Faster modular composition using two relation matrices

Vincent Neiger
Sorbonne Université, CNRS, LIP6
F-75005 Paris, France

Bruno Salvy
Inria, ENS de Lyon, CNRS, UCBL, LIP UMR 5668
Lyon, France

Éric Schost
University of Waterloo
Cheriton School of Computer Science
Waterloo, Canada

Gilles Villard
CNRS, ENS de Lyon, Inria, UCBL, LIP UMR 5668
Lyon, France

Abstract

Modular composition is the problem of computing the composition of two univariate polynomials modulo a third one. For a long time, the algebraic algorithm with the best complexity bound for this problem was that of Brent and Kung (1978). Recently, we improved this algorithm by computing and using a polynomial matrix that encodes a certain basis of algebraic relations between the polynomials. This is further improved here by making use of two polynomial matrices of smaller dimension. Under genericity assumptions on the input, this results in an algorithm using $\tilde{O}(n^{(\omega+3)/4})$ arithmetic operations in the base field, where ω is the exponent of matrix multiplication. With naive matrix multiplication, this is $\tilde{O}(n^{3/2})$, while with the best currently known exponent ω this is $O(n^{1.343})$, improving upon the previously most efficient algorithms.

Keywords

Modular composition; polynomial matrices; minimal bases.

ACM Reference Format:
Vincent Neiger, Bruno Salvy, Éric Schost, and Gilles Villard. 2026. Faster modular composition using two relation matrices. In *51st International Symposium on Symbolic and Algebraic Computation (ISSAC '26), July 13–17, 2026, Oldenburg, Germany*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3815436.3815451

1 Introduction

Given three polynomials $a, f \in \mathbb{K}[x]$ and $g \in \mathbb{K}[y]$ with coefficients in a field $\mathbb{K}$, modular composition asks to compute $g(a) \bmod f$.

Quasi-optimal solutions to this problem are known in special cases and complexity models. In particular, a famous algorithm of Kedlaya and Umans solves it in $n^{1+\epsilon}\log(q)^{1+o(1)}$ bit operations when $\mathbb{K} = \mathbb{F}_q$ is a finite field [23], where n is the degree of f; see also [5, 14] for detailed analyses and further improvements. Very recently, the case of formal power series (where $f = x^n$) was solved in quasi-linear complexity by Kinoshita and Li [24], in the algebraic model of computation, where the complexity is expressed in terms of the number of arithmetic operations in $\mathbb{K}$.

In the general case and in the algebraic complexity model, the first non-trivial algorithm was due to Brent and Kung [9]. It has been an open problem for a long time to improve upon it and reach a complexity exponent below 3/2 [10, 13]. This was achieved recently (see [30] where more motivation can be found). The present article brings further improvement to this general situation.

The main parameter in complexity estimates is $n = \deg(f)$. Indeed, we may focus on the case $\deg(a) < n$, to which the general case reduces via $g(a) \bmod f = g(a \bmod f) \bmod f$; and up to using a y^n-adic expansion of g, we may restrict to $\deg(g) < n$. We use the $\tilde{O}(\cdot)$ notation, which hides factors that are polylogarithmic in n.

Recall that using fast polynomial multiplication, sums, products and divisions modulo f can be computed in $\tilde{O}(n)$ operations in $\mathbb{K}$ [13]. Thus, the most natural algorithm for modular composition, based on Horner evaluation, has complexity $\tilde{O}(n^2)$, which is in agreement with the number $\Theta(n^2)$ of coefficients it computes.

Brent and Kung's algorithm. Hereafter, we use standard notation such as $\mathbb{K}[x]_{<m}$ for the set of univariate polynomials in x with coefficients in $\mathbb{K}$ and degree less than m, and $\mathbb{K}[x,y]_{<(m,d)}$ for the set of bivariate polynomials in x, y of bidegree less than (m, d).

Brent and Kung [9] introduced the following *baby steps-giant steps* algorithm, where $m = \lceil\sqrt{n}\rceil$.

(1.1) Reduce the polynomials $a, a^2, \ldots, a^m$ modulo f;
(1.2) Build $\bar{G}$ in $\mathbb{K}[y_0, y_1]_{<(m,m)}$ such that $g = \bar{G}(y, y^m)$, and write $\bar{G} = \bar{g}_0(y_0) + \bar{g}_1(y_0)y_1 + \cdots + \bar{g}_{m-1}(y_0)y_1^{m-1}$;
(1.3) Using the a^i's of Step (1.1), compute $\bar{G}(a, y_1) \bmod f$ via the multiplication of two matrices in $\mathbb{K}^{m\times m}$ and $\mathbb{K}^{m\times n}$, which simultaneously yields all $\bar{g}_i(a)$ modulo f;
(1.4) Deduce $g(a) = \bar{G}(a, a^m) \bmod f$ using Horner evaluation by evaluating y_1 at a^m modulo f.

This method uses $\tilde{O}(m^{\omega+1}) = \tilde{O}(n^{(\omega+1)/2})$ operations in $\mathbb{K}$, later improved by Huang and Pan using fast rectangular matrix multiplication [19]. The total number of coefficients computed by this method is of order $n^{3/2}$, which explains the previous barrier of 3/2 on the exponent of the algebraic complexity.

Polynomial matrix algorithms have recently been introduced in this context, to make use of bivariate polynomials. A simplified version of the algorithm in [30], which applies when $f(0) \neq 0$ and the input a is generic (in the Zariski sense), can be sketched as follows. It uses a smaller parameter $m = \lceil n^{1/3}\rceil$, and the complementary parameter $d = \lceil n/m\rceil$ which is of order $\Theta(m^2) = \Theta(n^{2/3})$.

In what follows, for a polynomial $g = \sum_i g_i x^i$ in $\mathbb{K}[x]$ and non-negative integers u, k, we write $[g]_u^k = g_u + g_{u+1}x + \cdots + g_{u+k}x^k$; $g \text{ rem } f$ denotes the remainder of g in the Euclidean division by f.

(2.1) Reduce $a^2, \ldots, a^m, a^{2m}, \ldots, a^{(m-1)m}$ modulo f;
(2.2) Compute $[x^i a^{-k} \text{ rem } f]_0^{m-1}$ for $0 \le i < m$ and $0 \le k < 2d$;

This work is licensed under a Creative Commons Attribution 4.0 International License.
ISSAC '26, Oldenburg, Germany
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2595-1/2026/07
https://doi.org/10.1145/3815436.3815451

(2.3) Deduce a basis of y-degree at most d for the $\mathbb{K}[y]$-module

$$\mathcal{M}_m = \{p(x,y) \in \mathbb{K}[x,y]_{<(m,\cdot)} \mid p(x,a) = 0 \bmod f\};$$

(2.4) Reduce g by this basis to obtain $G \in \mathbb{K}[x,y]_{<(m,d)}$ such that $G(x,a) = g(a)$ modulo f;
(2.5) Build $\bar{G}$ in $\mathbb{K}[x,y_0,y_1]_{<(m,m,m)}$ such that $G = \bar{G}(x,y,y^m)$, and write $\bar{G} = \sum_{0\le i<m} \bar{g}_i(x,y_0)y_1^i$;
(2.6) Using the a^i's of Step (2.1), compute $\bar{G}(x,a,y_1) \bmod f$, via the multiplication of matrices in $\mathbb{K}[x]^{m\times m}_{<m}$ and $\mathbb{K}[x]^{m\times d}_{<m}$, which simultaneously yields all $\bar{g}_i(x,a)$ modulo f;
(2.7) Deduce $g(a) = \bar{G}(x,a,a^m) \bmod f$ using Horner evaluation with a^m modulo f.

The basis in Step (2.3) is given as a matrix in $\mathbb{K}[y]^{m\times m}$, called a *relation matrix*. For any f and for a generic a, it has degree d (see Proposition 2.6), so that its total size is $O(n^{4/3})$ field elements. This is also an upper bound on the size of the other intermediate objects computed by the algorithm. In these conditions, the whole procedure uses $\tilde{O}(n^{(\omega+2)/3})$ operations in $\mathbb{K}$. This can be improved slightly by taking advantage of fast rectangular matrix products, and this forms the basis for a more complex algorithm that handles arbitrary input by Las Vegas randomization [30].

Modular composition with bivariate input. Steps (2.5) to (2.7) above are actually following an algorithm due to Nüsken and Ziegler to compute $G(x,a) \bmod f$ for a bivariate polynomial G [31]. This article presents an acceleration of this bivariate modular composition, under genericity assumptions. This is achieved thanks to another relation matrix, this time in $\mathbb{K}[x]^{\mu\times\mu}$, for a new parameter μ, which is computed efficiently as a basis of the $\mathbb{K}[x]$-module

$$\mathcal{N}_\mu = \{p(x,y) \in \mathbb{K}[x,y]_{<(\cdot,\mu)} \mid p(x,a) = 0 \bmod f\}.$$

The composition is then performed as follows:

(3.1) Reduce $a^{j\mu^i}$ modulo f, for $i = 0,1,2$ and $0 \le j < \mu$;
(3.2) Compute a small-degree basis of $\mathcal{N}_\mu$, and let δ be its degree;
(3.3) Simultaneously reduce the polynomials of (3.1) by this basis to obtain A_j, B_j in $\mathbb{K}[x,y]_{<(\delta,\mu)}$, for $0 \le j < \mu$, such that

$$A_j(x,a) = a^{j\mu} \bmod f \quad \text{and} \quad B_j(x,a) = a^{j\mu^2} \bmod f;$$

(3.4) Build the polynomial $\bar{G}$ in $\mathbb{K}[x,y_0,y_1,y_2]_{<(m,\mu,\mu,\mu)}$ such that $G(x,y) = \bar{G}(x,y,y^\mu,y^{\mu^2})$;
(3.5) Use $\bar{G}$ and the A_j's and B_j's of Step (3.3) to compute

$$G(x,a) = \bar{G}(x,a,a^\mu,a^{\mu^2}) \bmod f$$

by multiplying matrices in $\mathbb{K}[x,y]^{\mu\times\mu}_{<(m,\mu)}$ and $\mathbb{K}[x,y]^{\mu\times\mu}_{<(\lceil\delta/\mu\rceil,\mu)}$.

For any μ, and for a generic a, we prove that the degree δ of the relation matrix of Step (3.2) satisfies $\delta = \lceil n/\mu \rceil$. Choosing μ in $\Theta(d^{1/3})$, we obtain the following result.

Theorem 1.1. *For any f in $\mathbb{K}[x]$ of degree $n > 0$ and any m, d in $\mathbb{N}_{>0}$ such that $md \in \Theta(n)$, there exists a non-empty Zariski-open subset $\mathcal{V}_{f,d}$ of $\overline{\mathbb{K}}^n$ such that for $(a_0,\dots,a_{n-1})$ in $\mathcal{V}_{f,d} \cap \mathbb{K}^n$, $a = \sum_{0\le i<n} a_i x^i$, and any $G \in \mathbb{K}[x,y]_{<(m,d)}$, one can compute the composition $G(x,a) \bmod f$ in $\tilde{O}(md^{(\omega+2)/3})$ operations in $\mathbb{K}$.*

In the univariate case with $(m,d) = (1,n)$, this provides us with an alternative way to recover the operation count $\tilde{O}(n^{(\omega+2)/3})$ of the algorithm in Steps (2.1)-(2.7), at least for generic input. In both cases, the dimensions of the relation matrices are of order $n^{1/3}$.

