Whenever the flag is Cert, the matrix is a basis of $\mathcal{M}_m^{(a,f)}$.

(ii) (Completeness.) If $\mathrm{Hk}_{m,\lceil n/m \rceil}^{(a,f)}$ has rank n and $d \geq \lceil n/m \rceil$, the flag is Cert, and the matrix has degree at most d.

Soundness means that an accepted matrix is correct whatever the random elements. More generally, if $\operatorname{rank} \mathrm{Hk}_{m,d}^{(a,f)} = \deg \det \mathcal{M}_m^{(a,f)}$, the matrix returned is a weak Popov basis of $\mathcal{M}_m^{(a,f)}$ of degree at most d [21, Props. 5.6 and 7.1].

2.5 The Kaltofen–Shoup framework

Let $f \in \mathbb{F}_q[x]$ be monic and square-free of degree n, $\mathcal{A} = \mathbb{F}_q[x]/\langle f \rangle$, $\beta \in [0, 1]$, $l = \lceil n^\beta \rceil$ and $m_D = \lceil n/(2l) \rceil$. We use the five steps (D1)–(D5) of Algorithm D as described in Section 1.1, with the labels of [16]: the baby steps h_i, the giant steps G_j, the interval polynomials I_j, the coarse factorization $F_j = \gcd(f^*, I_j)$ and the fine factorization by the gcds with $G_j - h_i$. Only the giant steps (D2) involve modular composition. The baby steps (D1) cost $O(l\,\mathsf{M}(n) \log q) = n^{1+\beta+o(1)} \log q$ by repeated q-th powering, and the steps (D3)–(D5), which consist of a multipoint evaluation of $\prod_{i<l}(Y - h_i)$ over $\mathcal{A}$ at $G_1, \ldots, G_{m_D}$ followed by gcds, cost $n^{1+\beta+o(1)} + n^{2-\beta+o(1)}$ [16, proof of Thm. 2]. Kaltofen and Shoup compute (D2) by doubling, which we recall in Section 5.2. Square-free factorization of a polynomial of degree n over $\mathbb{F}_q$ uses $O(\mathsf{M}(n) \log n + n \log q)$ operations [37, §3], and equal-degree factorization is done by [37, Algorithm 3.6] with the trace computation of [37, §5]; we recall it in Section 5.3.

3 Batched composition with a fixed argument

Let $f \in \mathbb{K}[x]$ be monic of degree $n \geq 2$, $a \in \mathbb{K}[x]_{<n}$ arbitrary and $g_1, \ldots, g_k \in \mathbb{K}[y]_{<n}$. We compute all $g_i(a) \operatorname{rem} f$ in two phases: the stages of Table 4 that depend only on (f, a) and the random elements are run once, and the two compositions of Eq. (2.3) are then performed for all g_i together, by matrix products. The tool for the second phase is a module presentation of $\mathcal{A}$. The proofs of Theorem 3.6 and of the auxiliary statements on costs are in Appendix B.

3.1 Relation matrices as presentations

Let $P \in \mathbb{K}[y]^{N \times N}$ be nonsingular and row reduced, with row degrees $r_1, \ldots, r_N$, and let

$$Q_P = \mathbb{K}[y]^N / \mathcal{L}(P), \qquad V_P = \mathbb{K}[y]_{<r_1} \times \cdots \times \mathbb{K}[y]_{<r_N} \subseteq \mathbb{K}[y]^N.$$

The quotient Q_P is a $\mathbb{K}[y]$-module, and we write Y for the action of y on it.

Proposition 3.1 (normal forms in a presentation). *(a) The composite $V_P \hookrightarrow \mathbb{K}[y]^N \to Q_P$ is an isomorphism of $\mathbb{K}$-vector spaces, whose inverse is $[v] \mapsto v \operatorname{Rem} P$. In particular $\dim_{\mathbb{K}} Q_P = r_1 + \cdots + r_N = \deg \det P$.*

(b) Transport Y to V_P by this isomorphism. Then, for every $g \in \mathbb{K}[y]$ and $v \in \mathbb{K}[y]^N$,

$$(g\,v) \operatorname{Rem} P = g(Y)\,(v \operatorname{Rem} P).$$

(c) *Let* $\mathcal{L}(P) = \mathcal{M}_N^{(a,f)}$ *and let* $\mathcal{A}$ *be a* $\mathbb{K}[y]$*-module through* $y \cdot u = au$*. The map* $\iota : Q_P \to \mathcal{A}$*,* $[(v_0, \ldots, v_{N-1})] \mapsto \sum_i v_i(a)\, x^i \operatorname{rem} f$*, is an injective morphism of* $\mathbb{K}[y]$*-modules, with image* $\mathcal{A}_N = \sum_{i<N} x^i \mathbb{K}[a]$*, and*

$$g(a) \operatorname{rem} f = \iota(g(Y)[e_1]) \qquad (g \in \mathbb{K}[y]).$$

If a *generates the* $\mathbb{K}$*-algebra* $\mathcal{A}$*, then* ι *is an isomorphism and* $\deg \det P = n$*.*

Proof. (a) Existence and uniqueness of $v \operatorname{Rem} P$ (Section 2.2) say exactly that every class of Q_P has a unique representative in V_P. Since P is row reduced, $\deg \det P = \sum_i r_i$ by the predictable-degree property. (b) Both sides are elements of V_P congruent to $g\,v$ modulo the $\mathbb{K}[y]$-module $\mathcal{L}(P)$, so they are equal by (a). (c) The map $\mathbb{K}[y]^N \to \mathcal{A}$, $v \mapsto \sum_i v_i(a) x^i \operatorname{rem} f$, is $\mathbb{K}[y]$-linear when y acts on $\mathcal{A}$ through a, and its kernel is $\mathcal{M}_N^{(a,f)} = \mathcal{L}(P)$ by definition. Hence ι is well defined and injective, with the stated image. Since $\iota([e_1]) = 1$, we get $\iota(g(Y)[e_1]) = g(a) \cdot 1$. If a generates $\mathcal{A}$ then $\mathbb{K}[a] = \mathcal{A}$, so $\mathcal{A}_N = \mathcal{A}$ and $\deg \det P = \dim_{\mathbb{K}} Q_P = n$. □

Brent and Kung inside the presentation For a generating a and a basis P of $\mathcal{M}_N^{(a,f)}$, Proposition 3.1 identifies Q_P with $\mathcal{A}$, so that composition becomes evaluation at Y applied to $[e_1]$, as in [21]. For a proper full-rank submodule, such as that used by the one-matrix algorithm, Q_P need not be isomorphic to $\mathcal{A}$. However, the identities below still hold, because $\mathcal{L}(P)$ consists of relations and is a $\mathbb{K}[y]$-module.

We now apply Brent–Kung blocking in Q_P. Write $g = \sum_{j<J} g^{(j)} y^{jD}$ with $\deg g^{(j)} < D$, so that

$$g(Y)[e_1] = \sum_{j<J} g^{(j)}(Y)\, Y^{jD}[e_1].$$

In $\mathcal{A}$, the baby steps $a, a^2, \ldots, a^{D-1}$ have n coefficients each, and applying the $g^{(j)}$ requires the product of an $n \times D$ by a $D \times J$ matrix; this gives the Brent–Kung cost $n^{(\omega+1)/2}$. In Q_P, on the other hand, the same operation multiplies a representative $v \in V_P$ by the polynomial $g^{(j)}$, and Proposition 3.1(b) allows us to reduce only at the end:

$$g(Y)[e_1] = \Big[\sum_{j<J} g^{(j)}\, P_j \Big], \qquad P_j := (y^{jD} e_1) \operatorname{Rem} P.$$

The vectors P_j form a *remainder table*, which we compute once by repeated squaring of Y^D (Lemma 3.3). For a batch, we multiply this $N \times J$ table by the $J \times k$ matrix $(g_i^{(j)})$ of coefficient blocks, and then we divide by P (Lemma 3.4). With $N \approx J \approx m$ and $D = d = n/m$, the table has $O(n)$ coefficients per column and $O(mn)$ in total, and its construction costs $\tilde{O}(m^\omega d)$, which fits the precomputation budget. Hence the reduction of the whole batch is one rectangular product with entries of degree $O(d)$. We use this construction for both presentations of Eq. (2.3), that is, for $(\phi_\gamma, \mu_\gamma)$ and for (γ, f).

3.2 Shared precomputation

Lemma 3.2 (shared precomputation)**.** *Let* $m \in \{1, \ldots, n\}$*,* $d = \lceil n/m \rceil$ *and* $1 \le B \le 2d$*. From* $(f, a, \mathbf{r})$ *alone, with* $\mathbf{r} \in \mathbb{K}^{n+\max(m,2)}$*, one can compute with* $\tilde{O}(m^\omega d + c(n, m, d) + Bn)$ *operations either* Fail*, exactly when the first three stages of Table 4 fail, or the following data: the matrices* $P_\gamma = R^{(\gamma,f)}$ *and* P_ϕ*, the Popov basis of* $\mathcal{L}(R^{(\phi_\gamma,\mu_\gamma)})$*, which are row reduced of degree at most* $2d$*, with* $\mathcal{L}(P_\gamma) = \mathcal{M}_m^{(\gamma,f)}$ *and* $\mathcal{L}(P_\phi) \subseteq \mathcal{M}_{m'}^{(\phi_\gamma,\mu_\gamma)}$ *for some* $m' \le 2m$*; for each* $P \in \{P_\phi, P_\gamma\}$*, the division data of Lemma 2.2(a) with* $\delta = 4d$ *and the remainder table* $P_j = (y^{jd} e_1) \operatorname{Rem} P$*,* $0 \le j < \lceil n/d \rceil$*; and the powers* $\phi_\gamma^i \operatorname{rem} \mu_\gamma$ *and* $\gamma^i \operatorname{rem} f$ *for* $0 \le i \le B$*.*

Proof. The relation matrices come from the first three stages of Table 4, within the stated bound and with the stated properties [21, Props. 5.8 and 6.1]. Converting the second one to Popov form, the division data and the remainder tables each cost $\tilde{O}(m^\omega d)$, by Lemmas 2.2 and 2.3 and the remainder-doubling Lemma 3.3 below (with $N \le 2m$, $J \le m$, $D = d$, $\Delta_0 \le 2d$), and the powers cost $\tilde{O}(Bn)$. □

3.3 Remainder doubling

Fix a row-reduced $P \in \mathbb{K}[y]^{N\times N}$ with row degrees $(r_1, \dots, r_N)$ and entries of degree at most Δ_0, and write $\rho(v) = v \operatorname{Rem} P$, extended columnwise to matrices.

Lemma 3.3 (remainder doubling)**.** *Let $D \ge 1$, $\Pi^{(\nu)} = \rho(y^{2^\nu D} I_N)$ and $P_j = \rho(y^{jD} e_1)$ for $\nu, j \ge 0$.*

(a) ρ is $\mathbb{K}$-linear, $\rho(v) = \rho(v')$ whenever $v - v' \in \mathcal{L}(P)$, and $\rho(c\,v) = \rho(c\,\rho(v))$ for every $c \in \mathbb{K}[y]$.

(b) $\Pi^{(\nu+1)} = \rho(\Pi^{(\nu)}\Pi^{(\nu)})$ and $P_{j+2^\nu} = \rho(\Pi^{(\nu)} P_j)$ for all $\nu, j \ge 0$.

(c) For every $g = \sum_{j<J} g^{(j)} y^{jD}$ with $\deg g^{(j)} < D$, one has $g\,e_1 \equiv \sum_{j<J} g^{(j)} P_j$ modulo $\mathcal{L}(P)$; hence $\rho(g\,e_1) = \rho(\sum_{j<J} g^{(j)} P_j)$.

(d) $P_0, \dots, P_{J-1}$ can be computed by Algorithm 3.1 with $\tilde{O}((\Delta_0 + D)(N^\omega + \mathsf{MM}(N, N, J)))$ operations in $\mathbb{K}$.

Proof. (a) Uniqueness of the remainder gives the first two claims. Since $\mathcal{L}(P)$ is a $\mathbb{K}[y]$-module, $v - \rho(v) \in \mathcal{L}(P)$ implies $cv - c\rho(v) \in \mathcal{L}(P)$, hence $\rho(cv) = \rho(c\rho(v))$.

(b) For every $v \in \mathbb{K}[y]^N$,

$$y^{2^\nu D} v = \sum_i v_i\, y^{2^\nu D} e_i \equiv \sum_i v_i\, \Pi^{(\nu)} e_i = \Pi^{(\nu)} v \pmod{\mathcal{L}(P)},$$

because $y^{2^\nu D} e_i - \Pi^{(\nu)} e_i \in \mathcal{L}(P)$ by definition and $\mathcal{L}(P)$ is a $\mathbb{K}[y]$-module. Applying this to $v = \Pi^{(\nu)} e_i \equiv y^{2^\nu D} e_i$ gives $y^{2^{\nu+1} D} e_i \equiv y^{2^\nu D} \Pi^{(\nu)} e_i \equiv \Pi^{(\nu)} \Pi^{(\nu)} e_i$, which is the first identity, and applying it to $v = P_j \equiv y^{jD} e_1$ gives $y^{(j+2^\nu)D} e_1 \equiv \Pi^{(\nu)} P_j$, which is the second.

(c) $g\,e_1 = \sum_j g^{(j)} y^{jD} e_1 \equiv \sum_j g^{(j)} P_j$, since $\mathcal{L}(P)$ is a $\mathbb{K}[y]$-module.

(d) The loop invariant of Algorithm 3.1 follows from (b). Remainders have entries of degree less than $r_i \le \Delta_0$, so every product formed in the loop has entries of degree less than $2\Delta_0$, and an excess $\delta = 2\Delta_0$ is sufficient in Lemma 2.2, whatever the row degrees r_i. With the precomputation of Lemma 2.2(a), iteration ν costs $\tilde{O}(\Delta_0\, \mathsf{MM}(N, N, N + 2^\nu))$ for the product and the same for the division. Since $2^\nu < 2J$ and there are $O(\log J)$ iterations, the total is $\tilde{O}(\Delta_0(N^\omega + \mathsf{MM}(N, N, J)))$, to which the initial divisions (excess $D+1$) add $\tilde{O}((\Delta_0 + D)N^\omega)$. □

3.4 Batched evaluation

Lemma 3.4 (batched reduction)**.** *Let P be as in Section 3.3, $D \ge 1$, and suppose that $P_0, \dots, P_{J-1}$ and the precomputation of Lemma 2.2(a) with $\delta \ge \Delta_0 + D$ are known. Given $g_1, \dots, g_k \in \mathbb{K}[y]_{<JD}$, the vectors $\tilde{v}_i = \rho(g_i e_1)$, $1 \le i \le k$, can be computed with $\tilde{O}((\Delta_0 + D)(\mathsf{MM}(N, J, k) + \mathsf{MM}(N, N, k)))$ operations, by a linear straight-line program in the coefficients of the g_i. Each $\tilde{v}_i$ satisfies $\deg \tilde{v}_{i,l} < r_l \le \Delta_0$ and $g_i e_1 - \tilde{v}_i \in \mathcal{L}(P)$.*

Proof. Write $g_i = \sum_{j<J} g_i^{(j)} y^{jD}$ with $\deg g_i^{(j)} < D$ (no arithmetic), let $G = (g_i^{(j)})_{j,i} \in \mathbb{K}[y]^{J\times k}$, and compute $V = [P_0 | \cdots | P_{J-1}]\, G$, at cost $\tilde{O}((\Delta_0 + D)\mathsf{MM}(N, J, k))$. The entries of V have degree less than $\Delta_0 + D$, so an excess $\Delta_0 + D$ suffices in Lemma 2.2(a), which gives $\rho(V)$ within the stated cost. By Lemma 3.3(c), column i of $\rho(V)$ is $\rho(g_i e_1)$. Both steps multiply the variable matrix G, then V, by fixed matrices, so the computation is linear (Lemma 2.2(b)). □

Algorithm 3.1 RemainderTable(P, D, J)

Require: $P \in \mathbb{K}[y]^{N\times N}$ row reduced with entries of degree $\le \Delta_0$; integers $D, J \ge 1$; the precomputation of Lemma 2.2(a) for P with $\delta = \max(2\Delta_0, D+1)$
Ensure: $P_j = (y^{jD} e_1) \operatorname{Rem} P$ for $0 \le j < J$
1: $\Pi \leftarrow (y^D I_N) \operatorname{Rem} P$; $\quad C \leftarrow [\, e_1 \operatorname{Rem} P \,]$
2: $\nu \leftarrow 0$
3: **while** $2^\nu < J$ **do**
4: $\quad \triangleright$ here $\Pi = (y^{2^\nu D} I_N) \operatorname{Rem} P$ and $C = [P_0 | \cdots | P_{2^\nu - 1}]$
5: $\quad W \leftarrow \Pi \cdot [\, \Pi \mid C \,]$
6: $\quad [\, \Pi \mid C' \,] \leftarrow W \operatorname{Rem} P$; $\quad C \leftarrow [\, C \mid C' \,]$
7: $\quad \nu \leftarrow \nu + 1$
8: **end while**
9: **return** the first J columns of C

Algorithm 3.2 BatchedComposition$(f, a, (g_i)_{1\le i\le k}, m, B, \mathbf{r})$

Require: f monic of degree $n \ge 2$; $a \in \mathbb{K}[x]_{<n}$; $g_1, \ldots, g_k \in \mathbb{K}[y]_{<n}$; $m \in \{1, \ldots, n\}$; $1 \le B \le 2d$ with $d = \lceil n/m \rceil$; $\mathbf{r} \in \mathbb{K}^{n+\max(m,2)}$
Ensure: $(g_i(a) \operatorname{rem} f)_{1\le i\le k}$, or Fail
1: compute the data of Lemma 3.2 from $(f, a, m, B, \mathbf{r})$, or **return** Fail
2: $g_i \leftarrow g_i(y - r_1)$ for all i
3: $\tilde{v}_i \leftarrow g_i e_1 \operatorname{Rem} P_\phi$ for all i, and let $\tilde{g}_i(x, y)$ be the polynomial of $\tilde{v}_i$
4: $b_i \leftarrow \tilde{g}_i(x, \phi_\gamma) \operatorname{rem} \mu_\gamma$ for all i; rename x into y
5: $\tilde{w}_i \leftarrow b_i e_1 \operatorname{Rem} P_\gamma$ for all i, and let $\tilde{u}_i(x, y)$ be the polynomial of $\tilde{w}_i$
6: $b'_i \leftarrow \tilde{u}_i(x, \gamma) \operatorname{rem} f$ for all i
7: **return** $(b'_i(x - r_2))_{1\le i\le k}$

Lemma 3.5 (batched bivariate evaluation; after [21, Algorithms 3.4, 3.5, Lemma 3.3, Prop. 3.4])**.** *Let $h \in \mathbb{K}[x]$ be monic of degree n, $b \in \mathbb{K}[x]_{<n}$, and $\tilde{g}_1, \ldots, \tilde{g}_k \in \mathbb{K}[x, y]_{<(N,\Delta)}$. For $1 \le B \le \Delta$ and $s = \lceil \Delta/B \rceil$, all $\tilde{g}_i(x, b) \operatorname{rem} h$ can be computed with $\tilde{O}(N\,\mathsf{MM}(ks, B, \lceil n/N \rceil) + ks(n+N) + Bn)$ operations. The powers $b^j \operatorname{rem} h$, $j \le B$, are computed once, at cost $\tilde{O}(Bn)$, and are the only data depending on b; given them, the computation is a linear straight-line program in the coefficients of the $\tilde{g}_i$.*

Proof. Write $\tilde{g}_i = \sum_{u<s} \tilde{g}_{i,u}\, y^{Bu}$ with $\deg_y \tilde{g}_{i,u} < B$. Applied to the ks polynomials $\tilde{g}_{i,u}$, SimultaneousBivariateModularComposition [21, Algorithm 3.4] returns all $\tilde{g}_{i,u}(x, b) \operatorname{rem} h$. Its main step multiplies the $ks \times B$ matrix of coefficients of the $\tilde{g}_{i,u}$ by the $B \times \lceil n/N \rceil$ matrix of slices of length N of the powers $b^j \operatorname{rem} h$, over $\mathbb{K}[x]_{<N}$, at cost $\tilde{O}(N\,\mathsf{MM}(ks, B, \lceil n/N \rceil))$, while recombination and reduction modulo h cost $\tilde{O}(ks(n+N))$ and the powers cost $\tilde{O}(Bn)$. As in BivariateModularComposition [21, Algorithm 3.5], each $\sum_u (\tilde{g}_{i,u}(x, b) \operatorname{rem} h)\, \hat{b}^u$, with $\hat{b} = b^B \operatorname{rem} h$, is then evaluated by Horner's rule with $s-1$ multiplications modulo h, in total $\tilde{O}(ksn)$. Throughout, the coefficients of the $\tilde{g}_{i,u}$ are multiplied only by fixed data, namely the powers of b, $\hat{b}$ and the precomputed inverse of the reversal of h, so the computation is linear in them. □

If $N \le 2m$ and $d = \lceil n/m \rceil$, splitting the $\lceil n/N \rceil$ columns into blocks of d columns gives $N\,\mathsf{MM}(ks, B, \lceil n/N \rceil) = O(m\,\mathsf{MM}(ks, B, d))$. Thus, in all our uses, the cost of Lemma 3.5 is

$$\tilde{O}(m\,\mathsf{MM}(ks, B, d) + ksn + Bn).$$

3.5 The one-matrix theorem

Table 5: Batch curves: exponent $\Lambda(t)$ of the cost of $k = n^t$ compositions with one argument, with the full table of rectangular bounds (programs of Appendix A). "One matrix": Algorithm 3.2, arbitrary argument (Eq. (3.2)); "two matrices": Algorithm 4.1, generating argument (Eq. (4.2)); the same two-matrix curve holds for arbitrary arguments by Theorem 6.21; "more levels": $\Lambda_4 \le \Lambda_3$ of Theorems 6.10, 6.12, and 6.23, any argument (four levels at $t = 0$); "reshaping, gen.": Theorem 4.9, generating argument; "BK": batched Brent–Kung [16, Lemma 3] with rectangular products and at most n baby steps; "$t + \kappa$": k independent compositions [21]. The dual certificates and the comparison with batched Brent–Kung are in Appendix B.3.

t	one matrix, rect.	one matrix, square Λ_{sq}	two matrices, rect.	more levels, rect.	reshaping, gen.	BK, rect.	$t+\kappa$	$1+t$
0	1.4294	1.4572	1.4572	1.3309	1.6857	1.6251	1.4294	1
0.2	1.5275	1.5486	1.4572	1.4034	1.6857	1.7679	1.6294	1.2
0.5	1.6857	1.6857	1.5983	1.5983	1.6857	1.9899	1.9294	1.5
1	2.1251	2.1857	2.0834	2.0834	2	2.3714	2.4294	2
1.5	2.6251	2.6857	2.5834	2.5834	2.5	2.7947	2.9294	2.5
2	3.1251	3.1857	3.0834	3.0834	3	3.2501	3.4294	3

Theorem 3.6 (batched composition, arbitrary argument)**.** *Let $m \in \{1, \dots, n\}$, $d = \lceil n/m \rceil$, $1 \le B \le 2d$, $s = \lceil 2d/B \rceil$ and $k \ge 1$.*

(i) For every $\mathbf{r} \in \mathbb{K}^{n+\max(m,2)}$, Algorithm 3.2 returns either $(g_i(a) \operatorname{rem} f)_{1 \le i \le k}$ or Fail*, using*

$$\tilde{O}\Big(m^\omega d + c(n,m,d) + d\,\mathsf{MM}(m,m,k) + m\,\mathsf{MM}(ks,B,d) + ksn + Bn + kn\Big) \tag{3.1}$$

operations in $\mathbb{K}$.

(ii) The event Fail *is that of Proposition 2.4; in particular, it does not depend on the g_i. If f is separable and the entries of $\mathbf{r}$ are uniform and independent in a finite $S \subseteq \mathbb{K}$, then $\Pr[\mathsf{Fail}] \le 6n^2/\operatorname{card}(S)$.*

(iii) For $k = \lceil n^t \rceil$, the cost is $n^{\Lambda_1(t)+o(1)}$, where

$$\Lambda_1(t) = \min_{\substack{0 \le \eta \le 1 \\ 0 \le \varrho \le 1-\eta}} \max \left\{ \begin{array}{l} 1 - \eta + \omega(\eta,\eta,\eta), \quad \eta + \omega(\frac{1-\eta}{2}, \frac{1-\eta}{2}, 1-\eta), \\ 1 - \eta + \omega(\eta,\eta,t), \\ \eta + \omega(t+1-\eta-\varrho,\ \varrho,\ 1-\eta), \\ t + 2 - \eta - \varrho, \quad 1 + \varrho, \quad 1 + t \end{array} \right\}, \tag{3.2}$$

and $\omega(\cdot,\cdot,\cdot)$ may be replaced by any upper bound. We write $\hat{\Lambda}_1$ for the function obtained with U of Eq. (2.2) and the selected generators of Appendix A; then $\Lambda_1 \le \hat{\Lambda}_1$, and all numerical values of Λ_1 stated in this paper are values of $\hat{\Lambda}_1$. For instance, $\hat{\Lambda}_1(t) = t + (\omega_2 - 1)/2$ for $t \ge 1$.

(iv) With square blocks and suitable parameters, with $m = O(n^{1/2} + (nk)^{1/3})$, the cost is

$$\tilde{O}(n^{(\omega+2)/3} k^{(\omega-1)/3} + n^{\omega/2} k)$$

for every $k \ge 1$; if $\omega = 2$, it is $n^{o(1)}(n^{4/3}k^{1/3} + nk)$, in particular $(nk)^{1+o(1)}$ for $k \ge n^{1/2}$.

Remark 3.7 (numerical values and comparison). Table 5 lists the envelope values $\hat{\Lambda}_1$, with $\hat{\Lambda}_1(0) = \kappa$ and $\Lambda_1(t) \le \hat{\Lambda}_1(t) = t + (\omega_2 - 1)/2$ for $t \ge 1$. The latter gives the amortized exponent $(\omega_2 - 1)/2 < 1.1251$ per composition, against $\omega/2 < 1.1857$ with square blocks; at $t = 1$ the cost is $O(n^{2.1251})$, against n^ω for computing all powers followed by one product.

4 Generating arguments

The proofs of the statements of this section are in Appendix B.4.

4.1 Good position

Assume that f is separable and that a generates $\mathcal{A}$, that is, the minimal polynomial of a mod f has degree n; equivalently, a takes n distinct values at the roots of f in an algebraic closure $\overline{\mathbb{K}}$. We use the parameters

$$m = \bar{m} = \lceil n^{1/3} \rceil, \qquad d = \bar{d} = \lceil n/m \rceil.$$

They satisfy $d \leq \bar{m}^2$.

Lemma 4.1. *Let $h \in \mathbb{K}[x]$ be separable of degree n with roots $\zeta_1, \ldots, \zeta_n \in \overline{\mathbb{K}}$, and $b \in \mathbb{K}[x]_{<n}$ with $b(\zeta_i) = \lambda_i$. For $D, \bar{m} \geq 1$, the rank of the matrix of the coefficient vectors of $x^i b^j \operatorname{rem} h$, $i < D$, $j < \bar{m}$, is the rank of the $n \times D\bar{m}$ matrix $[\zeta_\nu^i \lambda_\nu^j]_{\nu \leq n;\,(i,j)}$ over $\overline{\mathbb{K}}$.*

Proof. Evaluation at the ζ_ν is an isomorphism of $\overline{\mathbb{K}}$-algebras $\overline{\mathbb{K}}[x]/\langle h \rangle \to \overline{\mathbb{K}}^n$, which maps $x^i b^j$ to $(\zeta_\nu^i \lambda_\nu^j)_\nu$, and ranks do not change under field extension. □

Proposition 4.2 (genericity after a random change of basis)**.** *Let $f \in \mathbb{K}[x]$ be separable of degree $n \geq 2$ and let $a \in \mathbb{K}[x]_{<n}$ generate $\mathcal{A}$. Let $m, \bar{m} \in \{1, \ldots, n\}$, $d = \lceil n/m \rceil$, $\bar{d} = \lceil n/\bar{m} \rceil$, and let $r_1, r_2 \in S$ and the coefficients of $\gamma \in \mathbb{K}[x]_{<n}$ be uniform and independent in a finite $S \subseteq \mathbb{K}$. Put $\hat{f} = f(x + r_2)$ and $\hat{a} = (a + r_1)(x + r_2)$. With probability at least $1 - 8n^2/\operatorname{card}(S)$ all of the following hold:*

(a) $\gcd(\hat{a}, \hat{f}) = 1$ *and* $\hat{f}(0) \neq 0$*;*

(b) $\gcd(\gamma, \hat{f}) = 1$*, and the minimal polynomial μ_γ of γ mod $\hat{f}$ has degree n;*

(c) $\mathrm{Hk}_{m,d}^{(\gamma, \hat{f})}$ *has rank n;*

(d) $\mathrm{Hk}_{m,d}^{(\phi_\gamma, \mu_\gamma)}$ *has rank n, where $\phi_\gamma \in \mathbb{K}[y]_{<n}$ satisfies $\phi_\gamma(\gamma) \equiv \hat{a} \bmod \hat{f}$;*

(e) the minimal bases of $\mathcal{N}_{\bar{m}}^{(\gamma, \hat{f})}$ and of $\mathcal{N}_{\bar{m}}^{(\phi_\gamma, \mu_\gamma)}$ have degree $\bar{d}$.

The core of the proof is the following specialization argument. Let $\xi_\nu = \gamma(\hat{\varphi}_\nu)$ at the roots $\hat{\varphi}_\nu$ of $\hat{f}$ and $\hat{\lambda}_\nu = \hat{a}(\hat{\varphi}_\nu)$, which are pairwise distinct. By the evaluation lemma (Lemma 4.1), condition (e) for $(\phi_\gamma, \mu_\gamma)$ asks for an $n \times n$ minor of $[\xi_\nu^i \hat{\lambda}_\nu^j]_{\nu;\, i < \bar{d},\, j < \bar{m}}$ to be nonzero; on the columns $c = i\bar{m} + j < n$, the specialization $\xi_\nu = \hat{\lambda}_\nu^{\bar{m}}$ turns this minor into the Vandermonde determinant of the $\hat{\lambda}_\nu$, so the minor is a nonzero polynomial in the coefficients of γ, and the Schwartz–Zippel lemma applies. The other conditions are handled in the same way, with [21, Props. 7.6 and 7.8] for (c) and (d).

Proposition 4.3 (non-generating arguments)**.** *Let f be separable and suppose that a takes some value λ at ℓ distinct roots of f. Then every basis of $\mathcal{N}_{\bar{m}}^{(a,f)}$ has degree at least ℓ, and if $\ell > m$ then* $\deg \det$ *of any basis of $\mathcal{M}_m^{(a,f)}$ is less than n. Both statements hold for $(\phi_\gamma, \mu_\gamma)$ for every γ satisfying Proposition 4.2(b).*

A value of multiplicity larger than $\bar{m}$ prevents any change of basis from making the algorithm of [22] applicable at these parameters. Frobenius images $x^{q^s} \operatorname{rem} f$ with f square-free are generators, which is all that factorization needs.

Lemma 4.4 (degree certificate)**.** *If* CandidateBasis(h, b, m, d) *[21, Algorithm 5.1] returns Cert with a matrix R, or if* ChangeOfBasis *[21, Algorithm 6.1] succeeds with output R, where $d = \lceil n/m \rceil$, then* $\deg R \leq d$*, and the same holds for the Popov form of R.*

Algorithm 4.1 BatchedCompositionGen$(f, a, (g_i)_{1\le i\le k}, r_1, r_2, \gamma)$

Require: f monic separable of degree $n \ge 2$; $a \in \mathbb{K}[x]_{<n}$ generating $\mathcal{A}$; $g_1, \dots, g_k \in \mathbb{K}[y]_{<n}$; $r_1, r_2 \in \mathbb{K}$, $\gamma \in \mathbb{K}[x]_{<n}$; $m = \bar{m} = \lceil n^{1/3} \rceil$, $d = \bar{d} = \lceil n/m \rceil$

Ensure: $(g_i(a) \operatorname{rem} f)_{1\le i\le k}$, or Fail

1: $a \leftarrow (a + r_1)(x + r_2)$, $f \leftarrow f(x + r_2)$; if $\gcd(a, f) \ne 1$ or $f(0) = 0$, **return** Fail
2: $(R_\gamma, \mu_\gamma, \phi_\gamma) \leftarrow$ ChangeOfBasis(f, γ, a, m, d), or **return** Fail; if $\mu_\gamma(0) = 0$, **return** Fail
3: $(R', \mathit{flag}) \leftarrow$ CandidateBasis$(\mu_\gamma, \phi_\gamma, m, d)$; if flag = NoCert, **return** Fail; $R_\phi \leftarrow$ Popov form of R'
4: **for** $(b, h, R) \in \{(\phi_\gamma, \mu_\gamma, R_\phi), (\gamma, f, R_\gamma)\}$ **do**
5: compute the Popov basis of $\mathcal{N}_{\bar{m}}^{(b,h)}$; if its degree is not $\bar{d}$, **return** Fail; compute $A_j^{(b,h)}$, $j < \bar{m}$
6: compute the division data of Lemma 2.2(a) for R with $\delta = 2d$, the remainder table $P_j^{(R)}$, $j < m$, and $b^\ell \operatorname{rem} h$ for $\ell < 2\bar{m}$
7: **end for**
8: $g_i \leftarrow g_i(y - r_1)$ for all i
9: $(u_i)_i \leftarrow$ Evaluate$((g_i)_i, \phi_\gamma, \mu_\gamma, R_\phi)$; rename x into y
10: $(b_i)_i \leftarrow$ Evaluate$((u_i)_i, \gamma, f, R_\gamma)$
11: **return** $(b_i(x - r_2))_{1\le i\le k}$

4.2 The two-matrix batch

The subroutine Evaluate$((g_i)_i, b, h, R)$ computes all $g_i(b) \operatorname{rem} h$ from the precomputed data for (b, h, R). It proceeds in three stages, a reduction followed by the giant and baby steps of a two-level evaluation, and each stage is a matrix product for the whole batch. Let $L' = \lceil \bar{d}/m \rceil$ and $d' = m + \bar{d}$.

