

Faster Modular Composition

VINCENT NEIGER, Sorbonne Université, France

BRUNO SALVY, Inria, France

ÉRIC SCHOST, University of Waterloo, Canada

GILLES VILLARD, CNRS, France

A new Las Vegas algorithm is presented for the composition of two polynomials modulo a third one, over an arbitrary field. When the degrees of these polynomials are bounded by n , the algorithm uses $O(n^{1.43})$ field operations, breaking through the $3/2$ barrier in the exponent for the first time. The previous fastest algebraic algorithms, due to Brent and Kung in 1978, require $O(n^{1.63})$ field operations in general, and $n^{3/2+o(1)}$ field operations in the special case of power series over a field of large enough characteristic. If cubic-time matrix multiplication is used, the new algorithm runs in $n^{5/3+o(1)}$ operations, while previous ones run in $O(n^2)$ operations.

Our approach relies on the computation of a matrix of algebraic relations that is typically of small size. Randomization is used to reduce arbitrary input to this favorable situation.

CCS Concepts: • Computing methodologies → Algebraic algorithms; • Theory of computation → Algebraic complexity theory;

Additional Key Words and Phrases: Symbolic computation, algorithm, complexity, modular composition, multivariate polynomial, multivariate relation

ACM Reference Format:

Vincent Neiger, Bruno Salvy, Éric Schost, and Gilles Villard. 2024. Faster Modular Composition. *Journal of the ACM* 71, 2, Article 11 (April 2024), 79 pages. <https://doi.org/10.1145/3638349>

1 INTRODUCTION

1.1 Problem and Result

Many fundamental operations over univariate polynomials of degree at most n with coefficients in a commutative ring \mathbb{A} can be computed in a number of arithmetic operations in \mathbb{A} that is quasi-linear in n [26]. It is the case for multiplication, division with remainder by a monic polynomial, multipoint evaluation, interpolation at points whose differences are units in \mathbb{A} , and greatest common divisors when \mathbb{A} is a field.

Implementations of Algorithms 3.1 to 8.1 are available at https://github.com/vneiger/faster_modular_composition_SageMath, based on the SageMath software (version ≥ 9.4 is required and is freely available at <https://www.sagemath.org/>). Authors' addresses: V. Neiger, Sorbonne Université, CNRS, LIP6, F-75005 Paris, France; e-mail: vincent.neiger@lip6.fr; É. Schost, University of Waterloo, David R. Cheriton School of Computer Science, Waterloo, Ontario, Canada N2L 3G1; e-mail: eschost@uwaterloo.ca; B. Salvy, Inria, Univ. Lyon, CNRS, ENS de Lyon, UCBL, LIP, F-69364 Lyon, cedex 07, France; e-mail: bruno.salvy@inria.fr; G. Villard, CNRS, Univ. Lyon, ENS de Lyon, Inria, UCBL, LIP, F-69364 Lyon, cedex 07, France; e-mail: gilles.villard@cnrs.fr.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 0004-5411/2024/04-ART11

<https://doi.org/10.1145/3638349>

In contrast with these operations, improving the cost bound for *modular composition* is a long-standing open question. Given three polynomials $a, f \in \mathbb{A}[x]$ and $g \in \mathbb{A}[y]$, with $\deg(a) < n$ and $\deg(g) < n$ where $n = \deg(f)$, and with f monic, this problem is to compute $g(a) \text{ rem } f$, where the “rem” operation takes the remainder of the Euclidean division.

Motivation. This operation arises in a variety of contexts. For instance, with $f = x^n$, it amounts to power series composition. For many applications of power series, composition is the bottleneck. This is the case for power series reversion, which can then be reduced to composition with a small overhead [17]. This is also the case of further operations such as solving families of functional equations [32].

The application of certain algebra morphisms also translates to modular composition. Over a field \mathbb{K} , for f and a in $\mathbb{K}[x]$, we denote by $a \bmod f \in \mathbb{K}[x]/\langle f \rangle$ the class of a modulo f . Then, for e and f in respectively $\mathbb{K}[y]$ and $\mathbb{K}[x]$, and for a \mathbb{K} -algebra morphism $\phi : \mathbb{K}[y]/\langle e \rangle \rightarrow \mathbb{K}[x]/\langle f \rangle$, if $\phi(y \bmod e) = a \bmod f$ then for g in $\mathbb{K}[y]$, the image $\phi(g \bmod e)$ is equal to $g(a) \bmod f$.

Over finite fields, with e and f the same polynomial and ϕ the Frobenius endomorphism, this results in modular composition playing an important role in algorithms for polynomial factorization [27, 48, 49]. Dedicated algorithms exist for modular composition over finite fields, with quasi-linear complexity (they are discussed later). Still, there remains a variety of questions that can be considered over arbitrary fields, and which are impacted by modular composition (or closely related operations such as power projection, discussed later as well): computing the minimal polynomial of an algebraic number [69–71], normal bases computations [30, 49], arithmetic operations with two algebraic numbers [12], computing with towers of algebraic extensions [36, 65, 66], Riemann–Roch space computations [1, 2], and so on.

Previous algorithms. The most famous algorithm in this area is that of Kedlaya and Umans, which achieves complexity $n^{1+\epsilon} \log^{1+o(1)}(q)$ *bit operations* for any given $\epsilon > 0$ [52, Corollary 7.2] when the field \mathbb{K} is the finite field \mathbb{F}_q . In contrast, we deal with an arbitrary field \mathbb{K} and count *arithmetic operations* in \mathbb{K} . In this context, the known algorithms have much higher complexity estimates.

Modular composition can be performed using Horner’s algorithm with modular reduction at each stage, which leads to a complexity in $\tilde{O}(n^2)$ operations if fast polynomial multiplication is used. The notation $c' = \tilde{O}(c)$ means that $c' = O(c \log^k(c))$ for some $k > 0$; in other words, logarithmic factors are dropped.

In 1978, Brent and Kung gave two algorithms that perform composition modulo x^n (the case of power series) [16, 17]. One relies strongly on Taylor expansion and runs in $\tilde{O}(n^{3/2})$ operations; the other one, using a baby steps/giant steps approach, uses $O(n^{(\omega+1)/2}) + \tilde{O}(n^{3/2})$ operations, where $\omega \leq 3$ is a feasible matrix multiplication exponent (two $n \times n$ matrices can be multiplied in $O(n^\omega)$ operations; the best known bound is 2.371552 [3, 22, 77]). This latter algorithm works verbatim and in the same complexity for composition modulo an arbitrary polynomial f of degree n not restricted to be x^n [27]. Both these algorithms have remained essentially the best ones since then. Huang and Pan used fast rectangular matrix multiplication in the central step of the baby steps/giant steps algorithm to reduce its complexity to $O(n^{\omega_2/2}) + \tilde{O}(n^{3/2})$ [41], where $\omega_2 \leq \omega + 1$ is a feasible exponent such that a $n \times n^2$ matrix can be multiplied by a $n^2 \times n$ matrix in $O(n^{\omega_2})$ operations. The currently best known value gives $\omega_2 \approx 3.250385$ [55, 56, 77], which makes the previous algebraic complexity bound $O(n^{1.626})$ for modular composition for an arbitrary f . Even assuming an optimal matrix multiplication, which means $\omega = 2$, these algorithms do not break the exponent barrier $3/2$.

Our result. The open problem 2.4 in the book of Bürgisser, Clausen, and Shokrollahi [19] asks whether Brent and Kung’s algorithm can be improved substantially. The research problem 12.19 in

von zur Gathen and Gerhard's book [26] asks for complexity in $\tilde{O}(n^{1.5})$ or better. Our main result answers both questions positively when \mathbb{A} is a field, with few extra hypotheses.

THEOREM 1.1. *Given $a, f \in \mathbb{K}[x]$ and $g \in \mathbb{K}[y]$ with coefficients in a field \mathbb{K} , with $\deg(f) = n$, $\deg(a)$ and $\deg(g)$ smaller than n , and a tuple r of $O(n^{1+1/3})$ field elements, Algorithm **MODULARCOMPOSITION** returns either $g(a) \bmod f$ or FAIL after $\tilde{O}(n^\kappa)$ arithmetic operations in \mathbb{K} , with*

$$\kappa = 1 + \frac{1}{\frac{1}{\omega-1} + \frac{2}{\omega_2-2}} < 1.43. \quad (1)$$

It returns FAIL with probability at most $(2n^4 + 18n^2)/\text{card}(S)$ when the entries of r are chosen uniformly and independently from a finite subset $S \subseteq \mathbb{K}$.