This type of bivariate composition is also at the core of Nüsken and Ziegler's bivariate multipoint evaluation [31, Algo. 11]. For this question, our improved algorithm leads to the following.

Theorem 1.2. *Let m, d, n be in $\mathbb{N}_{>0}$ with $md \in \Theta(n)$. For any $\xi = (x_1,\dots,x_n)$ in $\mathbb{K}^n$ with pairwise distinct x_i, there exists a non-empty Zariski-open subset $\mathcal{W}_{\xi,d}$ of $\overline{\mathbb{K}}^n$ such that for $(y_1,\dots,y_n)$ in $\mathcal{W}_{\xi,d} \cap \mathbb{K}^n$, any $G(x,y) \in \mathbb{K}[x,y]_{<(m,d)}$ can be evaluated at all $(x_i,y_i) \in \mathbb{K}^2$, for $1 \le i \le n$, in $\tilde{O}(md^{(\omega+2)/3})$ operations in $\mathbb{K}$.*

Theorems 1.1 and 1.2 are proved in §3. For comparison, Nüsken and Ziegler's method has complexity $\tilde{O}(md^{\omega_2/2})$, with no genericity assumption [31, Thm. 8]. There, ω_2 is such that one can multiply an $s\times s$ by an $s\times s^2$ matrix in $O(s^{\omega_2})$ operations. One can take $\omega_2 = \omega + 1$ and then our algorithm has a better complexity bound. This persists with the slightly better bounds known for ω_2 [1, 27, 35].

New univariate composition algorithm. For our question of computing $g(a) \bmod f$, we combine the ideas from both preceding algorithms, using relation matrices for both modules: after choosing our parameter m, we reduce g by the basis of $\mathcal{M}_m$ to obtain a bivariate polynomial G in $\mathbb{K}[x,y]_{<(m,d)}$, with $d = \lceil n/m \rceil$, then we use the algorithm of the previous paragraph to evaluate $G(x,a) \bmod f$.

This time, the optimal choice for m is $m = \lceil n^{1/4} \rceil$; this gives $d \in \Theta(n^{3/4})$ and allows us to set the parameter μ in the previous paragraph to $\mu = m$ (which is indeed in $\Theta(d^{1/3})$). The resulting algorithm looks as follows:

(4.1) Perform Steps (3.1)-(3.3), with $\mu = m$;
(4.2) Use the polynomials A_j and B_j to compute the truncations $[x^i a^{-k-1} \text{ rem } f]_0^{m-1}$ for $0 \le i < m$ and $0 \le k < 2d$;
(4.3) From these, compute a basis of $\mathcal{M}_m$ of degree at most d;
(4.4) Reduce g by this basis to obtain $G \in \mathbb{K}[x,y]_{<(m,d)}$ such that $G(x,a) = g(a)$ modulo f;
(4.5) Conclude with Steps (3.4)-(3.5)

Step (4.2) is the most technical one. The corresponding algorithm (in §4) improves on the one used in [30] by making use of the polynomials A_j and B_j introduced in the previous paragraph, and manipulates objects of size $O(mn)$ using $\tilde{O}(m^{\omega-1}n)$ arithmetic operations. This is the same complexity bound as the final composition in Step (4.5), in the same way as Steps (2.2) and (2.6) had the same complexity bound in our previous algorithm. These apparent coincidences are explained by complexity equivalences, discussed in §6. Altogether, we obtain the following result proved in §5.

Theorem 1.3. *For any f in $\mathbb{K}[x]$ of degree $n > 0$, there exists a non-empty Zariski-open subset $\mathcal{O}_f$ of $\overline{\mathbb{K}}^n$ such that, for $(a_0,\dots,a_{n-1})$ in $\mathcal{O}_f \cap \mathbb{K}^n$, $a = \sum_{0\le i<n} a_i x^i$, and any $g \in \mathbb{K}[x]_{<n}$, one can compute $g(a) \bmod f$ using $\tilde{O}(n^{(\omega+3)/4}) \subset O(n^{1.343})$ operations in $\mathbb{K}$.*

This complexity exponent satisfies

$$5/4 \le (\omega+3)/4 \le 1.343 \le 1.452 \le 3/2,$$

where the first approximation is obtained with the best known bound on ω (it improves upon the previous $O(n^{1.43})$ [30]); the second one uses Strassen's algorithm; the last one shows that for the first time, $3/2$ is reached even without fast matrix multiplication.

Along computations, the algorithm is able to detect whether genericity assumptions are met, and throw an error otherwise. Then one can fall back on other algorithms. It is natural to wonder

whether randomization could help and yield a Las Vegas algorithm, as it did in [30]; we leave this to future work.

2 Modules of relations and their bases

Let $a, f \in \mathbb{K}[x]$ be polynomials with $f \neq 0$ and let $n = \deg f$. The elements of the bivariate ideal $\mathcal{I} = \langle f(x), y - a(x)\rangle \subseteq \mathbb{K}[x, y]$ are called *relations* in what follows, and play an important role in our algorithms. In particular, we make use of relations of small degrees in x or y, by focusing on the $\mathbb{K}[y]$- and $\mathbb{K}[x]$-modules

$$\mathcal{M}_\mu = \mathcal{I}_{<(\mu,\cdot)} = \{p \in \mathbb{K}[x,y]_{<(\mu,\cdot)} \mid p(x,a) = 0 \bmod f\}, \quad (1)$$
$$\mathcal{N}_\mu = \mathcal{I}_{<(\cdot,\mu)} = \{p \in \mathbb{K}[x,y]_{<(\cdot,\mu)} \mid p(x,a) = 0 \bmod f\}. \quad (2)$$

Both are free and of rank μ. Their bases can be represented by nonsingular matrices in $\mathbb{K}[y]^{\mu\times\mu}$ and $\mathbb{K}[x]^{\mu\times\mu}$, respectively, each column containing the coefficients of a relation.

In this section, we show that for generic a, both $\mathcal{N}_\mu$ and $\mathcal{M}_\mu$ have bases of degree $\lceil n/\mu\rceil$ (Lemma 2.5 and Proposition 2.6), and we present algorithms computing such bases efficiently. We also show how a reduction by these bases allows one to compute the polynomials $A_j(x,y)$ and $B_j(x,y)$ for Step (3.3), as well as the polynomial G for Step (4.4) in Section 1 (Corollary 2.4 and Lemma 2.7).

2.1 Minimal bases

When dealing with free modules over a polynomial ring in one variable, one usually computes with bases (which we see as univariate polynomial matrices) having specific forms, to benefit from their smaller size and stronger properties. We first recall basic definitions and properties (see also [4, 21]).

Definition 2.1. Let $P \in \mathbb{K}[x]^{\mu\times\mu}$ be a polynomial matrix with no zero column and let d_j be the maximal degree of the entries in its column j, for $0 \le j < \mu$. We write $\deg(P)$ for $\max_{0\le j<\mu} d_j$. The matrix P is *column reduced* when $\deg(\det(P)) = d_0 + \cdots + d_{\mu-1}$. A basis of a free $\mathbb{K}[x]$-module is *minimal* if it is column reduced.

This terminology is explained by the fact that the column degrees of column reduced bases are lexicographically minimal among all bases of the module, up to permuting the columns to make these degrees nondecreasing. Minimal bases are not unique and the algorithms we use produce them in variants of Popov forms.

Definition 2.2. A matrix $P = (p_{ij}) \in \mathbb{K}[x]^{\mu\times\mu}$ with no zero column is in *weak Popov form* if for all $i \neq j$,

$$\deg(p_{ij}) \le \deg(p_{jj}), \quad \text{with strict inequality if } i > j. \quad (C_1)$$

It is in *Popov form* if moreover, for all i,

$$p_{ii} \text{ is monic and } \deg(p_{ii}) > \deg(p_{ij}), \quad i \neq j. \quad (C_2)$$

Finally, for a *shift* $\boldsymbol{s} = (s_0, \ldots, s_{\mu-1}) \in \mathbb{Z}^\mu$, the matrix is in *$\boldsymbol{s}$-Popov form* if it satisfies (C_2), while (C_1) is replaced by

$$\deg(p_{ij}) + s_i \le \deg(p_{jj}) + s_j, \quad \text{with strict inequality if } i > j.$$

For any given $\boldsymbol{s}$, a $\mathbb{K}[x]$-submodule of $\mathbb{K}[x]^\mu$ of rank μ admits a unique basis in $\boldsymbol{s}$-Popov form.

2.2 Computing modulo $\mathbb{K}[x]$-relations

For $\mu \ge 2$, $\mathcal{N}_\mu$ has a basis $\{f, y - a \text{ rem } f, \ldots, y^{\mu-1} - a^{\mu-1} \text{ rem } f\}$, which is not minimal. Its matrix is triangular, with determinant f. Thus, all bases of $\mathcal{N}_\mu$ have determinant λf for some $\lambda \in \mathbb{K} \setminus \{0\}$. It turns out that a minimal basis of $\mathcal{N}_\mu$ can be computed efficiently, along with reductions with respect to this basis.

Proposition 2.3 (Minimal basis & reduction). *Let $n \ge \mu > 0$, let $f \in \mathbb{K}[x]$ have degree n, and let $a \in \mathbb{K}[x]_{<n}$. The minimal basis in Popov form $R \in \mathbb{K}[x]^{\mu\times\mu}$ of $\mathcal{N}_\mu$ from Eq. (2) can be computed using $\tilde{O}(\mu^{\omega-1}n)$ operations in $\mathbb{K}$ and satisfies $\lceil n/\mu\rceil \le \deg(R) \le n$. Given ℓ polynomials $u_0, \ldots, u_{\ell-1}$ in $\mathbb{K}[x]_{<n}$, using $\tilde{O}(\lceil \ell/\mu\rceil \mu^{\omega-1}n)$ operations in $\mathbb{K}$ one can compute polynomials $U_0, \ldots, U_{\ell-1}$ in $\mathbb{K}[x,y]_{<(\delta,\mu)}$ such that $U_j(x,a) = u_j \bmod f$, where $\delta = \deg(R)$.*

Proof. Let $\bar{a}_i = a^i \text{ rem } f$, $0 \le i < \mu$. The Popov basis R (with shift $(0, \ldots, 0)$) for the module $\mathcal{N}_\mu$ of solutions of the linear system

$$\bar{a}_0 p_0 + \cdots + \bar{a}_{\mu-1} p_{\mu-1} = 0 \bmod f \quad (3)$$

can be computed in $\tilde{O}(\mu^{\omega-1}n)$ operations in $\mathbb{K}$ [28, Thm. 1.4]. The degree $\delta = \deg(R)$ satisfies $\lceil n/\mu\rceil \le \delta \le n$, since the sum of the column degrees of R is exactly $\deg(\det(R)) = \deg(f) = n$.

In fact, both R and $U_0, \ldots, U_{\ell-1}$ are computed in $\tilde{O}((\mu+\ell)^{\omega-1}n)$ operations in $\mathbb{K}$, using the same algorithm with the equation

$$\bar{a}_0 p_0 + \cdots + \bar{a}_{\mu-1} p_{\mu-1} - u_0 q_0 - \cdots - u_{\ell-1} q_{\ell-1} = 0 \bmod f \quad (4)$$

and the shift $\boldsymbol{s} = (0, \ldots, 0, n, \ldots, n) \in \mathbb{Z}^{\mu+\ell}$, where 0 appears μ times and n appears ℓ times, the unknowns being the p_i's and q_j's. Indeed, writing the corresponding basis as

$$P = \begin{bmatrix} P_0 & P_1 \\ Q_0 & Q_1 \end{bmatrix} \in \mathbb{K}[x]^{(\mu+\ell)\times(\mu+\ell)},$$

where P_0 is $\mu \times \mu$ and Q_1 is $\ell \times \ell$, we now show that $P_0 = R$, $Q_0 = 0$, $Q_1 = I_\ell$ and, most importantly, $P_1 = [U_{ij}]$, with $U_j = \sum_{0\le i<\mu} U_{ij} y^i$.