Reduction. First, we compute $\tilde{v}_i = g_i e_1 \operatorname{Rem} R$ for all i, by Lemma 3.4 with $D = d$ and $J = \lceil n/d \rceil \le m$. Let $G_i \in \mathbb{K}[x, y]_{<(m,d)}$ be the polynomial of $\tilde{v}_i$. By Proposition 3.1, $G_i(x, b) \equiv g_i(b) \bmod h$, and it remains to evaluate G_i, whose y-degree is now less than $d \le \bar{m}^2$.

Giant steps. Next, we write $G_i = \sum_{c<d} G_{i,[c]}(x) y^c$ and group its coefficients by blocks of $\bar{m}$: for $c_1 < \bar{m}$,

$$s_{i,c_1}(x, y_0) = \sum_{c_0<\bar{m}} G_{i,[c_0+\bar{m}c_1]}(x)\, y_0^{c_0} \in \mathbb{K}[x, y_0]_{<(m,\bar{m})},$$

with zero coefficients when $c_0 + \bar{m}c_1 \ge d$, so that $G_i(x, y) = \sum_{c_1} s_{i,c_1}(x, y)\, y^{\bar{m}c_1}$. The powers $b^{\bar{m}c_1}$ are replaced by the auxiliary polynomials A_{c_1}, which satisfy $A_{c_1}(x, b) \equiv b^{\bar{m}c_1} \bmod h$ and have x-degree less than $\bar{d}$. We split $A_{c_1} = \sum_{u<L'} A_{c_1,u}\, x^{um}$ with $A_{c_1,u} \in \mathbb{K}[x, y_0]_{<(m,\bar{m})}$, compute the $k \times L'$ product $[s_{i,c_1}]_{i,c_1} \cdot [A_{c_1,u}]_{c_1,u}$, and put $S_i = \sum_u x^{um}(\cdot)_{i,u} \in \mathbb{K}[x, y_0]_{<(d',2\bar{m})}$. Thus $S_i(x, b) \equiv G_i(x, b) \bmod h$.

Baby steps. Finally, we write $S_i = \sum_{\ell<2\bar{m}} S_{i,\ell}(x) y_0^\ell$ and split each power $b^\ell \operatorname{rem} h$ into $\lceil n/d' \rceil$ slices of degree less than d'. We multiply the $k \times 2\bar{m}$ matrix $[S_{i,\ell}]$ by the $2\bar{m} \times \lceil n/d' \rceil$ matrix of slices, recombine the slices, and reduce the k results modulo h.

Lemma 4.5. *Assume that h is monic of degree n, that $\mathcal{L}(R) = \mathcal{M}_m^{(b,h)}$ with R row reduced of degree at most d, and that $A_{c_1}(x, b) \equiv b^{c_1\bar{m}} \bmod h$ with $A_{c_1} \in \mathbb{K}[x, y_0]_{<(\bar{d},\bar{m})}$. Then* Evaluate *returns $g_i(b) \operatorname{rem} h$ for all i, using $\tilde{O}(n^{2/3}\mathsf{MM}(n^{1/3}, n^{1/3}, k) + kn)$ operations, by a linear straight-line program in the coefficients of the g_i.*

Theorem 4.6 (batched composition with two relation matrices)**.** *Let f be monic and separable of degree $n \geq 2$, $a \in \mathbb{K}[x]_{<n}$ a generator of $\mathcal{A}$, and $g_1, \ldots, g_k \in \mathbb{K}[y]_{<n}$. For every (r_1, r_2, γ), Algorithm 4.1 returns either $(g_i(a) \operatorname{rem} f)_i$ or* Fail*, using*

$$\tilde{O}(n^{(\omega+2)/3} + n^{2/3}\,\mathsf{MM}(n^{1/3}, n^{1/3}, k)) \tag{4.1}$$

operations in $\mathbb{K}$. If r_1, r_2 and the coefficients of γ are uniform and independent in a finite $S \subseteq \mathbb{K}$, then Fail *has probability at most $8n^2/\operatorname{card}(S)$, for every choice of the g_i; hence, for $\operatorname{card}(S) \geq 16n^2$, repetition gives a Las Vegas algorithm with the same expected cost. For $k \geq n^{1/3}$ the bound is $\tilde{O}(kn^{(\omega+1)/3})$, and if $\omega = 2$ it is $n^{o(1)}(n^{4/3} + nk)$. For $k = n^t$ the cost is $n^{\Lambda_2(t)+o(1)}$ with*

$$\Lambda_2(t) = \max\Big(\frac{\omega+2}{3},\ \frac{2}{3} + \omega(\tfrac{1}{3}, \tfrac{1}{3}, t)\Big). \tag{4.2}$$

Remark 4.7 (the single composition). For $k = 1$, Algorithm 4.1 costs $\tilde{O}(n^{(\omega+2)/3})$, which does not improve on [21]. The obstacle is the change of basis: at $m = \bar{m} = n^{1/4}$, the three-level evaluation of [22] leaves the truncated powers of [21, Algorithm 6.1] with cost $\tilde{O}(n^{1/4+3\omega_2/8}) = O(n^{1.4689})$, and even their optimized cost $\tilde{O}(n^{\kappa})$ exceeds $n^{(\omega+3)/4}$. Thus Proposition 4.2 alone does not give the generic single-composition exponent, while a batch pays for the change of basis only once. Section 6.1 accelerates the change of basis for generators, and Section 6.4 then handles every argument.

Remark 4.8 (why the second relation matrix helps a batch). The Nüsken–Ziegler evaluation in Algorithm 3.2 uses baby powers $b^{\ell} \operatorname{rem} h$ with n coefficients each, and its intermediate results have total size $k(d/B)n$. Algorithm 4.1 replaces the giant powers $b^{\bar{m}c_1}$ by the polynomials A_{c_1}, of x-degree $\bar{d}$ and size $\bar{d}\bar{m} = \Theta(n)$, so that the entries of the product have only $m\bar{m} = \Theta(n^{2/3})$ coefficients and each S_i has $\Theta(n)$. With three levels and $d > \bar{m}^2$, as in [22], the intermediate size becomes $\Theta(k(d/\bar{m}^2)n)$; in the cost model of Appendix A this gives a gain only for $k < n^{1/3}$ (Section 6.2), which is outside the range used in Theorem 1.1.

4.3 Large batches

For large batches, the reshaping method of Neiger, Rosenkilde and Solomatov evaluates each outer polynomial in quasi-linear time, once suitable relations are known [20]. For $I = \langle h, y - b\rangle$ and a reshaping sequence $\eta = (\eta_i)_{0 \leq i \leq \ell}$, that is, $\eta_{i-1} > \eta_i \geq \lfloor \frac{2}{3}\eta_{i-1} \rfloor$, an *$\eta$-reshaper* is a family $r_i = y^{\eta_i} - H_i(x, y) \in I$ with $\deg_y H_i < \mu_i := 2\eta_i - \eta_{i-1} + 1$; it is *balanced* if $\deg_x H_i \leq \lfloor n/\mu_i \rfloor + 1$. By [20, Thm. 6.1], if a balanced η-reshaper is known for an $(n, 1)$-reshaping sequence, that is, with $\eta_0 = n$ and $\eta_\ell = 1$, of length $O(\log n)$, then $g(b) \operatorname{rem} h$ is computed for every $g \in \mathbb{K}[y]_{<n}$ in $\tilde{O}(n)$ operations, by a linear straight-line program in the coefficients of g. Their construction of reshapers [20, Cor. 7.9] costs $\tilde{O}(n^{\omega})$ and its balancedness is proved for random arguments. We construct balanced reshapers for prescribed generators, using the relation module of each side according to the width μ_i.

Theorem 4.9 (large batches)**.** *Let $f \in \mathbb{K}[x]$ be monic and separable of degree $n \geq 2$ and let $a \in \mathbb{K}[x]_{<n}$ generate $\mathcal{A}$. There is a Las Vegas preprocessing using an expected $\tilde{O}(n^{(\omega+1)/2})$ operations in $\mathbb{K}$, depending only on (f, a) and random elements, after which $g(a) \operatorname{rem} f$ is computed for every $g \in \mathbb{K}[y]_{<n}$ in $\tilde{O}(n)$ operations, by a linear straight-line program. Hence k compositions with the argument a cost an expected $\tilde{O}(n^{(\omega+1)/2} + kn)$ operations, and k power projections of length n cost the same.*

Proof. Fix the $(n, 1)$-reshaping sequence $\eta_0 = n$, $\eta_i = \max(1, \lfloor 2\eta_{i-1}/3 \rfloor)$, stopped at the first $\eta_\ell = 1$; its length is $\ell = O(\log n)$ [20, §3]. Apply the shifts and the change of basis of Algorithm 4.1, at the dimension $\lceil n^{1/3} \rceil$, at cost $\tilde{O}(n^{(\omega+2)/3})$; then $g(a) \operatorname{rem} f$ is obtained from $u = g(\phi_\gamma) \operatorname{rem} \mu_\gamma$ and $u(\gamma) \operatorname{rem} \hat{f}$, as in Algorithm 4.1. For each of the two pairs $(b, h) \in \{(\phi_\gamma, \mu_\gamma), (\gamma, \hat{f})\}$ and each $i \leq \ell$, we compute H_i with $H_i(x, b) \equiv b^{\eta_i} \bmod h$, as follows.

If $\mu_i \le \sqrt{n}$, compute a minimal basis of $\mathcal{N}_{\mu_i}^{(b,h)}$, return Fail unless its degree is $\lceil n/\mu_i \rceil$, and reduce $b^{\eta_i} \operatorname{rem} h$ by it [22, Prop. 2.3]. This gives $H_i \in \mathbb{K}[x,y]_{<(\lceil n/\mu_i \rceil, \mu_i)}$, which satisfies the degree conditions since $\lceil n/\mu_i \rceil \le \lfloor n/\mu_i \rfloor + 1$, in $\tilde{O}(\mu_i^{\omega-1} n) \subseteq \tilde{O}(n^{(\omega+1)/2})$ operations.

If $\mu_i > \sqrt{n}$, put $D = \lceil n/\mu_i \rceil \le \sqrt{n} + 1$ and $d' = \lceil n/D \rceil \le \mu_i$. Compute a basis R of $\mathcal{M}_D^{(b,h)}$ of degree at most d' by CandidateBasis(h, b, D, d'), return Fail unless it returns Cert, convert it to Popov form, and return Fail unless its degree is at most d' (on Cert this always holds: the approximant basis of [21, Algorithm 5.1] has determinant degree $2Dd'$, its first D diagonal degrees add up to n, and the other D are at least $\deg R$, so that $n + D \deg R \le 2Dd'$ and $\deg R \le d'$). By Lemma 3.3(b) with block length 1, the remainder $(y^{\eta_i} e_1) \operatorname{Rem} R$ is computed from $O(\log n)$ products of $D \times D$ matrices of degree less than d', each of degree less than $2d'$, followed by divisions. Its D components have degree less than d', so that the corresponding polynomial H_i has $\deg_x H_i < D$ and $\deg_y H_i < d' \le \mu_i$, and $y^{\eta_i} - H_i$ is a relation because $\mathcal{L}(R) \subseteq \mathcal{M}_D^{(b,h)}$. The cost is $\tilde{O}(D^\omega d' + c(n, D, d'))$, where $D^\omega d' = O(D^{\omega-1} n) \subseteq O(n^{(\omega+1)/2})$ and $c(n, D, d') = O(D\,(n/D)^{\omega_2/2}) \subseteq O(n^{\omega_2/2}) \subseteq O(n^{(\omega+1)/2})$, since $\omega_2 \le \omega + 1$.

Every returned $r_i = y^{\eta_i} - H_i$ lies in $\langle h, y - b \rangle$ and satisfies the balancedness bounds, whether or not the random choices are good, so that every accepted output is correct by [20, Thm. 6.1]. The tests succeed when the conditions of Proposition 4.2 hold for the $2\ell + 2$ pairs of dimensions used, that is, (e) at the dimensions $\mu_i \le \sqrt{n}$ and (c), (d) at the dimensions (D, d'); by the proof of Proposition 4.2, the corresponding nonzero polynomials have total degree $O(n^2 \log n)$, so that Fail has probability $O(n^2 \log n / \operatorname{card}(S))$, and a set S of size $\Theta(n^2 \log n)$, with the extension-field convention of Theorem 6.1, gives an expected $O(1)$ attempts. The online phase consists of two reshaping evaluations and the shifts, in $\tilde{O}(n)$ operations; it is linear in g, and its transposition gives the power projections. □

For generating arguments, the batch curve is thus at most $\min\{\Lambda_4(t), \max(\frac{\omega+1}{2}, 1 + t)\}$, and $k = n$ compositions cost $\tilde{O}(n^2)$, against $n^{2.0834}$ with Theorem 4.6; for $k \ge n^{(\omega-1)/2}$ the cost is $\tilde{O}(nk)$, of the order of the output. The pair of exponents $(\frac{\omega+1}{2}, 1)$ for preprocessing and online evaluation also appears for multipoint evaluation at explicit point sets [30, 33]; here the algebra is given by f only, the argument is prescribed, and the preprocessing is certified. The factorization does not use these batches, whose preprocessing exceeds n^{e_2}.

5 Polynomial factorization

Throughout this section, $\mathbb{F}_q$ has characteristic p, and $\mathsf{M}(n) = n^{1+o(1)}$ is the cost of multiplying polynomials of degree less than n. The algorithm of Theorem 1.1 follows the three stages of Section 1.1: the square-free factorization, Algorithm D of Kaltofen and Shoup with giant steps computed by the batches of Theorem 4.6, and the equal-degree method of von zur Gathen and Shoup with the single compositions of [21]. Every argument of a composition is a Frobenius image, which generates the algebra, so that Proposition 4.2 applies, and the compositions of the equal-degree stage cost

$$n^{\kappa+o(1)}, \qquad \kappa \le (\omega + 2)/3 \le e_2,$$

which fits our total bound.

5.1 Extension of scalars

The batched algorithms draw random elements from a finite subset S of the base field and fail with probability $O(n^2 / \operatorname{card}(S))$, so that an extension of $\mathbb{F}_q$ is needed when q is not large compared with n^2. Indeed, if a square-free f splits completely over $\mathbb{F}_q$ and the coefficients of γ are uniform in $\mathbb{F}_q$, then the values of γ at the n roots are uniform and independent, and they are

Algorithm 5.1 BatchedGiantSteps(f, l)

Require: $f \in \mathbb{F}_q[x]$ monic square-free of degree n; $G_1 = h_l = x^{q^l} \operatorname{rem} f$ (from (D1)); $m_D = \lceil n/(2l) \rceil$
Ensure: $G_j = x^{q^{lj}} \operatorname{rem} f$ for $1 \le j \le m_D$
1: $s \leftarrow 1$
2: **while** $s < m_D$ **do**
3: $k \leftarrow \min(s, m_D - s)$
4: ▷ here $G_j = x^{q^{lj}} \operatorname{rem} f$ for $1 \le j \le s$
5: $(G_{s+1}, \ldots, G_{s+k}) \leftarrow (G_1(G_s) \operatorname{rem} f, \ldots, G_k(G_s) \operatorname{rem} f)$ by one call to a batched composition algorithm over $\mathbb{K}'$ with argument G_s, repeated with fresh random elements on Fail
6: $s \leftarrow s + k$
7: **end while**
8: **return** $(G_1, \ldots, G_{m_D})$

pairwise distinct, as the change of basis requires, with probability $\prod_{i<n}(1 - i/q) \le e^{-n(n-1)/(2q)}$, which tends to 0 when $n^2/q \to \infty$. An extension of degree $O(\log n)$ suffices.

Lemma 5.1. *Let $f \in \mathbb{F}_q[x]$ be monic and square-free of degree $n' \ge 2$, let $n \ge n'$ (in applications, the degree of the original input), let $c \ge 2$ be a fixed constant, $e = \max(1, \lceil \log_q(16n^c) \rceil)$ and $\mathbb{K}' = \mathbb{F}_{q^e}$. Then:*

(a) f is separable over $\mathbb{K}'$;

(b) for $a, g_1, \ldots, g_k \in \mathbb{F}_q[x]_{<n'}$, the remainders $g_i(a) \operatorname{rem} f$ computed in $\mathbb{K}'[x]$ belong to $\mathbb{F}_q[x]$ and coincide with the remainders computed in $\mathbb{F}_q[x]$; in particular, the extension only supplies random elements inside composition calls, and the Frobenius map remains the q-power map of $\mathbb{F}_q[x]/\langle f \rangle$;

(c) if a generates $\mathbb{F}_q[x]/\langle f \rangle$, then it generates $\mathbb{K}'[x]/\langle f \rangle$;

(d) running Algorithm 3.2, or, when a generates $\mathbb{F}_q[x]/\langle f \rangle$, Algorithm 4.1, over $\mathbb{K}'$ with $S = \mathbb{K}'$ costs the bound of Theorem 3.6, respectively Theorem 4.6, multiplied by $\tilde{O}(\log n)$, operations in $\mathbb{F}_q$, and fails with probability at most $8n'^2/q^e \le \frac{1}{2} n^{2-c}$; constructing $\mathbb{K}'$ costs $(\log n)^{O(1)} \log q$ operations, once for the whole computation.

Proof. (a) Since $\mathbb{F}_q$ is perfect, a square-free f satisfies $\gcd(f, f') = 1$: if an irreducible $\pi \in \mathbb{F}_q[x]$ divided f and f', then $f = \pi h$ with $\pi \nmid h$ gives $\pi \mid \pi' h$, hence $\pi' = 0$ and $\pi \in \mathbb{F}_q[x^p]$ would be a p-th power in $\mathbb{F}_q[x]$. A Bézout relation $uf + vf' = 1$ in $\mathbb{F}_q[x]$ remains valid in $\mathbb{K}'[x]$, so f has no multiple root in an algebraic closure. (b) Euclidean division by the monic polynomial f is unique in $\mathbb{K}'[x]$, and the remainder computed in $\mathbb{F}_q[x]$ satisfies its defining property in $\mathbb{K}'[x]$. (c) The powers $1, a, \ldots, a^{n'-1} \operatorname{rem} f$ form a basis of $\mathbb{F}_q[x]/\langle f \rangle$, i.e., the matrix of their coefficient vectors is invertible over $\mathbb{F}_q$. It remains invertible over $\mathbb{K}'$, so they form a basis of $\mathbb{K}'[x]/\langle f \rangle = \mathbb{K}' \otimes_{\mathbb{F}_q} \mathbb{F}_q[x]/\langle f \rangle$. (d) An irreducible polynomial of degree $e = O(\log n)$ over $\mathbb{F}_q$ is found with $(\log n)^{O(1)} \log q$ expected operations, and one operation of $\mathbb{K}'$ costs $\tilde{O}(e)$ operations of $\mathbb{F}_q$. The failure bounds are those of Theorem 3.6(ii) and Theorem 4.6, which apply by (a) and (c), with $\operatorname{card}(S) = q^e \ge 16n^c$. □

5.2 Distinct-degree factorization with batched giant steps

We keep the notation of Section 2.5: f is monic and square-free of degree n, $\mathcal{A} = \mathbb{F}_q[x]/\langle f \rangle$, σ is the Frobenius automorphism $u \mapsto u^q$ of $\mathcal{A}$, $l = \lceil n^\beta \rceil$ and $m_D = \lceil n/(2l) \rceil$. Steps (D1), (D3), (D4), (D5) of [16, Algorithm D] are kept unchanged. Step (D2) is computed as follows.

Lemma 5.2. *Algorithm 5.1 is correct. It performs at most* $\lceil \log_2 m_D \rceil$ *batched calls, each with an argument that generates* $\mathcal{A}$ *and a batch size* $k \le m_D/2+1$*. If* $C(n,k)$ *is the expected cost of a batch of size* k*, assumed nondecreasing in* k*, the expected cost of Algorithm 5.1 is* $\tilde{O}(C(n, m_D))$ *operations in* $\mathbb{F}_q$*.*

Proof. Correctness. If $G_i \equiv x^{q^{li}}$ and $G_s \equiv x^{q^{ls}}$ modulo f, then, since the coefficients of G_i lie in $\mathbb{F}_q$ and are fixed by $u \mapsto u^q$,

$$G_i(G_s) \equiv G_i(x^{q^{ls}}) = G_i(x)^{q^{ls}} \equiv x^{q^{l(i+s)}} \pmod f,$$

which proves that the invariant is preserved. The loop terminates with $s = m_D$ after at most $\lceil \log_2 m_D \rceil$ iterations, because s doubles until the last iteration.

Generators. The argument of each call is $G_s = \sigma^{ls}(x)$. Since σ is an automorphism of the $\mathbb{F}_q$-algebra $\mathcal{A}$ and x generates $\mathcal{A}$, so does $\sigma^{ls}(x)$: every $u \in \mathcal{A}$ is $u = \sigma^{ls}(\sigma^{-ls}(u)) = v(\sigma^{ls}(x))$ with $v \in \mathbb{F}_q[x]_{<n}$ the representative of $\sigma^{-ls}(u)$.

Cost. Each call has $k \le s \le m_D$ and $k \le m_D - s$, so $k \le m_D/2 + 1$. There are at most $\lceil \log_2 m_D \rceil$ calls, each of expected cost $C(n,k)\tilde{O}(\log n) \le C(n, m_D)\tilde{O}(\log n)$ by Lemma 5.1 and monotonicity, since a call that fails with probability at most $1/2$ is repeated at most twice on average. □

Proposition 5.3 (batched distinct-degree factorization)**.** *Let* Λ *be a batch curve, i.e., a nondecreasing function such that batches of* $k = \lceil n^t \rceil$ *compositions with one argument generating* $\mathcal{A}$ *can be computed by a Las Vegas algorithm in expected* $n^{\Lambda(t)+o(1)}$ *operations, with a failure probability at most* $1/2$ *when* $\operatorname{card}(S) \ge 16n^2$*. For every* $\beta \in [0,1]$*, the modified Algorithm D computes the distinct-degree factorization of a monic square-free* $f \in \mathbb{F}_q[x]$ *of degree* n *with an expected*

$$n^{1+\beta+o(1)} \log q + n^{2-\beta+o(1)} + n^{\Lambda(1-\beta)+o(1)} \tag{5.1}$$

operations in $\mathbb{F}_q$*.*

Proof. Correctness follows from Lemma 5.2 and [16, Lemma 1]. The costs of (D1) and (D3)–(D5) were recalled in Section 2.5. (D2) costs $n^{\Lambda(1-\beta)+o(1)}$ by Lemma 5.2, since $m_D \le n^{1-\beta}$. □

Corollary 5.4. *With the batch curve* Λ_2 *of Theorem 4.6 and* $\beta = (\omega+1)/6$*, the distinct-degree factorization of* f *costs an expected* $n^{(\omega+7)/6+o(1)} \log q$ *operations. With* Λ_2 *evaluated with the bound* $\omega(1,1,s) \le (2-s)\omega + (s-1)\omega_2$ *for* $s \in [1,2]$*, and* $\beta = (2\omega_2 - \omega - 1)/(3(\omega_2 - \omega + 1))$*, it costs* $n^{e_2(\omega,\omega_2)+o(1)} \log q$*, with* e_2 *as in Theorem 1.1, provided that* $s^* = (\omega_2 - 2\omega + 4)/(\omega_2 - \omega + 1) \in [1,2]$*.*

Proof. By Lemma 5.2 every argument generates $\mathcal{A}$, so Theorem 4.6 applies to every batch. With Lemma 5.1 its failure probability is at most $\frac{1}{2} n^{2-c} \le \frac{1}{2}$ (as $c \ge 2$), and its cost is nondecreasing in k. By Eq. (4.2) and homogeneity, $\Lambda_2(t) = \max\left(\frac{\omega+2}{3},\ \frac{2}{3} + \frac{1}{3}\omega(1,1,3t)\right)$.

Square blocks. For $t \ge 1/3$, $\omega(1,1,3t) \le \omega + 3t - 1$ (split the $n^{1/3} \times n^t$ factor into $n^{t-1/3}$ square blocks), so $\Lambda_2(t) \le t + (\omega+1)/3$, which is at least $(\omega+2)/3$. With $\beta = (\omega+1)/6$ and $t = 1-\beta = (5-\omega)/6 \ge 1/3$ (as $\omega \le 3$), the three terms of Eq. (5.1) have exponents $1+\beta = (\omega+7)/6$, $2-\beta = (11-\omega)/6 \le (\omega+7)/6$ (as $\omega \ge 2$), and $\Lambda_2(1-\beta) \le (1-\beta) + (\omega+1)/3 = (\omega+7)/6$.

Rectangular bound. Put $s = 3(1-\beta)$. Equating $1+\beta$ with $\frac{2}{3} + \frac{1}{3}((2-s)\omega + (s-1)\omega_2)$ gives $3 + 3\beta = 2 + (3\beta - 1)\omega + (2 - 3\beta)\omega_2$, that is, $3\beta(\omega_2 - \omega + 1) = 2\omega_2 - \omega - 1$. This is the stated β, and then $s = s^*$ and $1 + \beta = e_2(\omega, \omega_2)$; the condition $s^* \in [1,2]$ is the range in which the interpolation bound is valid. The remaining exponents are not larger, over the whole range: with $Q_\omega = \omega_2 - \omega + 1 > 0$ one checks the identities

$$\beta - \frac{1}{2} = \frac{\omega + \omega_2 - 5}{6Q_\omega}, \qquad e_2 - \frac{\omega+2}{3} = \frac{(3-\omega)(\omega_2 - \omega)}{3Q_\omega},$$

whose right-hand sides are nonnegative for $2 \le \omega \le 3$ and $\omega_2 \ge 3$ (and $\omega_2 \ge \omega$). Hence $2-\beta \le 1+\beta$, with equality at $(\omega,\omega_2)=(2,3)$, and $(\omega+2)/3 \le e_2$, with difference $1/6$ there. The identities $s^*-1=(3-\omega)/Q_\omega$ and $2-s^*=(\omega_2-2)/Q_\omega$ show that $s^* \in [1,2]$ whenever $\omega \le 3$. □

For the basic bounds $\omega < 2.371339$, $\omega_2 < 3.250035$ [1], Corollary 5.4 gives $\beta = 0.5551$, $s^* = 1.3346$, $k = n^{0.445}$ at the top level, and the exponents 1.5619 (square blocks) and $e_2 = 1.555125\ldots$ (rectangular). The same algorithm, analyzed with the full table, that is, with $\Lambda_2(1-\beta) \le \frac{2}{3} + \frac{1}{3}U(1,1,3(1-\beta))$ for the generators of Appendix A, has the exponent

$$e_2^{\text{full}} = \frac{24413604}{15719207} = 1.5531065\ldots < 1.5532,$$

at $\beta = 0.553107$, with $3(1-\beta) = 1.3407$ and top batch $k = n^{0.4469}$; here the active entries are $\omega(1,1,0.9)$ and $\omega(1,1,1.5)$, and the value is exact for the program by the certificate of Remark 5.8.

5.3 Equal-degree factorization

Let f' be a monic, square-free distinct-degree output of degree $n' = r'd'$, a product of r' irreducible polynomials of degree d'. We apply [37, Algorithm 3.6], in which the trace $\mathrm{Tr}(v) = \sum_{0 \le i < d'} v^{q^i}$ of a random $v \in \mathcal{A}' = \mathbb{F}_q[x]/\langle f' \rangle$ is computed by [37, Algorithm 5.2]. For $d' \ge 2$ and $\ell = \lfloor \log_2(d'-1) \rfloor$, Algorithm 5.2 performs at most $4\ell+2$ compositions $g(b) \operatorname{rem} f'$, each with an argument b of the form ξ^{q^j}, $\xi = x \bmod f'$. All other operations are multiplications, powerings by q and gcds [37, §§3, 5]. Since f' is square-free, every such argument generates $\mathcal{A}'$ (as in Lemma 5.2). We compute each composition with [21, Algorithm 8.1] over the field $\mathbb{K}'$ of Lemma 5.1; this suffices for Theorem 1.1, and Theorem 6.1 would only lower the cost of this stage.

Proposition 5.5 (equal-degree factorization)**.** *The irreducible factors of f' can be computed by a Las Vegas algorithm using an expected $n'^{\kappa+o(1)} + n'^{1+o(1)} \log q$ operations in $\mathbb{F}_q$, where $\kappa = 1 + 1/(1/(\omega-1) + 2/(\omega_2-2)) < 1.4294$. Summed over the distinct-degree outputs, the cost is $n^{\kappa+o(1)} + n^{1+o(1)} \log q$; with Theorem 6.1 instead of [21], it is $O(n^{1.3429} + n^{1+o(1)} \log q)$.*

Proof. Correctness is [37, Thm. 3.7, Lemma 5.3]. Since every composition returns the exact result or Fail, the algorithm is executed exactly as analysed there. The cases $d' = 1$ (the trace is v itself) and $r' = 1$ (nothing to split) are trivial, so we assume $d', r' \ge 2$. Let $\kappa' = \kappa$, or $\kappa' = (\omega+3)/4$ if Theorem 6.1 is used instead of [21]. Fix $0 < \varepsilon \le 3-\kappa'$ and let $C_T(t) = c_\varepsilon t^{\kappa'+\varepsilon}$ be a bound on the expected cost of one composition of degree t, including repetitions. For $a \ge 1$ it satisfies $aC_T(t) \le C_T(at) \le a^3 C_T(t)$, which is the hypothesis of [37, §5]. Then [37, Thm. 5.4] gives the expected cost $O(C_T(n') \log d' \log r' + \mathsf{M}(n') \log n' \log r' + \mathsf{M}(n') \log r' \log q)$; this bound holds for the expectation of the total, because the randomness of the compositions is independent of that of the splitting and the conditional expected cost of every composition is at most C_T. The sum over the outputs uses $\sum_i n_i'^{\kappa'+\varepsilon} \le n^{\kappa'+\varepsilon}$. □

Remark 5.6. The compositions of the equal-degree stage must also be replaced. Indeed, with Brent–Kung composition the same algorithm costs $n^{(\omega+1)/2+o(1)} + n^{1+o(1)} \log q$ [16, §2], and $(\omega+1)/2 > 1.6856$ exceeds the exponents of Theorem 1.1 and Corollary 5.9.

5.4 Proof of Theorem 1.1

Algorithm. (0) Divide f by its leading coefficient c, which is returned with the factorization. (1) Compute the square-free decomposition $f = f_1 f_2^2 \cdots f_n^n$ with f_i monic, square-free and pairwise coprime, using $O(\mathsf{M}(n) \log n + n \log q)$ operations [37, §3]. (2) For every $f_i \ne 1$, compute its

distinct-degree factorization by Proposition 5.3 with the batch curve Λ_2 and the parameter β of Corollary 5.4. (3) Factor every distinct-degree output by Proposition 5.5.

Cost. The functions $n \mapsto n^e$, $e \geq 1$, are superadditive, so the costs of (2) and (3) summed over the f_i are bounded by their values for a square-free polynomial of degree n. By Corollary 5.4 and Proposition 5.5 the expected total is

$$n^{(\omega+7)/6+o(1)} \log q + n^{\kappa+o(1)} + n^{1+o(1)} \log q,$$

and $\kappa \leq (\omega+2)/3 \leq (\omega+7)/6$ for feasible $2 \leq \omega \leq 3$, using $\min(\omega_2, \omega+1)$ in κ. The same holds with $e_2(\omega, \omega_2)$, since $e_2 \geq (\omega+2)/3$ by Corollary 5.4. If $\omega = 2$, then $\omega_2 = 3$ is feasible, $e_2(2,3) = 3/2$ and $\kappa = 4/3$.

Las Vegas property. Every composition call returns the correct result or Fail, and is repeated on Fail. The square-free and distinct-degree stages are otherwise deterministic, and the equal-degree stage is Las Vegas [37, Thm. 3.7]. Hence the output is always correct, and the bound holds for the expected cost. □

5.5 The choice of the parameter

The parameter β of Corollary 5.4 is the best one for Algorithm D with giant steps computed by Theorem 4.6, when rectangular products are charged by the bounds of Appendix A.