Here, we use an algebraic model of computation: roughly, basic arithmetic operations $\{+, -, \times, \div\}$ and zero-tests in the base field \mathbb{K} are counted at unit cost; for more details, see Section 2. As usual with probabilistic algorithms of Las Vegas type, the algorithm can be repeated until it succeeds, so that only its running time becomes a random variable.

We assume that the characteristic p of \mathbb{K} is known to the algorithms. For \mathbb{K} finite and of small cardinality q (namely, $q \leq 2n^4 + 18n^2$), the probability statement becomes vacuous. However, in such cases, one can work in a sufficiently large field, by constructing an extension of \mathbb{K} of degree $O(\log(n))$ efficiently (see [26, Section 14.9] and references therein). In this extension, each arithmetic operation can be performed in $\tilde{O}(\log(n))$ arithmetic operations in \mathbb{K} , so that the asymptotic complexity estimate is unaffected.

We also give a probabilistic algorithm of the Las Vegas type with the same complexity bound for computing an annihilating polynomial for $a \bmod f$, that is, a nonzero polynomial $g \in \mathbb{K}[y]$ such that $g(a) \bmod f = 0$.

The improvements brought by fast matrix multiplication on one hand and by fast rectangular matrix multiplication, on the other hand, are made clearer by noting that the exponent κ of composition satisfies

$$\frac{4}{3} \leq \kappa = 1 + \underbrace{\frac{1}{\frac{1}{\omega-1} + \frac{2}{\omega_2-2}}}_{<1.42945} \leq \underbrace{\frac{\omega+2}{3}}_{<1.4572} \leq \underbrace{\frac{5}{3}}_{<1.666667},$$

where the first approximation is obtained with the bounds on ω and ω_2 given above; the second one is obtained when no fast rectangular matrix multiplication is used, so that ω_2 simply becomes $\omega + 1$; the last one is obtained when no fast matrix multiplication is used, thus taking $\omega = 3$. In the latter case, our algorithm is the first subquadratic one for modular composition. In the other direction, considering the lower bounds $\omega \geq 2$ and $\omega_2 \geq 3$ shows that $\kappa \geq 4/3$, giving a lower bound on the complexity estimate that can be achieved by the algorithm designed in this work.

Main steps. To compute $g(a) \bmod f$, our approach relies on first computing a polynomial \tilde{g} of “small degree” such that $\tilde{g}(a) \bmod f = g(a) \bmod f$. If $a \bmod f$ has a minimal polynomial of small degree $\mu(a)$, then one can take $\tilde{g} = g \bmod \mu$. In general, such a μ may not exist, and a small degree *univariate* \tilde{g} may not exist either. However, generically, one can compute a set of *bivariate* polynomials $\mu_i(x, y)$ of “small degree” such that $\mu_i(x, a) \bmod f = 0$. These are called *relations*. From these, a small degree bivariate $\tilde{g}(x, y)$ is found via some type of reduction of g by all μ_i ’s simultaneously, ensuring $\tilde{g}(x, a) \bmod f = g(a) \bmod f$. Relations form one of the main ingredients of the new algorithm, and most of the new technical results are about them. On the algorithmic side, the coefficients of these relations are gathered into matrices called *matrices of relations* and we make heavy use of fast algorithms on polynomial matrices.

Here and throughout the article, genericity is understood in the Zariski sense: a property is generic if it holds outside of a hypersurface of the corresponding parameter space. A *randomized change of basis* brings f and a to a situation where “small” matrices of relations exist.

This probabilistic algorithm is proved to be correct for f separable (i.e., with no repeated roots in an algebraic closure $\overline{\mathbb{K}}$ of \mathbb{K}) and for f purely inseparable (i.e., with only one root in $\overline{\mathbb{K}}$). Modular composition modulo an arbitrary f is reduced to these two extreme cases by *separable decomposition* of f [57], Chinese remainder theorem, and a slight generalization of a technique called *untangling* [34]. The latter allows to transport the composition problem over \mathbb{K} modulo a factor of the separable decomposition, to a composition problem over a quotient algebra with purely inseparable modulus.

Complexity aspects. Under genericity conditions, $m \times m$ matrices of relations of “small degree” $d \leq \lceil n/m \rceil$ are shown to exist (the choice of m is optimized below). Their computation starts from the first m coefficients of the $2md$ polynomials $x^i a^k \bmod f$, for $0 \leq i < m$ and $0 \leq k < 2d$. With

$$c(n, m, d) = (m + n/d)d^{\omega_2/2}, \quad (2)$$

these coefficients can be computed in $\tilde{O}(m^2d + c(n, m, d))$ operations in \mathbb{K} (Section 3.3).

From these coefficients, a matrix of relations is obtained by *approximant bases* [31] in $\tilde{O}(m^\omega d)$ operations (Section 5.4). Given such a matrix and in the same complexity, linear system solving over $\mathbb{K}[x]$ [80] allows us to reduce the univariate $g \in \mathbb{K}[y]$ to a bivariate $\tilde{g} \in \mathbb{K}[x, y]$, of degrees smaller than m and d in x and y , such that $g(a) \equiv \tilde{g}(x, a) \bmod f$ (Section 4.2).

A generalization due to Nüsken and Ziegler [63] of Brent and Kung’s algorithm to the case of a bivariate polynomial $\tilde{g}(x, y) \in \mathbb{K}[x, y]$ finally computes the “uni-bivariate” composition $\tilde{g}(x, a) \bmod f$ using $\tilde{O}(c(n, m, d))$ operations in \mathbb{K} (Section 3.2).

Altogether, the costs of the various parts of the algorithm add up to $\tilde{O}(m^\omega d + c(n, m, d))$ operations in \mathbb{K} . Then choosing m and $d = \lceil n/m \rceil$ so as to minimize $m^\omega d + c(n, m, d)$ leads us to $m \sim n^\eta$, where

$$\eta = \frac{1}{1 + \frac{\omega-1}{(\omega_2-2)/2}}, \quad (3)$$

which is approximately 0.3131 with the bounds on ω and ω_2 given above and leads to the complexity estimate $\tilde{O}(n^\kappa)$ of Theorem 1.1.

1.2 Previous Algorithms in Special Cases

To compute $g(a) \bmod f$, previous known improvements upon Brent and Kung’s approach all have requirements on the input, either on some of the polynomials f , g , and a , or on the ring or field of coefficients—possibly with nonalgebraic algorithms.

1.2.1 Special Modulus.

Power Series. For the special case $f = x^n$ of power series, Brent and Kung’s second algorithm relying on Taylor expansion performs composition in only $\tilde{O}(n^{3/2})$ operations, provided $a'(0)$ and $(\lceil \sqrt{n \log(n)} \rceil)!$ are invertible in \mathbb{A} ; the assumption on $a'(0)$ can be weakened [32, Section 3.4.3]. In more variables, even in the specific case $g(x, a) \bmod f$ handled by the Nüsken–Ziegler algorithm, we do not know of any algorithm computing composition faster for power series than modulo arbitrary polynomials.

Faster composition in only $\tilde{O}(n)$ operations for $g(a) \bmod x^n$ is possible for many special cases of g : when g is a polynomial of degree $O(1)$, but also when it is a power series solution of a polynomial equation of degree $O(1)$ via Newton’s iteration, or when it is a solution of a

differential equation (e.g., exp), by first forming a differential equation for $g(a)$ and then solving it by Newton's iteration or other divide-and-conquer algorithms, generally in characteristic 0 or large enough [11, 17, 32, 58; 10, Section 13.4].

Similarly, still in the case when $f = x^n$, if furthermore a has specific properties, then composition of power series can be performed in $\tilde{O}(n)$ operations. This is the case when a is a polynomial of moderate degree [17] (it is a part of Brent and Kung's fast composition algorithm), an algebraic power series [32], but also for a class of truncated power series that can be obtained via shifts, reversals, scalings, multiplications by polynomials, exponentials, and logarithms [14].