Note first that, by definition of $\boldsymbol{s}$-Popov forms, the principal submatrices P_0 and Q_1 are in $(0, \ldots, 0)$- and $(n, \ldots, n)$-Popov form, respectively. Also, Eq. (C_2) implies $\deg(P) \le \deg(\det(P))$, while this is at most $\deg(f) = n$, by the same reasoning as for $\mathcal{N}_\mu$ above. Thus, the choice of shift $\boldsymbol{s}$ and the definition of $\boldsymbol{s}$-Popov form ensure that $\deg(Q_0) + n < \deg(P_0) \le n$, hence $Q_0 = 0$. This implies that P_0 is the $(0, \ldots, 0)$-Popov basis of the solutions of Eq. (3), hence $P_0 = R$ by uniqueness.

Next, since both $\det(P)$ and $\det(R)$ are equal to f up to multiplication by a nonzero constant, $\det(Q_1)$ is itself a constant, and thus $Q_1 = I_\ell$, since the identity matrix is the unique unimodular matrix in $(n, \ldots, n)$-Popov form.

As a result, writing $P_1 = [\bar{U}_{ij}]$ for the entries of P_1, the fact that the $(\mu + j)$-th column of P is a solution of Eq. (4) implies that

$$\sum_{i<\mu} \bar{a}_i \bar{U}_{ij} = u_j \bmod f, \text{ that is, } \sum_{i<\mu} \bar{U}_{ij} a^i = u_j \bmod f.$$

Finally, $\deg(\bar{U}_{ij}) < \deg(R_{ii})$ follows from the fact that P is in $\boldsymbol{s}$-Popov form. By uniqueness of P, there is only one set of polynomials $\bar{U}_{ij}$ satisfying these degree bounds and Eq. (4), hence $\bar{U}_{ij} = U_{ij}$.

The claimed cost bound comes from applying the above method on $\lceil \ell/\mu\rceil$ lists, each containing $\le \mu$ of the polynomials u_i. □

The composition algorithm, and its components in Sections 3 and 4, use the following consequence of Proposition 2.3.

Corollary 2.4. *Let $n \geq \mu > 0$, let $f \in \mathbb{K}[x]$ have degree n, let $a \in \mathbb{K}[x]_{<n}$, and let δ be the degree of a minimal basis of $\mathcal{N}_\mu$. Using $\tilde{O}(\mu^{\omega-1}n)$ operations in $\mathbb{K}$, one can compute polynomials A_j, B_j in $\mathbb{K}[x,y]_{<(\delta,\mu)}$ for $0 \leq j < \mu$, such that $A_j(x,a) = a^{j\mu} \bmod f$ and $B_j(x,a) = a^{j\mu^2} \bmod f$ for all j.*

Proof. By Proposition 2.3, the sought $\ell = 2\mu$ polynomials are found in $\tilde{O}(\mu^{\omega-1}n)$ from $a_j = a^{j\mu} \operatorname{rem} f$ and $a_{\mu+j} = a^{j\mu^2} \operatorname{rem} f$ for $j = 0, \ldots, \mu-1$, themselves computed iteratively in $\tilde{O}(\mu n)$ ops. □

The above results come from deterministic algorithms which support any a and f. In modular composition, we use the following bound that holds generically. The proof is in Appendix A.

Lemma 2.5. *Let $n \geq \mu > 0$ and $f \in \mathbb{K}[x]$ have degree n. For a generic $a \in \mathbb{K}[x]_{<n}$, any minimal basis of $\mathcal{N}_\mu$ has degree $\lceil n/\mu \rceil$.*

The genericity is characterized by a nonzero polynomial $\Phi_{f,\mu} \in \mathbb{K}[\bar{a}_0, \ldots \bar{a}_{n-1}]$ (the $\bar{a}_i$'s are new indeterminates) whose zero set must be avoided by the coefficients of a; it is given in Appendix A. The detection of non-generic input is straightforward by inspecting the degrees in a minimal basis of $\mathcal{N}_\mu$, whose efficient computation is independent from genericity aspects, as seen above.

2.3 Computing modulo $\mathbb{K}[y]$-relations

The module $\mathcal{M}_\mu$ from Eq. (1) is more delicate to compute with. We rely on our previous work on this module [30]. Minimal bases of $\mathcal{M}_\mu$ have degree $\lceil n/\mu \rceil$ for generic a (see Appendix B). However, for their computation, we exploit a truncation technique whose success requires additional genericity conditions.

Proposition 2.6 (Minimal basis [30, Algo. 5.1 and § 7.3.2]). *Let $n \geq \mu > 0$ and $f \in \mathbb{K}[x]$ have degree $n > 0$ with $f(0) \neq 0$. There is an algorithm which takes as input any $a \in \mathbb{K}[x]_{<n}$ with $\gcd(a, f) = 1$ along with the truncated powers*

$$[x^i a^{-k-1} \operatorname{rem} f]_0^{\mu-1} \quad \text{for } 0 \leq i < \mu \text{ and } 0 \leq k < 2\delta,$$

and, using $\tilde{O}(\mu^{\omega-1}n)$ operations in $\mathbb{K}$, either returns a minimal basis of $\mathcal{M}_\mu$ of degree $\delta = \lceil n/\mu \rceil$ for generic a or detects that a does not satisfy the genericity condition.

Proof. Algorithm 5.1 of [30] first computes the truncated powers of the proposition. Next, its Steps 2 and 3 compute a weak Popov matrix $R \in \mathbb{K}[y]^{\mu\times\mu}$ of degree at most 2δ in $\tilde{O}(\mu^{\omega-1}n)$ operations in $\mathbb{K}$ [30, Prop. 5.6 and its proof]. For a generic a and when $f(0) \neq 0$, this matrix R is a basis of $\mathcal{M}_\mu$ with determinant of degree n [30, § 7.3.2, p. 46]. It follows that $\deg(R) = \lceil n/\mu \rceil$ [30, Prop. 5.6].

The genericity in $a(x)$ is precisely characterized by a polynomial $\Delta_{f,\mu}$ [30, Prop. 7.6] and by the requirement $\gcd(a, f) = 1$, which can be characterized as the non-cancellation of the resultant $\operatorname{Res}(a, f)$. The detection of non-generic input is performed algorithmically by the flag NoCert in Algo. 5.1 of [30]. □

For the reduction of a polynomial in $\mathbb{K}[x,y]_{<(\mu,\cdot)}$ with respect to this basis, we rely on a minimal nullspace basis computation [36].

Lemma 2.7 (Reduction). *Given a nonsingular R in $\mathbb{K}[y]^{\mu\times\mu}$ of degree $\leq \delta$ and whose columns are relations of $\mathcal{M}_\mu$, and given a polynomial $g(x,y)$ in $\mathbb{K}[x,y]_{<(\mu,\cdot)}$ of y-degree in $O(n)$, one can compute a polynomial $G(x,y)$ in $\mathbb{K}[x,y]_{<(\mu,\delta)}$ such that $G(x,a) = g(x,a) \bmod f$, using $\tilde{O}(\mu^\omega(\delta + n/\mu))$ operations in $\mathbb{K}$.*

For a detailed algorithm in the specific case of this lemma, and a proof of the complexity bound, we refer to [30, Sec. 4.2].

3 Bivariate Modular Composition

In this section, we are given f in $\mathbb{K}[x]$ of degree n, a in $\mathbb{K}[x]_{<n}$, and $G \in \mathbb{K}[x,y]_{<(m,d)}$ for some positive integers m, d, and we prove the following proposition on the cost of computing $G(x,a) \bmod f$. Theorem 1.1 will easily follow (in that theorem, we suppose that md is $\Theta(n)$, and we assume a is generic).

Proposition 3.1. *Consider integers n, m, d and μ, with $d^{1/3} \leq \mu \leq n$. Given f in $\mathbb{K}[x]$ of degree n, a in $\mathbb{K}[x]_{<n}$, $G \in \mathbb{K}[x,y]_{<(m,d)}$, and the polynomials $(A_j)_{0\leq j<\mu}$ and $(B_j)_{0\leq j<\mu}$ from Corollary 2.4, one can compute the composition $G(x,a) \bmod f$ in $\tilde{O}(\mu^\omega(\delta + m\mu))$ operations in $\mathbb{K}$, where δ is the degree of a minimal basis of $\mathcal{N}_\mu$.*

3.1 Algorithm and proof of Prop. 3.1 & Thm. 1.1.

The principle is to use an inverse Kronecker substitution. Since $d \leq \mu^3$, we can write any index $i < d$ as $i = i_0 + i_1\mu + i_2\mu^2$ with all $0 \leq i_j < \mu$, and then map the monomial y^i to $y_0^{i_0} y_1^{i_1} y_2^{i_2}$. By linearity this gives a map

$$\begin{aligned} \mathbb{K}[x,y]_{<(m,d)} &\to \mathbb{K}[x,y_0,y_1,y_2]_{<(m,\mu,\mu,\mu)} \\ G(x,y) &\mapsto \bar{G}(x,y_0,y_1,y_2). \end{aligned}$$

The evaluation at $y = a \bmod f$ amounts to computing modulo the collection of relations $y^j = a^j \bmod f$ for $0 \leq j < d$, in the $\mathbb{K}[x]$-module $\mathbb{K}[x,y]_{<(\cdot,d)}$. Through the above map, for $0 \leq j < \mu$, these relations become

$$y_0^j = a^j \bmod f;\ y_1^j = A_j(x,y_0) \bmod f;\ y_2^j = B_j(x,y_0) \bmod f$$

that we use to compute $S(x,y_0) = \bar{G}(x,y_0,a^\mu,a^{\mu^2}) \bmod f$.

Step 1: computing partial sums. Writing the coefficients of $\bar{G}$ as

$$\bar{G}(x,y_0,y_1,y_2) = \sum_{0\leq i_1,i_2<\mu} s_{i_1 i_2}(x,y_0) y_1^{i_1} y_2^{i_2},$$

one first computes the sums

$$s_{i_2} = \sum_{0\leq i_1<\mu} s_{i_1 i_2}(x,y_0) A_{i_1}(x,y_0) \in \mathbb{K}[x,y_0]_{<(m+\delta,2\mu)}, \quad 0 \leq i_2 < \mu.$$

The simultaneous computation of the sums s_{i_2} can be expressed as the product of the $\mu\times\mu$ matrix with entries $s_{i_1 i_2} \in \mathbb{K}[x,y_0]_{<(m,\mu)}$ by the vector in $\mathbb{K}[x,y_0]^\mu_{<(\delta,\mu)}$ with entries A_{i_1}. Using the $y_0^{\lceil \delta/\mu \rceil}$-adic expansion of these polynomials, this vector can be expanded into an $\mu \times \mu$ matrix with entries in $\mathbb{K}[x,y_0]_{<(\lceil \delta/\mu \rceil,\mu)}$. This matrix product can thus be computed at the announced cost $\tilde{O}(\mu^\omega(\delta + m\mu))$.