Proposition 5.7. *Let $\hat{\Lambda}_2(t) = \max\left(\frac{\omega+2}{3}, \frac{2}{3} + U(\frac{1}{3}, \frac{1}{3}, t)\right)$, where U is the bound Eq. (2.2) built from the generators of Appendix A, so that $\hat{\Lambda}_2$ is the cost of the giant steps of Algorithm D computed by Theorem 4.6. Let $\Delta_\omega = \omega_2 - \omega$, $c = \frac{2\omega-\omega_2}{2}$ and $A = \frac{2+2\omega-\omega_2}{3}$, and assume that $z = (c, c, \Delta_\omega)$ satisfies $z \cdot \pi(g_j) \leq w_j$ for every generator (g_j, w_j) and every permutation π. Then*

$$\min_{0 \leq \beta \leq 1} \max\left\{1+\beta,\ 2-\beta,\ \hat{\Lambda}_2(1-\beta)\right\} = \frac{A + 2\Delta_\omega}{1+\Delta_\omega} = e_2(\omega, \omega_2).$$

The hypothesis holds for the basic bounds of Section 2.1 and for $(\omega, \omega_2, \alpha) = (2, 3, 1)$.

Proof. The upper bound is Corollary 5.4. For the lower bound, the hypothesis says that z is feasible for the dual of the linear program Eq. (2.2), so that $U(v) \geq z \cdot v$ for every v by weak duality. With $v = (\frac{1}{3}, \frac{1}{3}, t)$ this gives $U(\frac{1}{3}, \frac{1}{3}, t) \geq \frac{2c}{3} + \Delta_\omega t$, hence $\hat{\Lambda}_2(t) \geq A + \Delta_\omega t$. Geometrically, z is tight on $(1,1,1)$ and $(1,1,2)$, so that the line through the two points that define the interpolation Eq. (2.1) supports the whole curve $s \mapsto U(1,1,s)$. For every β, the maximum is therefore at least $1+\beta$ and at least $A + \Delta_\omega(1-\beta)$, hence at least their convex combination with weights $\frac{\Delta_\omega}{1+\Delta_\omega}$ and $\frac{1}{1+\Delta_\omega}$, which does not depend on β:

$$\frac{\Delta_\omega(1+\beta) + A + \Delta_\omega(1-\beta)}{1+\Delta_\omega} = \frac{A + 2\Delta_\omega}{1+\Delta_\omega} = \frac{5\omega_2 - 4\omega + 2}{3(\omega_2 - \omega + 1)}.$$

□

For the basic bounds of Section 2.1, we have $0 \leq c \leq \Delta_\omega \leq 1$, so that the largest value of $z \cdot \pi(g)$ over the permutations π puts the largest entries of g on the coordinate Δ_ω. The inequality is then an equality for $(1,1,1)$ and $(1,1,2)$, and it is strict for $(1,0,0)$, $(1,1,0)$ and $(1,1,\alpha)$, whose values Δ_ω, $c + \Delta_\omega$ and $\Delta_\omega + (1+\alpha)c$ are less than 0.88, 1.63 and 1.87.

Remark 5.8 (the full table). With the full table, $z = (c, c, \delta)$ with $c = \frac{1292557}{1700000}$ and $\delta = \frac{7219207}{8500000}$ satisfies $z \cdot \pi(g_j) \leq w_j$ for all 46 permuted generators, with equality exactly at $(0.9, 1, 1)$, $(1, 0.9, 1)$ and $(1, 1, 1.5)$. The proof of Proposition 5.7, with $A' = \frac{2}{3} + \frac{2c}{3}$ and δ in place of A and Δ_ω, gives the lower bound $(A' + 2\delta)/(1+\delta) = 24413604/15719207$ for the program, which is attained. Thus e_2^{full} is optimal for Algorithm D with two relation matrices and the full table, in the same sense as Proposition 5.7.

Corollary 5.9 (one relation matrix)**.** *With the batch curve Λ_1 of Theorem 3.6, which needs no assumption on the argument, the distinct-degree factorization, and the whole factorization, cost an expected $n^{3\omega/(\omega+2)+o(1)}\log q$ operations, and $O(n^{1.6214}\log q)$ with rectangular products and the full table (Appendix A).*

Proof. By Proposition B.3, for $t \le 1/2$ square blocks give $\Lambda_1(t) = \frac{\omega+2}{3} + \frac{\omega-1}{3}t$. Equating with $1+\beta$ at $t = 1-\beta$ gives $\beta^* = 2(\omega-1)/(\omega+2)$, which lies in $[1/2, 1)$ for $2 \le \omega < 4$, and $1+\beta^* = 3\omega/(\omega+2)$. Moreover $2-\beta^* = 6/(\omega+2) \le 3\omega/(\omega+2)$. The rectangular value is the optimum of the program of Appendix A with Λ_1 (at $\beta = 0.6213$, $\eta = 0.4531$, $\varrho = 0.4455$). The other stages are bounded as in the proof of Theorem 1.1. □

Remark 5.10 (scope). Theorem 1.1 holds over every finite field. For $\log q = n^{o(1)}$ and $p = n^{\rho_p}$, the algebraic routes for small characteristic derived from [26, 17, 4] are faster than e_2 only when $\rho_p < 0.0819$, even under the assumptions most favourable to them; the derivation is in the supplementary material. In particular, for $p \ge n^{1/12}$ our bound also improves on these routes.

For $\log q = n^{\rho+o(1)}$, optimize the additive bound Eq. (5.1) in β. With the envelope of the basic rectangular bounds, $A = (2+2\omega-\omega_2)/3$ and $\Delta_\omega = \omega_2 - \omega$, the exponent is

$$e(\rho) = \begin{cases} e_2 + \dfrac{\Delta_\omega}{1+\Delta_\omega}\rho, & 0 \le \rho \le (\omega_2-2)/3, \\ 1 + \dfrac{\omega_2}{6} + \dfrac{\rho}{2}, & (\omega_2-2)/3 \le \rho \le \omega_2/3, \\ 1+\rho, & \rho \ge \omega_2/3. \end{cases}$$

This exponent is smaller than that obtained with independent compositions by [21] for $\rho < \kappa$, and equal to it for $\rho \ge \kappa$, while the bound of von zur Gathen and Shoup is better for $2-\omega_2/3 < \rho < \omega_2/3$. Write e_{basic} for this exponent. With the full table, the exponent is

$$e_{\text{full}}(\rho) = \begin{cases} \frac{24413604}{15719207} + \frac{7219207}{15719207}\rho, & 0 \le \rho \le 0.098211, \\ \frac{2964417}{1910804} + \frac{227701}{477701}\rho, & 0.098211 \le \rho \le (\omega_2-2)/3, \\ 1 + \dfrac{\omega_2}{6} + \dfrac{\rho}{2}, & (\omega_2-2)/3 \le \rho \le \omega_2/3, \\ 1+\rho, & \rho \ge \omega_2/3, \end{cases}$$

where the first slope is $\delta/(1+\delta)$ with δ as in Remark 5.8 and the second comes from the segment of $U(1,1,\cdot)$ between 1.5 and 2. The derivation and the comparison with the bit model are in the supplementary material; the threshold 0.0819 above is computed with the basic bounds for all routes.

5.6 Large fields

When $\log q$ is large, the baby steps $h_i = x^{q^i} \bmod f$ of (D1), computed by repeated q-th powering, dominate. They can also be computed by batched compositions: since $h_{s+i} = h_i(h_s) \bmod f$, the doubling of Algorithm 5.1 applies to them with the arguments h_s, after a single powering $h_1 = x^q \bmod f$. Kaltofen and Shoup already compute baby and giant steps by modular composition after one powering [16, Lemma 4 and Section 5]; what changes here is the cost of the batches. This removes the factor $\log q$ from all steps but one.

Corollary 5.11 (large fields)**.** *There is a Las Vegas algorithm which, given $f \in \mathbb{F}_q[x]$ of degree n, returns its factorization into irreducible polynomials using an expected*

$$n^{\Lambda_2(1/2)+o(1)} + n^{1+o(1)}\log q, \qquad \Lambda_2(1/2) \le \frac{2}{3} + \frac{1}{3}\,\omega(1,1,\tfrac{3}{2}) \le \frac{2}{3} + \frac{\omega+\omega_2}{6},$$

operations in $\mathbb{F}_q$. With the full table this is $O(n^{1.5983}) + n^{1+o(1)}\log q$, with the basic bounds $O(n^{1.6036}) + n^{1+o(1)}\log q$, and it is $n^{3/2+o(1)} + n^{1+o(1)}\log q$ if $\omega = 2$. In particular, for $\log q =$

$n^{\rho+o(1)}$ *and* $H = \hat{\Lambda}_2(1/2)$*, the exponent is at most* $\min\{e(\rho), \max(H, 1+\rho)\}$*, where* $e = e_{\text{full}}$ *or* $e = e_{\text{basic}}$ *of Remark 5.10, with* H *computed from the same data. This is smaller than* $e(\rho)$ *for* $H - \frac{3}{2} < \rho < \omega_2/3$*, where* $H - \frac{3}{2} < 0.0983$ *with the full table and* $H - \frac{3}{2} < 0.1036$ *with the basic bounds, equal to it for* $\rho \geq \omega_2/3$*, and never larger than the exponent* $\max\{2, 1+\rho\}$ *of von zur Gathen and Shoup.*

Proof. We modify only the baby steps in the algorithm of Section 5.4, applied with $\beta = 1/2$. Let f be a square-free part of degree N, put $l = \lceil N^{1/2} \rceil$, and write the costs below with N in place of n; since $\sum N \leq n$ and the exponents are at least 1, the sum over the parts is bounded by the same expressions in n. We compute $h_1 = x^q \operatorname{rem} f$ by repeated squaring, with $N^{1+o(1)} \log q$ operations, and then $h_1, \dots, h_l$ by doubling:once $h_1, \dots, h_s$ are known, one batch with the argument h_s gives $h_{s+i} = h_i(h_s) \operatorname{rem} f$ for $i \leq \min(s, l-s)$. This identity holds because the coefficients of h_i lie in $\mathbb{F}_q$, exactly as in the proof of Lemma 5.2, and every argument h_s is a Frobenius image of x, hence a generator. By Lemma 5.2 applied to these batches, the baby steps cost $N^{\Lambda_2(1/2)+o(1)}$ besides the powering, and the giant steps, with batches of size at most $m_D/2 + 1 = O(N^{1/2})$, cost the same. Steps (D3)–(D5) cost $N^{3/2+o(1)}$, and the square-free and equal-degree stages cost $n^{\kappa+o(1)} + n^{1+o(1)} \log q$ by Proposition 5.5. Since $3/2$ and κ are at most $\Lambda_2(1/2)$, this proves the first bound. By Eq. (4.2) and Eq. (2.1) with $s = 3/2$,

$$\Lambda_2(1/2) = \max\Big(\frac{\omega+2}{3},\ \frac{2}{3} + \frac{1}{3}\,\omega(1, 1, \tfrac{3}{2})\Big) \leq \frac{2}{3} + \frac{\omega+\omega_2}{6},$$

since $\omega_2 \geq \omega$; with the basic bounds this is $1.6035623\ldots$, with the full table, where $(1, 1, 1.5)$ is a generator, $\frac{2}{3} + \frac{2.794633}{3} = 1.598211$, and it is $3/2$ for $(\omega, \omega_2) = (2, 3)$. For the basic bounds, the comparison with $e(\rho)$ follows from the three branches of Remark 5.10: on the first branch $e(\rho) = e_2 + \frac{\Delta_\omega}{1+\Delta_\omega}\rho$ exceeds H exactly for $\rho > (H - e_2)(1 + \Delta_\omega)/\Delta_\omega$, and this threshold equals $H - \frac{3}{2} = 0.1035623\ldots$ since $H = A + \Delta_\omega/2$ and $e_2 = H - \frac{\Delta_\omega}{1+\Delta_\omega}(H - \frac{3}{2})$; on the second, $e(\rho) \geq e((\omega_2 - 2)/3) > 1.75 > H$ and $e(\rho) > 1 + \rho$ for $\rho < \omega_2/3$; on the third, both equal $1 + \rho$. For the full table, $H = A' + \delta/2$ with the certificate of Remark 5.8, and the same identities show that the first branch of e_{full} meets H at $\rho = H - \frac{3}{2} = 0.098211$; the later branches are nondecreasing, exceed $1 + \rho$ for $\rho < \omega_2/3$, and equal $1 + \rho$ from $\rho = \omega_2/3$ on, where the input-powering term forces $\beta = 0$ and $\hat{\Lambda}_2(1) = 1 + \omega_2/3$. Finally, $\max(\Lambda_2(1/2), 1 + \rho) \leq \max(2, 1 + \rho)$ because $\Lambda_2(1/2) < 2$. □

Within this arrangement, $\Lambda_2(1/2)$ cannot be lowered by another choice of β: with baby and giant steps both computed by batched doubling, the charged exponent is at least $\max(\Lambda(\beta), \Lambda(1-\beta))$ for the batch curve Λ, and Λ is nondecreasing. Moreover, the dual vector of Proposition 5.7 gives $\hat{\Lambda}_2(1/2) \geq A + \Delta_\omega/2 = \frac{2}{3} + \frac{\omega+\omega_2}{6}$ for the basic bounds, and that of Remark 5.8 gives $\hat{\Lambda}_2(1/2) \geq A' + \delta/2 = 1.598211$ for the full table, so that the bound of Corollary 5.11 is exact in that cost model. This is not a lower bound for other algorithms. A hybrid that computes the small baby steps by powering and the others by doubling does not help either: minimizing $\max\{\min(1 + \beta + \rho, \Lambda(\beta)), \Lambda(1 - \beta), 2 - \beta, 1 + \rho\}$ over β gives exactly the smaller of the two exponents of Corollary 5.11.

6 Every argument

Throughout this section, f is separable; separability is used in the evaluation lemma (Lemma 4.1), in the coprimality of the factors computed in Section 6.4 and in counting fibers. The proofs are in Appendix C for Section 6.1 and in Appendix D for Section 6.4.

6.1 A faster change of basis and single composition

For a single composition, the change of basis of Table 4 costs $\tilde{O}(n^\kappa)$, which is larger than $n^{(\omega+3)/4}$ (Remark 4.7). Its bottleneck is the computation of the truncated sequences

$$[u\gamma^{-k-1} \operatorname{rem} f]_{<m}$$

and of the block powers of γ^{-1}. We compute them instead by the method of [22, Prop. 4.1], which the algorithm of that paper applies to $a^{-1} \bmod f$; the batched algorithms of Section 3 did not need it, because there the truncated powers [21, Algorithm 3.8], of cost $c(n, m, d)$, are paid once per batch. This method accepts an arbitrary multiplier u, but it requires the base γ^{-1} to be in good position. This follows from Proposition 4.2, because $r(x, y) \mapsto y^{\bar{m}-1} r(x, 1/y)$ is a degree-preserving isomorphism of $\mathbb{K}[x]$-modules from $\mathcal{N}_{\bar{m}}^{(\gamma,f)}$ onto $\mathcal{N}_{\bar{m}}^{(\gamma^{-1},f)}$ (Lemma 6.2). Keeping the certified reconstruction and using the dimension $m \approx n^{1/4}$, we obtain the generic cost for the powers, for the change of basis and for both subsequent compositions.

Theorem 6.1. *Let $\mathbb{K}$ be a field, let $f \in \mathbb{K}[x]$ be monic and separable of degree $n \geq 2$, and let $a \in \mathbb{K}[x]_{<n}$ generate $\mathbb{K}[x]/\langle f \rangle$. Let $g \in \mathbb{K}[y]_{<n}$. There is an algorithm which, given two elements $r_1, r_2 \in \mathbb{K}$ and a polynomial $\gamma \in \mathbb{K}[x]_{<n}$, returns either $g(a) \operatorname{rem} f$ or Fail, and uses*

$$\tilde{O}(n^{(\omega+3)/4})$$

operations in $\mathbb{K}$ on every execution. If r_1, r_2 and the n coefficients of γ are independent and uniform in a nonempty finite set $S \subseteq \mathbb{K}$, then

$$\Pr[\mathsf{Fail}] \leq \min\{1, 8n^2/\operatorname{card}(S)\}.$$

Consequently, when $\operatorname{card}(S) \geq 16n^2$, repetition gives a Las Vegas algorithm with the same expected complexity. Over $\mathbb{F}_q$, the expected bound in operations in $\mathbb{F}_q$ holds for every q, using a finite extension when necessary, as specified at the end of the proof.

We write $[u]_{<w} = u \operatorname{rem} x^w$ for coefficient truncation, always taken after reduction modulo the indicated modulus. The parameters used below are

$$m = \mu = \left\lceil (4n)^{1/4} \right\rceil, \qquad d = \lceil n/m \rceil, \qquad \ell^* = 2d + 2. \tag{6.1}$$

They satisfy $m = \Theta(n^{1/4})$, $d = \Theta(n^{3/4})$, $m \leq n$ and $\ell^* \leq m^3$, since $m^4 \geq 4n$ and $m \geq 2$ give $2d + 2 \leq 2n/m + 4 \leq m^3/2 + 4 \leq m^3$. We handle $n < 16$ directly; for $n \geq 16$ we have $3 \leq m \leq \sqrt{n}$, whence $d + 2m - 1 \leq n/3 + 2\sqrt{n} \leq n$ and $2d \leq n + 1$, so that every call satisfies the conditions of Lemma 6.3 with width at most $2m - 1$ and auxiliary degree d.

Lemma 6.2 (Inversion of the auxiliary relations)**.** *Let $h \in \mathbb{K}[x]$ have degree n and let $b \in \mathbb{K}[x]_{<n}$ be a unit modulo h. For every $1 \leq \mu \leq n$, the minimal bases of $\mathcal{N}_\mu^{(b,h)}$ and $\mathcal{N}_\mu^{(b^{-1},h)}$ have the same degree.*

Lemma 6.3 (Exact truncated and block powers)**.** *Let $h \in \mathbb{K}[x]$ have degree n, let $b \in \mathbb{K}[x]_{<n}$, and suppose that a minimal basis of $\mathcal{N}_\mu^{(b,h)}$ has degree δ. Assume $2\delta \leq n + 1$. For $1 \leq \mu \leq n$, the data of [22, Corollary 2.4] can be computed in $\tilde{O}(\mu^{\omega-1} n)$ operations. Given these data, for any multiplier $c \in \mathbb{K}[x]_{<n}$, positive width w with $w + \delta \leq n$, and $1 \leq \ell \leq \mu^3$, one can compute*

$$[cb^k \operatorname{rem} h]_{<w}, \qquad 0 \leq k < \ell,$$

in $\tilde{O}(\mu^\omega(\delta + w\mu))$ operations.

If $h(0) \neq 0$ and $2m - 1 + \delta \leq n$, the same data give all

$$\Gamma_{i,k} = [x^i b^k \operatorname{rem} h]_{<m}, \qquad 0 \leq i < m, \quad 0 \leq k < \ell,$$

in $\tilde{O}(\mu^\omega(\delta + (2m-1)\mu)) + O(m^2\ell)$ operations.

Algorithm 6.1 FastChangeOfBasis(h, γ, u) (Lemma 6.4), $n \geq 16$

Require: $h \in \mathbb{K}[x]$ monic of degree n, $h(0) \neq 0$; $\gamma, u \in \mathbb{K}[x]_{<n}$ arbitrary; $m = \lceil (4n)^{1/4} \rceil$, $d = \lceil n/m \rceil$

Ensure: Fail, or (R, χ, ϕ) with R a basis of $\mathcal{M}_m^{(\gamma,h)}$ of degree $\leq d$, χ the minimal polynomial of $\gamma \bmod h$, $\deg \chi = n$, and $\phi(\gamma) \equiv u \bmod h$

1: if $\gcd(\gamma, h) \neq 1$, **return** Fail; $b \leftarrow \gamma^{-1} \operatorname{rem} h$
2: compute a minimal basis N_b of $\mathcal{N}_m^{(b,h)}$; if $\deg N_b \neq d$, **return** Fail; compute its A_j, B_j
3: $r_k \leftarrow [u\gamma^{-k-1} \operatorname{rem} h]_{<m}$ for $k < 2d$, with the multiplier $bu \operatorname{rem} h$
4: $\Gamma_{i,k} \leftarrow [x^i b^k \operatorname{rem} h]_{<m}$ for $i < m$, $k < 2d+2$
5: run Steps 4–7 of [21, Algorithm 6.1] on the arrays built from the r_k and $\Gamma_{i,k}$, or **return** Fail
6: if $\deg R > d$, **return** Fail; **return** (R, χ, ϕ)

Lemma 6.4 (Certified fast change of basis)**.** *Let $h \in \mathbb{K}[x]$ be monic of degree $n \geq 2$ with $h(0) \neq 0$, and let $\gamma, u \in \mathbb{K}[x]_{<n}$. With the parameters (6.1), a modification of [21, Algorithm 6.1] returns either Fail or (R, χ, ϕ), where R is a basis of $\mathcal{M}_m^{(\gamma,h)}$ of degree at most d, χ is the minimal polynomial of $\gamma \bmod h$ and has degree n, and $\phi \in \mathbb{K}[y]_{<n}$ satisfies $\phi(\gamma) = u \bmod h$. Its cost on every execution is $\tilde{O}(n^{(\omega+3)/4})$. It succeeds if γ is a unit and a generator, the minimal bases of $\mathcal{N}_m^{(\gamma,h)}$ have degree d, and $\operatorname{rank} \mathrm{Hk}_{m,d}^{(\gamma,h)} = n$.*

Lemma 6.5 (Certified composition at the same cost)**.** *Let $h \in \mathbb{K}[x]$ be monic of degree $n \geq 2$ with $h(0) \neq 0$, let $b \in \mathbb{K}[x]_{<n}$ be a unit modulo h, and let $v \in \mathbb{K}[y]_{<n}$. There is a procedure of cost $\tilde{O}(n^{(\omega+3)/4})$ on every execution which returns either $v(b) \operatorname{rem} h$ or Fail. It succeeds if the minimal bases of $\mathcal{N}_m^{(b,h)}$ have degree d and $\operatorname{rank} \mathrm{Hk}_{m,d}^{(b,h)} = n$. A certified basis of $\mathcal{M}_m^{(b,h)}$ of degree at most d may instead be supplied.*

Lemma 6.6 (Simultaneous good position)**.** *Under the hypotheses of Theorem 6.1, set $h = f(x + r_2)$ and $u = (a + r_1)(x + r_2)$. With probability at least $1 - 8n^2/\operatorname{card}(S)$, the following hold: $h(0) \neq 0$; u is a unit modulo h; γ is a unit and a generator modulo h; and, writing χ for its minimal polynomial and $\phi(\gamma) = u \bmod h$, both pairs (γ, h) and (ϕ, χ) have full rank $\mathrm{Hk}_{m,d}$ and minimal $\mathcal{N}_m$-basis degree d. The same auxiliary degree conditions hold for their inverses.*

Remark 6.7. The change-of-basis lemma permits an arbitrary input element u, but Theorem 6.1 needs the generating hypothesis, which puts the subsequent pair (ϕ, χ) in good position. The Las Vegas question posed after Theorem 1.3 of [22] for arbitrary inputs is answered, for separable f, in Section 6.4.

6.2 Small batches

For one composition, Theorem 6.1 uses dimension $n^{1/4}$ and three evaluation levels, at cost $n^{(\omega+3)/4}$; for $k \leq n^{1/3}$ compositions, Theorem 4.6 uses dimension $n^{1/3}$ and two levels, at cost $n^{(\omega+2)/3}$. Between the two, the dimension can grow with the batch size. We first note that the lemmas of Section 6.1 do not depend on the particular dimension $\lceil (4n)^{1/4} \rceil$.

Lemma 6.8 (other dimensions)**.** *Let $n \geq 16$ and let m be an integer with $(4n)^{1/4} \leq m \leq n^{1/3}+1$; put $\mu = m$, $d = \lceil n/m \rceil$ and $\ell^* = 2d+2$. Then Lemmas 6.3, 6.4, and 6.6 and the computation of a certified basis in Lemma 6.5 hold with these parameters, with $\tilde{O}(m^\omega d)$ in place of $\tilde{O}(n^{(\omega+3)/4})$.*

Proof. The proofs in Appendix C use the parameters only through the inequalities $\ell^* \leq m^3$, $d + 2m - 1 \leq n$ and $2d \leq n + 1$, and through the cost terms $m^{\omega-1}n$, $m^\omega(d + m^2)$, $m^2\ell^*$, $m^\omega d$ and nm. The first inequality follows from $m^4 \geq 4n$, as in Section 6.1, and the other two from $3 \leq m \leq \sqrt{n}$, which holds for $n \geq 16$. Since $m \leq 2n^{1/3}$ and $d \geq n/m$, we have $m^2 \leq 8d$, so that

every cost term is $O(m^\omega d)$. The probability bound of Lemma 6.6 is a sum of degrees at most n, $n(n-1)/2$, $2n^2/m$, $2n^2$, $n(d-1)$ and $n(m-1)$; here $2n^2/m$, $n(d-1)$ and $n(m-1)$ are at most n^2 each, so that the sum is still at most $\frac{13}{2}n^2 + \frac{1}{2}n$. □

The evaluation now uses three levels, as in [22, Prop. 3.1], with the middle level shared by the whole batch. Let $h \in \mathbb{K}[x]$ be monic of degree n, $b \in \mathbb{K}[x]_{<n}$ and R a row-reduced basis of $\mathcal{M}_m^{(b,h)}$ of degree at most d, and suppose that the minimal bases of $\mathcal{N}_m^{(b,h)}$ have degree d, with the data $A_j, B_j \in \mathbb{K}[x,y]_{<(d,m)}$ of [22, Cor. 2.4], which satisfy $A_j(x,b) \equiv b^{jm}$ and $B_j(x,b) \equiv b^{jm^2}$ modulo h for $j < m$. Suppose also that the remainder table $P_j^{(R)}$, $j < \lceil n/d \rceil$, with $D = d$, the division data of Lemma 2.2(a) for R with $\delta = 2d$, and the powers $b^l \operatorname{rem} h$, $l < 3m$, are known. Put $\nu = \lceil d/m^2 \rceil \le m$. The subroutine Evaluate3$((g_i)_{i\le k}, b, h, R)$ proceeds as follows.

Reduction. Compute the polynomials $G_i \in \mathbb{K}[x,y]_{<(m,d)}$ of $g_i e_1 \operatorname{Rem} R$ by Lemma 3.4, so that $G_i(x,b) \equiv g_i(b) \bmod h$.

Middle level. Write the y-exponents of $G_i = \sum_c G_{i,[c]}(x) y^c$ as $c = c_0 + mc_1 + m^2 c_2$ with $c_0, c_1 < m$ and $c_2 < \nu$, and put

$$s_{i,c_2,c_1}(x,y_0) = \sum_{c_0<m} G_{i,[c_0+mc_1+m^2c_2]}(x)\, y_0^{c_0} \in \mathbb{K}[x,y_0]_{<(m,m)}.$$

Split $A_{c_1} = \sum_{u<\lceil d/m \rceil} A_{c_1,u} x^{um}$ with $A_{c_1,u} \in \mathbb{K}[x,y_0]_{<(m,m)}$, multiply the $k\nu \times m$ matrix $[s_{i,c_2,c_1}]$, with rows indexed by (i,c_2), by the $m \times \lceil d/m \rceil$ matrix $[A_{c_1,u}]$, and recombine the rows into $S_{i,c_2} = \sum_u x^{um} (\cdot)_{(i,c_2),u} \in \mathbb{K}[x,y_0]_{<(d+m,2m)}$.

Top level. Compute $V_i = \sum_{c_2<\nu} S_{i,c_2}(x,y_0)\, B_{c_2}(x,y_0) \in \mathbb{K}[x,y_0]_{<(2d+m,3m)}$.

Baby steps. Write $V_i = \sum_{l<3m} V_{i,l}(x) y_0^l$, split each $b^l \operatorname{rem} h$ into slices of degree less than $d'' = 2d+m$, multiply the $k \times 3m$ matrix $[V_{i,l}]$ by the matrix of slices, recombine and reduce the k results modulo h.

Lemma 6.9. *Under these hypotheses, Evaluate3 returns $g_i(b) \operatorname{rem} h$ for all i, using, besides the given data,*

$$\tilde{O}(d\,\mathsf{MM}(m,m,k) + m^2\mathsf{MM}(k\nu,m,\lceil d/m \rceil) + k\nu n + d\,\mathsf{MM}(k,m,m) + kn)$$

operations, by a linear straight-line program in the coefficients of the g_i.

Proof. By the choice of the exponents,

$$\begin{aligned} G_i(x,b) &= \sum_{c_2<\nu} b^{m^2c_2} \sum_{c_1<m} b^{mc_1} s_{i,c_2,c_1}(x,b) \\ &\equiv \sum_{c_2<\nu} B_{c_2}(x,b)\, S_{i,c_2}(x,b) = V_i(x,b) \pmod h, \end{aligned}$$

since $S_{i,c_2}(x,y_0) = \sum_{c_1} s_{i,c_2,c_1}(x,y_0) A_{c_1}(x,y_0)$ and $A_{c_1}(x,b) \equiv b^{mc_1}$; the baby steps then compute $V_i(x,b) \operatorname{rem} h$. The reduction costs $\tilde{O}(d\,\mathsf{MM}(m,m,k))$ by Lemma 3.4, with $J = \lceil n/d \rceil \le m$ and $D = d$. The middle level is a product with entries of size m^2, of cost $\tilde{O}(m^2\mathsf{MM}(k\nu,m,\lceil d/m \rceil))$, and the recombination costs $\tilde{O}(k\nu dm) = \tilde{O}(k\nu n)$. The top level consists of $k\nu$ products of bivariate polynomials of size $O(dm) = O(n)$, at cost $\tilde{O}(k\nu n)$ by Kronecker substitution. In the baby steps, $\lceil n/d'' \rceil \le m$, so that the product costs $\tilde{O}(d\,\mathsf{MM}(k,m,m))$, and the recombination and the reductions cost $\tilde{O}(kn)$. Every step multiplies data derived from the g_i by fixed data, which gives the linearity. □

Theorem 6.10 (small batches)**.** *Let $f \in \mathbb{K}[x]$ be monic and separable of degree $n \geq 16$, let $a \in \mathbb{K}[x]_{<n}$ generate $\mathcal{A}$, and let $g_1, \ldots, g_k \in \mathbb{K}[y]_{<n}$. For every integer m with $(4n)^{1/4} \leq m \leq n^{1/3} + 1$, with $d = \lceil n/m \rceil$ and $\nu = \lceil d/m^2 \rceil$, there is an algorithm which, given $r_1, r_2 \in \mathbb{K}$ and $\gamma \in \mathbb{K}[x]_{<n}$, returns either all $g_i(a) \operatorname{rem} f$ or* Fail*, using*

$$\tilde{O}(m^\omega d + d\,\mathsf{MM}(m,m,k) + m^2\mathsf{MM}(k\nu, m, \lceil d/m \rceil) + k\nu n + d\,\mathsf{MM}(k,m,m) + kn)$$

operations in $\mathbb{K}$ on every execution. If r_1, r_2 and the coefficients of γ are uniform and independent in a finite $S \subseteq \mathbb{K}$, then Fail *has probability at most $8n^2/\operatorname{card}(S)$, for every choice of the g_i. For $1 \leq k \leq n^{1/3}/4$ and $m = \lceil (4nk)^{1/4} \rceil$, square blocks give the cost*

$$\tilde{O}(n^{(\omega+3)/4} k^{(\omega-1)/4}),$$

which is $n^{o(1)}(n^{5/4}k^{1/4})$ if $\omega = 2$. For $k = n^t$ with $t \geq 0$, the cost is $n^{\Lambda_3(t)+o(1)}$, where

$$\Lambda_3(t) = \min_{1/4 \leq s \leq 1/3} \max\Big\{1 - s + \omega(s,s,s),\ 1 - s + \omega(s,s,t),\ 2s + \omega(t+1-3s, s, 1-2s),\ 2 + t - 3s,\ 1 + t\Big\}.$$

Proof. The algorithm is Algorithm 4.1 with three changes: the dimensions are $m = \bar{m}$ as given, the change of basis is that of Lemma 6.4, the certified basis for $(\phi_\gamma, \mu_\gamma)$ is computed as in Lemma 6.5, both with the parameters of Lemma 6.8, and Evaluate is replaced by Evaluate3. The data B_j, $j < \nu$, are available since $\nu \leq m$, that is, $d \leq m^3$. Correctness follows from Eq. (2.3) and Lemma 6.9, the failure probability from Lemma 6.8 applied to Lemma 6.6, and the precomputation, which also comprises the auxiliary bases, the Popov forms, the division data, the remainder tables and the powers $b^l \operatorname{rem} h$ for $l < 3m$, costs $\tilde{O}(m^\omega d + mn) = \tilde{O}(m^\omega d)$ by Lemma 6.8. The online cost is that of Lemma 6.9, applied twice.