Separable Polynomials. Ritzmann observed that for a separable modulus $f(x) = (x - \epsilon_1) \cdots (x - \epsilon_n)$ with distinct $\epsilon_1, \dots, \epsilon_n$ that are known, modular composition boils down to multipoint evaluation and interpolation [68], which can be computed in $\tilde{O}(n)$ arithmetic operations. When furthermore the ring of coefficients is \mathbb{Z} , he uses well-chosen ϵ_i 's to give an efficient algorithm for composing power series, in a nonalgebraic model of computation: if g and a over \mathbb{Z} have coefficients bounded in absolute value by K , then $g(a) \bmod x^n$ can be computed using $\tilde{O}(n^2 \log(K))$ bit operations, which is quasi-optimal since the output has bit size $\Omega(n^2 \log(K))$ in general.

Chinese Remainder Theorem. In our work, the cases of power series and of separable polynomials play an important role as well. We use the observation that if a factorization $f = f_1 \cdots f_s$ is known with the f_i 's relatively prime, then composing modulo f reduces to composing modulo each f_i and reconstructing the result via the Chinese remainder theorem. Several consequences of this observation have been discussed by van der Hoeven and Lecerf [35].

1.2.2 Special Rings or Fields. For power series over a ring \mathbb{A} of positive characteristic, Bernstein proposed an algebraic algorithm whose complexity is quasi-linear in n , with a constant factor that depends on the characteristic of the ring [6]. In particular, this algorithm is very efficient over rings whose characteristic is a product of small primes; if \mathbb{A} is a ring of prime characteristic p then the algorithm uses $\tilde{O}(np)$ operations in \mathbb{A} .

A further step forward was achieved by Umans in 2008 [72], with a new algorithm for modular composition modulo an arbitrary f , over finite fields of small characteristic: if p is $n^{o(1)}$, his algorithm uses $n^{1+o(1)}$ base field operations. Later, Kedlaya and Umans introduced new techniques for composition over finite rings of the form $(\mathbb{Z}/r\mathbb{Z})[z]/\langle h(z) \rangle$, for an integer r and h monic. For a finite field $\mathbb{K} = \mathbb{F}_q$, their algorithm runs in $n^{1+\epsilon} \log^{1+o(1)}(q)$ bit operations [52, Corollary 7.2].

As in Ritzmann's work, a key idea in References [52, 72] is to exploit fast multipoint evaluation, but this time in a multivariate setting. The composition $g(a) \bmod f$ is reduced to the evaluation at suitable points of a multivariate polynomial constructed from g by an inverse Kronecker substitution, decreasing degrees at the expense of increasing the number of variables. Umans' algorithm performs the evaluation using the properties of the Frobenius endomorphism [72, Thm 6], while Kedlaya and Umans' proceeds by lifting to characteristic zero (which requires working in a bit complexity model) [52, Thm 6.3]. These multipoint evaluation algorithms have been extended to arbitrary number of variables and arbitrary finite fields [7, 8]. For general fields, efficient analogues of these multivariate multipoint evaluation algorithms are currently unknown.

1.3 Related Questions

1.3.1 (Multivariate) Multipoint Evaluation. For simplicity, we limit the discussion to the case of a field; most of it extends to rings, with minor restrictions. The evaluation of a univariate $g \in \mathbb{K}[x]_{<n}$ at n points in the field \mathbb{K} , and conversely the interpolation of a polynomial of degree $< n$ from n values, are computable in quasi-linear complexity [26, Chap. 10]. For polynomials in at least two variables, however, the situation becomes tightly related to modular composition.

The motivation of Nüsken and Ziegler [63] was the evaluation of a polynomial $g \in \mathbb{K}[x, y]_{<(m, d)}$ at n points $(x_k, y_k)_{1 \leq k \leq n}$ in general position, with $md = O(n)$. Their algorithm first computes a univariate interpolation polynomial such that $a(x_k) = y_k$ for all k ; then the composition $b = g(x, a(x)) \text{ rem } f$, where $f = \prod_k (x - x_k)$; and concludes by a univariate multipoint evaluation of b at x_1, \dots, x_n . Since the univariate evaluation and interpolation are performed in essentially linear time, the complexity is dominated by the “uni-bivariate” modular composition $g(x, a) \text{ rem } f$.

The case when several points have the same x -value can be handled by an affine change of coordinates [63]; another approach, taken by Kedlaya and Umans, is to pick n suitable points t_1, \dots, t_n in \mathbb{K} , to compute two interpolation polynomials a_x and a_y in $\mathbb{K}[t]$, and thus reduce the evaluation to the fully bivariate modular composition $g(a_x, a_y) \text{ rem } f$, where now $f = \prod_k (t - t_k)$. This extends to an arbitrary number of variables and shows that multipoint evaluation in s variables reduces to multivariate modular composition in the same number of variables [52, Theorem 3.3].

As mentioned in Section 1.2.2, Kedlaya and Umans actually make a heavy use of a converse reduction [52, Theorem 3.1]. If g is a polynomial in $\mathbb{K}[x_1, \dots, x_s]$, the composition $g(a_1(x), \dots, a_s(x)) \text{ rem } f$ reduces to a multipoint evaluation of a polynomial of smaller degree in each of its variables, whose number is increased. For the univariate case of composition ($s = 1$) studied here, the smallest possible number of variables for evaluation would be 2, leading to a bivariate evaluation of a polynomial of degree \sqrt{n} at $\Theta(n^{3/2})$ points, which is too large for our target complexity. The next possible choice would be a polynomial of 3 variables in degree $n^{1/3}$ at $\Theta(n^{4/3})$ points. Unfortunately, we are not aware of a sufficiently efficient multipoint evaluation algorithm in 3 or more variables to make this approach succeed in the algebraic model.

1.3.2 Bivariate Ideals. Viewing the problem of computing $g(a) \bmod f$ as a problem of reduction of g modulo the ideal $\mathcal{I} = \langle y - a(x), f(x) \rangle$, we introduce bivariate polynomials in a different way from the inverse Kronecker substitution mentioned above. Gröbner bases are commonly used for reductions modulo multivariate ideals. A division with a remainder similar to that in Equation (6) below would be achieved via reduction by an appropriate Gröbner basis of \mathcal{I} , provided we could compute this basis and perform the reduction in good complexity. However, already the size of the Gröbner basis itself may be $\Theta(n^{3/2})$ (see the example below), hence exceed our target complexity.

For an ideal given by two generic bivariate polynomials of degree n (hence the ideal is of degree n^2) and the graded lexicographic order, van der Hoeven and Larriau avoid the use of an explicit Gröbner basis. They show that a *concise representation* of the basis of size only $\tilde{O}(n^2)$ is sufficient for reducing a polynomial modulo the ideal in time $\tilde{O}(n^2)$ [33]; the concise representation consists in particular of truncations of well chosen polynomials in the ideal. It is unclear to us whether a similar truncation strategy could be applied specifically to \mathcal{I} , whose degree is only n . Instead, the matrices of relations we compute give a set of small degree polynomials in \mathcal{I} that may not generate the whole ideal (see Section 4.1), but provide a process of complexity $\tilde{O}(n^\kappa)$ for the reduction modulo \mathcal{I} of Equation (6). These polynomials generate the same ideal as the first polynomials in the Gröbner basis of \mathcal{I} for the lexicographic order (see Corollary 4.3).

The concise representation of Gröbner bases has also been exploited by van der Hoeven and Lecerf for computing the minimal polynomial of the multiplication by y modulo \mathcal{J} , when $\mathcal{J} = \langle f_1, f_2 \rangle$ is generated by two generic polynomials $f_1, f_2 \in \mathbb{K}[x, y]$ and \mathbb{K} is a finite field [40, Section 4]. They apply the transposition principle to a bivariate modular composition map modulo \mathcal{J} , then compute the minimal polynomial from the resulting bivariate power projections [45, Section 6]. The evaluation of the composition map modulo \mathcal{J} is again in $\tilde{O}(n^2)$, thanks to the concise representation [40]. In our case of $\mathcal{I} = \langle y - a(x), f(x) \rangle$ and for a generic a , matrices of relations allow us to compute the minimal polynomial of the multiplication by y modulo \mathcal{I} in

complexity $\tilde{O}(n^k)$ (see Section 10.1); matrices of relations are obtained via a bivariate power projection process that can be regarded, in part, as dual to Nüsken and Ziegler's bivariate modular composition algorithm (Section 3.4.3).