Step 2: computing $S(x,y_0)$. The computation of

$$S(x,y_0) = \sum_{0\leq i_2<\mu} s_{i_2}(x,y_0) B_{i_2}(x,y_0) \in \mathbb{K}[x,y_0]_{<(m+2\delta,3\mu)}$$

is a straightforward loop. Each summand can be computed naively in $\tilde{O}(\mu(\delta+m))$ operations in $\mathbb{K}$; the total is a negligible $\tilde{O}(\mu^2(\delta+m))$.

Step 3: Horner evaluation. Finally, given $S(x,y_0)$, Horner evaluation at $y_0 = a \bmod f$ yields $S(x,a) \bmod f$ at a cost of $\tilde{O}((n+m+\delta)\mu)$ operations. Since $\delta \geq n/\mu$, this is $\tilde{O}(\mu^2\delta + m\mu)$, which is negligible compared to the cost of the other operations.

Altogether, the cost is $\tilde{O}(\mu^\omega(\delta + m\mu))$, which establishes Proposition 3.1.

In the statement of Theorem 1.1, our assumption is that $md \in \Theta(n)$, which implies $d \leq cn$ for some constant c, hence $d \leq n^3$ for n large enough, so the previous proposition applies. In this context, we choose $\mu = \lceil d^{1/3} \rceil \in \Theta(d^{1/3})$. If furthermore the coefficients of a belong to the Zariski open set $\mathcal{V}_{f,d}$ defined with the polynomial $\Phi_{f,\lceil d^{1/3} \rceil}$ from Lemma 2.5, the degree δ is equal to $\lceil n/\mu \rceil \in \Theta(n/d^{1/3})$. In addition to the cost in Proposition 3.1, we first need to compute all polynomials A_j and B_j. By Corollary 2.4, this is $\tilde{O}(\mu^{\omega-1}n)$ operations in $\mathbb{K}$, that is, $\tilde{O}(d^{(\omega-1)/3}n)$. This is $\tilde{O}(md^{(\omega+2)/3})$, since $md \in \Theta(n)$. The runtime for bivariate composition itself is $\tilde{O}(d^{\omega/3}(n/d^{1/3} + md^{1/3}))$ operations in $\mathbb{K}$, and using again $md \in \Theta(n)$, this is also $\tilde{O}(d^{\omega/3}(md^{2/3} + md^{1/3})) = \tilde{O}(md^{(\omega+2)/3})$. This proves Theorem 1.1.

3.2 Multipoint evaluation: proof of Thm. 1.2

Proof. It follows the steps of Nüsken and Ziegler's algorithm. From $(x_1, y_1), \ldots, (x_n, y_n)$, we first compute $f = \prod(x - x_i)$ and the interpolating polynomial $a \in \mathbb{K}[x]_{<n}$ such that $a(x_i) = y_i$ for all i, both in $\tilde{O}(n)$ operations in $\mathbb{K}$ [13]. Given G in $\mathbb{K}[x, y]_{<(m,d)}$ and from Theorem 1.1, if the coefficients of a are in $\mathcal{V}_{f,d} \cap \mathbb{K}^n$, $h = G(x, a) \bmod f$ is obtained in $\tilde{O}(md^{(\omega+2)/3})$ operations in $\mathbb{K}$. The desired values are then deduced by univariate multipoint evaluation of h at $x_1, \ldots, x_n$, again in $\tilde{O}(n)$ operations in $\mathbb{K}$. The points $(y_1, \ldots, y_n) \in \mathbb{K}^n$ for which the coefficients of a are in $\mathcal{V}_{f,d} \cap \mathbb{K}^n$ are in the image $\mathcal{W}_{\xi,d}$ of $\mathcal{V}_{f,d}$ by the invertible Vandermonde matrix constructed from $\xi = (x_1, \ldots, x_n)$, which is thus also a non-empty Zariski open set.

Arbitrary sets of points. Theorem 1.2 and Nüsken and Ziegler's results hold for points with pairwise distinct x-coordinate. For arbitrary sets of points, a natural approach (going back to [31]) is to apply a random change of coordinates, since with high probability, this guarantees that all x-coordinates are distinct. However, we do not expect this to be sufficient to have small-degree relation matrices needed for the target complexity bound, and further work will be needed here as well to obtain a Las Vegas algorithm.

When $m = d = \sqrt{n}$, Nüsken and Ziegler's algorithm has complexity $\tilde{O}(n^{(\omega_2+2)/4})$, which is in $O(n^{1.313})$, while Theorem 1.2 gives $\tilde{O}(n^{(\omega+5)/6})$, which is in $O(n^{1.229})$. Without changing variables, the best result we are aware of for arbitrary inputs is in [18], with a runtime of $\tilde{O}(n^{2-2/(\omega+1)}) \subset O(n^{1.41})$ operations in $\mathbb{K}$. Note that several other results are known in an *amortized* model [15–18, 29], some of them also requiring generic inputs.

4 Truncated powers

Step (4.2) of the new composition algorithm computes the truncated powers $[x^i a^{-k} \text{ rem } f]_0^{m-1}$ for $0 \leq i < m, 0 \leq k \leq 2d$. Using the small-degree polynomials A_j and B_j in Step (3.3) lets us improve the previous result for this computation [30, Prop. 3.6].

Proposition 4.1. *Consider integers n, m, d and μ, with $d^{1/3} \leq \mu \leq n$. Given f in $\mathbb{K}[x]$ of degree n, a in $\mathbb{K}[x]_{<n}$ and the polynomials $(A_j)_{0\leq j<\mu}$ and $(B_j)_{0\leq j<\mu}$ from Corollary 2.4, one can compute the truncations $[ba^k \text{ rem } f]_0^{m-1}$ for $0 \leq k < d$ using $\tilde{O}(\mu^\omega(\delta + m\mu))$ operations in $\mathbb{K}$, where δ is the degree of a minimal basis of $\mathcal{N}_\mu$.*

In this section, we use the bracket notation $[\cdot]_0^{m-1}$ with bivariate polynomials: it stands for the first coefficients m with respect to x. Also, while the proposition computes truncations of terms $ba^k \text{ rem } f$, we will apply this with a replaced by its inverse modulo f.

4.1 Simultaneous truncated products

The first ingredient is the simultaneous computation of several truncated products with a specific structure.

Lemma 4.2. *Let $n, \mu, \hat{\mu}, \delta, m \in \mathbb{N}_{>0}$ with $\hat{\mu} \in O(\mu^2)$ and $\delta \leq n$. Given f in $\mathbb{K}[x]$ of degree n, h in $\mathbb{K}[x, y]_{<(\delta,\hat{\mu})}$, and $\eta_1, \ldots, \eta_\mu$ in $\mathbb{K}[x, y]_{<(\delta,\mu)}$, one can compute all truncated products*

$$R_j = [x^{n-\delta} h\eta_j \text{ rem } f]_0^{m-1} \in \mathbb{K}[x, y]_{<(m,\mu+\hat{\mu}-1)}, \quad 1 \leq j \leq \mu$$

using $\tilde{O}(\mu^\omega(\delta + m\mu))$ operations in $\mathbb{K}$.

Proof. Use segmentation with respect to y to write

$$h(x, y) = \sum_{0\leq i<\bar{\mu}} h_i(x, y) y^{\mu i},$$

with $h_0, \ldots, h_{\bar{\mu}-1} \in \mathbb{K}[x, y]_{<(\delta,\mu)}$ and $\bar{\mu} \in O(\mu)$. Set $H_i = x^{n-\delta} h_i$. It is enough to compute, for all $0 \leq i < \bar{\mu}$ and $1 \leq j \leq \mu$,

$$R_{ij} = [H_i \eta_j \text{ rem } f]_0^{m-1} \in \mathbb{K}[x, y]_{<(m,2\mu-1)}.$$

Indeed, all $R_j = \sum_i R_{ij} y^{\mu i}$ can then be deduced in linear time with respect to the total size of the R_{ij}'s, which is in $O(m\mu^3)$. Since all products R_{ij} have y-degree less than $2\mu - 1$, we compute them as polynomials in $\mathbb{B}[x]$, with $\mathbb{B} = \mathbb{K}[y]/\langle y^{2\mu-1} \rangle$.

For the computation, we use Algo. 3.6 from [30]. While the original presentation assumes that we work over a field, rather than a ring such as $\mathbb{B}$, it is readily checked that all operations are performed in the ring. A more serious issue is that a direct application of Lemma 3.5 in [30] does not give the runtime we expect, as it does not take into account that $H_i \text{ rem } x^{n-\delta} = 0$ and $\deg_x(\eta_j) < \delta$. Thus, we revisit the steps of that algorithm, and analyze how these two properties improve its runtime.

Steps 2 and 3 compute $\bar{H}_i = x^{n-1} H_i(1/x, y)$, which has x-degree less than δ, and $\bar{\eta}_j = x^{n-1}\eta_j(1/x, y)$, which vanishes modulo $x^{n-\delta}$. This does not cost any arithmetic operation.

Step 3 also computes the power series expansions $\gamma_j = \bar{\eta}_j / \bar{f}$ modulo x^{n-1}, where $\bar{f} = x^n f(1/x)$. Since the x-valuation of $\bar{\eta}_j$ is $\geq n-\delta$, each such expansion takes $\tilde{O}(\delta)$ operations in $\mathbb{B}$, for a total of $\tilde{O}(\mu^2\delta)$ operations in $\mathbb{K}$.

Step 4 defines matrices P_1 and P_2: the ith row of P_1 is obtained by segmenting the coefficients of $\bar{H}_i$ with respect to x, into slices of x-degree less than m; P_2 is obtained from P_1 by discarding its last column. In the original presentation, these matrices have up to $O(\lceil n/m \rceil)$ nonzero columns, but here the rightmost ones are zero because $\bar{H}_i$ has x-degree $< \delta$. Discarding these columns in P_1 and P_2 leaves us with matrices $\bar{P}_1, \bar{P}_2$ for both, with $\bar{\mu}$ rows and $O(\lceil \delta/m \rceil)$ columns, and entries in $\mathbb{B}[x]_{<m}$.

Step 4 also defines matrices Q_1 and Q_2, obtained by segmenting the polynomials γ_j, as we did for $\bar{H}_i$. For valuation reasons, many bottom rows of these matrices are zero and can be discarded, leaving us with matrices $\bar{Q}_1, \bar{Q}_2$ with $O(\lceil \delta/m \rceil)$ rows and μ columns, and the same type of entries as those of $\bar{P}_1, \bar{P}_2$.

Step 5 performs the matrix products $\bar{P}_1\bar{Q}_1$ and $\bar{P}_2\bar{Q}_2$, as well as $\mu\bar{\mu}$ products of polynomials in $\mathbb{B}[x]_{<m}$; the latter take a total of $\tilde{O}(m\mu^3)$ operations in $\mathbb{K}$. For computing a product $\bar{P}_i\bar{Q}_i$, considering the two cases $m\mu \in O(\delta)$ and $\delta \in O(m\mu)$ leads to the complexity bounds $\tilde{O}(\mu^\omega\delta)$ and $\tilde{O}(m\mu^{\omega+1})$, respectively.

Step 6 is unchanged, and amounts to $O(\mu^2)$ multiplications in $\mathbb{B}[x]/\langle x^m\rangle$, that is, another $\tilde{O}(m\mu^3)$ operations in $\mathbb{K}$.

The four terms arising in the cost are thus $m\mu^3$, $\mu^2\delta$, and either $\mu^\omega\delta$ or $m\mu^{\omega+1}$. They are all in $O(\mu^\omega(\delta + m\mu))$. □

4.2 High part of a Euclidean remainder

The following lemma is extracted from the proof of [30, Lem. 3.5]. $\mathbb{B}$ is a commutative ring.