For the square-block bound, let $m = \lceil (4nk)^{1/4} \rceil$; then $m \leq n^{1/3} + 1$ because $4k \leq n^{1/3}$, and $k \leq m$ because $k^3 \leq 4n$. Up to constant factors, $m^\omega d = m^{\omega-1}n = n^{(\omega+3)/4}k^{(\omega-1)/4}$; $d\,\mathsf{MM}(m,m,k)$ and $d\,\mathsf{MM}(k,m,m)$ are at most dm^ω; $k\nu \leq kn/m^3 + 2k = O(m)$ and $\lceil d/m \rceil = \Omega(m)$, so that $m^2\mathsf{MM}(k\nu, m, \lceil d/m \rceil) = O(m^2 \cdot (n/m^3)\, m^\omega) = O(m^{\omega-1}n)$; and $k\nu n + kn = O(mn)$. For the exponent Λ_3, put $m = n^s$; the five terms are the exponents of $m^\omega d$, $d\,\mathsf{MM}(m,m,k)$, the middle level, the top level and the output, while $d\,\mathsf{MM}(k,m,m)$, the truncated powers $m^{\omega+2}$ and the powers mn are dominated, by symmetry of ω and since $3s \leq 1$. At $s = 1/3$ the five terms are at most $\max\{(\omega+2)/3,\ 2/3 + \omega(1/3, 1/3, t)\}$, so that $\Lambda_3 \leq \Lambda_2$ for every t. □

With the full table, the envelope values are $\hat{\Lambda}_3(0) = (\omega+3)/4 < 1.3429$, $\hat{\Lambda}_3(0.1) < 1.3660$, $\hat{\Lambda}_3(0.2) < 1.4034$ and $\hat{\Lambda}_3(0.3) < 1.4435$, against $\hat{\Lambda}_2(t) < 1.4572$ on $[0, 1/3]$, and $\hat{\Lambda}_3(1/3) = \hat{\Lambda}_2(1/3)$. For $t > 1/3$ the third level does not help in this cost model, for any $s \leq 1/3$. Indeed, let $z = (c, c, \delta)$ be an optimal dual vector of the program defining $\hat{U}(\frac{1}{3}, \frac{1}{3}, t)$, symmetric in its first two coordinates by symmetry of the program, so that $\hat{\Lambda}_2(t) = \frac{2}{3} + \frac{2c}{3} + \delta t$. Applied to the dimensions $(t+1-3s, s, 1-2s)$ of the middle level with δ on the first coordinate, it shows that the middle level is charged at least $\delta t + \delta + c + (2 - 3\delta - c)s$, whose minimum over $s \leq 1/3$ is $\hat{\Lambda}_2(t)$ as soon as $3\delta + c \geq 2$. For the basic bounds, $z = (c, c, \Delta_\omega)$ with $c = (2\omega - \omega_2)/2$ for $1/3 \leq t \leq 2/3$ and $z = (b, b, 1)$ with $b = (\omega_2 - 2)/2$ for $t \geq 2/3$, and both satisfy this inequality; for the full table, the curve is linear on the intervals of $3t$ with endpoints 1, 20/19, 3/2, 2 and ∞, and the exact dual vectors of these pieces, listed in Appendix A, satisfy $3\delta + c - 2 > 1.24$. Thus Λ_3 is the batch curve for generating arguments, and $\hat{\Lambda}_3 = \hat{\Lambda}_2$ for $t \geq 1/3$. The online phase is linear, so the transposed statements of Appendix E.1 hold for small batches as well; in particular k power projections of length $2n$ for a generating argument, and hence k inverse compositions as in Corollary E.3, cost $\tilde{O}(n^{(\omega+3)/4}k^{(\omega-1)/4})$. For arbitrary arguments, Theorem 6.23 gives both forward-composition bounds and, by transposition, the corresponding power-projection bounds; inverse composition as in Corollary E.3 retains its generator hypothesis. The factorization does not use small batches: its batch sizes are about $n^{0.447}$.

6.3 A fourth level

The dimension in Lemma 6.8 is at least $(4n)^{1/4}$ for two reasons: the truncated powers of Lemma 6.3 have length at most m^3, and the evaluation uses three digit levels. Both restrictions can be removed. In addition to A_j, B_j, let $C_j \in \mathbb{K}[x,y]_{<(d,m)}$, $j < m$, satisfy $C_j(x,b) \equiv b^{jm^3} \bmod h$; they are obtained, with the other data, by reducing $b^{jm^3} \operatorname{rem} h$ with [22, Prop. 2.3], so that all auxiliary data still cost $\tilde{O}(m^{\omega-1}n + mn)$.

Raw evaluation. For $w \geq 1$ and $L \leq m^4$, let $E_{w,L} \colon \mathbb{K}[x,y]_{<(w,L)} \to \mathbb{K}[x]_{<n}$, $G \mapsto G(x,b) \operatorname{rem} h$. Write the y-exponents as $c = c_0 + mc_1 + m^2c_2 + m^3c_3$ with $c_0, c_1, c_2 < m$ and $c_3 < \lceil L/m^3 \rceil \leq m$, form the blocks $s_{c_3,c_2,c_1} \in \mathbb{K}[x,y_0]_{<(w,m)}$ as in EVALUATE3, and compute successively

$$S_{c_3,c_2} = \sum_{c_1} s_{c_3,c_2,c_1} A_{c_1}, \qquad T_{c_3} = \sum_{c_2} S_{c_3,c_2} B_{c_2}, \qquad V = \sum_{c_3} T_{c_3} C_{c_3},$$

of bidegrees less than $(w+d, 2m)$, $(w+2d, 3m)$ and $(w+3d, 4m)$, and then $V(x,b) \operatorname{rem} h$ with the powers $b^l \operatorname{rem} h$, $l < 4m$. The first sum is a product of a $\lceil L/m^2 \rceil \times m$ matrix by an $m \times \lceil d/w \rceil$ matrix with entries of size wm, and the others are $O(\lceil L/m^2 \rceil)$ products of bivariate polynomials of size $O(n)$ with fixed polynomials. For $w = O(m)$ and $d = \Theta(n/m)$, this is a linear straight-line program of length

$$\tilde{O}(m^2 \mathsf{MM}(\lceil L/m^2 \rceil, m, \lceil d/m \rceil) + \lceil L/m^2 \rceil n + dm^2 + mn).$$

For a batch, the same fourth level extends EVALUATE3 to $d \leq m^4$: the rows of the middle product are indexed by (i, c_2, c_3), there are still $k\nu$ of them with $\nu = \lceil d/m^2 \rceil$, and the cost of Lemma 6.9 is unchanged.

Lemma 6.11 (truncated powers by transposition)**.** *Let $h \in \mathbb{K}[x]$ be monic of degree n with $h(0) \neq 0$, $b \in \mathbb{K}[x]_{<n}$, $w, L \geq 1$, and suppose that $E_{w,L}$ is computed by a linear straight-line program of length T. Then, for every $c \in \mathbb{K}[x]_{<n}$, the truncations $[cb^j \operatorname{rem} h]_{<w}$, $0 \leq j < L$, can be computed in $O(T) + \tilde{O}(n + wL)$ operations.*

Proof. If $w > n$, restrict the program to inputs of x-degree less than n by setting the other input coefficients to zero; this computes $E_{n,L}$ with length at most T, and the truncations for width w are those for width n followed by $w-n$ zero coefficients. We may thus assume $w \leq n$. Let $K \in \mathbb{K}^{n \times wL}$ be the matrix of $E_{w,L}$, whose columns are the coefficient vectors of $x^i b^j \operatorname{rem} h$. By [22, §6.1, Lemma 6.1 and Eq. (8)], the vector of all truncations equals $\operatorname{diag}(Q_w, \ldots, Q_w)\, K^{\mathsf{T}} P_n^{-1} \vec{c}$, where $P_n = M_{x^w,h} S$, S is the Hankel matrix of h, and $Q_w \in \mathbb{K}^{w \times w}$ is the product of the matrix $-[\operatorname{coeff}_i(x^j h)]_{i,j<w}$ by the reversal matrix. Since $h(0) \neq 0$, multiplication by x^{-w} modulo h and by S^{-1} cost $\tilde{O}(n)$, and each block Q_w, a truncated product by h, costs $\tilde{O}(w)$. By the transposition principle, K^{T} is applied in $O(T) + O(n + wL)$ operations. □

Theorem 6.12 (four levels)**.** *Let $n \geq 16$ and let m be an integer with $(4n)^{1/5} \leq m \leq n^{1/3} + 1$, $d = \lceil n/m \rceil$ and $\nu = \lceil d/m^2 \rceil$. The correctness and success statements of Lemma 6.8 extend to m when the truncated powers of Lemmas 6.4 and 6.5 are computed by Lemma 6.11, and their preprocessing cost is*

$$C_4(n,m) = \tilde{O}(m^\omega d + m^2 \mathsf{MM}(\nu, m, \lceil d/m \rceil) + \nu n).$$

Theorem 6.10 extends to m, with the same total cost bound and failure probability, when EVALUATE3 is extended by the fourth level, since $C_4(n,m)$ is dominated by that bound for every $k \geq 1$. Consequently $k = n^t$ compositions with a generating argument cost $n^{\Lambda_4(t)+o(1)}$, where Λ_4 is defined as Λ_3 with the minimum taken over $1/5 \leq s \leq 1/3$. In particular, one composition costs $n^{\kappa_4+o(1)}$ with

$$\kappa_4 = \Lambda_4(0) = \min_{1/5 \leq s \leq 1/4} \max\left\{1 + (\omega-1)s,\ 2s + \omega(1-3s, s, 1-2s),\ 2-3s\right\} \leq \frac{\omega+3}{4},$$

and $\kappa_4 = 5/4$ *if* $\omega = 2$.

Proof. Since $m^5 \geq 4n$ and $m \geq 3$, we have $2d+2 \leq 2n/m + 4 \leq m^4$, so that the truncated sequences of length $2d+2$ and the evaluation have at most four digit levels. In the change of basis, compute the sequences $[(bu \operatorname{rem} h)b^j \operatorname{rem} h]_{<m}$ and $[x^{m-1}b^j \operatorname{rem} h]_{<2m-1}$, $j < 2d+2$, for the inverse base $b = \gamma^{-1} \operatorname{rem} h$, by Lemma 6.11 with $w \in \{m, 2m-1\}$, and deduce the arrays $\Gamma_{i,k}$ by the downward recurrence of Lemma 6.3, in $O(m^2 d)$ operations; the same applies to the certified basis of Lemma 6.5. The raw evaluation with $w \leq 2m-1$ and $L = 2d+2$ costs the terms of Theorem 6.10 with $k = 1$. The proofs of Appendix C used the conditions $d + 2m - 1 \leq n$ and $2d \leq n+1$ only for the high and low parts of [22, Prop. 4.1], which are no longer used, and the other parameter conditions of Lemma 6.8 hold as before, with $m^2 \leq 8d$; in the probability bound, the degrees $2n^2/m$, $n(d-1)$ and $n(m-1)$ are still at most n^2, so that the same bound holds uniformly over this range of dimensions. The raw evaluation adds the two last terms of $C_4(n,m)$ to the preprocessing, and the terms dm^2, mn and $m^2 d$ are at most $m^\omega d$. The evaluation is that of Lemma 6.9 with the fourth level. The exponent follows as in Theorem 6.10, with the range of s extended to $s \geq 1/5$; at $t = 0$ the reduction, baby-step and output terms are at most $1+s$. Taking $s = 1/4$ gives $\kappa_4 \leq (\omega+3)/4$, since $\omega(1/4, 1/4, 1/2) \leq (\omega+1)/4$. If $\omega = 2$, the second term is $2 - 3s$ for $s \leq 1/4$, and the minimum is $5/4$ at $s = 1/4$. □

With the full table, $\hat{\kappa}_4 = 1.330827\ldots < 1.3309$, attained at $s = 0.2412$, where the first two terms are equal; with the basic bounds, $\hat{\kappa}_4 = 1.331434$. Thus the fourth level lowers the exponent of one composition below $(\omega+3)/4 < 1.3429$, the exponent of [22] for generic inputs; the gain comes from the rectangular product of the first level, and it vanishes if $\omega = 2$. The equivalence between bivariate evaluation and truncated powers used in Lemma 6.11 is that of [22, §6.1]; what is new is its use below the dimension $n^{1/4}$, the four-level cost analysis, and its certified use for prescribed and arbitrary arguments. The curve $\hat{\Lambda}_4$ is below $\hat{\Lambda}_3$ only for $t < 0.0363$, for instance $\hat{\Lambda}_4(0.02) = 1.337444$, and $\hat{\Lambda}_4 = \hat{\Lambda}_3$ for $t \geq 0.0363$. More levels do not help in this cost model: a dimension $s < 1/5$ costs at least the second-level term $2 + t - 3s > 1.4 + t$, whereas $s = 1/4$ costs at most $(\omega+3)/4 + t < 1.3429 + t$, since every term grows with slope at most 1 in t.

Using the four-level preprocessing and the algorithm of Theorem 6.23 with $k = 1$, the linear-program conclusion of Lemma E.6 holds with length and expected preprocessing cost $n^{\kappa_4 + o(1)}$ for every input length $N = O(n)$. Consequently one-form power projection as in Theorem E.2 and the three tasks of Corollary E.7 cost an expected $n^{\kappa_4+o(1)}$ operations, and Corollary E.9 holds with $n^{\kappa_4+o(1)}$ in place of $n^{(\omega+3)/4+o(1)}$; the separability, characteristic and extension-field conditions remain in force.

6.4 Bounded fibers

Throughout this subsection, let $f \in \mathbb{K}[x]$ be monic and separable of degree $n \geq 1$, and let $a \in \mathbb{K}[x]_{<n}$. By Proposition 4.3, an argument that takes the same value at many roots of f cannot be put in good position. We handle such arguments by separating the large fibers from the small ones. After a generic change of coordinates, both obstructions are determined by the sizes of the fibers (Lemma 6.13 and [21, Prop. 7.8]). Write

$$\ell_{f,a}(c) = \operatorname{card}\{z \in \overline{\mathbb{K}} : f(z) = 0,\ a(z) = c\}, \qquad L_{f,a} = \max_c \ell_{f,a}(c). \tag{6.2}$$

In particular, $L_{f,a} = 1$ means that a generates $\mathcal{A}$. By Corollary 6.14, after a random change of basis the auxiliary relation matrix has the generic degree

$$\max\{L_{f,a}, \lceil n/\mu \rceil\}.$$

Small fibers therefore do not affect this degree, but they may prevent the certificate of a full relation module; since the reduction needs only a nonsingular matrix of relations (Lemma 6.15),

we then certify a submodule instead, by [21, Algorithm 5.2]. For large fibers, we take a short relation

$$r(x,y) = \sum_i x^i c_i(y), \qquad r(x,a) \equiv 0 \bmod f, \qquad \deg_x r = \delta.$$

If a value c has more than δ preimages, then $r(x,c)$ vanishes identically, that is, $c_i(c) = 0$ for every i. Consequently, the content of r annihilates all heavy values (Lemma 6.18), and gcds over $\mathbb{K}$ separate a factor on which a takes few values from a cofactor with small fibers.

6.4.1 Relations and certificates

Lemma 6.13 (rank on lines)**.** *Let $c_1, \ldots, c_r$ be distinct elements of an extension of $\mathbb{K}$, let $\ell_1, \ldots, \ell_r \geq 1$ with $\sum_k \ell_k = n$, and let $d, \mu \geq 1$. Give the ℓ_k points of the line $Y = c_k$ independent abscissae $z_{k,1}, \ldots, z_{k,\ell_k}$. The matrix $[z_{k,s}^i c_k^j]_{(k,s);\, i<d,\, j<\mu}$ has generic rank $\min\{d\mu, \sum_k \min(\ell_k, d)\}$, also when the abscissae are required to be nonzero and pairwise distinct. In particular it has generic rank n if and only if $n \leq d\mu$ and $\max_k \ell_k \leq d$.*

Exchanging the variables gives [21, Lemmas 7.10 and 7.11] on the $\mathcal{M}$ side. The proof in Appendix D also bounds the degree of the nonvanishing minor.

Corollary 6.14. *Let $h \in \mathbb{K}[x]$ be monic separable of degree n, $b \in \mathbb{K}[x]_{<n}$, $1 \leq \mu \leq n$ and $D = \max\{L_{h,b}, \lceil n/\mu \rceil\}$. Let $\gamma = \sum_{i<n} T_i x^i$ with independent indeterminates T_i, let χ be its characteristic polynomial modulo h and $\phi(\gamma) = b \bmod h$. There is a nonzero polynomial $\Psi(T)$ over a splitting field, of degree at most $n(D-1)$, such that at every specialization where γ generates $\mathbb{K}[x]/\langle h \rangle$ and $\Psi \neq 0$, the minimal bases of $\mathcal{N}_\mu^{(\phi,\chi)}$ have degree D. The same holds for $\phi^{-1} \bmod \chi$ when b is a unit.*

Lemma 6.15 (relations in a submodule suffice)**.** *Let b, h be as above and let $R \in \mathbb{K}[y]^{m_R \times m_R}$ be nonsingular with $\mathcal{L}(R) \subseteq \mathcal{M}_{m_R}^{(b,h)}$. If $v - \tilde{v} \in \mathcal{L}(R)$ then $\sum_i v_i(b) x^i \equiv \sum_i \tilde{v}_i(b) x^i \bmod h$. Conversion of R to Popov form preserves $\mathcal{L}(R)$ and does not increase its degree [21, Lemma 5.5], and remainders modulo the Popov form have y-degree less than the corresponding row degrees. Consequently the reduction of [22, Lemma 2.7], the remainder tables of Section 3, EVALUATE of Section 4, and the evaluation of [22, Prop. 3.1] return correct results for R, at the costs stated for a basis of the same dimension and degree. Moreover, over a splitting field $\deg \det \mathcal{M}_m^{(b,h)} = \nu_m := \sum_c \min(\ell_{h,b}(c), m)$ for every separable pair, without genericity.*

Lemma 6.16 (sound certificate for a submodule)**.** *Let $h \in \mathbb{K}[x]$ be monic and separable of degree $M \geq 16$ with $h(0) \neq 0$, let b be a unit modulo h, and let m, d be as in (6.1) with n replaced by M. There is a procedure which returns either* Fail *or a nonsingular matrix $R \in \mathbb{K}[y]_{\leq d}^{m_R \times m_R}$ with $\mathcal{L}(R) \subseteq \mathcal{M}_{m_R}^{(b,h)}$ and $1 \leq m_R \leq 2m$, using $\tilde{O}(M^{(\omega+3)/4})$ operations on every execution. If the minimal bases of $\mathcal{N}_m^{(b,h)}$ have degree d and $\operatorname{rank} \mathrm{Hk}_{m,d}^{(b,h)} = \nu_m$, it returns* Fail *with probability at most $(m-1)/\operatorname{card}(S)$ over $m-2$ elements of S drawn uniformly and independently.*

Proposition 6.17 (bounded fibers)**.** *Let $f \in \mathbb{K}[x]$ be monic separable of degree $M \geq 2$ and $a \in \mathbb{K}[x]_{<M}$.*

(i) With $m = \lceil (4M)^{1/4} \rceil$ and $d = \lceil M/m \rceil$: if $L_{f,a} \leq d$, there is an algorithm which, given $g \in \mathbb{K}[y]_{<M}$ and independent uniform random elements of S consisting of r_1, r_2, the M coefficients of γ and, for $M \geq 16$, $m-2$ further elements for the submodule certificate, returns $g(a) \operatorname{rem} f$ or Fail *in $\tilde{O}(M^{(\omega+3)/4})$ operations on every execution, and fails with probability at most $\min\{1, 8M^2/\operatorname{card}(S)\}$.*

Algorithm 6.2 BoundedFiberComposition$(f, a, (g_i)_{i\le k}, J, mode)$ (Proposition 6.17 and Lemma 6.20), $M = \deg f \ge 16$

Require: f monic separable of degree M; $a \in \mathbb{K}[x]_{<M}$ (any); $g_i \in \mathbb{K}[y]_{<Jd}$; $mode \in \{\text{single}, \text{batch}\}$; $m = \lceil (4M)^{1/4} \rceil$ in single mode and $m = \lceil M^{1/3} \rceil$ in batch mode; $d = \lceil M/m \rceil$; random $r_1, r_2 \in S$, $\gamma \in S^M$, $m-2$ certificate coefficients in S

Ensure: $(g_i(a) \operatorname{rem} f)_i$, or Fail (never a wrong result)

1: $h \leftarrow f(x + r_2)$, $u \leftarrow (a + r_1)(x + r_2) \operatorname{rem} h$; if $h(0) = 0$ or $\gcd(u, h) \ne 1$, **return** Fail
2: $(R_\gamma, \chi, \phi) \leftarrow$ Algorithm 6.1 (single mode) or ChangeOfBasis (batch mode) on (h, γ, u), or **return** Fail; if $\chi(0) = 0$, $\gcd(\phi, \chi) \ne 1$ or $\deg R_\gamma > d$, **return** Fail
3: for $b \in \{\phi, \phi^{-1} \operatorname{rem} \chi\}$ modulo χ and $b \in \{\gamma, \gamma^{-1} \operatorname{rem} h\}$ modulo h, compute a minimal basis of $\mathcal{N}_m$; if its degree is not d, **return** Fail; otherwise compute its A_j, B_j
4: $(R', flag) \leftarrow$ CandidateBasis(χ, ϕ, m, d), with the truncated powers of ϕ^{-1} of Lemma 6.3 in single mode; if $\deg R' > d$, **return** Fail
5: if $flag =$ Cert, $R \leftarrow R'$; otherwise let r be column 1 of R' and s column 2 plus a random combination of the others; if $r(x, \phi) \not\equiv 0$, $s(x, \phi) \not\equiv 0 \bmod \chi$ or $\gcd_x(r, s) \ne 1$, **return** Fail; $R \leftarrow$ Sylvester matrix of (r, s)
6: $P \leftarrow$ Popov form of R, of dimension m_R; if $\deg P > d$, **return** Fail
7: compute the division data of Lemma 2.2(a) for P and R_γ, the remainder table $P_0, \dots, P_{J-1}$ of P, and in batch mode $\phi^\ell \operatorname{rem} \chi$ and $\gamma^\ell \operatorname{rem} h$ for $\ell < 2m$
8: $\tilde{v}_i \leftarrow g_i(y - r_1)e_1 \operatorname{Rem} P$ for all i, keeping all m_R components
9: $z_i \leftarrow \tilde{v}_i(x, \phi) \operatorname{rem} \chi$, by [22, Prop. 3.1] for each input in single mode and by Evaluate with x-width $2m$ in batch mode
10: $w_i \leftarrow z_i(\gamma) \operatorname{rem} h$ in the same way with R_γ; **return** $(w_i(x - r_2))_i$

(ii) With $m = \lceil M^{1/3} \rceil$ and $d = \lceil M/m \rceil$: if $L_{f,a} \le d$, the same holds for k polynomials $g_1, \dots, g_k \in \mathbb{K}[y]_{<M}$, with cost $\tilde{O}(M^{(\omega+2)/3} + d\,\mathsf{MM}(m, m, k))$, that is $\tilde{O}(M^{(\omega+2)/3} + kM^{(\omega+1)/3})$ with square blocks.

(iii) With $M \ge 16$, an integer m such that $(4M)^{1/4} \le m \le M^{1/3} + 1$, $d = \lceil M/m \rceil$ and $\nu = \lceil d/m^2 \rceil$: if $L_{f,a} \le d$, the same holds for k polynomials $g_1, \dots, g_k \in \mathbb{K}[y]_{<Jd}$, where $J \ge \lceil M/d \rceil$, with the random elements of (i) and cost

$$\tilde{O}(m^\omega d + dJm^{\omega-1} + d\lceil J/m \rceil \mathsf{MM}(m, m, k) + m^2 \mathsf{MM}(k\nu, m, \lceil d/m \rceil) + k\nu M + kJd).$$

In all cases the algorithm consists of a preprocessing phase, which depends only on (f, a) and the random elements and returns Fail or certified data, and an online phase. The certified data include a matrix R with

$$\det R \ne 0, \qquad \mathcal{L}(R) \subseteq \mathcal{M}_{m_R}^{(\phi,\chi)}, \qquad 1 \le m_R \le 2m, \qquad \deg R \le d$$

(equality with the full relation module is not needed). The failure event does not depend on the outer polynomials, and after a successful preprocessing the online phase is correct for every outer polynomial.

6.5 The content separator

Table 6: Summary of the cases of Section 6.4 for the three-level algorithms (the four-level extension used for Λ_4 in Theorem 6.23 lowers the dimension to $(4M)^{1/5}$ and replaces the long table by blocking), with $m_s = \lceil (4M)^{1/4} \rceil$, $m_b = \lceil M^{1/3} \rceil$, $d_s = \lceil M/m_s \rceil$, $d_b = \lceil M/m_b \rceil$, $\varepsilon_M = \min\{1, 8M^2/\operatorname{card}(S)\}$, and the separator parameters $H_N = \lfloor \frac{1}{4} \lfloor N/2 \rfloor^{\tau} \rfloor$, $t_N = \lfloor N/(H_N+1) \rfloor + 1$, $\tau \in \{3/4, 2/3\}$. All local moduli are separable. The degree column refers to a nonsingular matrix of relations, not necessarily a basis of the full module, whose dimension m_R may exceed m; for the separator, δ is the degree of the column used. A few-value piece needs only its annihilator P, of degree s.

case	degree of f	dimension m	dimension m_R	degree of relations	fiber condition	failure probability
Single bounded fibers	M	m_s	$1 \le m_R \le 2m_s$	$\le d_s$	$L_{f,a} \le d_s$	ε_M
Small batches	$M \ge 16$	$m_s \le m \le \lfloor M^{1/3} \rfloor + 1$	$1 \le m_R \le 2m$	$\le \lceil M/m \rceil$	$L_{f,a} \le \lceil M/m \rceil$	ε_M
Cube-root bounded fibers	M	m_b	$1 \le m_R \le 2m_b$	$\le d_b$	$L_{f,a} \le d_b$ cofactor	ε_M
Separator	N	t_N	—	$\delta \le H_N$	$L_{h',a} \le H_N \le s$ values;	0
Few-value piece	b	—	—	—	no fiber bound	0
Long terminal table	$M \le n$	m_b	$1 \le m_R \le 2m_b$	$\le d_b$	$L_{h_*,a} \le d_b$	ε_M

The first three rows are Proposition 6.17(i), (iii) and (ii), and the others are Lemmas 6.18 to 6.20. The long table uses the local width m_b and $J = \lceil n/d_b \rceil$, and its application is charged at the global width $\bar{m} = \lceil n^{1/3} \rceil$. Only the precomputation can fail, and for $\operatorname{card}(S) \ge 16n^2$ the expected number of attempts is $O(1)$.

Lemma 6.18 (content separator)**.** *Let $f \in \mathbb{K}[x]$ be monic separable of degree $N \ge 1$, $a \in \mathbb{K}[x]_{<N}$, and $1 \le t \le N$. In $\tilde{O}(Nt^{\omega-1})$ operations one can compute deterministically a nonzero $P \in \mathbb{K}[y]$ of degree less than t and a factorization $f = Fh'$ into coprime monic factors such that $P(a) \equiv 0 \bmod F$, so that a takes at most $\deg P$ values on the roots of F, and every value of a on the roots of h' has fiber size at most $\lfloor N/t \rfloor$ in f. No fiber is split between F and h'.*

Proof. Compute a column-reduced basis of $\mathcal{N}_t^{(a,f)}$ by [22, Prop. 2.3], a deterministic construction that applies to every a and costs $\tilde{O}(Nt^{\omega-1})$. Its determinant is a nonzero constant times f (the columns fe_0 and $e_j - (a^j \operatorname{rem} f)e_0$ form a basis), so its column degrees sum to N and some column has degree $\delta \le \lfloor N/t \rfloor$. Write that relation as $r = \sum_{i \le \delta} x^i c_i(y)$, $\deg c_i < t$, and put

$$P = \operatorname{monic} \gcd(c_0, \dots, c_\delta), \qquad F = \gcd(f, P(a) \operatorname{rem} f), \qquad h' = f/F.$$

If $\ell_{f,a}(c) > \delta$, the polynomial $r(x, c)$, of degree at most δ, vanishes at more than δ points and is zero. Hence $c_i(c) = 0$ for all i, and $P(c) = 0$ since gcds do not change under field extension. A root z of f is a root of F exactly when $P(a(z)) = 0$, a condition on the value $a(z)$ only. This proves the assertions, with F and h' coprime because f is separable. The gcd of the coefficients costs $\tilde{O}(N + t)$, the Horner evaluation of $P(a)$ costs $\tilde{O}(Nt)$, and the last gcd costs $\tilde{O}(N)$. □

The separation uses only linear algebra and gcds over $\mathbb{K}$, without characteristic polynomials, factorization or root extraction; the content P may also vanish at light values, which causes no difficulty.

A failed certificate need not reveal a factor of f. For example, take the irreducible polynomial and the argument

$$f = x^{16} + x^5 + x^3 + x^2 + 1 \in \mathbb{F}_2[x], \qquad a = x^{4369} \operatorname{rem} f \in \mathbb{F}_{16}.$$

All four fibers have four points, so that $\nu_3 = 12 < 16$, and every $\gcd(f, q(a))$ is 1 or f. In this case the separator gives $F = f$, and the few-value case handles the whole algebra.

Lemma 6.19 (few values)**.** *Let $F \in \mathbb{K}[x]$ be monic of degree $b \geq 1$, $a \in \mathbb{K}[x]_{<b}$, and let $P \in \mathbb{K}[y]$ of degree $s \geq 1$ satisfy $P(a) \equiv 0 \bmod F$. Then $g_1(a), \ldots, g_k(a) \operatorname{rem} F$, for $g_j \in \mathbb{K}[y]_{<n}$, can be computed in $\tilde{O}(kn + bs + \mathsf{MM}(b, s, k))$ operations.*

Proof. Since $g_j(a) \equiv (g_j \operatorname{rem} P)(a) \bmod F$, it suffices to compute $a^i \operatorname{rem} F$ for $i < s$ and to multiply the $b \times s$ matrix of their coefficients by the $s \times k$ matrix of coefficients of the remainders. □

6.6 Long outer polynomials and the general theorem

A factor of small degree M must still accept outer polynomials of degree up to n. Splitting these inputs into $\lceil n/M \rceil$ blocks would multiply the batch size. Instead, we lengthen the remainder table and apply it by a rectangular product at the global width $n^{1/3}$.

Lemma 6.20 (long outer polynomials on a factor with bounded fibers)**.** *Let $2 \leq M \leq n$, let $h \in \mathbb{K}[x]$ be monic and separable of degree M, $b \in \mathbb{K}[x]_{<M}$, $m = \lceil M^{1/3} \rceil$, $d = \lceil M/m \rceil$ and $\bar{m} = \lceil n^{1/3} \rceil$. If $L_{h,b} \leq d$, all $g_i(b) \operatorname{rem} h$ for $g_1, \ldots, g_k \in \mathbb{K}[y]_{<n}$ can be computed by a Las Vegas algorithm using an expected $\tilde{O}(n^{(\omega+2)/3} + \bar{m}^2 \mathsf{MM}(\bar{m}, \bar{m}, k))$ operations. Each preprocessing attempt uses the random choices of Proposition 6.17(ii) and fails with probability at most $\min\{1, 8M^2/\operatorname{card}(S)\}$; choose $\operatorname{card}(S) \geq 16n^2$, with the extension-field convention of Theorem 6.1 when necessary, and repeat the preprocessing on failure.*

Theorem 6.21 (arbitrary arguments)**.** *Let $f \in \mathbb{K}[x]$ be monic and separable of degree $n \geq 1$ and let $a \in \mathbb{K}[x]_{<n}$ be arbitrary. There is a Las Vegas algorithm which, given $g \in \mathbb{K}[y]_{<n}$, computes $g(a) \operatorname{rem} f$ using an expected $\tilde{O}(n^{(\omega+3)/4})$ operations in $\mathbb{K}$, and one which, given $g_1, \ldots, g_k \in \mathbb{K}[y]_{<n}$, computes all $g_j(a) \operatorname{rem} f$ using an expected*

$$\tilde{O}(n^{(\omega+2)/3} + n^{2/3} \mathsf{MM}(n^{1/3}, n^{1/3}, k))$$

operations, matrix dimensions being rounded up; for $k = n^t$ this is $n^{\Lambda_2(t)+o(1)}$, the curve of Theorem 4.6. The random elements and the treatment of small fields are those of Theorem 6.1, with the additional certificate coefficients of Proposition 6.17. Both algorithms first apply the deterministic content separator and then repeat certified compositions on failure (Algorithm 6.3).

The idea of the proof is as follows; the details are in Appendix D.7. With $\tau = 3/4$ for one polynomial and $\tau = 2/3$ for a batch, Algorithm 6.3 applies the content separator repeatedly, with auxiliary dimension about $N^{1-\tau}$ at local degree N, as long as the cofactor has degree less than $N/2$. The local degrees thus decrease geometrically, the recorded factors have annihilators of degree $O(N^{1-\tau})$ and are handled by Lemma 6.19, and the terminal factor h_* has fibers of size at most $\frac{1}{4}(\deg h_*)^\tau$, so that Proposition 6.17(i), with the outer polynomial split into blocks, or the long table of Lemma 6.20 applies to it. The residues are combined by Chinese remaindering.

Remark 6.22. The batched bound of Theorem 6.21 gives the full rectangular curve Λ_2 of Theorem 4.6 for arbitrary arguments; its proof uses only padding and partitioning of matrix products. Generating arguments remain the relevant case for factorization, where they occur at every giant step. The decomposition needs only annihilators for the few-value pieces, and it applies over imperfect fields, in any characteristic. Moduli with repeated factors are treated in Theorem E.11, which covers every modulus over a perfect field; for inseparable factors over imperfect fields, the Las Vegas algorithm of [21] at exponent κ still applies.

The bounds of Theorem 6.10 also extend to arbitrary arguments. The separator now uses an auxiliary dimension that grows with k, and the terminal factor is handled by Proposition 6.17(iii).