Note. For a sufficiently large field \mathbb{K} , take $f = (x-1) \cdots (x-n)$, where $n = k(k+1)/2$, and $a \in \mathbb{K}[x]$ the polynomial of degree smaller than n such that $a(i) = [\sqrt{2}i]$ for $1 \leq i \leq n$. Then the reduced Gröbner bases for the graded lexicographic order and for the lexicographic order, both with $y < x$, coincide. They contain one polynomial with leading term $x^i y^{k-i}$ for each $i \in \{0, \dots, k\}$. Counting the number of monomials of these polynomials shows that this basis has $k(k+1)(k+2)/3 + (k+1)$ monomials; this is of the order of $n^{3/2}$.

1.3.3 Modular Composition and Multipoint Evaluation with Precomputation. Quasi-linear modular composition $g(a) \bmod f$ is feasible after precomputations on (f, a) only, for a generic and f square free [61].

Likewise, after precomputations on the evaluation points and under genericity assumptions on them, quasi-linear multivariate multipoint evaluation is feasible [39], as well as quasi-linear bivariate interpolation [61]. Furthermore, for bivariate evaluation, genericity can be replaced by randomization [38].

In these works, the precomputation stages are at least as expensive as the fastest known corresponding modular composition or multipoint evaluation algorithms. They have a feature in common with our composition algorithm: from f, a (or from the evaluation points), they compute a set of polynomials that belong to $\langle y - a(x), f(x) \rangle$ (or vanish at the points), and allow for efficient degree reduction of the polynomial to compose with (or to evaluate). This set is either akin to several matrices of relations of $\mathcal{M}_m^{(a,f)}$ for a small number of m ranging from 1 to n [61], or is a collection of well-chosen polynomials in several Gröbner bases for subsets of the points so as to build a multivariate divide and conquer evaluation tree [38, 39].

1.4 Algorithmic Tools

Our work builds upon a sequence of earlier algorithmic progress that we now recall. We denote by $\mathbb{K}[x]_{<n}$ the set of univariate polynomials in x with coefficients in \mathbb{K} and degree less than n ; by $\mathbb{K}[x, y]_{<(r,s)}$ the bivariate polynomials in x, y of bidegree in (x, y) less than (r, s) .

1.4.1 Baby Steps/Giant Steps. One of the bottlenecks in algebraic approaches for evaluating g at a modulo f is the computation of successive powers $1, a, a^2, \dots$ modulo f , which leads to the question of minimizing the number of powers that are used. The solution used by Brent and Kung relies on a baby steps/giant steps scheme [17, 64], where only

$$1, a, \dots, a^{\lceil \sqrt{n} \rceil} \bmod f \quad \text{and} \quad a^{2\lceil \sqrt{n} \rceil}, a^{3\lceil \sqrt{n} \rceil}, \dots \bmod f$$

are computed. The former group forms the baby steps; the latter forms the giant steps. The problem is then reduced to about \sqrt{n} modular compositions “ $g_i(a) \bmod f$ ” for g_i of degree about \sqrt{n} . These compositions are all obtained simultaneously through the multiplication of two matrices of sizes roughly $\sqrt{n} \times \sqrt{n}$ and $\sqrt{n} \times n$. This is followed by a less expensive Horner evaluation step using the powers of $a^{\lceil \sqrt{n} \rceil}$. See Section 3.1 for a complete description.

1.4.2 Projection-Reconstruction. Wiedemann's algorithm [76] finds the minimal polynomial of a matrix $A \in \mathbb{K}^{n \times n}$ by considering the sequence $(v^T A^k w)_{k \geq 0}$, for two vectors v and w . This sequence is linearly recurrent and its generating function $h(y) = \sum_{k \geq 0} (v^T A^k w) / y^{k+1}$ is rational; for generic v and w , the denominator of $h(y)$ is the minimal polynomial μ_A of the matrix A . Writing $d \leq n$ for the degree of μ_A , this polynomial can be reconstructed efficiently from the first $2d$ terms

of the sequence by the Berlekamp–Massey algorithm or, equivalently, by the computation of a Padé approximant. Given the expansion in y^{-1} of a rational power series $h(y) = q(y)/\mu_A(y)$ with polynomials q and μ_A of degree at most $d - 1$ and d , this reconstructs the fraction (q, μ_A) as a solution of

$$(h(y) + O(y^{-2d-1}))\mu_A(y) - q(y) = O(y^{-d-1}). \quad (4)$$

If the degree of μ_A is unknown, one can use this approach with the upper bound $d = n$ instead.

Wiedemann’s algorithm can be combined with the baby steps/giant steps paradigm [44, Section 3; 69; 49, Algorithm AP]. In particular, when A is the matrix M_a of multiplication by $a \bmod f$ in the basis $\mathcal{B} = (1, x, \dots, x^{n-1})$ of $\mathbb{K}[x]/\langle f \rangle$, this was used by Shoup to compute the *minimal polynomial* of the polynomial a modulo f [69–71]. For irreducible f , Shoup used the vectors $v = w = 1$ (where 1 is the first column of the identity matrix), in which case the sequence $(v^T A^k w)_{k \geq 0}$ becomes the sequence of *power projections* $(\ell(1), \ell(a), \ell(a^2), \dots)$, where ℓ is the linear form that takes the coefficient of 1 of an element of $\mathbb{K}[x]/\langle f \rangle$ written on the basis \mathcal{B} . For an arbitrary f , Shoup used a random linear form ℓ , corresponding to a random choice of the vector v and $w = 1$.

In either case, the required $2d$ elements of the sequence can be obtained by left multiplication by v^T of a matrix whose columns are the coefficient vectors of $1, a, a^2, \dots$ modulo f . Now, the *right* multiplication of the exact same matrix by a vector of coefficients corresponds to modular composition. Using the *transposition principle*, Shoup described a baby steps/giant steps algorithm that computes the power projections for an arbitrary linear form $\ell : \mathbb{K}[x]/\langle f \rangle \rightarrow \mathbb{K}$ in the same complexity as that of Brent and Kung’s algorithm [69–71] (See Section 3.1.2.). This principle states that the existence of an algebraic algorithm for the multiplication of a matrix by a vector induces the existence of an algorithm for the product of the transpose of that matrix by a vector, both having essentially the same complexity [19, Theorem 13.20; 13].

The same idea is used by Shoup for another operation that we also need. Given a, b, f , the *inverse modular composition* asks for a polynomial g of least degree such that $g(a) \equiv b \bmod f$ or for a proof that no such g exists. This problem reduces to the computation of the power projections

$$(\ell(1), \ell(a), \dots, \ell(a^{2n-1})) \quad \text{and} \quad (\ell(b), \ell(ab), \dots, \ell(a^{n-1}b)),$$

again in the same complexity as that of modular composition, followed by the resolution of a linear system of Hankel type [69, Theorem 3.5]. The latter is known to be equivalent to Padé approximation [15], where Equation (4) generalizes to

$$\left(\sum_{k \geq 0} \frac{\ell(a^k)}{y^{k+1}} + O(y^{-2n-1}) \right) g(y) - q(y) = \sum_{k \geq 0} \frac{\ell(a^k b)}{y^{k+1}} + O(y^{-n-1}),$$

with unknowns a numerator $q(y) \in \mathbb{K}[y]_{<n}$ and the inverse composition $g(y) \in \mathbb{K}[y]_{\leq n}$.

1.4.3 Blocks for Speed and Structure. Coppersmith introduced a block version of Wiedemann’s algorithm [20]. There, the scalar sequence $(v^T A^k w)_{k \geq 0}$ is replaced by the matrix sequence $(V^T A^k W)_{k \geq 0}$ for two *matrices* $V \in \mathbb{K}^{n \times \ell}$ and $W \in \mathbb{K}^{n \times m}$: the generating function $H(y) = \sum_{k \geq 0} (V^T A^k W) / y^{k+1}$ is a rational $\ell \times m$ matrix.

Such a matrix admits an irreducible *matrix fraction description* $N(y)D(y)^{-1}$ with $N \in \mathbb{K}[y]^{\ell \times m}$ and $D \in \mathbb{K}[y]^{m \times m}$ two polynomial matrices (see Section 5.1.1), and the columns of the denominator matrix D form a basis of the $\mathbb{K}[y]$ -module of polynomial vectors $u \in \mathbb{K}[y]^m$ such that $\sum_{i \leq \deg(u)} V^T A^{k+i} W u_i = 0$ for all $k \geq 0$, where u_i denotes the coefficient of y^i in u [50, Lemma 2.8]. For $m = 1$, this module is the ideal generated by the minimal polynomial of the sequence in Wiedemann’s algorithm.