Lemma 4.3. *Let f be monic of degree n in $\mathbb{B}[x]$, let P be in $\mathbb{B}[x]_{<n}$, and let Q in $\mathbb{B}[x]_{<d}$ for some $d \in \{1,\dots,n\}$. Given Q and the high part $[P]_{n-t-d+1}^{t+d-2}$ for some $t \in \{1,\dots,n-d+1\}$, one can compute the high part $[PQ \bmod f]_{n-t}^{t-1}$ in $\tilde{O}(d+t)$ operations in $\mathbb{B}$.*

Proof. Write the Euclidean division $PQ = hf + r$, thus with $r = PQ \bmod f$, and consider the reversed polynomials given by

$$\bar{P} = x^{n-1}P(1/x), \quad \bar{Q} = x^{d-1}Q(1/x),$$
$$\bar{f} = x^n f(1/x), \quad \bar{r} = x^{n-1}r(1/x), \quad \bar{h} = x^{d-2}h(1/x),$$

all being in $\mathbb{B}[x]$. By construction, we have $\bar{P}\bar{Q} = \bar{h}\bar{f} + x^{d-1}\bar{r}$. Our input gives us access to $\bar{P} \bmod x^{d-1}$, so that $\bar{h}$ can be obtained by power series division $(\bar{P}\bar{Q}/\bar{f}) \bmod x^{d-1}$ in $\tilde{O}(d)$ operations. Once h is known, the high part of r is obtained in $\tilde{O}(t+d)$ operations by

$$[r]_{n-t}^{t-1} = \left[[P]_{n-t-d+1}^{t+d-2}Q - h[f]_{n-t-d+1}^{t+d-2}\right]_{d-1}^{t-1}. \qquad \square$$

4.3 New TruncatedPowers algorithm

We turn to the proof of Proposition 4.1. For $k \geq 0$, we write $c_k = ba^k \bmod f \in \mathbb{A}$, where $\mathbb{A} = \mathbb{K}[x]/\langle f\rangle$. These are the coefficients of the generating function

$$D(y) = \frac{b}{1-ay} \in \mathbb{A}[[y]].$$

Our goal can then be restated as computing $[c_k]_0^{m-1}$ for $k < d$, that is, $[D \bmod y^d]_0^{m-1}$; this has size $\Theta(md)$. Computing the first d coefficients of D before truncating in x would yield an intermediate object of size $\Theta(nd)$, which is too large for our target complexity.

We fix $\mu = \lceil d^{1/3}\rceil$. In particular $d \leq \mu^3$, and we focus on computing $[c_k]_0^{m-1}$ for all $0 \leq k < \mu^3$. The key ingredient lies in the following lemma, which shows how to obtain new coefficients c_k, of high indices, from previously known ones. It makes use of the polynomials $(A_j)_{1\leq j<\mu}$ and $(B_j)_{1\leq j<\mu}$ from Corollary 2.4, and of the y-reversed counterparts of these polynomials, also in $\mathbb{K}[x,y]_{<(\delta,\mu)}$:

$$\alpha_j = y^{\mu-1}A_j(x,1/y) \quad \text{and} \quad \beta_j = y^{\mu-1}B_j(x,1/y)$$

for $1 \leq j < \mu$. We often see α_j and β_j as being in $\mathbb{A}[y]_{<\mu}$, as for instance in the following lemma.

Lemma 4.4. *For $1 \leq j < \mu$, and for any $k \geq \mu - 1$, the coefficient of y^k in $D\alpha_j$ is $c_{j\mu+k-(\mu-1)}$, while that of $D\beta_j$ is $c_{j\mu^2+k-(\mu-1)}$.*

Proof. We prove the claim for α_j; the proof is similar for β_j. Let $a_{j,k}$ be the coefficient of y^k in A_j, so that $\alpha_j = a_{j,\mu-1} + \cdots + a_{j,0}y^{\mu-1}$. Then, a direct expansion shows that for any $k \geq \mu-1$, the coefficient of y^k in the product $D\alpha_j$, working modulo f, is

$$ba^{k-(\mu-1)}a_{j,0} + \cdots + ba^{k-1}a_{j,\mu-2} + ba^k a_{j,\mu-1}$$
$$= ba^{k-(\mu-1)}A_j(x,a) = ba^{j\mu+k-(\mu-1)} = c_{j\mu+k-(\mu-1)}. \qquad \square$$

Thus, if we know $D \bmod y^N$, multiplying it by α_j gives us all coefficients $c_{j\mu}, c_{j\mu+1}, \dots, c_{j\mu+N-1-(\mu-1)}$. The same holds for β_j, but with a shift of $j\mu^2$ instead of $j\mu$. Consider the truncations

$$D_0 = D \bmod y^{\mu+2(\mu-1)} = D \bmod y^{3\mu-2},$$
$$D_1 = D \bmod y^{\mu^2+\mu-1}, \quad D_2 = D \bmod y^{\mu^3}.$$

We see D_0 and D_1 as being in $\mathbb{B}_0[x]$ and $\mathbb{B}_1[x]$ respectively, where

$$\mathbb{B}_0 = \mathbb{K}[y]/\langle y^{3\mu-2}\rangle \quad \text{and} \quad \mathbb{B}_1 = \mathbb{K}[y]/\langle y^{\mu^2+\mu-1}\rangle.$$

Recall that our goal is to compute $[D_2]_0^{m-1}$.

According to the above remarks, multiplying D_0 by $\alpha_1,\dots,\alpha_{\mu-1}$ in $\mathbb{B}_0[x]$ gives us all coefficients of D_1 (some of them are computed twice), and multiplying D_1 by $\beta_1,\dots,\beta_{\mu-1}$ in $\mathbb{B}_1[x]$ gives us all coefficients of D_2 (with no repeated calculations this time).

Still, computing D_1 and D_2 in this manner is too costly for our target. Instead, our algorithm below exploits the fact that we only seek low degree coefficients of D_2. It starts from D_0 and computes the high part $[D_1]_{n-\delta}^{\delta-1}$, then deduces the low part $[D_1]_0^{m-1}$, and eventually $[D_2]_0^{m-1}$.

Step 1: all of D_0. Each product $c_k = ba^k \bmod f$ for $0 \leq k < 3\mu-2$ is obtained in $\tilde{O}(n)$ operations in $\mathbb{K}$, hence a total in $\tilde{O}(\mu n)$ operations in $\mathbb{K}$ for D_0.

Step 2: high part $[D_1]_{n-\delta}^{\delta-1}$. For this computation, it is enough to compute all high parts $[D_0\alpha_j]_{n-\delta}^{\delta-1}$ in $\mathbb{B}_0[x]$. For each of these products, we use Lemma 4.3 with $t = \delta$ and input $[D_0]_{n-2\delta+1}^{2\delta-2}$. Each product uses $\tilde{O}(\delta)$ operations in $\mathbb{B}_0$, that is, in y-degree $O(\mu)$, for a total of $\tilde{O}(\mu^2\delta)$ operations in $\mathbb{K}$.

Step 3: low parts of D_1 and D_2. This step computes $[D_1]_0^{m-1}$ and $[D_2]_0^{m-1}$, which boils down to computing the low parts of all products $D_0\alpha_j$ for the former, and $D_1\beta_j$ for the latter. Splitting D_0 gives

$$[D_0\alpha_j]_0^{m-1} = [D_0]_0^{m-1}[\alpha_j]_0^{m-1} \bmod x^m + \left[x^{n-\delta}[D_0]_{n-\delta}^{\delta-1}\alpha_j \bmod f\right]_0^{m-1} \tag{5}$$

in $\mathbb{B}_0[x]$, and similarly

$$[D_1\beta_j]_0^{m-1} = [D_1]_0^{m-1}[\beta_j]_0^{m-1} \bmod x^m + \left[x^{n-\delta}[D_1]_{n-\delta}^{\delta-1}\beta_j \bmod f\right]_0^{m-1}, \tag{6}$$

in $\mathbb{B}_1[x]$. For the products in Eq. (5), we need the low and high parts of D_0, which are known. For the ones in Eq. (6), we need the high part of D_1, which we obtained in Step 2, and its low part, which we derive from using Eq. (5) with $j = 1,\dots,\mu-1$.

The products in Eq. (5) have lower cost than those in Eq. (6) since the degree of D_0 is smaller than that of D_1. Thus we focus on Eq. (6), for $1 \leq j < \mu$. The first term can be computed directly using μ bivariate multiplications modulo $(x^m, y^{\mu^2+\mu-1})$, for a total of $\tilde{O}(m\mu^3)$ operations. For the second term, we use Lemma 4.2 with $\hat{\mu} = \mu^2 + \mu - 1$, and with a runtime of $\tilde{O}(\mu^\omega(\delta + m\mu))$.

Altogether, the cost is $\tilde{O}(\mu n + \mu^2\delta + m\mu^3 + \mu^\omega(\delta + m\mu))$. Since $\delta \geq n/\mu$ (see Proposition 2.3), this is bounded by $\tilde{O}(\mu^\omega(\delta + m\mu))$.

5 Composition algorithm. Proof of Theorem 1.3

To prove our main result, we start with f of degree n in $\mathbb{K}[x]$, and we first establish the claim under the assumption $f(0) \neq 0$. Our first condition on a is that it satisfies $\gcd(a, f) = 1$ (this is a Zariski-generic property in the coefficients of a).

We choose $m = \lceil n^{1/4} \rceil$ and $d = \lceil n/m \rceil$, which implies both inequalities $d^{1/3} \leq m \leq n$. We suppose that a satisfies the genericity conditions of Lemma 2.5 and Proposition 2.6 for $\mu = m$, so in particular the minimal bases of $\mathcal{N}_m$ and $\mathcal{M}_m$ have degree d. Then, consider the following detailed presentation of the algorithm sketched in Section 1:

(1) Compute $\tilde{a} = a^{-1} \bmod f$, which exists since $\gcd(a, f) = 1$;
(2) Compute a minimal basis of $\mathcal{N}_m$, with degree d;
(3) Reduce a^{jm^i} modulo f, for $i = 0, 1, 2$ and $0 \leq j < m$;
(4) Simultaneously reduce the polynomials of (3) by the basis of (2) to obtain A_j, B_j in $\mathbb{K}[x, y]_{<(d,m)}$, for $0 \leq j < m$, such that

$$A_j(x, a) = a^{jm} \bmod f \quad \text{and} \quad B_j(x, a) = a^{jm^2} \bmod f;$$

(5) Use the polynomials A_j and B_j to compute the truncations $[x^i a^{-k-1} \text{ rem } f]_0^{m-1}$ for $0 \leq i < m$ and $0 \leq k < 2d$;
(6) From these, compute a basis of $\mathcal{M}_m$ of degree at most d;
(7) Reduce g by this basis to obtain $G \in \mathbb{K}[x, y]_{<(m,d)}$ such that $G(x, a) = g(a) \bmod f$;
(8) Use G and the A_j's and B_j's to compute $g(a) \bmod f$.

The first step takes $\tilde{O}(n)$ operations in $\mathbb{K}$. Computing the basis of $\mathcal{N}_m$ at Step 2 and deducing the polynomials A_j and B_j at Steps 3 and 4 altogether take $\tilde{O}(m^{\omega-1}n)$ operations, by Proposition 2.3 and Corollary 2.4. With our choice of m, this is $\tilde{O}(n^{(\omega+3)/4})$.