Algorithm 6.3 ArbitraryComposition$(f, a, (g_i)_{i\le k})$ (Theorem 6.21)

Require: f monic separable of degree n; $a \in \mathbb{K}[x]_{<n}$ arbitrary; $g_i \in \mathbb{K}[y]_{<n}$; $\tau = 3/4$ ($k = 1$) or $\tau = 2/3$ (batch)

Ensure: $(g_i(a) \operatorname{rem} f)_i$ (Las Vegas: internal calls are repeated on Fail)

$\mathcal{R} \leftarrow [\,]$; $h \leftarrow f$
loop
  $N \leftarrow \deg h$; if $N < 16$, append $(h, \text{annihilator of } a \operatorname{rem} h)$ to $\mathcal{R}$, $h_* \leftarrow 1$, **break**
  $H \leftarrow \lfloor \frac{1}{4} \lfloor N/2 \rfloor^{\tau} \rfloor$, $t_N \leftarrow \lfloor N/(H+1) \rfloor + 1$
  $r \leftarrow$ column of least degree δ of a column-reduced basis of $\mathcal{N}_{t_N}^{(a \operatorname{rem} h, h)}$; write $r = \sum_{i \le \delta} x^i c_i(y)$
  $P \leftarrow \gcd_i(c_i)$; $F \leftarrow \gcd(h, P(a) \operatorname{rem} h)$; $h' \leftarrow h/F$; if $F \neq 1$, append (F, P) to $\mathcal{R}$
  if $\deg h' = 0$, $h_* \leftarrow 1$, **break**; if $\deg h' \ge N/2$, $h_* \leftarrow h'$, **break**
  $h \leftarrow h'$
end loop
for $(F, P) \in \mathcal{R}$ **do**
  $w_{i,F} \leftarrow (g_i \operatorname{rem} P)(a) \operatorname{rem} F$, using the powers $a^j \operatorname{rem} F$, $j < \deg P$
end for
if $h_* \neq 1$, with $M = \deg h_*$ **then**
  in single mode, split g_1 into $\lceil n/M \rceil$ blocks of degree less than M, compose each by Algorithm 6.2 with $J = \lceil M/d \rceil$, and recombine by Horner's rule in $a^M \operatorname{rem} h_*$; in batch mode, apply Algorithm 6.2 with the long table $J = \lceil n/d \rceil$; repeat each call on Fail
end if
return the CRT combination of all residues modulo the pairwise coprime $F \in \mathcal{R}$ and h_*

Theorem 6.23 (small batches, arbitrary arguments)**.** *Let $f \in \mathbb{K}[x]$ be monic and separable of degree $n \ge 1$ and let $a \in \mathbb{K}[x]_{<n}$ be arbitrary. There are Las Vegas algorithms which, given $g_1, \dots, g_k \in \mathbb{K}[y]_{<n}$, compute all $g_j(a) \operatorname{rem} f$ using an expected*

$$\tilde{O}(n^{(\omega+3)/4} k^{(\omega-1)/4}) \quad \text{if } 1 \le k \le n^{1/3}, \qquad n^{\Lambda_4(t)+o(1)} \quad \text{if } k = n^t,\ t \ge 0,$$

operations in $\mathbb{K}$, with $\Lambda_4 \le \Lambda_3$ as in Theorem 6.12 and the random elements of Theorem 6.21. In particular, one composition costs an expected $n^{\kappa_4+o(1)} = O(n^{1.3309})$ operations.

The idea of the proof is as follows; the details are in Appendix D.8. At a local degree N with $k \le N^{1/3}/8$, the separator uses the auxiliary dimension $\Theta((Nk)^{1/4})$ instead of $\Theta(N^{1/3})$, so that the terminal factor, of degree $M \ge N/2$, has fibers of size at most $\lceil M/m \rceil$ for $m = \lceil (4Mk)^{1/4} \rceil$. It is handled by Proposition 6.17(iii) with the long remainder table $J = \lceil n/d \rceil$, whose application costs $\tilde{O}(nm^{\omega-1})$. At a local degree $N < 512k^3$, the parameters of Theorem 6.21 are used, and their costs are bounded by $\tilde{O}(nk^{\omega-1})$, which is at most $n^{(\omega+3)/4} k^{(\omega-1)/4}$ because $k^3 \le n$. For the curve Λ_4, the dimension exponent s is fixed by t, and the long tables are avoided: on the terminal factor, of degree $M = n^{\mu}$, the outer polynomials are cut into $\lceil n/M \rceil$ blocks of degree less than M. The pieces then cost $n^{\mu F(u)+o(1)}$ with $u \le (t+1-\mu)/\mu$, where F is the maximum in the definition of Λ_4 at this s, and $\mu F(u) \le F(t)$ because F grows with slope at most 1 and $F(t) \ge 1 + t$. For fixed (f, a, k) and successful random choices, every step that involves the outer polynomials is linear in their coefficients, so transposition as in Appendix E.1 also gives k power projections of length $2n$ for an arbitrary argument within the same bound.

7 Experiments

We implemented the one-matrix batched composition in SageMath over $\mathbb{F}_{8388593}$, on top of the reference implementation of ChangeOfBasis and MatrixOfRelations by the authors of

[21]; all outputs were checked against FLINT, and timings are medians of interleaved single-core runs. For $n = 8192$ and $k = 91$, the batch was 60.2 times faster than k independent compositions by the algorithm of [21] (whose cost was extrapolated from the first 16 inputs), and 2.19 times faster than sharing only the precomputation. At these sizes, however, batched Brent–Kung composition remains 22 times faster, so the asymptotic gains are not yet visible in practice. For the same reason, our prototype of the distinct-degree stage is faster than with independent compositions but slower than with Brent–Kung. The experiment also illustrates the role of the extension of scalars: for a polynomial of degree 4096 with all roots in $\mathbb{F}_{8388593}$, where the bound of Lemma 5.1 is vacuous, three attempts without extension failed.

The algorithms for arbitrary arguments were tested on 80 batches with generating arguments and on 6390 compositions with arbitrary arguments, all with correct results; the tests include every case of Table 6, at local degrees at least 16. The minimal polynomials and the irreducibility tests of Appendix E.3 were also correct on all tested inputs. The programs and the complete data are in the supplementary material.

8 Open problems

Within Algorithm D with two relation matrices, the exponent e_2 is optimal for the cost model of Proposition 5.7, and e_2^{full} for the full table by Remark 5.8. Improving it requires better bounds on $\omega(1, 1, s)$ near the active entries $s = 0.9$ and $s = 1.5$, a different evaluation scheme, or a different arrangement of baby and giant steps, as in [27]. In the cost model of Appendix A, unbalanced dimensions $m \neq \bar{m}$ do not help at the batch sizes used for factorization, a third evaluation level helps only for fewer than $n^{1/3}$ compositions (Section 6.2), and a fourth one only for fewer than $n^{0.037}$, while a fifth one does not help at all (Section 6.3). In the four-level bound κ_4, the binding terms are the precomputation and the first level; lowering κ_4 requires either a cheaper computation of the auxiliary relations or a better bound for the rectangular product of the first level. When the baby steps are also batched, the exponent $\Lambda_2(1/2)$ of Corollary 5.11 is optimal for that arrangement in the same cost model; for $0.0983 < \rho < \hat{\Lambda}_2(1/2) - 1 < 0.5983$, an improvement again requires a better batch curve near $t = 1/2$, while for larger ρ the bound is $n^{1+\rho+o(1)}$, the cost of computing $x^q \operatorname{rem} f$ by powering. For large batches with a generating argument, the preprocessing exponent $(\omega + 1)/2$ of Theorem 4.9 is that of the relation modules of width $\sqrt{n}$; we do not know whether it can be lowered while keeping a quasi-linear online phase, nor whether Theorem 4.9 extends to arbitrary arguments.

Our bounds use the evaluation at the roots of separable factors. The composition and power-projection bounds apply to every modulus supplied with a separable power decomposition (Theorem E.11); such a decomposition exists over every perfect field, and over finite fields it is computed with an additional cost $n^{1+o(1)} \log q$. Over imperfect fields, factors $h(x^{p^e})^\ell$ with $e > 0$ need a composition modulo an inseparable polynomial over a product of fields, and extending our bounds to them remains open.

A The rectangular exponent programs

All exponents refer to n, with the factor $\log q$ omitted; it multiplies only the term $n^{1+\beta}$ of step (D1) when the baby steps are computed by powering, and only a term $n^{1+o(1)}$ when they are computed by doubling. The programs `exponent_lp.py` (one relation matrix, Kaltofen–Shoup with Brent–Kung, and the plug-in bound) and `e4_lp.py` (two relation matrices) use the basic bounds, and `full_table.py` recomputes every program with the full table, after checking that it reproduces the basic values; all are in the supplementary material.

The bound on rectangular products We use the function U of Eq. (2.2) with the generators

$$(1,1,1) \mapsto \omega, \quad \pi(1,1,2) \mapsto \omega_2, \quad \pi(1,1,\alpha) \mapsto 2, \quad \pi(1,1,0) \mapsto 2, \quad \pi(1,0,0) \mapsto 1,$$

where π ranges over the permutations of the coordinates, $\omega = 2.371339$, $\omega_2 = 3.250035$ [1] and $\alpha = 0.321334$ [35] (the basic bounds), and, for the full table, also $\pi(1,1,k) \mapsto w_k$ for the eleven other entries $k \neq 1, 2$ of [1, Table 1], for instance $w_{0.9} = 2.293941$ and $w_{1.5} = 2.794633$; this gives 46 permuted generators. We keep $\omega = 2.371339$; the entry 2.371177 of that table is the bound announced in [11]. The constraint $\sum_j \lambda_j g_j \geq v$ encodes sub-additivity of $\omega(\cdot)$ together with padding by zeros, and splitting a dimension into blocks is the special case of adding $\pi(1,0,0)$. With the basic bounds, along the ray $(1,1,t)$ this is the convex piecewise linear function through $(\alpha, 2)$, $(2/3, \frac{2}{3}\omega_2)$, $(1, \omega)$, $(2, \omega_2)$, of slope 1 beyond $t = 2$ and constant 2 below α, and the node at $2/3$ comes from the permuted generators, $(1, 1, \frac{2}{3}) = \frac{1}{3}(1,2,1) + \frac{1}{3}(2,1,1)$, and lies strictly below the chord from $(\alpha, 2)$ to $(1, \omega)$. Since U is itself the value of a linear program, the exponent optimization is also a linear program, with objective the maximum of finitely many exponents, each affine in the parameters plus values of U.

One relation matrix Take $\beta, \eta, \varrho \in [0,1]$, $t = 1-\beta$, $m = n^\eta$, $d = n^{1-\eta}$, $B = n^\varrho$, subject to $\varrho \leq 1-\eta$. The objective is the maximum of the following exponents. Division by the relation matrix is charged $d\,\mathsf{MM}(m,m,k)$, as in Lemma 2.2(a) and Eq. (3.2), and the square-block alternative is $\lceil k/m \rceil m^\omega d$, which differs only for $k > m$.

term	exponent	source
(D1)	$1+\beta$	Section 2.5
(D3)–(D5)	$2-\beta$	Section 2.5
$m^\omega d$	$1-\eta+U(\eta,\eta,\eta)$	Lemmas 3.2 and 3.3
$c(n,m,d)$	$\eta+U(\frac{1-\eta}{2},\frac{1-\eta}{2},1-\eta)$	Lemma 3.2
$d\,\mathsf{MM}(m,m,k)$	$1-\eta+U(\eta,\eta,t)$	Lemma 3.4
$m\,\mathsf{MM}(ks,B,d)$	$\eta+U(t+1-\eta-\varrho,\varrho,1-\eta)$	Lemma 3.5
ksn, Bn, kn	$t+2-\eta-\varrho$, $\quad 1+\varrho$, $\quad 1+t$	Lemma 3.5, Algorithm 3.2

With the full table the optimum is 1.621335, at $\beta = 0.6213$, $\eta = 0.4531$, $\varrho = 0.4455$ (with the basic bounds, 1.622743 at $\beta = 0.6227$, $\eta = 0.4541$, $\varrho = 0.4616$), and the active terms are (D1), $m^\omega d$ and $m\,\mathsf{MM}(ks,B,d)$. With square blocks only the optimum is $3\omega/(\omega+2) = 1.627423$ (Corollary 5.9). For Theorem 3.6(iii) alone, drop (D1)–(D5) and fix t, with no other constraint, and this computes $\hat{\Lambda}_1$. For instance $\hat{\Lambda}_1(0) = 1.429338 = \kappa$, $\hat{\Lambda}_1(0.2) = 1.527459$ (basic: 1.531870), $\hat{\Lambda}_1(0.5) = 1.685669$, $\hat{\Lambda}_1(1) = 2.125018$, and $\hat{\Lambda}_1(t) = t + (\omega_2 - 1)/2 = t + 1.1250175$ for $t \geq 1$, with the dual certificate of Remark 3.7, which is a statement about U.

Two relation matrices With $m = \bar{m} = n^{1/3}$ and $t = 1-\beta \geq 1/3$, the exponents are $1+\beta$, $2-\beta$, $(\omega+2)/3$ and $\frac{2}{3} + U(\frac{1}{3},\frac{1}{3},t)$ (Theorem 4.6). With the basic bounds, along $(1,1,s)$, $s = 3t \in [1,2]$, U is the segment from $(1,\omega)$ to $(2,\omega_2)$, which gives the closed form $e_2(\omega,\omega_2)$ of

Corollary 5.4: 1.555125 at $\beta = 0.555125$, $s^* = 1.3346$, optimal for the program by the exact dual certificate of Proposition 5.7. With the full table the optimum is $e_2^{\text{full}} = 24413604/15719207 = 1.5531066$ at $\beta = 0.5531066$, $3(1-\beta) = 1.3406802$, optimal by the certificate of Remark 5.8, which `full_table_dual.py` checks in exact arithmetic on all 46 constraints. We also solved extended programs numerically, allowing $m \neq \bar{m}$ under $m\bar{m}^2 \geq n$ or adding a third evaluation level, with all rectangular products and intermediate sums charged; they return the same values at $m = \bar{m} = n^{1/3}$.

Three levels lower the batch curve for small batches, but not at the batch size $k = n^{0.447}$ that matters for factorization. The program `three_level_lp.py` computes $\hat{\Lambda}_3$ of Theorem 6.10, with the terms listed there and the dominated ones; with the full table it returns $\hat{\Lambda}_3(0) = 1.342835 = (\omega+3)/4$, $\hat{\Lambda}_3(0.1) = 1.365917$, $\hat{\Lambda}_3(0.2) = 1.403361$, $\hat{\Lambda}_3(0.3) = 1.443456$ (basic: 1.367879, 1.406122, 1.444365) and $\hat{\Lambda}_3(1/3) = \hat{\Lambda}_2(1/3)$. For $t \geq 1/3$ it returns $\hat{\Lambda}_2(t)$, in agreement with the dual argument at the end of Section 6.2, which `three_level_dual.py` checks exactly for the basic bounds and `full_table_nogain.py` checks exactly for the full table: on the pieces $3t \in [1, \frac{20}{19}]$, $[\frac{20}{19}, \frac{3}{2}]$, $[\frac{3}{2}, 2]$ and $[2, \infty)$ the vectors $(c, \delta) = (\frac{38699}{50000}, \frac{823379}{1000000})$, $(\frac{1292557}{1700000}, \frac{7219207}{8500000})$, $(\frac{1428427}{2000000}, \frac{227701}{250000})$ and $(\frac{250007}{400000}, 1)$ are feasible, tight at both ends, and satisfy $3\delta + c - 2 > 1.24$; the node $\frac{20}{19}$ comes from $\frac{10}{19}((0.9,1,1)+(1,0.9,1))$. With square blocks it returns $(\omega+3)/4+(\omega-1)t/4$, and $5/4 + t/4$ if $\omega = 2$. The curve $\hat{\Lambda}_4$ of Theorem 6.12 is the same program with $s \geq 1/5$; with the full table it returns $\hat{\kappa}_4 = \hat{\Lambda}_4(0) = 4609149047263/3463372000000 = 1.3308270\ldots$ at $s = 835517/3463372 = 0.2412$, certified by the dual vector $(\frac{835517}{1000000}, \frac{207259}{500000}, 1)$ of the first-level product, where the precomputation term $1 + (\omega-1)s$ and the first-level term $2s + U(1-3s, s, 1-2s)$ are equal; $\hat{\Lambda}_4(0.02) = 1.337444$, and $\hat{\Lambda}_4 = \hat{\Lambda}_3$ for $t \geq 30326/835517 = 0.03630$ (basic: 1.331434 at $s = 0.2417$, 1.338182, 0.0329). The single-composition route of Table 1 then has exponent $1 + \hat{\kappa}_4/2 = 1.665414$.

Large fields In Corollary 5.11, $\beta = 1/2$ and the exponent is $\max\{\Lambda_2(1/2), 1+\rho\}$ for $\log q = n^\rho$. With the full table, $(1, 1, \frac{3}{2})$ is a generator and $\hat{\Lambda}_2(1/2) = \frac{2}{3} + \frac{2.794633}{3} = 1.598211$; the crossover with the exponent of Remark 5.10 is at $\rho = 0.098211$, and the first branch of that exponent is $1.5531066 + 0.4592603\rho$. With the basic bounds, $3t = 3/2$ lies on the segment from $(1, \omega)$ to $(2, \omega_2)$, $\hat{\Lambda}_2(1/2) = \frac{2}{3} + \frac{\omega+\omega_2}{6} = 1.603562$ and the crossover is at $\rho = 0.10356$. The scripts `large_q_check.py` and `full_table_logq.py` verify these values.

Brent–Kung giant steps The parameters are β and τ, with n^τ baby powers as in [16, Lemma 3], $t = 1-\beta$, and $0 \leq \tau \leq 1+t$. The terms are $1+\beta$, $2-\beta$, and $U(1, \tau, t+1-\tau)$ for the product of an $n \times n^\tau$ matrix by an $n^\tau \times kn^{1-\tau}$ matrix. Then Horner evaluation contributes $t+2-\tau$, and the powers contribute $1+\tau$. The optimum is 1.781251 at $\beta = 0.7813$, $\tau = 0.6094$ (basic: 1.785273). With the value $\omega = 2.375477$ of [10] and square blocks it is 1.814982, the value 1.815 of [16, Thm. 1].

Automorphism projection The program of Theorem E.4 has the terms of $\Lambda(1-\beta)$ and $\Lambda(\beta)$, κ, and $U(1-\beta, 1, \beta)$; with the baby steps computed by powering, $\Lambda(\beta)$ is replaced by $1+\beta$. With $\Lambda = \Lambda_2$, the optimum is $\omega_2/2 = 1.6250175$ at $\beta = \frac{1}{2}$, where $U(\frac{1}{2}, 1, \frac{1}{2})$ is the only active term, for both data sets. Similarly, with $\Lambda = \Lambda_1$ for the giant steps and the baby steps computed by powering, it is 1.651762 at $\beta = 0.5635$ (basic: 1.6553591 at $\beta = 0.5591$); with Λ_1 for both, it would be $\hat{\Lambda}_1(1/2) = 1.6856695$. Using one transposed composition per level instead of a batch gives the optimum 1.718858 at $\beta = 0.7105$ (basic: 1.728226).

Gain decomposition With the full table, the independent-composition route improves from 1.781251 to 1.714669 using [21], to 1.671417 using Theorem 6.21, and to 1.665414 using Theorem 6.12, and batching with two relation matrices then gives 1.553107. Alternatively, batching the construction of [21] with one relation matrix gives 1.621335 (the square-block value

is 1.627423), before the two-matrix improvement to 1.553107. If $\omega = 2$, the corresponding single-composition and batching steps are $5/3 \to 13/8 \to 3/2$.

B Proofs for batched composition

B.1 Monotonicity of the batch curves

Lemma 5.2 and Proposition 5.3 use that the batch costs are nondecreasing in k. For Λ_1, the program Eq. (3.2) depends on t only through the right-hand sides of its constraints. Increasing t shrinks the feasible set, and the value of a linear program is convex in a right-hand-side parameter. Similarly, for Λ_2 it follows from Eq. (4.2), since $\omega(\frac{1}{3}, \frac{1}{3}, t)$ is nondecreasing in t. In both cases the cost of a batch of size k is at least the cost of any smaller batch up to a constant factor, by padding the batch with zero polynomials.

B.2 The one-matrix cost and square-block optimization

Proof. Correctness. Suppose that Lemma 3.2 does not return Fail. By Proposition 2.4 and the discussion before it, μ_γ has degree n, $\phi_\gamma(\gamma) \equiv a \bmod f$ and $\gcd(\phi_\gamma, \mu_\gamma) = \gcd(a, f) = 1$, for the shifted data of the normalization (Table 4). Since $\deg g_i < n \le \lceil n/d \rceil d$, Lemma 3.4 applies with $D = d$ and $J = \lceil n/d \rceil$, and gives $g_i e_1 - \tilde{v}_i \in \mathcal{L}(P_\phi) \subseteq \mathcal{M}_{m'}^{(\phi_\gamma, \mu_\gamma)}$, meaning that $g_i(y) - \tilde{g}_i(x, y)$ vanishes at $y = \phi_\gamma$ modulo μ_γ. Also, $\deg_y \tilde{g}_i < 2d$ and $\deg_x \tilde{g}_i < m'$, so Lemma 3.5 (with $h = \mu_\gamma$, $b = \phi_\gamma$, $N = m'$, $\Delta = 2d$) returns $b_i = g_i(\phi_\gamma) \operatorname{rem} \mu_\gamma$. Then $b_i(y) = g_i(\phi_\gamma(y)) + u_i(y)\mu_\gamma(y)$ for some u_i, and evaluating at γ modulo f gives $b_i(\gamma) \equiv g_i(a) \bmod f$, since $\mu_\gamma(\gamma) \equiv 0$ and $\phi_\gamma(\gamma) \equiv a$, as in Eq. (2.3). The second stage computes $b_i(\gamma) \operatorname{rem} f$ by the same argument with (f, γ, P_γ), $\mathcal{L}(P_\gamma) = \mathcal{M}_m^{(\gamma, f)}$ and $\deg b_i < n$. Then undoing the shifts as in the final stage of Table 4 and [21, Remarks 3.8, 5.7] gives $g_i(a) \operatorname{rem} f$.

Cost. The costs indicated in Algorithm 3.2 follow from Lemmas 3.2, 3.4, and 3.5 and the remark after Lemma 3.5, with $N \le 2m$, $J \le m$, $\Delta_0 \le 2d$, $D = d$, using $\mathsf{MM}(2m, m, k) = O(\mathsf{MM}(m, m, k))$. Their sum is Eq. (3.1).

Failure. Fail can only be returned at the first line, i.e., by the first three stages of Table 4, and all subsequent steps are deterministic and never fail. By Proposition 2.4 this event depends on $(f, a, \mathbf{r})$ only and has probability at most $6n^2/\operatorname{card}(S)$ when f is separable; on its complement all k outputs are correct.

Exponent. Put $m = \lceil n^\eta \rceil$, $B = \lceil n^\varrho \rceil$ (so $s = \Theta(n^{1-\eta-\varrho})$) and $k = \lceil n^t \rceil$ in Eq. (3.1). By definition of $\omega(\cdot,\cdot,\cdot)$, $\mathsf{MM}(n^u, n^v, n^w) = O(n^{\omega(u,v,w)+\varepsilon})$ for every $\varepsilon > 0$, and $c(n, m, d) = O(m\, d^{\omega_2/2})$ with $d^{\omega_2/2} = n^{\omega((1-\eta)/2,(1-\eta)/2,1-\eta)}$ by homogeneity, the bound ω_2 being replaceable by any valid bound for $\omega(1, 1, 2)$. The seven terms of Eq. (3.2) are the exponents of the seven terms of Eq. (3.1), in this order, and the constraints are $m \le n$ and $B \le 2d$. Rounding m, d, B to integers changes each term by a constant factor. □

Remark B.1. For $k \le m$ the third term of Eq. (3.1) is at most $m^\omega d$, and for $k > m$ square blocking bounds it by $\lceil k/m \rceil m^\omega d$. The program Eq. (3.2) charges instead the sharper rectangular cost $d\,\mathsf{MM}(m, m, k)$ of Lemma 2.2(a), with which all numerical values below are computed. The batch size enters the algorithm only through the products of Lemmas 3.4 and 3.5.

We now solve the program Eq. (3.2) in closed form when only square products are used, i.e., when $\omega(\cdot)$ is replaced by the bound U_{sq} obtained from Eq. (2.2) with the points $(1,1,1) \mapsto \omega$, $(1,1,0) \mapsto 2$, $(1,0,0) \mapsto 1$ and their permutations.

Lemma B.2. *Let $2 \le \omega \le 3$ and $v \in \mathbb{R}^3_{\ge 0}$ with sorted entries $x \ge y \ge z$. Then $U_{\mathrm{sq}}(v) = x + y + (\omega - 2)z$.*

Proof. For the upper bound, $v = z(1,1,1) + (y-z)e' + (x-y)e''$, where e' has ones at the positions of x, y and e'' at the position of x, and the cost of this combination is $\omega z + 2(y - z) + (x - y)$.

For the lower bound, take the weight vector w equal to 1 at the positions of x and y and to $\omega-2 \in [0,1]$ at the position of z. It satisfies $w \cdot g \le \mathrm{val}(g)$ for every generator g, with equality for $(1,1,1)$. The values are at most 2 and 1 for the permutations of $(1,1,0)$ and $(1,0,0)$, respectively, because $\omega - 2 \le 1$. Thus the vector is dual feasible, with value $w \cdot v = x + y + (\omega-2)z$. □

Proposition B.3 (closed form and optimality in the square-block model)**.** *Let $\omega \in [2,3]$ be a feasible exponent. For every $k \ge 1$, Algorithm 3.2 can be run with cost*

$$\tilde{O}(n^{(\omega+2)/3}k^{(\omega-1)/3} + n^{\omega/2}k),$$

with $m = B = \lceil (nk)^{1/3} \rceil$ if $k \le n^{1/2}$ and $m = B = \lceil n^{1/2} \rceil$ if $k \ge n^{1/2}$. The first term dominates exactly when $k \le n^{1/2}$. Moreover, with U_{sq} in place of $\omega(\cdot)$, the value of Eq. (3.2) is exactly

$$\Lambda_{\mathrm{sq}}(t) = \max\Big(\frac{\omega+2}{3} + \frac{\omega-1}{3}\,t,\quad t + \frac{\omega}{2}\Big),$$

i.e., no choice of (η, ϱ) does better in this cost model.

Proof. Write $L_1(t) = \frac{\omega+2}{3} + \frac{\omega-1}{3}t$ and $L_2(t) = t + \frac{\omega}{2}$. They agree at $t = \frac{1}{2}$, and L_1 has slope $\frac{\omega-1}{3} < 1$, so $\max(L_1, L_2)$ is L_1 on $[0, \frac{1}{2}]$ and L_2 on $[\frac{1}{2}, \infty)$. We use Lemma B.2 throughout, and write $T_{\mathrm{P1}}, \dots, T_{\mathrm{out}}$ for the seven terms of Eq. (3.2) in order.

Upper bound, $t \le \frac{1}{2}$. Take $\eta = \varrho = \frac{1+t}{3}$, and then $\varrho \le 1 - \eta$ and $t + 1 - \eta - \varrho = \frac{1+t}{3}$. We obtain $T_{\mathrm{P1}} = 1 + (\omega-1)\eta = L_1(t)$ and $T_{\mathrm{P2}} = \eta + \frac{\omega+1}{2}(1-\eta) \le 1 + (\omega-1)\eta$ because $\eta \ge \frac{1}{3}$. Also $T_{\mathrm{A}} = 1 - \eta + 2\eta + (\omega-2)t \le T_{\mathrm{P1}}$ since $t \le \eta$, and $T_{\mathrm{B1}} = \eta + (1-\eta) + \varrho + (\omega-2)\varrho = L_1(t)$. The remaining terms satisfy $T_{\mathrm{B2}} = \frac{4+t}{3} \le L_1(t)$ and $T_{\mathrm{B3}} = 1 + \varrho \le L_1(t)$, with $T_{\mathrm{out}} = 1 + t \le L_1(t)$ as $t \le \frac{1}{2}$.

Upper bound, $t \ge \frac{1}{2}$. Taking $\eta = \varrho = \frac{1}{2}$ gives $T_{\mathrm{P1}} = \frac{\omega+1}{2}$, $T_{\mathrm{P2}} = \frac{1}{2} + \frac{\omega+1}{4}$, $T_{\mathrm{A}} = \frac{1}{2} + t + \frac{1}{2} + \frac{\omega-2}{2} = L_2(t)$, $T_{\mathrm{B1}} = \frac{1}{2} + U_{\mathrm{sq}}(t, \frac{1}{2}, \frac{1}{2}) = L_2(t)$, $T_{\mathrm{B2}} = T_{\mathrm{out}} = t + 1$, $T_{\mathrm{B3}} = \frac{3}{2}$, and all are at most $L_2(t)$.

Lower bound. Let (η, ϱ) be feasible and $\delta = 1 - \eta$. By Lemma B.2 and $\min(t + \delta - \varrho, \varrho) \le \frac{t+\delta}{2}$,

$$T_{\mathrm{B1}} \ge \eta + (t + 2\delta) - (3-\omega)\tfrac{t+\delta}{2} = 1 + \tfrac{\omega-1}{2}(t + 1 - \eta),$$

so $\frac{1}{3}T_{\mathrm{P1}} + \frac{2}{3}T_{\mathrm{B1}} \ge \frac{1}{3}(1 + (\omega-1)\eta) + \frac{2}{3}\big(1 + \frac{\omega-1}{2}(t+1-\eta)\big) = L_1(t)$, and the maximum is at least $L_1(t)$ for every t. Let now $t \ge \frac{1}{2}$. If $\eta \le t$, then $T_{\mathrm{A}} = 1 + t + (\omega-2)\eta$ and, using $\min(t + \delta - \varrho, \varrho) \le \delta$, $T_{\mathrm{B1}} \ge t + \omega - 1 - (\omega-2)\eta$, and the average of the two is $L_2(t)$. If $\eta > t$, then $T_{\mathrm{P1}} = 1 + (\omega-1)\eta > 1 + (\omega-1)t \ge L_2(t)$, the last inequality being $(\omega-2)(t - \frac{1}{2}) \ge 0$.

Finally, $n^{L_1} \ge n^{L_2}$ if and only if $(k^2/n)^{(4-\omega)/6} \le 1$, i.e., $k \le n^{1/2}$. For the cost bound with the stated parameters, apply Theorem 3.6(i). Every term of Eq. (3.1) is $\tilde{O}(n^{\max(L_1, L_2)})$ by the preceding computation with integer parameters. Use square blocks $\mathsf{MM}(u,v,w) = O(\lceil u/z \rceil \lceil v/z \rceil \lceil w/z \rceil z^{\omega})$, where $z = \min(u,v,w)$, and $\omega_2 \le \omega + 1$. □

Corollary B.4 (the case $\omega = 2$)**.** *If $n \times n$ matrices can be multiplied with $n^{2+o(1)}$ operations, then for every $k \ge 1$,*

$$C(n,k) \le n^{o(1)}(n^{4/3}k^{1/3} + nk);$$

in particular $C(n,k) = (nk)^{1+o(1)}$, the size of the output, for every $k \ge n^{1/2}$.

Proof. Apply Proposition B.3 with $\omega = 2 + \varepsilon$ for every $\varepsilon > 0$, and observe that nk is the number of output coefficients. □

Proof of Theorem 3.6(iv). Run Algorithm 3.2 with the parameters of Proposition B.3, using a random tuple of $n+m$ entries with $m = O(n^{1/2} + (nk)^{1/3})$. The cost follows from Proposition B.3, and for $\omega = 2$ from Corollary B.4. □

B.3 Details of the numerical comparison

Table 5 lists values of $\hat{\Lambda}_1$. At $t = 0$, $\hat{\Lambda}_1(0) = \kappa$, the exponent of [21, Thm. 1.1] with the same bounds. For $t \geq 1$,

$$\Lambda_1(t) \leq \hat{\Lambda}_1(t) = t + \frac{\omega_2 - 1}{2} \qquad (t \geq 1),$$

where ω_2 is the stated upper bound, so that the amortized cost is $n^{(\omega_2-1)/2}$ per composition, below the square value $n^{\omega/2}$. Indeed, taking $\eta = \varrho = \frac{1}{2}$ gives $T_{\mathrm{A}} = T_{\mathrm{B1}} = \frac{1}{2} + U(\frac{1}{2}, \frac{1}{2}, t) \leq t + \frac{\omega_2-1}{2}$ by splitting the third dimension into blocks of size n^1. Conversely, take the weight vector $(1, c, c)$ with $c = \frac{\omega_2-2}{2}$, placing its first coordinate at the largest entry. This vector and its permutations are dual feasible for Eq. (2.2) with the generators of Appendix A, giving $T_{\mathrm{A}} \geq t + 1 + (\omega_2 - 3)\eta$ and $T_{\mathrm{B1}} \geq t + \omega_2 - 2 + (3 - \omega_2)\eta$, whose average is $t + \frac{\omega_2-1}{2}$. Against batched Brent–Kung with square blocks and at most n baby powers, the difference between its exponent and $\Lambda_{\mathrm{sq}}(t)$ is

$$\begin{cases} \dfrac{\omega - 1}{6}(1 + t), & 0 \leq t \leq \frac{1}{2}, \\ \dfrac{1 - (3 - \omega)t}{2}, & \frac{1}{2} \leq t \leq 1, \\ \dfrac{\omega - 2}{2}, & t \geq 1. \end{cases}$$

The last branch follows by computing all n baby powers and blocking the product of an $n \times n$ by an $n \times k$ coefficient matrix. For example, at $t = 1$ the one-matrix algorithm costs $n^{2.125018}$ against n^{ω} for the direct method, which computes all powers of a followed by one product. If $\omega = 2$, both costs are $n^{2+o(1)}$, the size of the output. For $k < n^{1/2}$, the optimum of Proposition B.3 refers to square blocking.