For $1 \leq m \leq \ell \leq n$, the matrix D contains more information: for example, for generic V and W , its invariant factors are the m invariant factors of largest degree of the characteristic matrix

$yI_n - A$ [50, Theorem 2.12], the highest degree one being the minimal polynomial of A . Consequently, the determinant of D has degree the sum ν_m of the degrees of these m invariant factors, which implies that $\nu_m \leq n$.

The computation of D can be achieved in two steps, which are matrix versions of the methods used for $m = 1$ in Section 1.4.2. Writing d for the degree of D , it is sufficient to compute the first $2d$ matrices of the sequence $(V^T A^k W)_{k \geq 0}$, which can be done by a baby steps/giant steps approach [50]. Next, D is obtained by matrix fraction reconstruction, solving

$$\left(V^T (yI_n - A)^{-1} W + O(y^{-2d-1}) \right) D(y) - N(y) = O(y^{-d-1})$$

for the unknown $N \in \mathbb{K}[y]^{\ell \times m}$ and $D \in \mathbb{K}[y]^{m \times m}$ of degrees at most $d - 1$ and d ; this can be done efficiently by a generalization of Padé approximation called minimal approximant bases, whose properties are recalled in Section 5.2. (See References [50, 51] for bibliographic pointers to algorithms that compute minimal linear generators of matrix sequences.) The parameter d plays a major role in the efficiency of both steps: it is usually unknown *a priori*, and might be as large as $\Theta(n)$. Yet, the interest of this block approach lies in the fact that, for generic V and W and $\ell \geq m$, the matrix D has degree $d = \lceil \nu_m/m \rceil \leq \lceil n/m \rceil$ [74, Corollary 6.4].

1.4.4 Efficient Projections and Small Bivariate Polynomials. Special choices of the matrices V and W above, with identity blocks, lead to efficient projections and have been shown to be effective in the context of black-box matrix inversion [24]. Even simpler matrices, $X = (I_m \ 0)^T$ and $Y = (0 \ I_m)^T$ in $\mathbb{K}^{n \times m}$ with $m \in \{1, \dots, n\}$, have been used by Villard in his fast algorithm for the bivariate resultant of two bivariate polynomials f and g in $\mathbb{K}[x, y]$ [75]. In this context, for generic f and g , this choice of X and Y is sufficient to ensure that the denominator matrix D contains m “small” polynomials in the ideal of $\mathbb{K}[x, y]$ generated by f and g .

1.5 Overview of the Core Algorithm

When $a \bmod f$ has a minimal polynomial μ_a of small degree, μ_a can be computed efficiently using power projections $(\ell(1), \ell(a), \ell(a^2), \dots)$ by Shoup’s algorithm, since few terms in the sequence are needed (see Sections 1.4.2 and 3.1.3). Then, for composition, one uses the identity $g(a) \equiv \hat{g}(a) \bmod f$, where $\hat{g} = g \bmod \mu_a$. Since \hat{g} has small degree, this reduces the number of powers of $a \bmod f$ that need be considered.

Our algorithm can be viewed as a block or bivariate version of this approach, *replacing the univariate polynomial μ_a by a collection of m small bivariate polynomials in the ideal generated by $y - a(x)$ and $f(x)$* , for a fixed parameter m . In a generic situation, while μ_a has degree n , there exists such a collection with degrees $m-1$ and $\lceil n/m \rceil$ in x and y . This collection is represented as a matrix in $\mathbb{K}[y]^{m \times m}$ and is found efficiently by exploiting the structure of the matrix of multiplication by $a \bmod f$.

Matrices of Relations. Let $M_a \in \mathbb{K}^{n \times n}$ be the matrix of multiplication by $a \bmod f$ in the basis $(1, x, \dots, x^{n-1})$. Following Section 1.4.3, in the special case where $A = M_a$, if V is a generic matrix in $\mathbb{K}^{n \times \ell}$, and W is the matrix $X = (I_m \ 0)^T$ with $m \leq \ell$ and $m \leq n$, the block Wiedemann approach yields a denominator matrix $D \in \mathbb{K}[y]^{m \times m}$ whose columns represent a basis of the $\mathbb{K}[y]$ -module

$$\mathcal{M}_m^{(a,f)} = \{r(x, y) = r_0(y) + \dots + r_{m-1}(y)x^{m-1} \mid r(x, a(x)) \equiv 0 \bmod f(x)\}; \quad (5)$$

this follows, for instance, from [74, Lemma 4.2]. The elements of this module are algebraic relations of degree less than m in x satisfied by $a \bmod f$ (Section 5.1). We call *matrix of relations* any

nonsingular matrix $R_m^{(a,f)} \in \mathbb{K}[y]^{m \times m}$ whose columns are the coefficients of polynomials in $\mathcal{M}_m^{(a,f)}$ (Section 4.1.1), that is, any nonsingular right multiple of D .

Given a matrix of relations $R_m^{(a,f)}$, the composition $g(a) \bmod f$ is obtained in two steps.

- First, by *polynomial matrix division* [43, Theorem 6.3-15, p. 389], there exist vectors $v, w \in \mathbb{K}[y]^m$ such that

$$(g(y) \ 0 \ \cdots \ 0)^\top = R_m^{(a,f)} w + v, \quad (6)$$

where $\deg(v) < d$ and d is an upper bound on $\deg(R_m^{(a,f)})$; finding such vectors takes $\tilde{O}(m^\omega(d + n/m))$ operations (Section 4.2) [80]. Then, by design, the bivariate polynomial

$$\tilde{g}(x, y) = v_1(y) + \cdots + v_m(y)x^{m-1}$$

has degree less than m and d in x and y , and is such that $g(a) \equiv \tilde{g}(x, a) \bmod f$.

- The polynomial \tilde{g} can then be evaluated at $y = a \bmod f$ by the Nüsken–Ziegler algorithm in $\tilde{O}(c(n, m, d))$ operations, with $c(\cdot)$ from Equation (2) (Proposition 3.4).

Truncated Sequence of Projections. In the block Wiedemann approach, using X as our right projection matrix, we need the first $2d$ elements of the matrix sequence $(VM_a^k X)_{k \geq 0}$, which amounts to a type of bivariate power projections (see Section 1.3.2). Unfortunately, we do not know how to obtain them efficiently enough for an arbitrary V . Choosing $V = X^\top$, we design a baby steps/giant steps algorithm in Section 3.3 that runs in $\tilde{O}(c(n, m, d) + m^2 d)$ operations. With this choice, by fraction reconstruction the sequence $(X^\top M_a^k X)_{k \geq 0}$ yields a denominator D that is a basis of the $\mathbb{K}[y]$ -module

$$\mathcal{M}_{m,m}^{(a,f)} = \left\{ r(x, y) \in \mathbb{K}[x, y]_{<(m, \cdot)} \mid [a(x)^k r(x, a(x)) \bmod f(x)]_0^{m-1} = 0 \text{ for all } k \geq 0 \right\},$$

where $[\cdot]_0^{m-1}$ is the projection on $\text{Span}(1, x, \dots, x^{m-1})$. The inclusion $\mathcal{M}_m^{(a,f)} \subseteq \mathcal{M}_{m,m}^{(a,f)}$ holds but may be strict, leading to a denominator D that is not a matrix of relations.

Matrices of Relations of Small Degree. For an arbitrary f with $f(0) \neq 0$ (this is not really a restriction, see Remark 3.8) and a generic a , two important properties hold (see Section 7.3): the above inclusion of modules is an equality—making the algorithm correct—and a basis $R_m^{(a,f)}$ of degree $d = \lceil n/m \rceil$ of $\mathcal{M}_m^{(a,f)}$ can be reconstructed from the first $2d$ elements of the sequence $(X^\top M_a^k X)_{k \geq 0}$ —making the algorithm fast.

The reconstruction is done via minimal approximant bases in Sections 5.2 and 5.3. Directly extending Section 1.4.3, we would solve the equation at infinity

$$\left(X^\top (yI_n - M_a)^{-1} X + O(y^{-2d-1}) \right) R_m^{(a,f)}(y) - N(y) = O(y^{-d-1}), \quad (7)$$

for unknown matrices N and $R_m^{(a,f)}$ of degree at most $d - 1$ and d . For technical reasons coming from the reconstruction algorithm, we actually use an expansion at $y = 0$ rather than at infinity, so that the sequence we use involves powers of M_a^{-1} instead of M_a (see Remark 5.7).