To perform Step 5, we apply the algorithm of Proposition 4.1 twice, with parameters $n, m' = 2m-2$ (instead of m), d and $\mu = m$ (so that $\delta = d$), and input polynomials f, $\tilde{a}$, and $b = x^{m-1} \bmod f$, resp. $b = x^{m-1}\tilde{a}^d \bmod f$. As in the proof of Theorem 1.1, the runtime is $\tilde{O}(md^{(\omega+2)/3})$, which is also $\tilde{O}(n^{(\omega+3)/4})$ for our choice of m.

Concatenating the results gives $[x^{m-1}a^{-k} \text{ rem } f]_0^{2m-2}$ for $0 \leq k < 2d-1$. From this, Algo. 3.8 and Prop.3.7 in [30] show that we can deduce the truncations $[x^i a^{-k-1} \text{ rem } f]_0^{m-1}$, for $0 \leq i < m$ and $0 \leq k < 2d$, in time $\tilde{O}(m^2 d) = \tilde{O}(n^{5/4})$.

For Step 6, thanks to $f(0) \neq 0$, Proposition 2.6 allows us to compute a minimal basis of $\mathcal{M}_m$ in time $\tilde{O}(m^{\omega-1}n) = \tilde{O}(n^{(\omega+3)/4})$. Lemma 2.7 then allows us to perform Step 7 in the same asymptotic runtime. Finally, Step 8 is handled by Proposition 3.1, using $\tilde{O}(md^{(\omega+2)/3})$ operations, which is again in $\tilde{O}(n^{(\omega+3)/4})$.

The Zariski-open set O_f is defined from the polynomial $\Phi_{f,\lceil n^{1/4} \rceil}$ from Lemma 2.5, the polynomial $\Delta_{f,\lceil n^{1/4} \rceil}$ from [30, Prop. 7.6] that we mentioned in the proof of Proposition 2.6, and by the requirement $\gcd(a, f) = 1$. This establishes the theorem for f such that $f(0) \neq 0$.

Now, let f be arbitrary of degree n, and write it as $f = x^\alpha f^*$, with f^* of degree $n^* = n - \alpha$ and $f^*(0) \neq 0$; let also $\bar{a}_0, \ldots, \bar{a}_{n-1}$ be indeterminates (that represent the coefficients of a).

Let Δ be a polynomial in $\mathbb{K}[\bar{b}_0, \ldots, \bar{b}_{n^*-1}]$ that defines the complement of O_{f^*}, and set $\Delta^*(\bar{a}_0, \ldots, \bar{a}_{n-1}) = \Delta(r_0, \ldots, r_{n^*-1})$, where $r_0, \ldots, r_{n^*-1}$ are the coefficients of the remainder of $\bar{a}_0 + \cdots + \bar{a}_{n-1}x^{n-1}$ by f^*. In view of the equality $\Delta^*(\bar{a}_0, \ldots, \bar{a}_{n^*-1}, 0, \ldots, 0) = \Delta(\bar{a}_0, \ldots, \bar{a}_{n^*-1})$, we deduce that Δ^* is nonzero.

Suppose then that the coefficients of a do not cancel Δ^*. To compute $h = g(a) \bmod f$, it is enough to compute $\hat{a} = a \bmod x^\alpha$ and $a^* = a \bmod f^*$, then $\hat{h} = g(\hat{a}) \bmod x^\alpha$ and $h^* = g(a^*) \bmod f^*$, and finally recover h by Chinese Remaindering. Computing $\hat{a}$, a^* and Chinese Remaindering, take quasi-linear time in n, so it remains to discuss the cost of computing $\hat{h}$ and h^*. In both cases, we use segmentation on g.

Computation of $\hat{h}$. Write $g = \sum_{i \leq \lceil n/\alpha \rceil} g_i(y) y^{i\alpha}$, with all g_i of degree less than α. Using the Kinoshita-Li algorithm [24], we can compute all $g_i(\hat{a}) \bmod x^\alpha$ in $\tilde{O}(\alpha(n/\alpha)) = \tilde{O}(n)$ operations. Then, after the additional computation of $\hat{a}^\alpha \bmod x^\alpha$, we can use Horner's scheme to recover $\hat{h} = g(\hat{a}) \bmod x^\alpha$ in softly linear time in n.

Computation of h^*. The same procedure applies. Since the coefficients of a do not cancel Δ^*, the coefficients of a^* do not cancel Δ, so a single composition by a^* modulo f^* takes $\tilde{O}(n^{*(\omega+3)/4})$ operations. The rest of the analysis is similar, and shows that we can obtain h^* in time $\tilde{O}(n^{(\omega+3)/4})$.

6 Complexity equivalences

The problem of modular composition is closely related to those of power projections and characteristic polynomials in quotient algebras. See, in particular, [33, 34] and [22, Sec. 6]. This results in equivalences between some of the subproblems addressed here that could be used to develop alternative algorithms. First, Propositions 3.1 and 4.1 show that two subproblems have the same complexity bound. In fact, there is a general relation between them: Section 6.1 shows that, in terms of straight-line programs, the problems of truncated powers and bivariate composition are essentially equivalent, when $f(0) \neq 0$. This is shown on the right side of Fig. 1. Next, Section 6.2 explains that, in generic cases, bivariate composition could be replaced by computations of specific characteristic polynomials obtained from minimal bases. This provides another alternative path for composition, shown on the left side of Fig. 1.

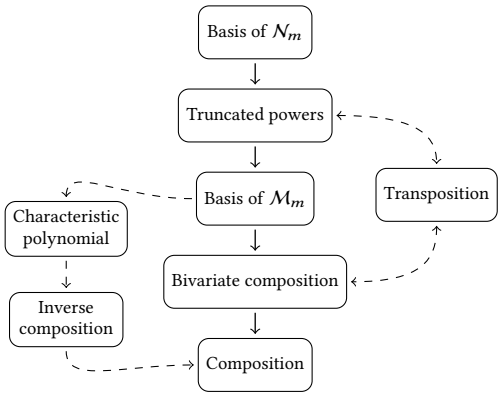

Figure 1: Subproblems involved in the new modular composition algorithm. The dotted lines indicate possible alternative paths.

6.1 Truncated powers vs bivariate composition

The transposition principle for linear straight-line programs [11, Thm 13.20] states that the product of a matrix by a vector has essentially the same cost as the product of the transpose of this matrix by a vector. This leads to complexity equivalences between linear problems that are solved by straight-line programs, like our algorithms in Sections 3 and 4. The equivalence in this section is proved by looking more closely at the linear maps involved in the computations.

For two polynomials $t, u \in \mathbb{K}[x]$ with $\deg(u) = l$, we denote by $M_{t,u} \in \mathbb{K}^{l\times l}$ the matrix of multiplication by t modulo u in the basis $(1, x, \ldots, x^{l-1})$. The problems of bivariate modular composition and truncated powers (with the parameters m and d) may be addressed using the following block Krylov matrices [30, Sec. 3.4]:

$$K_{m,d} = \begin{bmatrix} X & M_{a,f}X & \cdots & M_{a,f}^{d-1}X \end{bmatrix} \in \mathbb{K}^{n\times(md)}$$

and

$$L_{m,d} = \begin{pmatrix} X^{\mathsf{T}} \\ \vdots \\ X^{\mathsf{T}}M_{a,f}^{d-1} \end{pmatrix} \in \mathbb{K}^{(md)\times n},$$

where $X = [\mathrm{I}_m\ 0]^{\mathsf{T}} \in \mathbb{K}^{n\times m}$. For a given a, the matrix $K_{m,d}$ represents the linear map of bivariate composition for $G \in \mathbb{K}[x,y]_{<(m,d)}$ as in Proposition 3.1, while $L_{m,d}$ represents that of truncated powers for $b \in \mathbb{K}[x]_{<n}$, as in Proposition 4.1. Note that $L_{m,d} \neq K_{m,d}^{\mathsf{T}}$. Still, we now argue that truncated powers and bivariate composition are essentially equivalent problems when $f(0) \neq 0$.

The following lemma makes use of two auxiliary matrices. The *reversal matrix* $J_m \in \mathbb{K}^{m\times m}$ is the matrix of the permutation $(m, \ldots, 1)$. The triangular Hankel matrix $S \in \mathbb{K}^{n\times n}$ is defined by $S_{i,j} = f_{i+j-1}$ if $(n - i + 1) \geq j$ and $S_{i,j} = 0$ otherwise. Note that S is invertible, since $\deg(f) = n$. It is a symmetrizer for a [26, p. 455], that is, it satisfies [7, Lem. 2.5]

$$SM_{a,f}^{\mathsf{T}} = M_{a,f}S. \tag{7}$$

Lemma 6.1. *For $m \in \{1, \ldots, n\}$ and $d \geq 1$ we have*

$$L_{m,d} \cdot P_n = \mathrm{diag}(Q_m, \ldots, Q_m) \cdot K_{m,d}^{\mathsf{T}} \tag{8}$$

with $Q_m = -M_{f,x^m}J_m$ and $P_n = M_{x^m,f}S \in \mathbb{K}^{n\times n}$; Q_m and P_n depend only on f and are invertible if $f(0) \neq 0$.

Proof. The matrix Q_m arises from the product [26, Eq. (5), p. 456]

$$M_{x^m,f}S = \begin{pmatrix} Q_m & 0 \\ 0 & T_{n-m} \end{pmatrix} \in \mathbb{K}^{n\times n},$$

with $T_{n-m} = J_{n-m}M_{\hat{f},x^{n-m}}$ with $\hat{f} = x^n f(1/x)$. Hence

$$Q_m\, X^{\mathsf{T}}\, S^{-1} = X^{\mathsf{T}}\, M_{x^m,f}.$$

Using Eq. (7) with a^k, for any integer $k \geq 0$, gives

$$X^{\mathsf{T}}M_{a^k,f}^{\mathsf{T}} = X^{\mathsf{T}}S^{-1}M_{a^k,f}S. \tag{9}$$

Combining both equations gives $Q_m\, X^{\mathsf{T}}M_{a^k,f}^{\mathsf{T}} = X^{\mathsf{T}}\, M_{x^m,f}M_{a^k,f}S$, for an arbitrary $k \geq 1$, which is also

$$Q_m\, (X^{\mathsf{T}}M_{a^k,f}^{\mathsf{T}}) = (X^{\mathsf{T}}\, M_{a^k,f})\, P_n \in \mathbb{K}^{m\times n},$$

where $P_n = M_{x^m,f}S \in \mathbb{K}^{n\times n}$. This concludes the proof of Eq. (8).

Up to sign, the determinants of Q_m and T_{n-m} are f_0^m and f_n^{n-m}. Thus both Q_m and P_n are invertible if $f(0) \neq 0$. □

Applying the transposition principle to the algorithm in Section 3 gives a multiplication by $K_{m,d}^{\mathsf{T}}$ in essentially the same complexity as bivariate modular composition. Multiplying $\mathrm{diag}(Q_m, \ldots, Q_m)$, P_n, and their inverses by vectors has quasilinear complexity by reduction to polynomial multiplication [6, Ch. 2], hence Eq. (8) gives an algorithm of the same complexity for truncated powers. The reverse direction works in the same way, by transposing the algorithm in Section 4 in order to multiply $L_{m,d}^{\mathsf{T}}$ by vectors.

There exist techniques to transpose algorithms in a systematic way [8]. We have not checked whether they would produce exactly the same algorithms as those presented in Sections 3 and 4.