B.4 Generating arguments

B.4.1 Proof of Proposition 4.2

Proof. Let $\varphi_1, \ldots, \varphi_n \in \overline{\mathbb{K}}$ be the roots of f, pairwise distinct, and $\lambda_\nu = a(\varphi_\nu)$, pairwise distinct since a generates $\mathcal{A}$. The roots of $\hat{f}$ are $\hat{\varphi}_\nu = \varphi_\nu - r_2$, and $\hat{a}(\hat{\varphi}_\nu) = \lambda_\nu + r_1 =: \hat{\lambda}_\nu$, still pairwise distinct. For γ, write $\xi_\nu = \gamma(\hat{\varphi}_\nu)$; the map from the coefficient vector of γ to $(\xi_\nu)_\nu$ is linear and invertible (Vandermonde), so a nonzero polynomial in the ξ_ν of degree D is a nonzero polynomial of degree D in the coefficients of γ, to which the Schwartz–Zippel lemma applies over $\overline{\mathbb{K}}$.

(a) The two tests fail only at the roots of $\mathrm{Res}_x(a + r_1, f)$ and of $f(r_2)$, two nonzero polynomials of degree n in r_1 and in r_2, giving failure probability at most $2n/\operatorname{card}(S)$.

Fix r_1, r_2 such that (a) holds, so that the remaining events concern γ only.

(b) Use $\Delta_0 = \mathrm{Res}(\gamma, \hat{f})$, of degree n, and $\Delta_1 = \prod_{\nu<\nu'}(\xi_\nu - \xi_{\nu'})$, of degree $n(n-1)/2$. Off their zero sets, the ξ_ν are nonzero and pairwise distinct, $\mu_\gamma = \prod_\nu (y - \xi_\nu)$ is separable of degree n, and $\mu_\gamma(0) \neq 0$.

(c) By [21, Prop. 7.6], which uses $\hat{f}(0) \neq 0$, there is a nonzero polynomial $\Delta_{\hat{f},m}$ of degree at most $2n^2/m$ off whose zero set the rank is n.

(d) Under (b), $\mathbb{K}[x]/\langle \hat{f} \rangle \to \mathbb{K}[y]/\langle \mu_\gamma \rangle$, $\gamma \mapsto y$, is an isomorphism mapping $\hat{a}$ to ϕ_γ, with $\phi_\gamma(\xi_\nu) = \hat{\lambda}_\nu$. Thus ϕ_γ takes n pairwise distinct values at the n distinct nonzero roots of μ_γ. By [21, Prop. 7.8], applied with all multiplicities equal to 1, there is a polynomial $\Gamma(\bar{\xi}, \bar{\lambda})$ such that, for every fixed tuple λ of pairwise distinct values, $\Gamma(\bar{\xi}, \lambda)$ is a nonzero polynomial of degree at most $2n^2$ in $\bar{\xi}$, and $\operatorname{rank} \mathrm{Hk}_{m,d}^{(\phi_\gamma, \mu_\gamma)} = n$ whenever $\Gamma(\xi, \hat{\lambda}) \neq 0$ and $d \geq \lceil n/m \rceil$.

(e) By Lemma 2.6, and since every minimal basis of $\mathcal{N}_{\bar{m}}$ has degree at least $\lceil n/\bar{m} \rceil$ (Proposition 2.5), it suffices that the matrices of Lemma 2.6 with $D = \bar{d}$ have rank n.

For $(\phi_\gamma, \mu_\gamma)$, under (b), Lemma 4.1 with $h = \mu_\gamma$, $\zeta_\nu = \xi_\nu$ and $b = \phi_\gamma$ reduces this to the rank of $E(\xi) = [\xi_\nu^i \hat{\lambda}_\nu^j]_{\nu;\, i<\bar{d},\, j<\bar{m}}$. Each $c \in \{0, \ldots, n-1\}$ is written uniquely as $c = i\bar{m} + j$ with

$j < \bar{m}$, and then $i < \bar{d}$ because $c < n \le \bar{d}\bar{m}$. Let $\Psi_1(\xi)$ be the $n \times n$ minor of E on these n columns, which is a polynomial in ξ of degree at most $n(\bar{d}-1) \le n^2$. At the point $\xi_\nu = \hat{\lambda}_\nu^{\bar{m}}$ the entry in column c is $\hat{\lambda}_\nu^{i\bar{m}+j} = \hat{\lambda}_\nu^{c}$, so Ψ_1 takes the value of the Vandermonde determinant of the pairwise distinct $\hat{\lambda}_\nu$, which is nonzero; hence Ψ_1 is a nonzero polynomial, even if the $\hat{\lambda}_\nu^{\bar{m}}$ are not pairwise distinct.

Similarly, for $(\gamma, \hat{f})$, Lemma 4.1 with $h = \hat{f}$, $\zeta_\nu = \hat{\varphi}_\nu$ and $b = \gamma$ reduces it to the rank of $E'(\xi) = [\hat{\varphi}_\nu^i \xi_\nu^j]_{\nu;\, i<\bar{d},\, j<\bar{m}}$. Writing $c = i + j\bar{d}$ with $i < \bar{d}$ (and then $j < \bar{m}$), let $\Psi_2(\xi)$ be the minor on these columns, of degree at most $n(\bar{m}-1) \le n^2$ in ξ, and at $\xi_\nu = \hat{\varphi}_\nu^{\bar{d}}$ it is the Vandermonde determinant of the pairwise distinct $\hat{\varphi}_\nu$.

The product $\Delta_0\Delta_1\Delta_{\hat{f},m}\Gamma\Psi_1\Psi_2$ is a nonzero polynomial in the coefficients of γ of degree at most $n + n(n-1)/2 + 2n^2 + 2n^2 + 2n^2$, and off its zero set (b)–(e) hold. Adding the $2n/\operatorname{card}(S)$ of (a), the failure probability is at most $(6.5n^2 + 2.5n)/\operatorname{card}(S) \le 8n^2/\operatorname{card}(S)$ for $n \ge 2$. □

B.4.2 Proof of Proposition 4.3

Proof. Let $\varphi_1, \ldots, \varphi_\ell$ be roots with $a(\varphi_\nu) = \lambda$. By Lemma 4.1, restricted to these coordinates the vectors $x^i a^j$ $(i < D,\ j < \bar{m})$ become $\lambda^j(\varphi_\nu^i)_{\nu \le \ell}$, which span a space of dimension at most D. Full rank therefore forces $D \ge \ell$, and Lemma 2.6 gives the first claim. The second is the formula $\deg\det \mathcal{M}_m^{(a,f)} = \sum_\nu \min(\ell_\nu, m)$ of Lemma 6.15, where ℓ_ν are the multiplicities of the values of a. A change of basis preserves the multiset of values, since $\phi_\gamma(\xi_\nu) = \hat{a}(\hat{\varphi}_\nu)$. □

B.4.3 Proof of Lemma 4.4

Proof. In both algorithms R is the leading $m \times m$ block of an unshifted weak Popov approximant basis P of $(S, -I_m)$ at order $2d$, of dimension $2m$. Its column degrees sum to $\deg\det P \le 2md$, because the quotient by the approximant module injects into m series truncated at order $2d$. For ChangeOfBasis, write the shifted approximant basis as $\left(\begin{smallmatrix} P & u_0 \\ z & \lambda \end{smallmatrix}\right)$; the shift $(0, \ldots, 0, 2d)$ and reducedness force $\deg z + 2d < \deg P \le 2d$, hence $z = 0$, so that the extracted block P is again the unshifted basis. To return Cert, or to avoid Fail in the change of basis [21, Algorithm 5.1, Step 3; Algorithm 6.1, Step 5], the certificate requires two conditions: the diagonal degrees of the leading block R sum to n (to at least n in ChangeOfBasis), and each of the m rightmost columns of P has degree at least $\deg R$. The second condition is a separate part of the test, since the weak Popov property permits very unequal column degrees. Hence

$$2md \;\ge\; \sum_{j=1}^{2m} \operatorname{cdeg}_j P \;\ge\; n + m\deg R,$$

that is, $\deg R \le 2d - n/m < d+1$, and since degrees are integers, $\deg R \le d$. The Popov form has degree at most $\deg R$ by Lemma 2.3. □

B.4.4 Proof of Lemma 4.5

Proof. By Lemma 3.4, $g_i e_1 - \tilde{v}_i \in \mathcal{M}_m^{(b,h)}$, so that $G_i(x,b) \equiv g_i(b) \bmod h$, and $\deg_y G_i < d$ because R has degree at most d. Since $d \le \bar{m}^2$, every $c < d$ is $c_0 + \bar{m}c_1$ with $c_0, c_1 < \bar{m}$, and the giant steps give

$$G_i(x,b) = \sum_{c_1<\bar{m}} s_{i,c_1}(x,b)\, b^{\bar{m}c_1} \equiv \sum_{c_1<\bar{m}} s_{i,c_1}(x,b)\, A_{c_1}(x,b) = S_i(x,b) \pmod h,$$

of which the baby steps compute the remainder $\sum_\ell S_{i,\ell}\,(b^\ell \operatorname{rem} h) \operatorname{rem} h$. For the cost, with $m = \bar{m} = \lceil n^{1/3} \rceil$ we have $d, \bar{d}, d' \in \Theta(n^{2/3})$ and $L', \lceil n/d' \rceil \in \Theta(n^{1/3})$. The reduction costs $\tilde{O}(d\,\mathsf{MM}(m,m,k))$ by Lemma 3.4. The giant steps multiply a $k \times \bar{m}$ by a $\bar{m} \times L'$ matrix with entries in $\mathbb{K}[x,y_0]_{<(m,\bar{m})}$, at cost $\tilde{O}(m\bar{m}\,\mathsf{MM}(k,\bar{m},L'))$, and the baby steps multiply a $k \times 2\bar{m}$ by

a $2\bar{m} \times \lceil n/d' \rceil$ matrix over $\mathbb{K}[x]_{<d'}$, at cost $\tilde{O}(d'\,\mathsf{MM}(k, \bar{m}, \lceil n/d' \rceil))$; the recombinations and reductions add $\tilde{O}(kn)$. By the symmetry of MM, all three products cost $\tilde{O}(n^{2/3}\mathsf{MM}(n^{1/3}, n^{1/3}, k))$. The computation is linear because every step multiplies data derived from the g_i by fixed data (R through Lemma 2.2(b), the remainder table, the A_{c_1}, the powers $b^\ell \operatorname{rem} h$, and h). □

B.4.5 Proof of Theorem 4.6

Proof. Correctness. If no Fail is returned, then $f(0) \neq 0$ and $\gcd(a, f) = 1$ for the shifted data. By [21, Prop. 6.1], R_γ is the Popov basis of $\mathcal{M}_m^{(\gamma,f)}$, μ_γ has degree n, and $\phi_\gamma(\gamma) \equiv a \bmod f$. The flag is Cert, so R' is a basis of $\mathcal{M}_m^{(\phi_\gamma,\mu_\gamma)}$ (Proposition 2.7). Its Popov form R_ϕ is also a basis, row reduced of degree at most $\deg R'$ (Lemma 2.3). Moreover $\deg R_\gamma, \deg R_\phi \leq \bar{d}$ by Lemma 4.4, which gives $\deg_y G_i < d$ in Evaluate. Finally the $\mathcal{N}$-bases have degree $\bar{d}$, so Proposition 2.5 provides $A_{c_1} \in \mathbb{K}[x, y_0]_{<(\bar{d},\bar{m})}$. Lemma 4.5 applied to $(\phi_\gamma, \mu_\gamma, R_\phi)$ gives $u_i = g_i(\phi_\gamma) \operatorname{rem} \mu_\gamma$, and as in the proof of Theorem 3.6, $u_i(\gamma) \equiv g_i(a) \bmod f$, and Lemma 4.5 applied to (γ, f, R_γ) gives $b_i = u_i(\gamma) \operatorname{rem} f$. Undoing the shifts gives the result.

Cost. The precomputation costs $\tilde{O}(m^\omega d + c(n, m, d) + \bar{m}^{\omega-1}n + \bar{m}n)$ by [21, Prop. 6.1], Propositions 2.5 and 2.7 and Lemmas 2.2, 2.3, and 3.3. With $m = \bar{m} = \lceil n^{1/3} \rceil$ this is $\tilde{O}(n^{(\omega+2)/3} + n^{(\omega_2+1)/3}) = \tilde{O}(n^{(\omega+2)/3})$, since $\omega_2 \leq \omega + 1$. The two calls to Evaluate cost $\tilde{O}(n^{2/3}\mathsf{MM}(n^{1/3}, n^{1/3}, k) + kn)$ by Lemma 4.5, and $kn \leq n^{2/3}\mathsf{MM}(n^{1/3}, n^{1/3}, k)$ by our convention on MM. The exponent Eq. (4.2) follows from the definition of $\omega(\cdot, \cdot, \cdot)$ and homogeneity.

Failure. All Fail exits occur in the precomputation and depend on (f, a, r_1, r_2, γ) only. Suppose all events (a)–(e) of Proposition 4.2 hold, with $m = \bar{m} = \lceil n^{1/3} \rceil$. The first line succeeds by (a). By (b), (c) and [21, Props. 6.1 and 7.1], ChangeOfBasis succeeds and returns a basis of degree at most d, with $\mu_\gamma(0) \neq 0$ by (b). Then, by (d) and Proposition 2.7, CandidateBasis returns Cert and a basis of degree at most d. Its Popov form has degree at most d by Lemma 2.3, and the $\mathcal{N}$-bases have degree $\bar{d}$ by (e). Hence $\Pr[\mathsf{Fail}] \leq 8n^2/\operatorname{card}(S)$ by Proposition 4.2.

Special cases. For $k \geq n^{1/3}$, square blocks give

$$n^{2/3}\mathsf{MM}(n^{1/3}, n^{1/3}, k) = O(\lceil k/n^{1/3} \rceil n^{2/3} n^{\omega/3}) = O(kn^{(\omega+1)/3}).$$

If $\omega = 2$, then $\omega(\frac{1}{3}, \frac{1}{3}, t) = \max(\frac{2}{3}, \frac{1}{3} + t)$, so that $\Lambda_2(t) = \max(\frac{4}{3}, 1 + t)$. □

C Proofs for fast certified change of basis

Throughout this appendix, the parameters are those of (6.1), and relation vectors are written as columns.

C.1 Proof of Lemma 6.2

Proof. For a polynomial of y-degree less than μ, put $Jr(x, y) = y^{\mu-1} r(x, y^{-1})$, a $\mathbb{K}[x]$-linear involution which preserves x-degree. If $r(x, b) = 0 \bmod h$, then $(Jr)(x, b^{-1}) = b^{-(\mu-1)} r(x, b) = 0 \bmod h$. Thus J gives an isomorphism between the two modules, reversing the μ rows of a basis matrix in coefficient vectors. It therefore preserves each column degree and the rank of the column-leading matrix, and carries a column-reduced basis to a column-reduced basis. This proves the assertion, including when the column degrees are unequal. □

C.2 Proof of Lemma 6.3

Proof. Proposition 2.3 and Corollary 2.4 of [22] apply to every b, h and construct polynomials $A_j, B_j \in \mathbb{K}[x, y]_{<(\delta,\mu)}$ satisfying

$$A_j(x, b) = b^{j\mu} \bmod h, \qquad B_j(x, b) = b^{j\mu^2} \bmod h, \qquad 0 \leq j < \mu.$$

For the first assertion, apply [22, Prop. 4.1] with base b, multiplier c, width w and length μ^3, and truncate the sequence; in that statement the degree parameter belongs to the base, and the multiplier is arbitrary. The auxiliary data represent $b^{j\mu}$ and $b^{j\mu^2}$ by polynomials of bidegree less than (δ, μ); by the conditions $2\delta \le n+1$ and $w+\delta \le n$, the high parts use the coefficients of indices $n-2\delta+1$ to $n-1$ and the low parts those of indices below w. The initial $O(\mu)$ modular products, the high and low parts and the products of polynomial matrices cost $\tilde{O}(\mu n + \mu^2\delta + w\mu^3 + \mu^\omega(\delta + w\mu))$, which is $\tilde{O}(\mu^\omega(\delta + w\mu))$ since $n \le \mu\delta$ and $\omega \ge 2$.

For the block assertion, use this result once with $c = x^{m-1} \operatorname{rem} h$ and width $2m-1$. For each k set

$$T_{m-1,k} = [x^{m-1} b^k \operatorname{rem} h]_{<2m-1},$$
$$T_{i-1,k} = \frac{T_{i,k} - T_{i,k}(0)h/h(0)}{x} \operatorname{rem} x^{m+i-1} \qquad (i = m-1, \dots, 1).$$

The numerator has zero constant coefficient. Moreover, if $u = x^{i-1} b^k \operatorname{rem} h$ and $v = xu \operatorname{rem} h$, writing the leading coefficient of h as h_n gives $v = xu - u_{n-1}h/h_n$, and hence $v - v(0)h/h(0) = xu$. Truncating this identity proves by downward induction that $T_{i,k} = [x^i b^k \operatorname{rem} h]_{<m+i}$. Only the first $m+i$ coefficients of h are needed at an iteration. Then $\Gamma_{i,k} = [T_{i,k}]_{<m}$, and the recurrence uses $O(m^2)$ operations for each k. It is exactly the reconstruction in [21, Algorithm 3.8 and Proposition 3.7], and only its first truncated-power call has been replaced. □

C.3 Proof of Lemma 6.4

Proof. For $n < 16$, run the original Algorithm 6.1 with the additional test $\deg R \le d$, at cost bounded by an absolute constant. Correctness and sufficient success conditions follow from Propositions 6.1 and 7.1 of [21]. For $n \ge 16$, we retain Step 1 of Algorithm 6.1, which computes $b = \gamma^{-1} \operatorname{rem} h$, and compute a minimal basis of $\mathcal{N}_m^{(b,h)}$ by [22, Proposition 2.3]. If its degree is not d, we return Fail before computing any truncated powers; otherwise we compute its A_j, B_j data. Under the success hypotheses, this degree test passes by Lemma 6.2.

We replace Step 2 by Lemma 6.3 with multiplier $c = bu \operatorname{rem} h$, width m, and length $2d$, obtaining exactly $r_k = [u\gamma^{-k-1} \operatorname{rem} h]_{<m}$ for $0 \le k < 2d$. Replace Step 3 by the block construction of that lemma with base b and length $\ell^* = 2d+2$, then form the original arrays:

$$s(y) = \sum_{k=0}^{2d-1} \operatorname{coeff}_m(r_k) y^k, \qquad S(y) = \sum_{k=0}^{2d-1} S_k y^k,$$
$$S_k = -[\operatorname{coeff}_m(\Gamma_{0,k+1}) \ \cdots \ \operatorname{coeff}_m(\Gamma_{m-1,k+1})].$$

Here coeff_m denotes a column of m coefficients, so $s \in \mathbb{K}[y]^m_{<2d}$ and $S \in \mathbb{K}[y]^{m\times m}_{<2d}$ have exactly the signs, columns, and lengths required by Algorithm 6.1. We keep Steps 4–7 of Algorithm 6.1, that is, the reconstruction, both certification tests, the test on $\deg \chi$, the Hermite form and the recovery of ϕ, and we return Fail if $\deg R > d$. By [21, Proposition 6.1], every returned triple has the claimed meaning, with no condition on u.

Under the success hypotheses, [21, Proposition 7.1] shows that the truncated and full relation modules have determinantal degree n, and that $X^{\mathsf{T}}(yI - M_\gamma)^{-1}X$ is describable in degree d. The success clause of Proposition 6.1 therefore applies, and gives $\deg R \le d$ as well.

The costs below include the unchanged steps of that proposition, with logarithmic factors

suppressed in the right column.

gcd, inversion, and the two multipliers	$\tilde{O}(n)$,
auxiliary basis and A_j, B_j	$\tilde{O}(m^{\omega-1}n)$,
Step 2, arbitrary-multiplier powers	$\tilde{O}(m^{\omega}(d+m^2))$,
Step 3, the single wide sequence	$\tilde{O}(m^{\omega}(d+(2m-1)m))$,
Step 3, block reconstruction	$O(m^2(2d+2))$,
Step 4, shifted approximant basis	$\tilde{O}(m^{\omega}d)$,
Step 5, certification and degree tests	$O(m^2d)$,
Step 6, Hermite form	$\tilde{O}(m^{\omega}d)$,
Step 7, recovery of ϕ	$\tilde{O}(nm)$.

For the recovery, if T is the upper triangular Hermite form and $\bar{\phi}$ the recovered vector remainder, then $\phi = \bar{\phi}_1 - \sum_{j=2}^{m} T_{1,j}\bar{\phi}_j \operatorname{rem} \chi$, which needs $m-1$ products of degree $O(n+d)$. Both $m^{\omega-1}n$ and $m^{\omega}d$ have exponent $(\omega+3)/4$, and the other possible exponents are $(\omega+2)/4$ and $5/4$, which do not exceed it for $\omega \geq 2$. The auxiliary basis is computable within its displayed cost even on bad inputs, and testing its degree before use makes the cost uniform for the complete procedure in Algorithm 6.1. □

C.4 Proof of Lemma 6.5

Proof. For $n < 16$ use Horner evaluation, which always succeeds and has bounded cost, and assume henceforth that $n \geq 16$. Compute the auxiliary bases and the A_j, B_j data separately for b and $b^{-1} \operatorname{rem} h$, rejecting any basis of degree different from d before use, at cost $\tilde{O}(m^{\omega-1}n)$ by [22, Proposition 2.3 and Corollary 2.4]. The two degree tests are equivalent by Lemma 6.2.

If no relation basis is supplied, compute $[x^i b^{-k-1} \operatorname{rem} h]_{<m}$ for $i < m$, $k < 2d$ using Lemma 6.3, and feed them to the approximant reconstruction and certificate of Candidate-Basis [21, Algorithm 5.1]; we require the flag Cert and a returned basis of degree at most d, returning Fail otherwise. Propositions 5.6 and 7.1 of [21] imply both that certification is sound and that these tests pass under the stated rank hypothesis. This step costs $\tilde{O}(m^{\omega}d)$ in addition to the powers, whose cost is given in Lemma 6.4.

Next, reduce v by the certified basis using [22, Lemma 2.7], which in $\tilde{O}(m^{\omega}(d+n/m))$ operations produces $V \in \mathbb{K}[x,y]_{<(m,d)}$ satisfying $V(x,b) = v(b) \bmod h$. Then [22, Proposition 3.1], using the data for b, evaluates $V(x,b) \operatorname{rem} h$ in $\tilde{O}(m^{\omega}(d+m^2))$ operations, with the hypothesis $d \leq m^3$ following from (6.1). All computations after the certified relation basis are deterministic and exact. The same cost comparison as in Lemma 6.4 proves the bound. □

C.5 Proof of Lemma 6.6

Proof. Let $\varphi_1, \ldots, \varphi_n$ be the distinct roots of f in an algebraic closure, and put $\lambda_i = a(\varphi_i)$, whose values are pairwise distinct since a is a generator. The bad choices of r_2 for $h(0) \neq 0$ and of r_1 for u to be a unit are the zeros of two nonzero polynomials of degree n, giving probability at most $2n/\operatorname{card}(S)$. Condition on any remaining choices of these shifts, and write $\zeta_i = \varphi_i - r_2$ and $\eta_i = \lambda_i + r_1$, both lists consisting of distinct nonzero elements.

Put $\xi_i = \gamma(\zeta_i)$, and note that evaluation at the ζ_i is an invertible linear map on the coefficients of γ, so nonzero polynomials in the ξ_i remain nonzero with the same degree after this substitution. The product of $\prod_i \xi_i$ and $\prod_{i<j}(\xi_i - \xi_j)$ has degree $n + n(n-1)/2$ and ensures that γ is a unit and a generator, giving $\chi(y) = \prod_i (y - \xi_i)$, $\chi(0) \neq 0$, and $\phi(\xi_i) = \eta_i$.

For (γ, h), [21, Prop. 7.6] supplies a nonzero polynomial of degree at most $2n^2/m$ ensuring $\operatorname{rank} \mathrm{Hk}_{m,d} = n$. For (ϕ, χ), [21, Prop. 7.8] with multiplicities one gives a polynomial $\Gamma(\xi, \eta)$ of degree at most $2n^2$ in ξ. For every fixed tuple of distinct η_i it is nonzero as a polynomial in ξ, and outside its zero set the second Hankel rank is n.

It remains to control the auxiliary modules. By [22, Lemma A.1], degree at most d is equivalent to rank n of the coefficient matrix of $x^i b^j \operatorname{rem} h$ with $i < d$, $j < m$. Evaluation at the roots preserves rank, and for (ϕ, χ) the resulting matrix is $[\xi_\nu^i \eta_\nu^j]_{\nu;(i,j)}$. Select the n columns $c = im + j$, $0 \le c < n$, whose minor has degree at most $n(d-1)$ in ξ. At $\xi_\nu = \eta_\nu^m$ it is the Vandermonde determinant on the distinct η_ν, and hence is a nonzero polynomial, as in the proof of Proposition 4.2. For (γ, h) use $[\zeta_\nu^i \xi_\nu^j]$, and select the columns $c = i + jd$, $0 \le c < n$. The corresponding minor has degree at most $n(m-1)$ in ξ and is nonzero at $\xi_\nu = \zeta_\nu^d$, again by the Vandermonde determinant. Every minimal $\mathcal{N}_m$ basis has degree at least d, because its determinant has degree n [22, Proposition 2.3], so the two degrees are exactly d.

The product of the six nonzero polynomials just described has total degree at most

$$n + \frac{n(n-1)}{2} + \frac{2n^2}{m} + 2n^2 + n(d-1) + n(m-1) \le \frac{13}{2}n^2 + \frac{1}{2}n.$$

Schwartz–Zippel applies over the algebraic closure with the coefficient variables independently sampled from S. Adding the two shift failures bounds the total failure probability by $(13n^2/2 + 5n/2)/\operatorname{card}(S) \le 8n^2/\operatorname{card}(S)$ for $n \ge 2$. The inverse conditions follow from Lemma 6.2, without adding any bad event. □

C.6 Proof of Theorem 6.1

Proof of Theorem 6.1. For $n < 16$, direct Horner evaluation proves the assertion, with no failure, while for $n \ge 16$ we form $h = f(x + r_2)$, $u = (a + r_1)(x + r_2)$, and $v(y) = g(y - r_1)$. Return Fail if $h(0) = 0$ or $\gcd(u, h) \ne 1$. Apply Lemma 6.4 to (h, γ, u), obtaining (R_γ, χ, ϕ) or returning Fail. Check $\chi(0) \ne 0$ and $\gcd(\phi, \chi) = 1$, and return Fail if either check fails. Use Lemma 6.5 first to compute $z = v(\phi) \operatorname{rem} \chi$ and then to compute $w = z(\gamma) \operatorname{rem} h$, supplying R_γ in the latter call. Any failure terminates this attempt; otherwise we return $w(x - r_2)$.

Every successful execution has $\chi(\gamma) = 0 \bmod h$ and $\phi(\gamma) = u \bmod h$ by the certified change of basis. Then the certified compositions give

$$w = z(\gamma) = v(\phi(\gamma)) = v(u) = g(a(x + r_2)) \pmod h.$$

Undoing the shift gives $g(a) \operatorname{rem} f$. Lemma 6.6 implies that all the stated tests and the three certified procedures succeed on a single event of probability at least $1 - 8n^2/\operatorname{card}(S)$.

Each procedure has cost $\tilde{O}(n^{(\omega+3)/4})$ on every execution. The shifts, gcd tests, and final inverse shift cost $\tilde{O}(n)$, and polynomial translation can be performed by divide and conquer in arbitrary characteristic. These costs include sampling $n + 2$ field elements under the sampling convention of Section 2. Independent repetition has expected at most two attempts if $\operatorname{card}(S) \ge 16n^2$.

Finally, for $\mathbb{K} = \mathbb{F}_q$, if $q \ge 16n^2$ use $S = \mathbb{F}_q$. Otherwise choose the least positive e such that $q^e \ge 16n^2$, construct $E = \mathbb{F}_{q^e}$, and use $S = E$. Here $e = O(\log n)$, and arithmetic in E costs $\tilde{O}(e)$ operations in $\mathbb{F}_q$. An irreducible polynomial defining E can be constructed by the Las Vegas algorithm of [36, Section 14.9], and at this degree its expected cost is $(\log n)^{O(1)} \log q$, as in the extension-field convention of Section 2. Since the construction is used only when $q < 16n^2$, this overhead is polylogarithmic in n and is absorbed by the asserted bound. Separability and the linear independence of $1, a, \ldots, a^{n-1}$ persist under extension of scalars. The computed remainder has coefficients in $\mathbb{F}_q$, by uniqueness of the remainder of the original composition, so extracting those coefficients costs $\tilde{O}(ne)$. This proves the expected base-field bound. □

D Proofs for every argument

D.1 Proof of Lemma 6.13

Proof. On the line $Y = c_k$ the rows are evaluations of polynomials of degree less than d in one variable, so this line contributes at most $\min(\ell_k, d)$ to the rank, and the number of columns

is $d\mu$. Let $r_{\min}$ be the minimum and keep $s_k \le \min(\ell_k, d)$ rows on line k, $\sum s_k = r_{\min}$. Then list the kept rows line by line and give them the labels $0, 1, \dots, d-1, 0, 1, \dots$ cyclically. A line receives each label at most once, and each label is used at most $\lceil r_{\min}/d \rceil \le \mu$ times. Work over an infinite extension containing the c_k and choose distinct $\tau_0, \dots, \tau_{d-1}$ there. Specialize the abscissa of a row with label h to τ_h and pass to the Lagrange basis $(u_h)_{h<d}$ at the τ_h. Then keep the columns $u_h(X)Y^j$, $j < r_h$, where r_h counts the rows with label h. The resulting square matrix is block diagonal, its hth block being the Vandermonde matrix of r_h distinct values among the c_k. Hence a $r_{\min}$-minor is a nonzero polynomial, of degree at most $r_{\min}(d-1)$ in the abscissae, and multiplying it by $\prod z_{k,s} \prod (z_{k,s} - z_{k',s'})$ gives the statement with nonzero distinct abscissae. □

D.2 Proof of Corollary 6.14

Proof. At the roots ζ_ν of h, the pair (ϕ, χ) has roots $\xi_\nu = \gamma(\zeta_\nu)$ and values $b(\zeta_\nu)$, with the profile of b. By Lemma 2.6 and Lemma 4.1, the degree in question is the least D such that $[\xi_\nu^i \phi(\xi_\nu)^j]_{i<D, j<\mu}$ has rank n, and Lemma 6.13 computes it. The map from the coefficients of γ to (ξ_ν) is an invertible linear substitution over the splitting field, so the minor stays nonzero with the same degree. The statement for ϕ^{-1} follows from Lemma 6.2. □

D.3 Proof of Lemma 6.15

Proof. The first assertion is the definition of a relation, and the next two are standard properties of Popov forms. The cited procedures and Lemmas 3.3, 3.4, and 4.5 use R only through these congruences and degree bounds, and none of them uses $\deg \det R = n$. For the last assertion, $\mathbb{K}[y]^m / \mathcal{M}_m$ is isomorphic to the span of $x^i a^j$, $i < m$, which splits into the fibers by Lemma 4.1. On a fiber of size ℓ, the vectors x^i, $i < m$, span a space of dimension $\min(\ell, m)$ by a Vandermonde argument. □

D.4 Proof of Lemma 6.16

Proof. Compute the auxiliary bases and data for b and b^{-1}, rejecting a basis of degree other than d before use (Lemma 6.2). Then compute the truncated powers of b^{-1} by Lemma 6.3 and run Steps 2–3 of [21, Algorithm 5.1], returning Fail if the candidate has degree larger than d. If the flag is Cert, return the candidate. Otherwise run the certificate of [21, Algorithm 5.2]. Let r be the first column of the candidate, read as a polynomial in x and y, and s the second column plus a random combination of the others. Check $r(x, b) \equiv s(x, b) \equiv 0 \bmod h$ by [22, Prop. 3.1] with the data of b. For $m \ge 3$, check $\gcd_x(r, s) = 1$ over $\mathbb{K}(y)$ and return the Sylvester matrix of r and s. For $m \le 2$, return the verified columns. Return Fail if any step fails.

On every execution, the returned columns are x-shifts of verified relations, so they lie in $\mathcal{M}_{m_R}^{(b,h)}$. The gcd test makes the Sylvester matrix nonsingular, of dimension $\deg_x r + \deg_x s \le 2(m-1)$ and degree at most d, whether or not the untested columns of the candidate are relations. The powers cost $\tilde{O}(M^{(\omega+3)/4})$ as in Lemma 6.4. The two verifications cost $\tilde{O}(m^\omega(d + m^2))$, and reconstruction and the resultant cost $\tilde{O}(m^\omega d + m^2 d)$.