Beyond generic cases, a relevant quantity is

$$\nu_m^{(a,f)} = \deg(\sigma_1) + \cdots + \deg(\sigma_m), \quad (8)$$

where $\sigma_1, \dots, \sigma_n \in \mathbb{K}[y]$ are the invariant factors of $yI_n - M_a$, ordered by decreasing degree. This quantity is at most n , and it is the degree of the determinant of any basis of $\mathcal{M}_m^{(a,f)}$ (Proposition 4.1). In favorable situations, working with $d = \lceil \nu_m^{(a,f)}/m \rceil$, and *a fortiori* with $\lceil n/m \rceil$, is sufficient to obtain such a basis $\mathcal{M}_m^{(a,f)}$.

1.6 Probabilistic Algorithm for f Separable or Purely Inseparable

Our probabilistic algorithm aims at bringing arbitrary inputs to the favorable situation mentioned above, by means of a random change of basis. For a polynomial $\gamma \in \mathbb{K}[x]$ such that the minimal polynomial μ_γ of $\gamma \bmod f$ has degree n , the powers $(1, \gamma, \dots, \gamma^{n-1}) \bmod f$ form a basis of $\mathbb{A} = \mathbb{K}[x]/\langle f \rangle$. This induces a \mathbb{K} -algebra isomorphism:

$$\phi_\gamma : \mathbb{A} \rightarrow \mathbb{K}[y]/\langle \mu_\gamma \rangle$$

that maps γ to y , and more generally $u \in \mathbb{A}$ to v such that $v(y) \equiv u \bmod f$.

Using ϕ_γ allows us to transport our problem of modular composition to the right-hand side. For a in $\mathbb{K}[x]_{<n}$ and g in $\mathbb{K}[y]$, to find $g(a) \bmod f$, this boils down to the following (see Algorithm 8.1):

- a forward change of basis: through inverse modular composition, compute $\alpha \in \mathbb{K}[y]_{<n}$ such that $a = \alpha(\gamma) \bmod f$; this step also determines the minimal polynomial μ_γ ;
- a modular composition in the new basis: compute $\beta = g(\alpha) \bmod \mu_\gamma$;
- a backward change of basis: the modular composition $\beta(\gamma) \bmod f$, which equals $g(a) \bmod f$.

Computational Aspects. The second and third steps are modular compositions. They can be performed efficiently by the approach of Section 1.5, by finding and using matrices of relations $R_m^{(y,f)}$ and $R_m^{(\alpha,\mu_\gamma)}$, as long as certain genericity assumptions hold; this aspect is discussed below.

The first step, for the forward change of basis, is an instance of inverse modular composition and the calculation of a minimal polynomial. As mentioned in Section 1.4.2, Shoup's solutions recover both α and μ_γ from the power projections $(\ell(1), \ell(y), \dots, \ell(y^{2n-1}))$ and $(\ell(a), \ell(ya), \dots, \ell(y^{n-1}a))$, in the complexity of Brent and Kung's modular composition algorithm. Using matrices of relations we achieve a lower complexity, for a generic γ , as follows.

- (1) *Matrix of relations and minimal polynomial.* Generalizing the power projections of γ , the algorithm of Section 1.5 computes the first $2d$ terms of $(X^T M_\gamma^k X)_{k \geq 0}$, where $d = \lceil n/m \rceil$, and then reconstructs a basis $R_m^{(y,f)}$ of $\mathcal{M}_m^{(y,f)}$ by solving Equation (7) (with γ instead of a). This basis gives in particular the minimal polynomial μ_γ , which appears as an entry of the Hermite normal form of this basis (Proposition 4.1).
- (2) *Bivariate inverse composition.* The use of projections $(\ell(a), \ell(ya), \dots, \ell(y^{n-1}a))$ is directly generalized by computing the first $2d$ terms of $(X^T M_\gamma^k M_a \mathbf{1})_{k \geq 0}$, where $\mathbf{1}$ is the first column of X , and solving

$$\left(X^T (yI_n - M_\gamma)^{-1} X + O(y^{-2d-1}) \right) v_{\tilde{\alpha}}(y) - v_N(y) = X^T (yI_n - M_\gamma)^{-1} M_a \mathbf{1} + O(y^{-d-1}) \quad (9)$$

for polynomial vectors v_N and $v_{\tilde{\alpha}}$ in $\mathbb{K}[y]^m$ of degree less than d ; the entries of the vector $v_{\tilde{\alpha}}$ are the coefficients of a bivariate polynomial $\tilde{\alpha}(x, y)$ of small degree such that $\tilde{\alpha}(x, y) \equiv a \bmod f$.

As for Equation (9), we actually work with an expansion at $y = 0$ rather than infinity.

- (3) *Bivariate $\tilde{\alpha}$ to univariate α .* The situation is now symmetric to that of the composition algorithm of Section 1.5: we consider again Equation (6), where now g is unknown (it is α), v is known (it is $\tilde{\alpha}(x, y)$) and both $R_m^{(a,f)}$ and v have degree at most d , so that the polynomial matrix problem can be solved in $\tilde{O}(m^\omega(d + n/m))$ operations.

This approach is detailed in Algorithm CHANGEOFBASIS, with the steps reordered and combined so as to retrieve both $R_m^{(y,f)}$ and $v_{\tilde{\alpha}}$ from a single fraction reconstruction.

Probabilistic Aspects. For a generic γ , one has $\deg(\mu_\gamma) = n$, so the isomorphism ϕ_γ is well defined. Using the Schwartz–Zippel lemma, it is straightforward to control the probability of having $\deg(\mu_\gamma) < n$.

For generic γ , we can then follow the approach described in Section 1.5 to perform the last step, modular composition by γ , with the desired complexity. The quantitative aspects can be worked out as well, and similar considerations hold for the first step, inverse modular composition by γ .

However, the composition in the second step, $g(\alpha) \text{ rem } \mu_\gamma$, is more delicate to analyze. We need the equality of modules $\mathcal{M}_m^{(\alpha, \mu_\gamma)} = \mathcal{M}_{m,m}^{(\alpha, \mu_\gamma)}$, and that a matrix of relations in this module can be reconstructed from the first $2\lceil v_m^{(\alpha, \mu_\gamma)} / m \rceil \leq 2\lceil n/m \rceil$ elements of the corresponding matrix sequence; the analysis is made difficult by the fact that both α and μ_γ are nonlinear functions of the random element γ .

We prove that this happens for a generic γ in two cases: when f is *separable* in Section 8.3, and when f is *purely inseparable*, with extra conditions, in Section 8.4; the latter case covers power series composition with $f = x^n$. In both situations, there is a nonzero polynomial Δ in n variables such that the constraints above hold if Δ does not vanish at the coefficients of γ . We choose a random γ , and the probability of failure is again bounded by the Schwartz–Zippel lemma.

We do not have a proof that a generic γ satisfies our requirements for an arbitrary f . Our algorithm for the general case proceeds by reduction to the two extreme cases above, separable and purely inseparable polynomials.

From Monte Carlo to Las Vegas. At this stage, we have a probabilistic algorithm of Monte Carlo type, that runs in the announced complexity and returns the correct result with a controlled probability of error. The next question is to modify the algorithm so that it detects and reports the unlucky choices of γ for which its result would be incorrect.

In order to certify the result obtained for a random choice of $\gamma \in \mathbb{A}$, it would be sufficient to check the following properties:

- (1) the computed matrix $R_m^{(\gamma, f)}$ is a basis of relations of $\mathcal{M}_m^{(\gamma, f)}$;
- (2) the minimal polynomial of γ modulo f has degree n ;
- (3) the computed matrix $R_m^{(\alpha, \mu_\gamma)}$ is a basis of relations of $\mathcal{M}_m^{(\alpha, \mu_\gamma)}$.

However, we do not know how to check that all the columns of a matrix belong to the ideal $\langle f(x), y - \gamma(x) \rangle$ or $\langle \mu_\gamma(x), y - \alpha \rangle$ in sufficiently low complexity and in a deterministic way. The matrix $R_m^{(\gamma, f)}$ is easier to deal with: as it is expected to behave like in the generic case, its expected degree structure is known and the matrix can be certified by degree considerations (Item (ii) of Proposition 5.4, and Proposition 6.1). From there, the minimal polynomial of γ can be computed efficiently via the Hermite normal form of $\mathcal{M}_m^{(\gamma, f)}$, and it remains to check that it has degree n .