6.2 Characteristic polynomial vs composition

For any f and a generic a, the determinant of a minimal basis of $\mathcal{M}_\mu$, made monic, is the characteristic polynomial χ_a of a modulo f [30, Prop. 10.1]. Proposition 2.6 can thus be extended to the computation of χ_a, using also $\tilde{O}(\mu^{\omega-1}n)$ operations [25]. In Appendix C, we show that the characteristic polynomial and the modular composition problems are essentially equivalent in the model of computation by straight-line programs. In particular, a technique commonly used for parameterizing algebraic varieties enables us to reduce the modular composition problem and its inverse (see [33, Thm. 3.5; 30, Sec. 6.2]) to characteristic polynomial computations.

References

[1] Josh Alman, Rand Duan, Virginia Vassilevska Williamns, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. 2025. More asymmetry yields faster matrix multiplication. In *Proc. SODA*. SIAM, 2005–2039. doi:10.1137/1.9781611978322.63

[2] Maria-Emilia Alonso, Eberhard Becker, Marie-Françoise Roy, and Thorsten Wörmann. 1996. Zeros, multiplicities, and idempotents for zero-dimensional systems. In *Algorithms in algebraic geometry and applications*. Springer, 1–15. doi:10.1007/978-3-0348-9104-2_1

[3] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. 2006. *Algorithms in Real Algebraic Geometry*. Springer, 2nd edition. doi:10.1007/3-540-33099-2

[4] Bernhard Beckermann, George Labahn, and Gilles Villard. 1999. Shifted normal forms of polynomial matrices. In *Proc. ISSAC*. ACM Press, 189–196. doi:10.1145/309831.309929

[5] Vishwas Bhargava, Sumanta Ghosh, Mrinal Kumar, and Chandra Kanta Mohapatra. 2023. Fast, Algebraic Multivariate Multipoint Evaluation in Small Characteristic and Applications. *J. ACM* 70, 6 (2023). doi:10.1145/3625226

[6] Dario Bini and Victor Y. Pan. 1994. *Polynomial and matrix computations*. Birkhäuser. doi:10.1007/978-1-4612-0265-3

[7] Alin Bostan, Claude-Pierre Jeannerod, Christophe Mouilleron, and Éric Schost. 2017. On matrices with displacement structure: generalized operators and faster algorithms. *SIAM J. on Matrix Analysis and Applications* 38, 3 (2017), 733–775. doi:10.1137/16M1062855

[8] Alin Bostan, Grégoire Lecerf, and Éric Schost. 2003. Tellegen's principle into practice. In *Proc. ISSAC*. ACM Press, 37–44. doi:10.1145/860854.860870

[9] Richard P. Brent and Hsiang-Tsung Kung. 1978. Fast algorithms for manipulating formal power series. *J. ACM* 25, 4 (1978), 581–595. doi:10.1145/322092.322099

[10] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. 1997. *Algebraic complexity theory*. Springer-Verlag, Berlin.

[11] Peter Bürgisser, Michael Clausen, and Mohammad Amin Shokrollahi. 1997. *Algebraic complexity theory*. Grundlehren der mathematischen Wissenschaften, Vol. 315. Springer. doi:10.1007/978-3-662-03338-8

[12] Joachim von zur Gathen and Jürgen Gerhard. 1997. Fast algorithms for Taylor shifts and certain difference equation. In *Proc. ISSAC*. ACM Press, 40–47. doi:10.1145/258726.258745

[13] Joachim von zur Gathen and Jürgen Gerhard. 1999. *Modern computer algebra*. Third edition 2013. Cambridge University Press. doi:10.1017/CBO9781139856065

[14] Joris van der Hoeven and Grégoire Lecerf. 2020. Fast multivariate multi-point evaluation revisited. *J. Complexity* 56 (2020). doi:10.1016/j.jco.2019.04.001

[15] Joris van der Hoeven and Grégoire Lecerf. 2021. Amortized bivariate multi-point evaluation. In *Proc. ISSAC*. ACM Press, 179–185. doi:10.1145/3452143.3465531

[16] Joris van der Hoeven and Grégoire Lecerf. 2021. Fast amortized multi-point evaluation. *J. Complexity* (2021), 101574. doi:10.1016/j.jco.2021.101574

[17] Joris van der Hoeven and Grégoire Lecerf. 2023. Amortized multi-point evaluation of multivariate polynomials. *J. Complexity* 74 (2023). doi:10.1016/j.jco.2022.101693

[18] Joris van der Hoeven and Grégoire Lecerf. 2025. Faster multi-point evaluation over any field. *Applicable Algebra in Engineering, Communication and Computing* (2025). doi:10.1007/s00200-025-00704-7

[19] Xiohan Huang and Victor Y. Pan. 1998. Fast rectangular matrix multiplication and applications. *J. Complexity* 14 (1998), 257–299. doi:10.1006/jcom.1998.0476

[20] Nathan Jacobson. 2009. *Basic Algebra I.* Dover Publications Inc. https://store.doverpublications.com/0486471896.html Second Edition W.H. Freeman 1985.

[21] Thomas Kailath. 1980. *Linear Systems.* Prentice-Hall.

[22] Erich Kaltofen. 2000. Challenges of symbolic computation: my favorite open problems. *J. Symb. Comput.* 29, 6 (2000), 891–919. doi:10.1006/jsco.2000.0370

[23] Kiran S. Kedlaya and Christopher Umans. 2011. Fast polynomial factorization and modular composition. *SIAM J. on Computing* 40, 6 (2011). doi:10.1137/08073408X

[24] Yasunori Kinoshita and Baitian Li. 2024. Power series composition in near-linear time. In *Proc. FOCS*. 2180–2185. doi:10.1109/FOCS61266.2024.00127

[25] George Labahn, Vincent Neiger, and Wei Zhou. 2017. Fast, deterministic computation of the Hermite normal form and determinant of a polynomial matrix. *J. Complexity* 42 (2017), 44–71. doi:10.1016/j.jco.2017.03.003

[26] Peter Lancaster and Miron Tismenetsky. 1985. *The theory of matrices.* Academic Press. https://shop.elsevier.com/books/the-theory-of-matrices/lancaster/978-0-08-051908-1 Second edition.

[27] François Le Gall. 2024. Faster rectangular matrix multiplication by combination loss analysis. In *Proc. SODA*. SIAM, 3765–3791. doi:10.1137/1.9781611977912.133

[28] Vincent Neiger. 2016. Fast computation of shifted Popov forms of polynomial matrices via systems of modular polynomial equations. In *ISSAC'16*. ACM, 365–372. doi:10.1145/2930889.2930936

[29] Vincent Neiger, Johan Rosenkilde, and Grigory Solomatov. 2020. Generic bivariate multi-point evaluation, interpolation and modular composition with precomputation. In *Proc. ISSAC*. ACM Press, 388–395. doi:10.1145/3373207.3404032

[30] Vincent Neiger, Bruno Salvy, Éric Schost, and Gilles Villard. 2024. Faster modular composition. *J. ACM* 71, 2 (2024). doi:10.1145/3638349

[31] Michael Nüsken and Martin Ziegler. 2004. Fast multipoint evaluation of bivariate polynomials. In *ESA 2004*. Springer. doi:10.1007/978-3-540-30140-0_49

[32] Adrien Poteaux and Éric Schost. 2013. Modular composition modulo triangular sets and applications. *Computational Complexity* 22, 3 (2013), 463–516. doi:10.1007/s00037-013-0063-y

[33] Victor Shoup. 1994. Fast construction of irreducible polynomials over finite fields. *J. Symb. Comput.* 17, 5 (1994), 371–391. doi:10.1006/jsco.1994.1025

[34] Victor Shoup. 1999. Efficient computation of minimal polynomials in algebraic extensions of finite fields. In *Proc. ISSAC*. ACM Press. doi:10.1145/309831.309859

[35] Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. 2024. New bounds for matrix multiplication: from alpha to omega. In *Proc. SODA*. SIAM, 3792–3835. doi:10.1137/1.9781611977912.134

[36] Wei Zhou, George Labahn, and Arne Storjohann. 2012. Computing minimal nullspace bases. In *Proc. ISSAC*. ACM Press, 366–373. doi:10.1145/2442829.2442881

A Generic bases of $\mathcal{N}_\mu$: proof of Lemma 2.5

The proof is based on fairly standard results, such as the links between Popov forms and controllability realizations [21, Scheme II, p. 427 & Sec. 6.7.2, p. 481]. Let $f \in \mathbb{K}[x]$ be of degree n. By identifying elements of the quotient $a \in \mathbb{K}[x]/\langle f\rangle$ with the corresponding coefficient vector $v_a \in \mathbb{K}^n$, we view $\mathbb{K}^n$ as a $\mathbb{K}[x]$-module through $x \cdot v_a = v_{xa}$. We denote by $M_x \in \mathbb{K}^{n\times n}$ the matrix of multiplication by x modulo f in the basis $(1, x, \ldots, x^{n-1})$, so that $x \cdot v_a = M_x v_a$. This allows us to relate polynomial relations between the powers of a mod f to linear dependencies between vectors in Krylov subspaces [20, Sec. 3.10].

Lemma A.1. *Let V_a be the matrix $[v_1\ v_a\ \cdots\ v_{a^{\mu-1}}] \in \mathbb{K}^{n\times\mu}$. The degree of a minimal basis of $\mathcal{N}_\mu$ is the smallest integer δ such that*

$$\operatorname{rank}([V_a\ M_xV_a\ \cdots\ M_x^{\delta-1}V_a]) = \dim(\operatorname{Span}(\{M_x^k v_{a^j}, k \in \mathbb{N}, 0 \le j < \mu\})) = n.$$

Proof. The latter dimension is indeed n, since the span contains the independent vectors $\{M_x^k v_1, 0 \le k < n\} = \{v_1, v_x, \ldots, v_{x^{n-1}}\}$.

We first show that the degree of a minimal basis of $\mathcal{N}_\mu$ cannot be less than δ. By contradiction, assume there exists a minimal basis $R \in \mathbb{K}[x]^{\mu\times\mu}$ of $\mathcal{N}_\mu$ whose degree is $\ell < \delta$. Let $(\ell_1, \ldots, \ell_\mu)$ be the column degrees of B and $L \in \mathbb{K}^{\mu\times\mu}$ be its leading matrix, that is, $L_{i,j}$ is the coefficient of degree ℓ_j of R_{ij}. Since R is column reduced, L is invertible. Then $P = R\,\mathrm{diag}(x^{\ell-\ell_1}, \ldots, x^{\ell-\ell_\mu})L^{-1}$ has the form

$$P = P_0 + xP_1 + \cdots + x^{\ell-1}P_{\ell-1} + x^\ell I_\mu, \quad P_k \in \mathbb{K}^{\mu\times\mu}.$$

Since the columns of R are in $\mathcal{N}_\mu$, those of P are too, meaning that

$$[1\ \ a\ \ \cdots\ \ a^{m-1}]R = [1\ \ a\ \ \cdots\ \ a^{m-1}]P = 0 \bmod f. \tag{10}$$

This translates as $V_aP_0 + M_xV_aP_1 + \cdots + M_x^{\ell-1}V_aP_{\ell-1} + M_x^\ell V_a = 0$. From this identity, any vector in $\{M_x^k v_{a^j}, k \ge \ell, 0 \le j < \mu\}$ can be expressed as a linear combination of the columns of the matrix $[V_a\ M_xV_a\ \cdots\ M_x^{\ell-1}V_a]$, which contradicts the minimality of δ.