If the hypotheses hold, [21, Props. 7.1 and 5.6] show that the candidate is a basis of $\mathcal{M}_m^{(b,h)}$ of degree at most d, also when $\nu_m < M$ (and $d \ge \lceil \nu_m / m \rceil$ since $\nu_m \le M$). Then the verifications succeed. The basis polynomials have no common factor of positive x-degree, since a $\mathbb{K}[y]$-combination of them, given by the adjugate, is a nonzero polynomial in y alone. Each of the at most $m-1$ irreducible factors of r of positive x-degree therefore divides s only on a proper affine subspace of the random coefficients, which is hit with probability at most $1/\operatorname{card}(S)$ [21, Prop. 5.8]. □

D.5 Proof of Proposition 6.17

Proof. (i) Small M are handled by Horner evaluation. Form $h = f(x+r_2)$, $u = (a+r_1)(x+r_2)$, $v = g(y-r_1)$, and return Fail if $h(0) = 0$ or $\gcd(u, h) \neq 1$. The profile of u modulo h is that of a modulo f. Apply Lemma 6.4 to (h, γ, u): the multiplier u is arbitrary there, and the output is a certified basis R_γ, the minimal polynomial χ of γ, of degree M, and ϕ with $\phi(\gamma) = u \bmod h$. Check $\chi(0) \neq 0$ and $\gcd(\phi, \chi) = 1$. Then apply Lemma 6.16 to (χ, ϕ), reduce ve_1 by the Popov form of the returned matrix and evaluate the remainder at ϕ modulo χ (Lemma 6.15), giving $z = v(\phi) \operatorname{rem} \chi$. Finally compute $z(\gamma) \operatorname{rem} h$ by Lemma 6.5 with the supplied R_γ and undo the shift. Every accepted output is correct, by the certificates and Lemma 6.15, and every step costs $\tilde{O}(M^{(\omega+3)/4})$.

For the probability, condition on the shifts, whose failure probability is at most $2M/\operatorname{card}(S)$, and let ζ_ν be the roots of h. Writing $\xi_\nu = \gamma(\zeta_\nu)$, we bound the bad event by a product of nonzero polynomials in the coefficients of γ. The factor ensuring $\xi_\nu \neq 0$ and pairwise distinctness has degree $M + M(M-1)/2$. The polynomial of [21, Prop. 7.6] ensures $\operatorname{rank} \mathrm{Hk}_{m,d}^{(\gamma,h)} = M$ and has degree at most $2M^2/m$, and that of [21, Prop. 7.8] ensures $\operatorname{rank} \mathrm{Hk}_{m,d}^{(\phi,\chi)} = \nu_m$ and has degree at most $2M^2$. It is nonzero for every fixed tuple of distinct values. The minor of Corollary 6.14 for (ϕ, χ) has degree at most $M(d-1)$, using $L_{f,a} \leq d$. The minor for (γ, h) uses the columns $i + jd < M$ at $\xi_\nu = \zeta_\nu^d$ and has degree at most $M(m-1)$. On the complement of their zero sets all tests of Lemmas 6.4, 6.5, and 6.16 succeed, except for the Sylvester test, whose conditional failure probability is at most $(m-1)/\operatorname{card}(S)$. The total is at most $(\frac{13}{2}M^2 + \frac{7}{2}M - 1)/\operatorname{card}(S) \leq 8M^2/\operatorname{card}(S)$, as in Lemma 6.6.

(ii) Handle $M < 16$ directly. Otherwise perform the translations and unit tests of (i). Run [21, Algorithm 6.1] on (h, γ, u, m, d), with its certification tests, obtaining (R_γ, χ, ϕ) or Fail, then check $\chi(0) \neq 0$, $\gcd(\phi, \chi) = 1$ and $\deg R_\gamma \leq d$. In place of the Cert-only candidate basis for (χ, ϕ), run [21, Algorithm 5.2] on (χ, ϕ, m, d) with $m-2$ fresh random coefficients, convert its output to Popov form R_ϕ, and return Fail if $\deg R_\phi > d$. Its dimension m_R satisfies $1 \leq m_R \leq 2m$ and $\mathcal{L}(R_\phi) \subseteq \mathcal{M}_{m_R}^{(\phi,\chi)}$. The degree test is needed because Algorithm 5.2 guarantees only degree $2d$ on an arbitrary execution. For each of (ϕ, χ) and (γ, h) compute a minimal basis of $\mathcal{N}_m$ and reject unless its degree is d, before computing and using its A_j data.

Then use the online steps of Algorithm 4.1 with these certified matrices and data. The full Algorithm 4.1 requires a generating argument and rejects NoCert, so we do not call it here. Each reduction uses the actual dimension m_R, block length d and $J = \lceil M/d \rceil \leq m$. In the bivariate evaluation, coefficient arrays are padded to x-width $W = 2m$ (no relation matrix is padded or truncated), so that in Lemma 4.5 the A_j are split into $\lceil d/W \rceil$ slices of width W and $d' = W + d$, and the identities are unchanged. Since m_R may exceed m, all m_R components of the remainder must be kept. The congruences follow from Lemma 6.15.

Both routines of [21] cost $\tilde{O}(m^\omega d + (m + M/d)d^{\omega_2/2}) \subseteq \tilde{O}(M^{(\omega+2)/3})$ since $\omega_2 \leq \omega + 1$, and Popov conversion, degree tests, auxiliary data, division precomputation and remainder tables fit the same bound. Since $W = \Theta(m)$ and $d = \Theta(m^2)$, the two bivariate products cost

$$\tilde{O}(Wm\,\mathsf{MM}(k, m, \lceil d/W \rceil) + (W+d)\,\mathsf{MM}(k, 2m, \lceil M/(W+d) \rceil)) \subseteq \tilde{O}(d\,\mathsf{MM}(m, m, k)),$$

and square blocking gives the bound. The good event is that of (i) at these dimensions: [21, Prop. 7.8] for $\operatorname{rank} \mathrm{Hk}_{m,d}^{(\phi,\chi)} = \nu_m$ and Corollary 6.14 for the auxiliary degree of (ϕ, χ), together with the polynomials of Proposition 4.2 for (γ, h). On this event the output of Algorithm 5.2 has degree at most d, so the degree test adds no bad event. The Sylvester test fails with conditional probability at most $(m-1)/\operatorname{card}(S)$, giving total failure probability at most $\min\{1, 8M^2/\operatorname{card}(S)\}$.

(iii) The preprocessing is that of (i) at the dimension m: Lemma 6.4 for (h, γ, u), the auxiliary bases and data for ϕ, ϕ^{-1} and γ, γ^{-1}, and Lemma 6.16 for (χ, ϕ). Their proofs use the parameters only as listed in the proof of Lemma 6.8, so that they hold at this dimension

with cost $\tilde{O}(m^\omega d)$. Then compute the Popov form P of the returned matrix, of dimension $m_R \le 2m$, its division data with $\delta = 2d$, the remainder table P_j, $j < J$, the same data for R_γ with $j < \lceil M/d \rceil$, and the powers $\phi^l \operatorname{rem} \chi$ and $\gamma^l \operatorname{rem} h$ for $l < 3m$. By Lemma 3.3, splitting the J columns into blocks of m, the tables cost $\tilde{O}(d(m^\omega + \mathsf{MM}(m_R, m_R, J))) = \tilde{O}(m^\omega d + dJm^{\omega-1})$, and the powers cost $\tilde{O}(mM)$.

The online phase reduces the translated inputs by Lemma 3.4, at cost $\tilde{O}(d(\mathsf{MM}(m_R, J, k) + \mathsf{MM}(m_R, m_R, k)))$, which is $\tilde{O}(d\lceil J/m \rceil \mathsf{MM}(m, m, k))$ after padding m_R to $2m$, and evaluates the remainders at ϕ modulo χ, then at γ modulo h, by Evaluate3. The remainders have all m_R components, so that coefficient arrays are padded to x-width $W = 2m$: then $s_{i,c_2,c_1} \in \mathbb{K}[x, y_0]_{<(W,m)}$, $S_{i,c_2} \in \mathbb{K}[x, y_0]_{<(d+W,2m)}$, $V_i \in \mathbb{K}[x, y_0]_{<(2d+W,3m)}$, and the slices of the powers have degree less than $2d + W$. The identities of Lemma 6.9 are unchanged by Lemma 6.15, and its costs change by constant factors only. Reading the inputs costs $O(kJd)$.

The good event is that of (i) at the dimension m. Its polynomials have degrees at most $2M$ for the shifts, $M + M(M-1)/2$, $2M^2/m$, $2M^2$, $M(d-1)$ and $M(m-1)$, and the Sylvester test adds $(m-1)/\operatorname{card}(S)$. Since $2M^2/m$, $M(d-1)$ and $M(m-1)$ are at most M^2, the total is at most $(\frac{11}{2}M^2 + \frac{5}{2}M + m)/\operatorname{card}(S) \le 8M^2/\operatorname{card}(S)$. On this event all tests succeed, as in (i). □

D.6 Proof of Lemma 6.20

Proof. The construction uses the local blocking width $m = \lceil M^{1/3} \rceil$, and the application uses the global width $\bar{m} = \lceil n^{1/3} \rceil$ (as in Section 4). For $M < 16$ use Horner's rule. Otherwise run the certified preprocessing of Proposition 6.17(ii), with all its tests, and make one change: for the certified matrix R_ϕ, of dimension $m_R \le 2m$ and degree at most d, build the remainder table $P_j = (y^{jd} e_1) \operatorname{Rem} R_\phi$ for $0 \le j < J = \lceil n/d \rceil$ instead of $j < \lceil M/d \rceil$, and the table for R_γ is unchanged. The entries have degree less than d and their doubling products less than $2d$, so Lemma 3.3 applies with the same division excess, at cost $\tilde{O}(d(m_R^\omega + \mathsf{MM}(m_R, m_R, J))) \subseteq \tilde{O}(dm^\omega + nm^{\omega-1}) \subseteq \tilde{O}(n^{(\omega+2)/3})$, by splitting the J columns into blocks of m. Reduce the translated outer polynomials $g_i(y - r_1)$, of degree less than n, by Lemma 3.4 with block length d, and the remainders have m_R components of degree less than d, independently of n, so the two evaluations of Proposition 6.17(ii) keep their cost $\tilde{O}(d\,\mathsf{MM}(m, m, k))$. Then the reduction costs $\tilde{O}(d(\mathsf{MM}(m_R, J, k) + \mathsf{MM}(m_R, m_R, k)))$. Since $m_R \le 2\bar{m}$, padding and splitting the inner dimension into blocks of $\bar{m}$ give $\mathsf{MM}(m_R, J, k) = O(\lceil J/\bar{m} \rceil \mathsf{MM}(\bar{m}, \bar{m}, k))$, and $d\lceil J/\bar{m} \rceil \le d + (n+d)/\bar{m} \le 3\bar{m}^2$ because $d \le m^2 \le \bar{m}^2$ and $n \le \bar{m}^3$. Hence the online phase costs $\tilde{O}(\bar{m}^2 \mathsf{MM}(\bar{m}, \bar{m}, k))$, which also absorbs the $\tilde{O}(kn)$ input and output work. The larger table is deterministic and adds no bad event, and once the preprocessing has succeeded, the online phase is correct for every outer polynomial of degree less than n. □

D.7 Proof of Theorem 6.21

Proof. The algorithm is Algorithm 6.3. Let $\tau = 3/4$ for the single composition and $\tau = 2/3$ for the batch, and put $E = 1 + (1-\tau)(\omega - 1)$ and $B = 1 + (1-\tau)(\omega - 2)$, thus $(E, B) = (\frac{\omega+3}{4}, \frac{\omega+2}{4})$ or $(\frac{\omega+2}{3}, \frac{\omega+1}{3})$.

Decomposition. Starting with the current modulus $h = f$, of degree N, proceed as follows. If $N < 16$, compute an annihilator of $a \operatorname{rem} h$ of degree at most N by linear algebra on $1, a, \ldots, a^N \operatorname{rem} h$, record h with it, and stop. Otherwise, let

$$H_N = \left\lfloor \tfrac{1}{4} \lfloor N/2 \rfloor^\tau \right\rfloor \ge 1, \qquad t_N = \left\lfloor \frac{N}{H_N + 1} \right\rfloor + 1,$$

so that $\lfloor N/t_N \rfloor \le H_N$ and $t_N = \Theta(N^{1-\tau})$, and apply Lemma 6.18 to $(h, a \operatorname{rem} h, t_N)$. Record F with P if $F \ne 1$. Let $M = \deg h'$. Stop if $M = 0$, or if $M \ge N/2$, in which case call h' the terminal factor. Otherwise continue with $h = h'$. The degrees decrease geometrically, so there

are $O(\log n)$ steps, and the separators cost $\sum \tilde{O}(Nt_N^{\omega-1}) = \tilde{O}(n^E)$. The recorded factors and the terminal factor are pairwise coprime with product f, and the terminal factor h_*, of degree M, has $L_{h_*,a} \le \lfloor N/t_N \rfloor \le H_N \le \frac{1}{4}M^\tau$.

Few-value pieces. On a recorded factor of degree $b \le N$ with annihilator of degree $s = O(N^{1-\tau})$ (or $O(1)$ when $N < 16$), Lemma 6.19 costs $\tilde{O}(kn + bs + \mathsf{MM}(b,s,k))$. For the single composition ($k = 1$) this is $\tilde{O}(n + N^{5/4})$. For the batch, pad s to $s' = \lceil CN^{1/3} \rceil \ge s$ for a fixed constant C and split the b rows into blocks of s': $\mathsf{MM}(b,s,k) = O(\lceil b/s' \rceil \mathsf{MM}(s',s',k)) = O(N^{2/3}\mathsf{MM}(\bar{m},\bar{m},k))$ with $\bar{m} = \lceil n^{1/3} \rceil$. Summed over the geometric sequence of degrees, the pieces cost $\tilde{O}(n^E)$, respectively $\tilde{O}(kn + n^{4/3} + n^{2/3}\mathsf{MM}(\bar{m},\bar{m},k))$.

Terminal factor. If $M = 1$, evaluate directly. For the single composition, $m = \lceil (4M)^{1/4} \rceil < 3M^{1/4}$ gives $d = \lceil M/m \rceil > M^{3/4}/3 \ge L_{h_*,a}$, so Proposition 6.17(i) applies to $(h_*, a \operatorname{rem} h_*)$. The outer polynomial has degree less than n, possibly larger than M. Write $g = \sum_{j<\lceil n/M \rceil} y^{jM} g_j$ with $\deg g_j < M$ and compose each block. Horner recombination in $a^M \operatorname{rem} h_*$ costs $\tilde{O}(n)$. The compositions cost $\lceil n/M \rceil \tilde{O}(M^E) \subseteq \tilde{O}(n^E)$ because $E > 1$. For the batch, $m = \lceil M^{1/3} \rceil \le 2M^{1/3}$ gives $d \ge M^{2/3}/2 \ge L_{h_*,a}$, and Lemma 6.20 applied to the original outer polynomials gives $\tilde{O}(n^{(\omega+2)/3} + \bar{m}^2\mathsf{MM}(\bar{m},\bar{m},k))$.

Recombination. The residues modulo the pairwise coprime factors are combined by the Chinese remainder theorem with a subproduct tree, in $\tilde{O}(kn)$ operations. Every step except the calls to Proposition 6.17 and Lemma 6.20 is deterministic and exact, and those calls are repeated until they succeed, each with expected $O(1)$ attempts. By linearity of expectation the expected total cost is $\tilde{O}(n^E)$ for one outer polynomial, and $\tilde{O}(n^{(\omega+2)/3} + n^{2/3}\mathsf{MM}(\bar{m},\bar{m},k))$ for k, since $n^{2/3}\mathsf{MM}(\bar{m},\bar{m},k) \ge kn$ and $n^{4/3} \le n^{(\omega+2)/3}$. □

D.8 Proof of Theorem 6.23

Proof. Put $\varepsilon = (\omega-1)/4$ and $T = n^{1+\varepsilon}k^\varepsilon = n^{(\omega+3)/4}k^{(\omega-1)/4}$. If $k > n^{1/3}/8$, then Theorem 6.21 gives $\tilde{O}(n^{(\omega+2)/3})$ with square blocks, and $n^{(\omega+2)/3} = n^{1+\varepsilon}(n^{1/3})^\varepsilon = O(T)$. Assume $k \le n^{1/3}/8$. We use two inequalities, valid for every $N \le n$: $N^{1+\varepsilon}k^\varepsilon \le T$, and

$$Nk^{\omega-1} = Nk^\varepsilon k^{3\varepsilon} \le nk^\varepsilon n^\varepsilon = T,$$

since $k^3 \le n$. Call a local degree $N \ge 16$ *small* if $k \le N^{1/3}/8$, and *large* otherwise, so that $N < 512k^3$ in the large case. The algorithm is Algorithm 6.3 with the following changes.

Decomposition. At a small degree N, put $H_N = \lfloor \frac{1}{4}\lfloor N/2 \rfloor^{3/4}k^{-1/4} \rfloor$ and $t_N = \lfloor N/(H_N+1) \rfloor + 1$. Since $4k^{1/4} \le 4 \cdot 8^{-1/4}N^{1/12} \le \lfloor N/2 \rfloor^{3/4}$ for $N \ge 16$, we have $H_N \ge 1$, and $t_N = \Theta((Nk)^{1/4})$, so that the separator of Lemma 6.18 costs $\tilde{O}(Nt_N^{\omega-1}) = \tilde{O}(N^{1+\varepsilon}k^\varepsilon)$. At a large degree, use the parameters of the batch case of Theorem 6.21, $t_N = \Theta(N^{1/3})$, at cost $\tilde{O}(N^{(\omega+2)/3})$, which is $\tilde{O}(Nk^{\omega-1})$ since $N^{1/3} < 8k$. The degrees decrease geometrically, so the decomposition costs $\tilde{O}(T)$.

Few-value pieces. A recorded factor of degree $b \le N$ has an annihilator of degree $s < t_N$, and Lemma 6.19 costs $\tilde{O}(kn + bs + \mathsf{MM}(b,s,k))$. At a small degree, pad s to $s' = \lceil C(Nk)^{1/4} \rceil$ for a fixed constant C; since $k^3 \le N$, also $k \le s'$, and $\mathsf{MM}(b,s',k) = O(\lceil b/s' \rceil s'^\omega) = O(N^{1+\varepsilon}k^\varepsilon)$. At a large degree, pad s to $s' = \lceil CN^{1/3} \rceil$; then $\mathsf{MM}(b,s',k) = O(\lceil b/s' \rceil \lceil k/s' \rceil s'^\omega) = O(Nk\, s'^{\omega-2}) = O(Nk^{\omega-1})$. Summed over the $O(\log n)$ pieces, this is $\tilde{O}(T)$, including the term kn.

Terminal factor. Let h_*, of degree $M \ge N/2$, be the terminal factor obtained at the degree N; we may assume $M \ge 16$, since otherwise Horner's rule costs $O(kn)$. If N is small, put $m = \lceil (4Mk)^{1/4} \rceil$ and $d = \lceil M/m \rceil$. Since $k \le N^{1/3}/8 \le M^{1/3}/4$, we have $(4M)^{1/4} \le m \le M^{1/3} + 1$ and $k \le m$, and $m \le 2(4Mk)^{1/4}$ gives

$$d \ge \frac{M}{m} \ge \frac{M^{3/4}k^{-1/4}}{2\sqrt{2}} \ge \frac{1}{4}\Big\lfloor \frac{N}{2} \Big\rfloor^{3/4} k^{-1/4} \ge H_N \ge L_{h_*,a}.$$

Apply Proposition 6.17(iii) to $(h_*, a \operatorname{rem} h_*)$ with $J = \lceil n/d \rceil$, to the original outer polynomials. Its terms are bounded as follows: $m^\omega d = O(Mm^{\omega-1})$ and $dJm^{\omega-1} \le (n+d)m^{\omega-1}$, both

$O(nm^{\omega-1}) = O(T)$; $d\lceil J/m\rceil \mathsf{MM}(m,m,k) \le (n/m+d)m^\omega = O(T)$ since $k \le m$; $k\nu = O(m)$ as in Theorem 6.10, so that $m^2\mathsf{MM}(k\nu, m, \lceil d/m\rceil) = O(Mm^{\omega-1})$ and $k\nu M = O(mM)$; and $kJd = O(kn)$. If N is large, then $L_{h_*,a} \le \frac{1}{4}M^{2/3}$ as in the proof of Theorem 6.21, and we apply the algorithm of Lemma 6.20. By its proof, with $m = \lceil M^{1/3}\rceil$, $d = \lceil M/m\rceil$ and $J = \lceil n/d\rceil$, the preprocessing costs $\tilde{O}(M^{(\omega+2)/3} + nm^{\omega-1})$ and the online phase $\tilde{O}(d\lceil J/m\rceil \mathsf{MM}(m,m,k))$. Since $m = O(k)$, this is $\tilde{O}(nk^{\omega-1} + (n/m+d)\lceil k/m\rceil m^\omega) = \tilde{O}(nk^{\omega-1}) = \tilde{O}(T)$.

Recombination. The Chinese remaindering costs $\tilde{O}(kn)$. As in Theorem 6.21, only the calls to Proposition 6.17 and Lemma 6.20 can fail, each with expected $O(1)$ attempts, and the expected total cost is $\tilde{O}(T)$.

The curve Λ_4. Let $k = n^t$, and let $s \in [1/5, 1/3]$ attain the minimum defining $\Lambda_4(t)$. For $u \ge 0$ let $F(u)$ be the maximum of the five terms in the definition of Λ_4 at this s and with t replaced by u, so that $\Lambda_4(t) = F(t)$. Each term is nondecreasing in u and increases by at most δ when u increases by δ, since $\omega(a,b,c+\delta) \le \omega(a,b,c) + \delta$ and $\omega(a+\delta,b,c) \le \omega(a,b,c) + \delta$ by splitting a dimension; moreover $F(u) \ge 1 + u$. Hence, for $0 < \mu \le 1$ and $u \ge t$ with $\mu u \le t + 1 - \mu$,

$$\mu F(u) \le \mu(F(t) + u - t) \le \mu F(t) + (1-\mu)(1+t) \le F(t).$$

Whenever the selected width satisfies $m < (4M)^{1/4}$, we use the extension of Proposition 6.17(iii) to $(4M)^{1/5} \le m$ obtained from Theorem 6.12: the truncated powers of b^{-1} in the certificate of Lemma 6.16 are computed by Lemma 6.11, and its two candidate relations are verified by the raw four-level evaluation of Section 6.3. Its preprocessing costs $C_4(M,m)$, and its total batch cost and failure bound are those used below.

The algorithm is Algorithm 6.3 with the following changes. At the current degree $N \ge 16$, the separator uses $H_N = \lfloor \frac{1}{4}\lfloor N/2\rfloor^{1-s}\rfloor \ge 1$ and $t_N = \lfloor N/(H_N+1)\rfloor + 1 = \Theta(N^s)$, at cost $\tilde{O}(Nt_N^{\omega-1})$, whose exponent relative to N is the first term of F. A few-value piece of degree $b \le N$ costs $\tilde{O}(kn + bt_N + (b/t_N+1)\mathsf{MM}(t_N,t_N,k))$ after reducing the outer polynomials modulo its annihilator, and the last term has the exponent of the second term of F at $u = \log k/\log N$. On the terminal factor h_*, of degree $M \ge N/2$, we cut each g_i into $\lceil n/M\rceil$ blocks of degree less than M; the $k' = k\lceil n/M\rceil$ blocks are composed together, and the results are recombined by Horner's rule in $a^M \operatorname{rem} h_*$, in $\tilde{O}(kn)$ operations. If $M < 64$, the blocks are composed by Horner's rule. Otherwise we apply Proposition 6.17(iii) to the k' blocks with $J = \lceil M/d\rceil \le m$ and $m = \max\{\lceil M^s\rceil, \lceil (4M)^{1/5}\rceil\}$, which lies in $[(4M)^{1/5}, M^{1/3}+1]$ since $(4M)^{1/5} \le M^{1/3}$ for $M \ge 64$. Since $m \le 2\sqrt{2}\,M^s$ and $M \ge \lfloor N/2\rfloor$, we have $d \ge M/m \ge \frac{1}{4}\lfloor N/2\rfloor^{1-s} \ge H_N \ge L_{h_*,a}$. The terms of Proposition 6.17(iii) are then those of Theorem 6.10 at degree M, dimension $\Theta(M^s)$ and batch size k'.

For the summation, express all dimensions as powers of n. If a piece has degree $N = n^\mu$, the exponent triples of its matrix products are $(\mu s, \mu s, \mu u)$, $(\mu u + \mu(1-3s), \mu s, \mu(1-2s))$ and their permutations, together with those of the preprocessing, where $\mu u = t$ for the separator and the few-value pieces and $\mu u \le t+1-\mu+o(1)$ for the terminal blocks. All of them lie in the fixed cube $[0, t+2]^3$. The function ω is 1-Lipschitz in each coordinate, so padding each dimension to the next point of a finite grid in this cube gives the bounds $n^{\omega(\cdot)+\varepsilon}$ with a constant independent of the piece. By homogeneity, $\omega(\mu a, \mu b, \mu c) = \mu\,\omega(a,b,c)$, so that the local work of every piece is at most $n^{\mu F(u)+\varepsilon} \le n^{F(t)+\varepsilon}$, by the inequality above. Pieces of bounded degree cost $O(kn)$. There are $O(\log n)$ pieces, and the reductions modulo the annihilators, the cutting and recombination of the blocks and the Chinese remaindering cost $\tilde{O}(kn) = n^{1+t+o(1)} \le n^{F(t)+o(1)}$. Only the certified preprocessing can fail, and it is repeated with expected $O(1)$ attempts. Letting $\varepsilon \to 0$, the expected total cost is $n^{\Lambda_4(t)+o(1)}$. □

E Further results

E.1 Transposition: batched power projection

Throughout this subsection, $f \in \mathbb{K}[x]$ is monic and separable of degree n, $a \in \mathbb{K}[x]_{<n}$ and $\mathcal{A} = \mathbb{K}[x]/\langle f\rangle$. Let $T_N \in \mathbb{K}^{n\times N}$ be the matrix whose i-th column is the coefficient vector of $a^i \operatorname{rem} f$, $0 \le i < N$. Then composition is $g \mapsto T_N \vec{g}$ and power projection is $\vec{\ell} \mapsto T_N^{\mathsf{T}}\vec{\ell}$, where $\vec{\ell}$ is the vector $(\ell(x^i))_{i<n}$ of a linear form ℓ on $\mathcal{A}$, and the batched versions are $I_k \otimes T_N$ and $I_k \otimes T_N^{\mathsf{T}}$.

Lemma E.1 (linearity of the online phase)**.** *Fix $(f, a, \mathbf{r})$ and suppose that Lemma 3.2 did not return* Fail*. Then the part of Algorithm 3.2 after its first line is a linear straight-line program in the coefficients of $(g_1, \dots, g_k)$: it uses only additions, subtractions and multiplications by elements computed in Lemma 3.2, and no zero test or division whose operand depends on the g_i. The same holds for Algorithm 4.1 after its precomputation.*

Proof. The shift $g \mapsto g(y - r_1)$ and the final shift $b \mapsto b(x - r_2)$ are fixed linear maps computed by $\tilde{O}(n)$ linear programs (divide-and-conquer polynomial translation, whose products have one fixed operand). The remaining steps are linear by Lemmas 3.4, 3.5, and 4.5. By contrast, [21, Algorithm 4.1, Step 2] computes a kernel basis of a matrix containing the vector $g\,e_1$, which is not a linear function of g; this is why we use the remainder table. □

Theorem E.2 (batched power projection)**.** *Let $N = O(n)$ and let $\ell_1, \dots, \ell_k$ be linear forms on $\mathcal{A}$, given by their vectors $\vec{\ell}_j$. With the parameters of Proposition B.3, the values $\ell_j(a^i)$, $1 \le j \le k$, $0 \le i < N$, can be computed by a Las Vegas algorithm with the failure event of Theorem 3.6 and cost $\tilde{O}(n^{(\omega+2)/3}k^{(\omega-1)/3} + n^{\omega/2}k)$; with the parameters optimized as in Theorem 3.6(iii), the cost is $n^{\Lambda_1(t)+o(1)}$ for $k = n^t$. If a generates $\mathcal{A}$, the same holds with the cost and failure probability of Theorem 4.6.*

Proof. Run the precomputation of Lemma 3.2 (resp. of Algorithm 4.1), which does not involve the input. If it fails, return Fail. Otherwise, by Lemma E.1, the online phase is a linear straight-line program Γ that computes $I_k \otimes T_N$. For $N > n$ we use Lemmas 3.3 and 3.4 with $J = \lceil N/d\rceil = O(m)$ blocks, which changes the costs by constant factors only. The correctness of Γ on every input is the correctness statement of Theorem 3.6 (resp. Theorem 4.6), which holds for all g once the precomputation has succeeded. By the transposition principle [7, Thm. 13.20], which bounds the linear complexity of the transpose by that of the matrix plus the difference between the numbers of rows and columns (with a correction for zero rows and columns), $I_k \otimes T_N^{\mathsf{T}}$ has a linear straight-line program of length $|\Gamma| + O(k(n+N))$; reversing the computation graph of Γ constructs it in time $O(|\Gamma|)$ [5], and applying it to $(\vec{\ell}_1, \dots, \vec{\ell}_k)$ gives the projections. □

Corollary E.3 (batched inverse composition)**.** *Let $a \in \mathbb{K}[x]_{<n}$ generate $\mathcal{A}$, with minimal polynomial μ_a of degree n, and let $b_1, \dots, b_k \in \mathcal{A}$. The unique $h_1, \dots, h_k \in \mathbb{K}[y]_{<n}$ with $h_j(a) \equiv b_j \bmod f$, together with μ_a, can be computed by a Las Vegas algorithm using the cost of Theorem E.2 for $k+1$ forms and $N = 2n$, plus $\tilde{O}(kn)$, with failure probability at most $(6n^2+n)/\operatorname{card}(S)$ when f is separable and the one-matrix version of Theorem E.2 is used, and $(8n^2+n)/\operatorname{card}(S)$ with the generator version.*

Proof. Choose a random linear form ℓ (coordinates of $\vec{\ell}$ in S). For $b \in \mathcal{A}$ put $N_b(Z) = \ell\big(b\,\frac{\mu_a(Z)-\mu_a(a)}{Z-a}\big)$, applying ℓ coefficientwise in Z. This polynomial has degree less than n and is determined by μ_a and the values $\ell(ba^i)$, $i < n$. If $b = h(a)$, then, since $\mu_a(a) = 0$,

$$h(a)\frac{\mu_a(Z)-\mu_a(a)}{Z-a} - h(Z)\frac{\mu_a(Z)-\mu_a(a)}{Z-a} = -\frac{h(Z)-h(a)}{Z-a}\,\mu_a(Z),$$

and applying ℓ gives

$$N_{h(a)} \equiv h\, N_1 \pmod{\mu_a}, \tag{E.1}$$

the identity behind Shoup's inverse composition [25, §3]. The algorithm computes the vectors of the forms $\ell_j = \ell(b_j\,\cdot)$ by transposed multiplications by b_j modulo f, at cost $\tilde{O}(n)$ each [25, 5], and applies Theorem E.2 with the $k+1$ forms $\ell, \ell_1, \dots, \ell_k$ and $N = 2n$ to obtain $\ell(a^i)$ for $i < 2n$ and $\ell(b_j a^i)$ for $i < n$. It then computes the minimal polynomial μ of the sequence $(\ell(a^i))_{i<2n}$ by a fast Berlekamp–Massey algorithm, in $\tilde{O}(n)$ operations, and returns Fail if $\deg\mu < n$; since μ divides μ_a, the equality $\deg\mu = n$ certifies $\mu = \mu_a$. Finally, it computes N_1, its inverse modulo μ_a, and $h_j = N_{b_j} N_1^{-1} \operatorname{rem} \mu_a$. Correctness follows from Eq. (E.1), since the generating series of $(\ell(a^i))_i$ is N_1/μ_a, so that $\gcd(N_1, \mu_a) = 1$ when $\deg\mu = n$.

The test $\deg\mu = n$ fails if and only if $\operatorname{Res}(\mu_a, N_1) = 0$. The coefficients of N_1 are linear forms in the coordinates of $\vec{\ell}$, so $\operatorname{Res}(\mu_a, N_1)$ is a homogeneous polynomial of degree at most n in these coordinates. To see that it is nonzero, let ℓ_0 map a^{n-1} to 1 and a^i to 0 for $i < n-1$; this form exists because $(a^i)_{i<n}$ is a basis of $\mathcal{A}$. Writing $\mu_a = \sum_{i\le n} \mu_i Z^i$ with $\mu_n = 1$, we obtain $\frac{\mu_a(Z)-\mu_a(a)}{Z-a} = \sum_{j<n} Z^j \sum_{i>j} \mu_i a^{i-1-j}$, whose only term in a^{n-1} has $i = n$, $j = 0$. Hence $N_1 = 1$ for $\ell = \ell_0$. By the Schwartz–Zippel lemma, the test fails with probability at most $n/\operatorname{card}(S)$, and adding the failure probability of Theorem 3.6 gives the claim. □

E.2 Normal bases

Let $f \in \mathbb{F}_q[x]$ be irreducible of degree n, $\mathcal{A} = \mathbb{F}_{q^n}$ and Q the matrix of σ in the basis $(x^i)_{i<n}$. Kaltofen and Shoup reduce the selection of a normal element and the conversions between normal and polynomial coordinates to *automorphism projection*, the computation of $(u(\sigma^i(\theta)))_{0\le i<K}$, $K = O(n)$, for a linear form u and $\theta \in \mathcal{A}$, and to its transpose [16, §§3–4, Thm. 6]. Set $t = \lceil n^\beta \rceil$ and $m_A = \lceil K/t \rceil$. Their Algorithm AP is again a baby-step/giant-step method. It computes the baby steps $\sigma^i(\theta)$, $i < t$, by repeated q-th powering, the element $x^{q^t} \operatorname{rem} f$ by $O(\log t)$ compositions, and the giant steps $\vec{u}^{\mathsf{T}} Q^{tj}$, $j < m_A$, by the doubling of [16, Lemma 8]; finally, it multiplies the $m_A \times n$ matrix of giant steps by the $n \times t$ matrix of baby steps. In the doubling, once the rows with $j < k'$ are known, one applies $Q^{tk'}$ to all of them; this is a batched power projection with the single argument $x^{q^{tk'}} \operatorname{rem} f$ and k' forms, followed by one composition giving $x^{q^{2tk'}} \operatorname{rem} f$. As in Section 5.6, we compute the baby steps by batched compositions instead of powering. With $h_s = x^{q^s} \operatorname{rem} f$ and $\theta_i = \sigma^i(\theta)$, we have $\sigma^s(u) = u(h_s) \operatorname{rem} f$ for every $u \in \mathcal{A}$, since the coefficients of u lie in $\mathbb{F}_q$; hence one batch with the argument h_s maps $h_1, \dots, h_s$ and $\theta_0, \dots, \theta_{s-1}$ to $h_{s+1}, \dots, h_{2s}$ and $\theta_s, \dots, \theta_{2s-1}$.