The other matrix, $R_m^{(\alpha, \mu_\gamma)}$, carries more information about α and cannot be expected to behave as predictably as $R_m^{(\gamma, f)}$. Our approach is to extract from its columns two small degree polynomials r and s in $\mathbb{K}[x, y]$. Since only two such polynomials are considered, they can be checked to vanish at $\alpha \bmod \mu_\gamma$ by the Nüsken–Ziegler algorithm without affecting the asymptotic cost. Then, these two polynomials are used to construct a Sylvester matrix that can be used for composition instead of $R_m^{(\alpha, \mu_\gamma)}$, without increasing the overall complexity (Algorithm 5.2).

Note. Equivalently, the randomization of our probabilistic algorithm can be seen as a change of projection. Indeed, let $P \in \mathbb{K}^{n \times n}$ have its j th column formed by the coefficients of $\gamma^{j-1} \text{ rem } f$. If $\gamma \bmod f$ generates $\mathbb{K}[x]/\langle f \rangle$ and M_α is the matrix of multiplication by $\alpha \bmod \mu_\gamma$ with basis

$(1, y, \dots, y^{n-1})$, then the multiplications by α and by a are related by

$$M_\alpha = P^{-1} M_a P. \quad (10)$$

Hence

$$X^T M_\alpha^k X = (X^T P^{-1}) M_a^k (P X),$$

which, for instance, on the right side, leads to considering the first m columns of P instead of X for projecting. This amounts to kinds of structured projections $(V^T M_a^k W)_{k \geq 0}$, i.e. with matrices V and W in a special proper subset of $\mathbb{K}^{n \times m}$.

1.7 Algorithm for the General Case

The algorithm of Section 1.6 is proved to work when f is either separable, or purely inseparable (for the latter, with extra conditions that are dealt with in Section 8.4). In Section 9, we address the general case, by first computing a separable decomposition of f [57], yielding a factorization into a product into pairwise coprime terms of the form $h_i(x^{p^{e_i}})^{\ell_i}$, with h_i separable and e_i, ℓ_i integers (here, p is the characteristic of \mathbb{K}).

Working modulo each factor separately, we are thus left with the question of composition modulo a polynomial of the form $h(x^{p^e})^\ell$, with h separable (all such results are eventually recombined via the Chinese remainder theorem).

For a modulus of the form $h(x)^\ell$, van der Hoeven and Lecerf showed how composition can be reduced to ℓ compositions modulo h , the computation of an annihilating polynomial modulo h , and a power series composition at precision ℓ with coefficients in $\mathbb{L} = \mathbb{K}[x]/\langle h(x) \rangle$ [34]. We extend this result to the case of moduli of the form $h(x^{p^e})^\ell$ in Section 9.4, involving essentially the same steps. The first two operations (compositions and annihilating polynomial modulo h) are directly handled by our results so far, but this is not quite the case for the latter, power series composition with coefficients in \mathbb{L} .

Our algorithms are written assuming they work over a field, as they perform zero-tests and inversions (compare this with Brent and Kung's algorithms, for instance, which apply over a ring). If h is irreducible, \mathbb{L} is a field, but if h is only assumed to be separable, then \mathbb{L} is only a product of fields. The *dynamic evaluation* paradigm [21] explains how an algorithm written for inputs lying in a field can carry over to inputs in a product of fields, but the original approach induces cost overheads that go beyond our cost target. Using van der Hoeven and Lecerf's efficient dynamic evaluation strategy [37], we show how our algorithm for power series adapts to this situation (Section 9.2) without affecting the asymptotic runtime.

1.8 Outline

Section 2 introduces some notation and our computational model. Section 3 details baby steps/giant steps techniques used in our composition algorithm: known ones such as in Brent and Kung's composition, and new ones such as for computing truncated powers, which give access to $(X^T M_a^k X)_{k \geq 0}$. Section 4 studies matrices of relations and how they are used in our composition algorithm, whereas Section 5 shows how to compute them efficiently by matrix fraction reconstruction under some assumptions on (f, a, m) . Section 6 presents an algorithm for the change of basis of Section 1.6: it finds the minimal polynomial μ_y and an inverse composition α such that $\alpha(y) \equiv a \pmod{f}$, under assumptions on (f, y, m) . Section 7 studies these assumptions, and in particular gives precise generic situations where they hold. Section 8 describes our main randomized composition algorithm and proves its correctness for a separable f and for a purely inseparable f (generalizing $f = x^n$); then Section 9 handles the general case of composition modulo any f . Finally, in Section 10, we state resulting complexity improvements for several variants of modular composition and other related problems.

2 PRELIMINARIES

Notation. In this article, \mathbb{K} is an arbitrary field. For bivariate polynomials in variables x and y , \deg_x and \deg_y give the degree in x and in y . For any polynomial or power series $p = \sum_i p_i x^i$, we use the following notation for a “slice” of it: $[p]_j^k = p_j + p_{j+1}x + \cdots + p_{j+k}x^k$. The ideal generated by polynomials f_1, \dots, f_k in an ambient ring (which will be clear from the context) is denoted by $\langle f_1, \dots, f_k \rangle$.

Vectors, such as elements of \mathbb{K}^m or $\mathbb{K}[y]^m$, are seen as column vectors by default; when row vectors are considered this is explicit in our notation, e.g., $\mathbb{K}^{1 \times m}$ or $\mathbb{K}[y]^{1 \times m}$. We often identify a polynomial $g_0(y) + \cdots + g_{m-1}(y)x^{m-1}$ in $\mathbb{K}[x, y]_{<(m, \cdot)}$ with the column vector $(g_0 \ \cdots \ g_{m-1})^\top$ in $\mathbb{K}[y]^m$ of its coefficients on the basis $(1, x, \dots, x^{m-1})$ of the $\mathbb{K}[y]$ -module $\mathbb{K}[x, y]_{<(m, \cdot)}$.

For a and f in $\mathbb{K}[x]$, M_a denotes the matrix of the linear map of multiplication by a in $\mathbb{K}[x]/\langle f(x) \rangle$ with basis $(1, x, \dots, x^{n-1})$, and μ_a , resp. χ_a , denotes the minimal, resp. characteristic polynomial of a in $\mathbb{K}[x]/\langle f(x) \rangle$ (that is, the minimal and characteristic polynomials of M_a).

Whenever the context is sufficiently clear, particularly in Sections 4, 5 and 7, notation such as $\mathcal{M}_m^{(a,f)}$, $v_m^{(a,f)}$ defined in the introduction is shortened into \mathcal{M}_m , v_m . We keep the superscripts in important statements.

Computational Model. Our algorithms are written in pseudocode, using standard syntax elements (for loops, if statements, …). Informally, we count all arithmetic operations $\{+, -, \times, \div\}$ and zero-tests in \mathbb{K} at unit cost. The underlying complexity model is the *computation tree* [19, Section 4.4].

A computation tree over \mathbb{K} is a binary tree whose nodes are partitioned into *input nodes* that form an initial segment of the tree starting at the root, *computation nodes* with outdegree 1, *branching nodes* with outdegree 2 and *output nodes* at the leaves. To each node is associated a label. Computation nodes are labeled by constants in \mathbb{K} or operations in $\{+, -, \times, \div\}$, in which case they also carry references to two previous input or computation nodes; branching nodes are labeled by zero-tests, referring to some previously computed quantity. Each leaf v is labeled with a sequence of references $(u_1, \dots, u_{\ell(v)})$ to previous input or computation nodes. The *cost* of a computation tree is its height τ , that is, the maximum length of a path from the root to a leaf.

It then makes sense to *evaluate* a computation tree at an element of \mathbb{K}^s – called *input to the tree*, where s is the number of input nodes, following a path from the root to a leaf. After the input nodes, the path is constructed as follows. Each computation node is assigned a value derived from the label it carries, when it is defined. Otherwise, e.g., in case of a division by 0, the path stops. At a branching node the path branches left or right depending on whether the value it refers to vanishes or not. At a leaf v with label $(u_1, \dots, u_{\ell(v)})$, the output of the computation is the tuple of the values computed at nodes $u_1, \dots, u_{\ell(v)}$. In that case, the computation tree is called *evaluable* at the input. Overall, the computation requires at most τ arithmetic operations in \mathbb{K} . An algorithm is called *quasi-linear* when the height of its computation tree is linear (up to logarithmic factors) in the number of inputs. It is called *quasi-optimal* when this height is linear (up to logarithmic factors) in the number of inputs *plus* the maximum number of values returned by the output nodes.