We follow a similar path to show that a minimal basis of $\mathcal{N}_\mu$ has degree at most δ. By definition of δ, the columns of $M_x^\delta V_a$ are combinations of those of $[V_a\ M_xV_a\ \cdots\ M_x^{\delta-1}V_a]$. Therefore there exist matrices $P_0, \ldots, P_{\delta-1} \in \mathbb{K}^{\mu\times\mu}$ such that $V_aP_0 + M_xV_aP_1 + \cdots + M_x^{\delta-1}V_aP_{\delta-1} + M_x^\delta V_a = 0$. By construction, the matrix $P = P_0 + \cdots + x^{\delta-1}P_{\delta-1} + x^\delta I_\mu$ is nonsingular and its columns are in $\mathcal{N}_\mu$. It follows, by minimality, that any minimal basis of $\mathcal{N}_\mu$ has degree at most $\deg(P) = \delta$. □

Lemma 2.5 can now be proven by considering $a = x^\delta$, where $\delta = \lceil n/\mu\rceil$. Decompose f as $f = \sum_{0\le i<\mu} f^{(i)}a^i$, with $\deg(f^{(i)}) < \delta$ for $i < \mu - 1$ and $\deg(f^{(\mu-1)}) \le \delta$. In the identity

$$\begin{bmatrix} f & -a & -a^2 & \cdots & -a^{\mu-1} \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ -1 & a & & & f^{(1)} \\ & \ddots & & & \vdots \\ & & -1 & & \vdots \\ & & & \ddots & \vdots \\ & & & a & \\ & & & -1 & f^{(\mu-1)} \end{bmatrix} = \begin{bmatrix} a & 0 & \cdots & 0 & f^{(0)} \\ -1 & a & & & f^{(1)} \\ & \ddots & & & \vdots \\ & & -1 & & \vdots \\ & & & \ddots & \vdots \\ & & & a & \\ & & & -1 & f^{(\mu-1)} \end{bmatrix}$$

in $\mathbb{K}[x]^{\mu\times\mu}$, the first matrix in the product is a basis of $\mathcal{N}_\mu$ and the second one is unimodular. It follows that the right-hand side is a basis of $\mathcal{N}_\mu$ of degree δ, and that minimal bases of $\mathcal{N}_\mu$ have degree at most δ. Hence, by Lemma A.1, $[V_a\ M_xV_a\ \cdots\ M_x^{\delta-1}V_a]$ has maximal rank n. Now consider a generic a. A fortiori, the latter matrix still has rank n, and by Lemma A.1 again, any minimal basis of $\mathcal{N}_\mu$ has degree at most δ. Finally, since the sum of the column degrees of a minimal basis of $\mathcal{N}_\mu$ is exactly the degree of its determinant, i.e. $\deg(f) = n$, such a basis cannot have a degree less than $\lceil n/\mu\rceil$. Therefore, minimal bases of $\mathcal{N}_m$ have degree exactly $\lceil n/\mu\rceil$. Based on the above, if $\bar{a} = \sum_{0\le i<n} \bar{a}_i x^i$, where the $\bar{a}_i$'s are new indeterminates, then $[V_{\bar{a}}\ M_xV_{\bar{a}}\ \cdots\ M_x^{\delta-1}V_{\bar{a}}]$ has a nonzero n-minor $\Phi_{f,\mu} \in \mathbb{K}[\bar{a}_0, \ldots \bar{a}_{n-1}]$. A set of polynomials a for which minimal bases of $\mathcal{N}_m$ have degree exactly $\delta = \lceil n/\mu\rceil$ is thus given by those whose coefficients avoid the zero set of $\Phi_{f,\mu}$.

B Generic bases of $\mathcal{M}_\mu$

We show that for any $f \in \mathbb{K}[x]$ of degree n, and for generic a, any minimal basis of $\mathcal{M}_\mu$ defined by Eq. (1) has degree $\delta = \lceil n/\mu\rceil$. We use techniques very similar to those in Appendix A, which we do not repeat in detail. Let M_a be the matrix of multiplication by a modulo f. We now view $\mathbb{K}^n$ as a $\mathbb{K}[y]$-module through $y \cdot v = M_a v$ for $v \in \mathbb{K}^n$. By swapping the roles of a and x in Lemma A.1 we obtain the following. Since the minimal polynomial of a may have

a degree less than n, the statement now involves a dimension that may not be equal to n. However, this dimension will be n in generic cases.

Lemma B.1. *Let $V_x = [v_1\ v_x\ \cdots\ v_{x^{\mu-1}}] = [\mathrm{I}_\mu\ 0]^\mathsf{T} \in \mathbb{K}^{n\times\mu}$. The degree of a minimal basis of $\mathcal{M}_\mu$ is the smallest integer δ such that*

$$\begin{aligned}&\operatorname{rank}([V_x\ M_aV_x\ \cdots\ M_a^{\delta-1}V_x])\\ &\qquad = \dim(\operatorname{Span}(\{M_a^k v_{x^j}, k\in\mathbb{N}, 0\le j<\mu\})) \le n.\end{aligned}$$

Lemma B.1 can be proved by reasoning analogous to that of the proof of Lemma A.1. Relation matrices P for $\mathcal{M}_\mu$ are now matrices in $\mathbb{K}[y]^{\mu\times\mu}$. Equation (10) is reinterpreted in the form

$$[1\ \ x\ \ \cdots\ \ x^{\mu-1}]P = 0 \bmod \langle f(x), y-a(x)\rangle,$$

which now translates as $V_xP_0 + M_aV_xP_1 + \cdots + M_a^{\ell-1}V_xP_{\ell-1} + M_a^\ell V_xP_\ell = 0$, when P has degree ℓ. The degree of minimal bases of $\mathcal{M}_\mu$ in generic cases follows from Lemma B.1 by considering a of degree μ. For such an a, the matrix $[V_x\ M_aV_x\ \cdots\ M_a^{\delta-1}V_x]$ has rank n. Indeed, since a has degree μ, its first n columns form an invertible upper triangular matrix. It can finally be deduced that there exists a polynomial $\Psi_{f,\mu} \in \mathbb{K}[\bar{a}_0,\dots\bar{a}_{n-1}]$ whose avoidance of the zero set makes it possible to characterize polynomials a for which minimal bases of $\mathcal{M}_m$ have degree exactly $\delta = \lceil n/\mu\rceil$.

C Characteristic polynomial vs composition

Here, we argue informally that for generic a and f (in the Zariski sense), the problem of computing the characteristic polynomial of a modulo f and modular composition have roughly the same complexity. We restrict to algebraic algorithms that can be written as straight-line programs. In this case, each problem can be solved by an algorithm whose cost is a constant multiple of that of the other problem, plus $\tilde{O}(n)$ operations. We then explain how our approach could exploit this equivalence. Note that, generically, f and the characteristic polynomial χ_a of a are separable, hence the minimal polynomial of a modulo f coincides with χ_a.

The reduction of the (characteristic) minimal polynomial to composition is frequently used; it can be inferred from Shoup's work [33, 34]. The minimal polynomial of a modulo f can be computed using a randomized algorithm, which requires one modular composition (allowing g to have degree $2n-1$) and the computation of the minimal polynomial of a linearly recurrent sequence in $\mathbb{K}^\mathbb{N}$ [12, Sec. 12.3]. When χ_a is separable, a deterministic algorithm can be derived using the trace instead of random linear forms as was done in [34]. (See, for example, [32, Sec. 4.3]).

We now sketch the reduction in the opposite direction, via the problem of inverse modular composition. Given a and f as before, and for a polynomial $h \in \mathbb{K}[x]_{<n}$, inverse modular composition is the problem of computing $g \in \mathbb{K}[y]_{<n}$ such that $g(a) = h \bmod f$ [33, Thm. 3.5; 30, Sec. 6.2]. Assuming that the minimal polynomial of a mod f has degree n, the polynomial g always exists and is unique. To address inverse modular composition, we introduce a new variable z and consider the polynomial $a + zh$. Using that $\chi_a \in \mathbb{K}[y]$ is separable, h and the characteristic polynomial $\chi_{a,h} \in \mathbb{K}[y,z]$ of $a + zh$ modulo f can be related as follows ([2, Lem. 2.8] or [3, Sec. 12.4, Eq. 12.4], valid for an arbitrary field):

$$-\frac{\left(\left(\frac{\partial}{\partial z}\chi_{a,h}\right)_{z=0}\right)(a)}{\chi_a'(a)} = h \bmod f. \tag{11}$$

If $\bar{\chi}_a$ is the inverse of χ_a' modulo χ_a, then

$$g = -\bar{\chi}_a\left(\frac{\partial}{\partial z}\chi_{a,h}\right)_{z=0} \operatorname{rem} \chi_a \tag{12}$$

is the solution to the problem of inverse composition for h.

A straight-line program C that computes χ_a gives rise to a program C_z that computes $\chi_{a,h}$ modulo $\langle f, z^2\rangle$, giving the derivative at $z=0$ in the left-hand side of Eq. (11). The divisions that occur in C_z are those already present in C and the overall cost is multiplied by a constant (see e.g., the proof of [11, Lem. (7.2)]). Note that here we have considered straight-line programs, but a situation where the tests on elements modulo z^2 can be done on the coefficients of valuation 0 could be suitable. Once the above derivative is known, the solution g from Eq. (12) can be computed after applying the extended Euclidean algorithm to produce $\bar{\chi}_a$.

Finally, modular composition can be reduced to inverse modular composition. Indeed, if $\alpha \in \mathbb{K}[y]_{<n}$ and $\gamma \in \mathbb{K}[x]_{<n}$ satisfy

$$\alpha(a) = x \bmod f,\ \gamma(\alpha) = g \bmod \chi_a, \tag{13}$$

then we obtain $\gamma = g(a)$ rem f in two inverse modular compositions within the announced complexity.

We remark that Eq. (13) defines the $\mathbb{K}$-algebra isomorphism

$$\phi_a : \mathbb{K}[x]/\langle f\rangle \to \mathbb{K}[y]/\langle\chi_a\rangle \tag{14}$$

that maps x to α and a to y, and more generally u to v such that $v(a) = u \bmod f$. Therefore, $p(\alpha, y) = 0 \bmod \chi_a$, with $p \in \mathbb{K}[x,y]$, if and only if $p(x,a) = 0 \bmod f$. This implies, in particular, that the bases of $\mathcal{M}_\mu$ and $\mathcal{N}_\mu$ for α modulo χ_a (after the substitution of "y" by "x", and "x" by "y" to make notation match), coincide with the bases of $\mathcal{N}_\mu$ and $\mathcal{M}_\mu$ for a modulo f, respectively.

We now discuss our composition approach in light of this computational equivalence. The determinant of a minimal basis of $\mathcal{M}_\mu$ is a multiple of the minimal polynomial of a mod f [30, Prop. 4.1], hence gives the characteristic polynomial when the two coincide. Since this determinant can be computed using $\tilde{O}(\mu^{\omega-1}n)$ operations [25], obtaining χ_a does not incur an additional cost beyond the minimal basis cost. According to Eq. (13), modular composition reduces to two applications of Eq. (12), which essentially corresponds to the computation of the characteristic polynomials of $a + zx$ modulo $\langle f, z^2\rangle$ and $\alpha + zg$ modulo $\langle\chi_a, z^2\rangle$. To the extent that the computation of a minimal basis of $\mathcal{M}_m$ and its determinant are written in terms of straight-line programs (the isomorphism of Eq. (14) ensures the relevance for both a and α), we obtain a different version of the last part of our algorithm, where computing the two characteristic polynomials above and applying Eqs. (12) and (13) yields the result of the final bivariate modular composition of Steps (4.4)-(4.5).