Theorem E.4 (automorphism projection and normal bases)**.** *With the generator version of Theorem E.2 in the doubling and the baby steps computed by batched compositions, automorphism projection, its transpose, the selection of a normal element and the conversions of [16, Thm. 6] are computed by Las Vegas algorithms using, for every $\beta \in [0,1]$, an expected*

$$n^{\max\{\Lambda_2(1-\beta),\ \Lambda_2(\beta),\ \omega(1-\beta,1,\beta),\ \kappa\}+o(1)} + n^{1+o(1)} \log q$$

operations in $\mathbb{F}_q$, where Λ_2 is the batch curve Eq. (4.2). Taking $\beta = \frac{1}{2}$ gives

$$n^{E_{\mathrm{AP}}+o(1)} + n^{1+o(1)} \log q, \qquad E_{\mathrm{AP}} = \max\Big\{\frac{\omega_2}{2},\ \frac{2}{3} + \frac{\omega+\omega_2}{6},\ \frac{\omega+2}{3},\ \kappa\Big\},$$

and $E_{\mathrm{AP}} = \omega_2/2 < 1.6251$ with the bounds of Section 2.1, for both data sets; if $\omega = 2$ the cost is $n^{3/2+o(1)} + n^{1+o(1)} \log q$. With the batch curve Λ_1 of Theorem 3.6, which needs no genericity argument, and the baby steps computed by powering, the same method gives $O(n^{1.6518} + n^{1.5635} \log q)$.

Proof. The element $h_1 = x^q \operatorname{rem} f$ costs $n^{1+o(1)} \log q$ by repeated squaring. The doubling described above has $\lceil \log_2 t \rceil$ levels; each is a batch of at most $2t$ compositions with an argument

h_s, which generates the field $\mathcal{A}$, so that the baby steps θ_i, $i < t$, and $h_t = x^{q^t} \operatorname{rem} f$ cost $n^{\Lambda_2(\beta)+o(1)}$ by Theorem 4.6, over the extension of Lemma 5.1 if q is small. The doubling has $\lceil \log_2 m_A \rceil$ levels, and the level with k' forms applies $I_{k'} \otimes T^{\mathsf{T}}$, where T is the composition matrix of the argument $x^{q^{tk'}} \operatorname{rem} f$. Since this argument is a Frobenius image of x, it generates the field $\mathcal{A}$ (Lemma 5.2), so that the generator version of Theorem E.2 applies, over the extension of Lemma 5.1 if q is small. Each level therefore costs $n^{\Lambda_2(\log_n k')+o(1)} + n^{\kappa+o(1)}$, and by monotonicity the doubling costs $n^{\max(\Lambda_2(1-\beta),\kappa)+o(1)}$; it is correct because composition with $x^{q^{ts}} \operatorname{rem} f$ has matrix Q^{ts}, whose transpose maps $\vec{u}^{\mathsf{T}} Q^{tj}$ to $\vec{u}^{\mathsf{T}} Q^{t(j+s)}$ [16, Lemma 8]. The final product costs $\mathsf{MM}(n^{1-\beta}, n, n^{\beta}) = n^{\omega(1-\beta,1,\beta)+o(1)}$. Automorphism evaluation is the transpose of automorphism projection [16, §3]: the $O(\log n)$ precomputations depend only on f and q, and once they succeed the algorithm is linear in $\vec{u}$ (Lemma E.1), so that transposition applies as in the proof of Theorem E.2; the baby steps depend only on f, q and θ. The three normal-basis problems reduce to these two operations, the minimal polynomials of the resulting scalar sequences and a Hankel system, with an expected $O(\max(\log_q n, 1))$ trials [16, Thms. 4–6]; with fast algorithms for linear recurrences and Hankel systems, each trial costs $\tilde{O}(n)$ besides the projections.

At $\beta = \frac{1}{2}$, homogeneity and symmetry give $\omega(\frac{1}{2}, 1, \frac{1}{2}) = \frac{1}{2}\omega(1, 2, 1) = \frac{\omega_2}{2}$, and $\Lambda_2(\frac{1}{2}) = \max\left(\frac{\omega+2}{3}, \frac{2}{3} + \frac{1}{3}\omega(1, 1, \frac{3}{2})\right) \le \max\left(\frac{\omega+2}{3}, \frac{2}{3} + \frac{\omega+\omega_2}{6}\right)$ by Eq. (2.1), which gives E_{AP}. The choice $\beta = \frac{1}{2}$ minimizes $\omega(1-\beta, 1, \beta)$, which is convex in β and invariant under $\beta \mapsto 1-\beta$, and also $\max\{\Lambda_2(\beta), \Lambda_2(1-\beta)\}$, since Λ_2 is nondecreasing; hence no other β lowers the exponent of this algorithm. If $\omega = 2$, then $\omega_2 = 3$ and the maximum is $3/2$, attained by the first two terms, while the last two equal $4/3$. The values for Λ_1 are given by the program of Appendix A. □

Remark E.5 (comparison and prior work). With one transposed composition of [21] at each level of the doubling instead of a batch, the same program gives 1.7189, and batched Brent–Kung with the same rectangular bounds gives 1.7813; the bound in [16, Thm. 6] is 1.815, using [10], with the additional term $n^{1+\beta+o(1)} \log q$ of the baby steps computed by powering. Huang and Pan improve the rectangular products in the distinct-degree and black-box Berlekamp analyses of [16] [14, Thms. 10.6, 10.7]. Kaltofen and Shoup omit the final product from their Eq. (8), using a rectangular argument valid for $\beta \ge 2/3$; we retain it because our optimum has $\beta = \frac{1}{2}$. With the generator curve Λ_2, the bottleneck is no longer the composition but this single rectangular product, and a further improvement of this application would have to avoid forming that product. We are not aware of better bounds for these problems in the algebraic model over an arbitrary $\mathbb{F}_q$; the deterministic normal-basis algorithms of Giesbrecht, Jamshidpey and Schost [12] are analysed in the bit model.

E.3 Further consequences

The single composition of Theorem 6.21 computes a linear map in the outer polynomial once the data depending on (a, f) have been certified. Transposing this linear program gives power projection, from which we obtain minimal and characteristic polynomials. Over finite fields we also obtain iterated Frobenius and an irreducibility test. Our results apply to every argument modulo a separable polynomial, using the reductions recalled in Section 1.3.

Lemma E.6 (a linear single composition)**.** *In the algorithms of Theorems 6.1 and 6.21, replace the reduction step by Popov conversion, the remainder table of Lemma 3.3 and Lemma 3.4 with $k = 1$. This does not change the cost $\tilde{O}(n^{(\omega+3)/4})$ or the failure event, and once every step that depends only on (f, a) and the random elements has succeeded, the map $g \mapsto g(a) \operatorname{rem} f$ on $\mathbb{K}[y]_{<N}$, $N = O(n)$, is computed by a linear straight-line program of length $\tilde{O}(n^{(\omega+3)/4})$.*

Proof. The replacement is needed because the reduction of [22, Lemma 2.7] uses an input-dependent minimal nullspace basis, and so it is not a linear straight-line program in the outer polynomial. In Lemma 6.5 and Proposition 6.17(i), the relation matrix is certified before g is read. Let m_R be its actual dimension, so that $1 \le m_R \le 2m$, and convert it to its Popov form

P by Lemma 2.3, at cost $\tilde{O}(m_R^\omega d)$, with degree still at most d. Then use block length d, $J = \lceil N/d \rceil = O(m)$, and the precomputation of Lemma 2.2(a) with $\delta = 2d$. This precomputation and the remainder table cost $\tilde{O}(d(m_R^\omega + \mathsf{MM}(m_R, m_R, J))) \subseteq \tilde{O}(m^\omega d)$. By Lemma 3.4, reducing $g\, e_1$ then costs $\tilde{O}(d(\mathsf{MM}(m_R, J, 1) + \mathsf{MM}(m_R, m_R, 1))) \subseteq \tilde{O}(m^2 d)$, linearly in g, and produces a polynomial of x-degree less than m_R and y-degree less than d, keeping all m_R components. Next, apply [22, Prop. 3.1] with x-width m_R and auxiliary dimension $\mu = m$: its hypotheses hold since $d \le m^3$, and its cost is $\tilde{O}(m^\omega(d + m_R m)) = \tilde{O}(m^\omega d)$ because $m_R \le 2m$ and $m^2 = O(d)$. For fixed auxiliary data the computation is linear in the polynomial being evaluated. With $m = \Theta(n^{1/4})$ and $d = \Theta(n^{3/4})$ these bounds are $\tilde{O}(n^{(\omega+3)/4})$, and bounded local degrees are handled directly.

In Theorem 6.21, the decomposition, the annihilators, the powers $a^i \operatorname{rem} F$ of Lemma 6.19, the power $a^M \operatorname{rem} h_*$ and the subproduct tree depend only on (f, a). Reduction modulo the annihilators, products by these fixed matrices, the terminal blocking with Horner's rule in a^M, and Chinese remaindering are linear in g. For an outer input length $N = O(n)$, the terminal blocking of Theorem 6.21 costs $\lceil N/M \rceil\, \tilde{O}(M^{(\omega+3)/4}) \subseteq \tilde{O}(n^{(\omega+3)/4})$. All repetitions and certification tests involve only the fixed data, and once they succeed, no test or division in the online computation depends on g. □

Corollary E.7 (power projection and minimal polynomials)**.** *Let $f \in \mathbb{K}[x]$ be monic and separable of degree n and let $a \in \mathbb{K}[x]_{<n}$ be arbitrary. The following can be computed by Las Vegas algorithms using an expected $\tilde{O}(n^{(\omega+3)/4})$ operations in $\mathbb{K}$ (over small fields, with the extension-field convention of Theorem 6.1):*

(i) for a linear form ℓ on $\mathcal{A}$ and $N = O(n)$, the values $\ell(a^i)$, $0 \le i < N$;

(ii) the minimal polynomial μ_a of a mod f;

(iii) if $\mathbb{K}$ has characteristic 0 or $p > n$, the characteristic polynomial χ_a.

Proof. (i) Run the precomputation of Lemma E.6 until it succeeds and transpose the online linear straight-line program, as in the proof of Theorem E.2. Its graph computes the correct map for every outer polynomial, so its transpose is correct for every ℓ, and the transposition adds $O(n + N)$ operations.

(ii) Choose $\operatorname{card}(S) \ge 16n^2$ and the coordinates of ℓ independently and uniformly in S, and compute $\ell(a^i)$ for $i < 2n$ by (i). The fast Berlekamp–Massey algorithm returns the monic minimal recurrence polynomial μ_ℓ of this sequence in $\tilde{O}(n)$ operations. The sequence has order at most n, so $2n$ terms suffice and $\mu_\ell \mid \mu_a$. Writing $D = \deg \mu_a$, the relevant $D \times D$ Hankel determinant is a nonzero polynomial of degree $D \le n$ in the coordinates of ℓ, so $\mu_\ell = \mu_a$ with probability at least $1 - n/\operatorname{card}(S)$ (see also [21, Lemma 3.2]). Evaluate $\mu_\ell(a) \operatorname{rem} f$ with the certified linear program of Lemma E.6 for input length $2n$, which permits $\deg \mu_\ell = n$. If the result is zero, then $\mu_a \mid \mu_\ell$, hence $\mu_\ell = \mu_a$, and otherwise repeat with a fresh form. Since every composition precomputation is repeated on Fail, acceptance is exact.

(iii) Put $F(t) = t^n f(t^{-1})$, so that $F(0) = 1$. The first n coefficients of $n - tF'(t)/F(t) = \sum_{i \ge 0} \operatorname{Tr}(x^i) t^i$ give the vector of the trace form in $\tilde{O}(n)$ operations, without division by integers. By (i) with $N = n + 1$, compute $s_i = \operatorname{Tr}(a^i) = \sum_\nu a(\zeta_\nu)^i$ for $1 \le i \le n$, and these are the power sums of the eigenvalues of multiplication by a. Writing $\chi_a = z^n + c_1 z^{n-1} + \cdots + c_n$ with $c_0 = 1$, Newton's identities $k c_k = -\sum_{j=1}^{k} c_{k-j} s_j$ recover χ_a in $\tilde{O}(n)$ operations by their fast form [5], and the divisions by $1, \ldots, n$ are valid under the hypothesis on the characteristic. □

Remark E.8. For fixed f and generic a, [22, §6.2] already obtains the characteristic polynomial at this exponent. Its Appendix C discusses a generic straight-line-program equivalence. [21, Thm. 1.1] gives Las Vegas modular composition at exponent κ for every input, and its Prop. 10.1 and §10.1.2 state additional hypotheses for certified minimal and characteristic polynomials.

Here the improved scope is every argument modulo a separable f. In small characteristic, power sums alone need not determine χ_a, because a fiber whose size is divisible by p contributes nothing to them. We do not treat this case, and the bound $p > n$ in (iii) cannot be replaced by $p \geq n$.

Corollary E.9 (iterated Frobenius and irreducibility)**.** *Let $f \in \mathbb{F}_q[x]$ be nonconstant of degree n, normalized to be monic.*

(i) If f is square-free, then for $k \geq 1$, $x^{q^k} \operatorname{rem} f$ can be computed by a Las Vegas algorithm using an expected $n^{(\omega+3)/4+o(1)} \log k + n^{1+o(1)} \log q$ operations in $\mathbb{F}_q$.

(ii) Whether f is irreducible can be decided by a Las Vegas algorithm using an expected $n^{(\omega+3)/4+o(1)} + n^{1+o(1)} \log q$ operations in $\mathbb{F}_q$.

Proof. (i) Compute $G_1 = x^q \operatorname{rem} f$ by repeated squaring, in $n^{1+o(1)} \log q$ operations. Since the coefficients of G_i are fixed by the q-power map, $G_{i+j} = G_i(G_j) \operatorname{rem} f$ [16, Lemma 4], so a binary ladder gives G_k with at most $2\lfloor \log_2 k \rfloor$ compositions. Since f is square-free, the q-power map is an automorphism of $\mathcal{A}$. The arguments G_j are images of x under automorphisms and generate $\mathcal{A}$, so Theorem 6.1 applies to each composition with its extension-field convention. The extension supplies random elements, while q remains the cardinality of the original field.

(ii) A polynomial of degree one is irreducible. For $n \geq 2$, reject if $\gcd(f, f') \neq 1$, since irreducible polynomials over finite fields are separable, and otherwise f is square-free. By Rabin's criterion [36, Cor. 14.35 and Algorithm 14.36], f is irreducible exactly when $G_n = x \operatorname{rem} f$ and $\gcd(G_{n/r} - x, f) = 1$ for every prime divisor r of n. Then compute G_1 once and reuse it for the $O(\log n)$ targets, each requiring $O(\log n)$ compositions by (i). The gcds cost $\tilde{O}(n \log n)$. This is the classical reduction analyzed in [36, Thm. 14.37], with our composition bound. □

Remark E.10. With one composition of [21] at a time, the same algorithms cost $n^{\kappa+o(1)}$ plus $n^{1+o(1)} \log q$, and with Brent–Kung compositions $n^{(\omega+1)/2+o(1)}$, or $n^{\omega_2/2+o(1)}$ with rectangular multiplication. In the bit model, [17, §8.2] already gives almost-linear irreducibility testing, and its small-characteristic algebraic construction gives almost-linear bounds in field operations for fixed finite fields, once the costs of extension and inverse Frobenius are included [17, §6.1, Cor. 7.2]. Our bound holds uniformly in the characteristic, in the algebraic model of this paper.

E.4 Moduli with repeated factors

The separability of f is used only through the evaluation at its roots. For moduli with repeated factors, the reduction of [21, §9], which follows van der Hoeven and Lecerf, reduces composition modulo h^ℓ to compositions modulo the separable polynomial h; with the batches of Theorem 6.23, the exponents of the separable case are preserved. A *separable power decomposition* of f is a factorization $f = \prod_{i \leq s} h_i^{\ell_i}$ into pairwise coprime monic separable polynomials h_i; it exists when $\mathbb{K}$ is perfect, for instance finite.

Theorem E.11 (repeated factors)**.** *Let $f \in \mathbb{K}[x]$ be monic of degree n, given with a separable power decomposition, and let $a \in \mathbb{K}[x]_{<n}$ be arbitrary. There is a Las Vegas algorithm which, given $g_1, \ldots, g_k \in \mathbb{K}[y]_{<n}$ with $k = n^t$, computes all $g_j(a) \operatorname{rem} f$ using an expected $n^{\Lambda_4(t)+o(1)}$ operations in $\mathbb{K}$; in particular, one composition costs $n^{\kappa_4+o(1)}$. Over $\mathbb{F}_q$, a separable power decomposition of an arbitrary f can be computed in $n^{1+o(1)} \log q$ operations, so that the same bounds hold for every modulus, with an additional term $n^{1+o(1)} \log q$.*

Proof. By the Chinese remainder theorem it suffices to treat each $f_i = h_i^{\ell_i}$, of degree $n_i = d_i \ell_i$, and to recombine in $\tilde{O}(kn)$ operations; $a \operatorname{rem} f_i$ for all i is computed by a remainder tree in $\tilde{O}(n)$ operations. Fix $f = h^\ell$ with h separable of degree d, and put $L = \mathbb{K}[\theta]/\langle h \rangle$, a product of fields.

The untangling $\psi_{h,\ell}\colon \mathbb{K}[x]/\langle h^\ell\rangle \to \mathbb{K}[\theta,z]/\langle h(\theta),(z-\theta)^\ell\rangle$, $x\mapsto z$, is an isomorphism since h is separable, and it and its inverse are applied in $\tilde{O}(n)$ operations [21, Lemmas 9.3 and 9.4]. Let $A=\psi_{h,\ell}(a)$ and $\alpha = a \operatorname{rem} h$, so that $A\equiv\alpha \bmod (z-\theta)$.

First compute the minimal polynomial μ of α modulo h, of degree at most d, by Corollary E.7 with the exponent κ_4 of Section 6.3; then $\mu(A)^\ell\equiv 0$, so that every g_j may be replaced by $g_j \operatorname{rem}\mu^\ell$. These reductions, for all j and all factors, use a remainder tree with moduli of total degree at most n, in $\tilde{O}(kn)$ operations. Next compute $\tilde{G}_j=\psi_{\mu,\ell}(g_j\operatorname{rem}\mu^\ell)=\sum_{i<\ell}\tilde{G}_{j,i}(\gamma)y^i$ with $\deg\tilde{G}_{j,i}<\deg\mu$, in $\tilde{O}(\ell d)$ operations each (untangling does not require separability), and the $k\ell$ compositions $G_{j,i}=\tilde{G}_{j,i}(\alpha)\operatorname{rem}h$ by Theorem 6.23. As in [21, §9.3.2, Eq. (35)], $G_j(\theta,y)=\sum_i G_{j,i}(\theta)y^i$ satisfies $G_j(\theta,A)\equiv g_j(A)$ in $L[z]/\langle(z-\theta)^\ell\rangle$.

It remains to evaluate $G_j(\theta,A)$ modulo $(z-\theta)^\ell$ over L. Substituting $z=\theta+T$, write $A(\theta,\theta+T)=\alpha+\Delta(T)$ with $\Delta(0)=0$, and compute the Taylor shift $H_j(T)=G_j(\theta,\alpha+T)\in L[T]_{<\ell}$, in $\tilde{O}(\ell)$ operations in L by quasi-linear translation over a commutative ring. Then $G_j(\theta,A)\equiv H_j(\Delta(T)) \bmod T^\ell$, a power series composition over the commutative ring L, which costs $\tilde{O}(\ell)$ operations in L [18, Thm. 1]. Each operation in L costs $\tilde{O}(d)$, so that this step and the tangling cost $\tilde{O}(kn)$ in total.

For the cost, the minimal polynomials cost $\sum_i d_i^{\kappa_4+o(1)}\le n^{\kappa_4+o(1)}$. With s the dimension exponent attaining $\Lambda_4(t)$ and F the maximum of its five terms, as in the proof of Theorem 6.23, treat the factors with $d_i=1$ directly, in $\tilde{O}(k\ell_i)$ operations, since then the compositions modulo h_i are evaluations at a point; for the others write $k=n_i^{t_i}$, $d_i=n_i^{\mu_i}$ and $k\ell_i=d_i^{u_i}$, so that $\mu_i u_i=t_i+1-\mu_i$ and $t_i\ge t$. The batch of $k\ell_i$ compositions modulo h_i costs $d_i^{F(u_i)+o(1)}=n_i^{\mu_i F(u_i)+o(1)}\le n_i^{F(t_i)+o(1)}$, by the inequality $\mu F(u)\le F(t)$ proved there. Since F grows with slope at most 1, $n_i^{F(t_i)}\le n_i^{F(t)-t}n_i^{t_i}=n_i^{F(t)-t}n^t$, and since $F(t)-t\ge 1$ and $\sum_i n_i=n$, the sum over the factors is at most $n^{F(t)-t}n^t=n^{F(t)}$; the uniformity of $o(1)$ is obtained as in that proof. Only the minimal polynomials and the batches use random elements, and they are repeated on failure.

Over $\mathbb{F}_q$, the square-free decomposition of f [36, §14.6] costs $\tilde{O}(n)$ operations except for the extraction of pth roots of the coefficients of polynomials in x^p, which costs $O(\log q)$ operations per coefficient since $c^{1/p}=c^{q/p}$; since $\mathbb{F}_q$ is perfect, the square-free parts are separable. □

For an imperfect field, factors $h(x^{p^e})^\ell$ with $e>0$ remain, and the reduction of [21, §9.4] then also needs a composition modulo $(z^{p^e}-\theta)^\ell$ over L, an inseparable modulus, for which the general exponent κ of [21] is available. Over a perfect field other than a finite field, the theorem applies once a separable power decomposition is supplied; its computation requires pth roots, whose cost depends on the representation of the field.

E.5 The limit of the Kaltofen–Shoup framework

As pointed out in [16, §1], Algorithm D cannot reach an exponent below 3/2 when its baby steps and interval polynomials are written explicitly: steps (D1) and (D3) write $n(l+1)$ and nm_D field elements, and $lm_D\ge n/2$ gives $n(l+m_D)\ge 2n\sqrt{lm_D}\ge\sqrt{2}\,n^{3/2}$. More precisely, let Λ be a continuous nondecreasing batch curve with $\Lambda(0)\le 2$ and $\Lambda(t)\ge 1+t$, the latter because a batch of size k writes kn elements. Optimized over β, the bound of Proposition 5.3 has exponent $\min_\beta\max\{1+\beta,2-\beta,\Lambda(1-\beta)\}$, in which the middle term is never the unique maximum, since $\Lambda(1-\beta)\ge 2-\beta$. As $1+\beta$ is increasing and $\Lambda(1-\beta)$ is nonincreasing, the minimum is $1+\beta^*$, where $\beta^*\in[0,1]$ is the solution of $1+\beta=\Lambda(1-\beta)$, and it equals 3/2 exactly when $\Lambda(1/2)\le 3/2$. This holds for Λ_1 and Λ_2 if $\omega=2$ (Corollary B.4 and Theorem 4.6). For batched Brent–Kung composition, $\Lambda(t)=\frac{\omega+1}{2}+\frac{\omega-1}{2}t$ gives the exponent $(3\omega-1)/(\omega+1)$ of [16, Thm. 3]. Algorithms of a different form reach 3/2 in the bit model [17] and in small characteristic [26], and Umans and Wang use a different arrangement of baby and giant steps [27].

References

[1] J. Alman, R. Duan, V. Vassilevska Williams, Y. Xu, Z. Xu, and R. Zhou, *More asymmetry yields faster matrix multiplication*, in Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2025, pp. 2005–2039, https://doi.org/10.1137/1.9781611978322.63, https://arxiv.org/abs/2404.16349.

[2] E. R. Berlekamp, *Factoring polynomials over large finite fields*, Math. Comp., 24 (1970), pp. 713–735, https://doi.org/10.1090/S0025-5718-1970-0276200-X.

[3] V. Bhargava, S. Ghosh, Z. Guo, M. Kumar, and C. Umans, *Fast multivariate multipoint evaluation over all finite fields*, J. ACM, 71 (2024), pp. 1–32, https://doi.org/10.1145/3652025.

[4] V. Bhargava, S. Ghosh, M. Kumar, and C. K. Mohapatra, *Fast, algebraic multivariate multipoint evaluation in small characteristic and applications*, J. ACM, 70 (2023), pp. 1–46, https://doi.org/10.1145/3625226.

[5] A. Bostan, G. Lecerf, and É. Schost, *Tellegen's principle into practice*, in Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation (ISSAC '03), ACM, 2003, pp. 37–44, https://doi.org/10.1145/860854.860870.

[6] R. P. Brent and H. T. Kung, *Fast algorithms for manipulating formal power series*, J. ACM, 25 (1978), pp. 581–595, https://doi.org/10.1145/322092.322099.

[7] P. Bürgisser, M. Clausen, and M. A. Shokrollahi, *Algebraic Complexity Theory*, vol. 315 of Grundlehren der mathematischen Wissenschaften, Springer, Berlin, Heidelberg, 1997, https://doi.org/10.1007/978-3-662-03338-8.

[8] D. G. Cantor and E. Kaltofen, *On fast multiplication of polynomials over arbitrary algebras*, Acta Inform., 28 (1991), pp. 693–701, https://doi.org/10.1007/BF01178683.

[9] D. G. Cantor and H. Zassenhaus, *A new algorithm for factoring polynomials over finite fields*, Math. Comp., 36 (1981), pp. 587–592, https://doi.org/10.1090/S0025-5718-1981-0606517-5.

[10] D. Coppersmith and S. Winograd, *Matrix multiplication via arithmetic progressions*, J. Symbolic Comput., 9 (1990), pp. 251–280, https://doi.org/10.1016/S0747-7171(08)80013-2.

[11] E. Dupont, M. Eisenberger, B. Kozlovskii, A. Mehrabian, F. J. R. Ruiz, A. See, R. Zhou, J. Alman, V. Vassilevska Williams, and M. Balog, *Improving the matrix multiplication exponent with modern optimization and AlphaEvolve*, 2026, https://arxiv.org/abs/2608.16884.

[12] M. Giesbrecht, A. Jamshidpey, and É. Schost, *Fast deterministic normal bases and circulant polynomial determinants*, 2026, https://arxiv.org/abs/2607.00313.

[13] Z. Guo, A. K. Narayanan, and C. Umans, *Algebraic problems equivalent to beating exponent 3/2 for polynomial factorization over finite fields*, in 41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016), vol. 58 of LIPIcs, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016, pp. 47:1–47:14, https://doi.org/10.4230/LIPIcs.MFCS.2016.47, https://arxiv.org/abs/1606.04592.

[14] X. Huang and V. Y. Pan, *Fast rectangular matrix multiplication and applications*, J. Complexity, 14 (1998), pp. 257–299, https://doi.org/10.1006/jcom.1998.0476.

[15] E. Kaltofen and A. Lobo, *Factoring high-degree polynomials by the black box Berlekamp algorithm*, in Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC '94), 1994, pp. 90–98, https://doi.org/10.1145/190347.190371.

[16] E. Kaltofen and V. Shoup, *Subquadratic-time factoring of polynomials over finite fields*, Math. Comp., 67 (1998), pp. 1179–1197, https://doi.org/10.1090/S0025-5718-98-00944-2.

[17] K. S. Kedlaya and C. Umans, *Fast polynomial factorization and modular composition*, SIAM J. Comput., 40 (2011), pp. 1767–1802, https://doi.org/10.1137/08073408X.

[18] Y. Kinoshita and B. Li, *Power series composition in near-linear time*, in 2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, 2024, pp. 2180–2185, https://doi.org/10.1109/FOCS61266.2024.00127.

[19] F. Le Gall and F. Urrutia, *Improved rectangular matrix multiplication using powers of the Coppersmith–Winograd tensor*, in Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2018, pp. 1029–1046, https://doi.org/10.1137/1.9781611975031.67.

[20] V. Neiger, J. Rosenkilde, and G. Solomatov, *Generic bivariate multi-point evaluation, interpolation and modular composition with precomputation*, in Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation (ISSAC '20), ACM, 2020, pp. 388–395, https://doi.org/10.1145/3373207.3404032, https://arxiv.org/abs/2003.12468.

[21] V. Neiger, B. Salvy, É. Schost, and G. Villard, *Faster modular composition*, J. ACM, 71 (2024), pp. 1–79, https://doi.org/10.1145/3638349, https://arxiv.org/abs/2110.08354.

[22] V. Neiger, B. Salvy, É. Schost, and G. Villard, *Faster modular composition using two relation matrices*, in Proceedings of the 2026 International Symposium on Symbolic and Algebraic Computation (ISSAC '26), ACM, 2026, pp. 315–324, https://doi.org/10.1145/3815436.3815451, https://arxiv.org/abs/2601.17422.

[23] V. Neiger and T. X. Vu, *Computing canonical bases of modules of univariate relations*, in Proceedings of the 2017 ACM International Symposium on Symbolic and Algebraic Computation (ISSAC '17), ACM, 2017, pp. 357–364, https://doi.org/10.1145/3087604.3087656.

[24] É. Schost, *Modular composition for decomposable polynomials*, ACM Commun. Comput. Algebra, 59 (2025), pp. 106–109, https://doi.org/10.1145/3832046.3832055.

[25] V. Shoup, *Fast construction of irreducible polynomials over finite fields*, J. Symbolic Comput., 17 (1994), pp. 371–391, https://doi.org/10.1006/jsco.1994.1025.

[26] C. Umans, *Fast polynomial factorization and modular composition in small characteristic*, in Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing (STOC '08), ACM, 2008, pp. 481–490, https://doi.org/10.1145/1374376.1374445.

[27] C. Umans and S. Wang, *A number-theoretic conjecture implying faster algorithms for polynomial factorization and integer factorization*, 2025, https://arxiv.org/abs/2511.10851.

[28] J. van der Hoeven and G. Lecerf, *Modular composition via factorization*, J. Complexity, 48 (2018), pp. 36–68, https://doi.org/10.1016/j.jco.2018.05.002.

[29] J. van der Hoeven and G. Lecerf, *Amortized bivariate multi-point evaluation*, in Proceedings of the 2021 International Symposium on Symbolic and Algebraic Computation (ISSAC '21), ACM, 2021, pp. 179–185, https://doi.org/10.1145/3452143.3465531.

[30] J. van der Hoeven and G. Lecerf, *Fast amortized multi-point evaluation*, J. Complexity, 67 (2021), p. 101574, https://doi.org/10.1016/j.jco.2021.101574.

[31] J. van der Hoeven and G. Lecerf, *Amortized multi-point evaluation of multivariate polynomials*, J. Complexity, 74 (2023), p. 101693, https://doi.org/10.1016/j.jco.2022.101693.

[32] J. van der Hoeven and G. Lecerf, *Univariate polynomial factorization over finite fields with large extension degree*, Appl. Algebra Engrg. Comm. Comput., 35 (2024), pp. 121–149, https://doi.org/10.1007/s00200-021-00536-1.

[33] J. van der Hoeven and G. Lecerf, *Faster multi-point evaluation over any field*, Appl. Algebra Engrg. Comm. Comput., (2025), https://doi.org/10.1007/s00200-025-00704-7. Online first.

[34] J. van der Hoeven and G. Lecerf, *Fast modular composition using spiroids*, J. Complexity, 96 (2026), p. 102073, https://doi.org/10.1016/j.jco.2026.102073.

[35] V. Vassilevska Williams, Y. Xu, Z. Xu, and R. Zhou, *New bounds for matrix multiplication: from alpha to omega*, in Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2024, pp. 3792–3835, https://doi.org/10.1137/1.9781611977912.134, https://arxiv.org/abs/2307.07970.

[36] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, Cambridge University Press, 2013, https://doi.org/10.1017/CBO9781139856065.

[37] J. von zur Gathen and V. Shoup, *Computing Frobenius maps and factoring polynomials*, Comput. Complexity, 2 (1992), pp. 187–224, https://doi.org/10.1007/BF01272074.