A computation tree takes inputs of fixed length. In order to solve a problem for inputs of arbitrary size and characteristic, we need a *family* of trees, parametrized by the input size and the characteristic. Every algorithm we describe using pseudocode in this article, and all algorithms that we rely on from the literature, can be described by a family of computation trees.

The translation from pseudocode to computation tree is usually rather direct, and as is customary in the literature, we do not do it explicitly. In a nutshell, for loops and recursive calls are “unrolled”; if statements that test whether a computed quantity vanishes yield branching nodes, and so on. Some operations in our pseudocode may not be directly available in our model (as

we only allow arithmetic operations in \mathbb{K} and zero-test), but they can be rewritten in a way that complies with our requirements. This is, for instance, the case when we compute the degree of a polynomial (as in Euclid's GCD algorithm): this can be achieved by scanning its coefficients, in order of decreasing degree, until a nonzero one is found. We also invoke a result by van der Hoeven and Lecerf [37] on the transformation of computation trees for *directed evaluation* in Section 9.2; the translation from pseudocode to tree also applies to their algorithm.

The families of trees that we build for modular composition with arbitrary degree and characteristic are *uniform*, in the sense that an appropriate tree description is generated from the pseudocode and any given degree n and characteristic p .

We allow our algorithms to return flags (such as FAIL, or CERT/NoCERT). This can be done in this model, by returning constants in the vector of outputs, such as 1 for FAIL and 0 otherwise.

Finally, several of our algorithms rely on randomization; however, we do not want to introduce another arithmetic operation for the selection of random field elements. One reason for this is that the result by van der Hoeven and Lecerf [37] mentioned above is explicitly written in a deterministic model. Instead, “random” field elements are given to our procedures as extra input parameters.

3 SIMULTANEOUS MODULAR OPERATIONS BY MATRIX MULTIPLICATION

A key ingredient in fast modular composition algorithms is to turn the problem into the simultaneous evaluation of polynomials of smaller degree, and exploit the structure brought by this simultaneity using matrix multiplication. In this section, after reviewing Brent and Kung's original algorithm and giving a direct extension of it, we use this idea in two further contexts: Nüsken and Ziegler's bivariate modular composition algorithm, and the computation of truncations of powers of the form $a^k \bmod f$. Both arise in our algorithms, and are bottlenecks in their complexity.

3.1 Brent and Kung's Algorithm

3.1.1 Modular Composition. We start with a review of Brent and Kung's algorithm to compute $g(a) \bmod f$, pointing out the impact of rectangular matrix multiplication [41] and how the runtime depends on the degrees of both f and g [69, Fact 3.1]. This can be seen as an introduction to the Nüsken–Ziegler algorithm, which generalizes this approach to a bivariate g .

PROPOSITION 3.1. *Given polynomials $f \in \mathbb{K}[x]$ of degree n , a in $\mathbb{K}[x]_{<n}$ and g in $\mathbb{K}[y]_{<d}$, Algorithm **MODULARCOMPOSITION-BRENTKUNG** computes $g(a) \bmod f$ using $\tilde{O}((1+n/d)d^{\omega_2/2})$ operations in \mathbb{K} .*

PROOF. Correctness follows from noticing that at Step 7, $b_i \equiv g_{ir} + g_{ir+1}a + \dots + g_{ir+r-1}a^{r-1} \pmod{f}$ holds for all i , where g_j is the coefficient of degree j in g for all j . The cost of the algorithm comes from $\Theta(d^{1/2})$ multiplications modulo f , which use $\tilde{O}(nd^{1/2})$ operations in \mathbb{K} , and a matrix product in sizes $s \times r$ and $r \times n$, with both s and r in $\Theta(d^{1/2})$. This product can be done through $\lceil n/d \rceil \leq n/d+1$ matrix products in sizes $s \times r$ and $r \times d$, each of which takes $O(d^{\omega_2/2})$ operations in \mathbb{K} . \square

Note. In the analysis, dividing the matrix product into blocks, as we did, is suboptimal. Using rectangular matrix multiplication directly, the runtime can be described by the finer estimate $\tilde{O}(d^{\omega_2 \log(n)/\log(d)}/2)$. Here, the notation ω_θ is a feasible exponent for rectangular matrix multiplication for any real number θ : there is an algorithm that multiplies an $n \times \lceil n^\theta \rceil$ matrix by an $\lceil n^\theta \rceil \times n$ matrix using $O(n^{\omega_\theta})$ operations [56]. However, this refinement complicates notation, and would not be of use for our main results. The same remark holds for several other runtime estimates in this section, such as Lemmas 3.3 and 3.5.

ALGORITHM 3.1 MODULARCOMPOSITION-BRENTKUNG(f, a, g)

Input: f of degree n in $\mathbb{K}[x]$, a in $\mathbb{K}[x]_{<n}$, g in $\mathbb{K}[y]_{<d}$

Output: $g(a) \text{ rem } f$

```

1:  $r \leftarrow \lceil d^{1/2} \rceil$ ,  $s \leftarrow \lceil d/r \rceil$ 
2:  $\hat{a}_0 \leftarrow 1$ 
3: for  $i = 1, \dots, r$  do  $\hat{a}_i \leftarrow a \cdot \hat{a}_{i-1} \text{ rem } f$  ▷  $\hat{a}_i = a^i \text{ rem } f$ 
4:  $A \leftarrow \text{matrix } (\text{coeff}(\hat{a}_i, j))_{\substack{0 \leq i < r \\ 0 \leq j < n}} \text{ in } \mathbb{K}^{r \times n}$  ▷ coefficient of degree  $j$  of  $\hat{a}_i$ 
5:  $G \leftarrow \text{matrix } (\text{coeff}(g, ir + j))_{\substack{0 \leq i < s \\ 0 \leq j < r}} \text{ in } \mathbb{K}^{s \times r}$ 
6:  $B = (b_{i,j})_{\substack{0 \leq i < s \\ 0 \leq j < n}} \leftarrow GA \text{ in } \mathbb{K}^{s \times n}$ 
7: for  $i = 0, \dots, s-1$  do  $b_i \leftarrow b_{i,0} + \dots + b_{i,n-1}x^{n-1}$ 
8: return  $b_0 + b_1\hat{a}_r + \dots + b_{s-1}\hat{a}_r^{s-1} \text{ rem } f$  ▷ Horner evaluation

```

ALGORITHM 3.2 POWERPROJECTION($f, a, d, (r_i)_{0 \leq i < n}$)

Input: f of degree n in $\mathbb{K}[x]$, a in $\mathbb{K}[x]_{<n}$, d in \mathbb{N} , $(r_i)_{0 \leq i < n}$ in \mathbb{K}^n

Output: $(\ell(1), \ell(a), \dots, \ell(a^{d-1} \bmod f))$, with $\ell(b_0 + \dots + b_{n-1}x^{n-1}) = r_0b_0 + \dots + r_{n-1}b_{n-1}$

3.1.2 Power Projection. The transposition principle implies the existence of an algorithm **POWERPROJECTION** with the same asymptotic runtime as Algorithm **MODULARCOMPOSITION-BRENTKUNG** and with the following signature [69].

Whereas seeing the details of Algorithm **MODULARCOMPOSITION-BRENTKUNG** is useful as a preamble to the Nüsken–Ziegler algorithm, Algorithm **POWERPROJECTION** only plays the role of a subroutine in one other algorithm given just below. Moreover, giving its pseudocode would require us to introduce concepts such as transposed product, that would not be used any further in this text. We refer the reader to Reference [71], which gives all details but uses classical matrix arithmetic (with $\omega_2 = 4$), so the runtime of that version is $\tilde{O}(d^2 + nd)$ instead of $\tilde{O}((1 + n/d)d^{\omega_2/2})$.

3.1.3 Small Minimal Polynomial. Modular composition can be sped up when the minimal polynomial μ_a of a modulo f has degree at most d , for some (small) integer $d \leq n$. To compute $g(a) \text{ rem } f$, the idea is that once μ_a is known, $\tilde{g} = g \text{ rem } \mu_a$ can be computed, and then $g(a) \equiv \tilde{g}(a) \bmod f$ (see e.g. [35, Section 4.1]). The computation of the latter by Proposition 3.1 benefits from \tilde{g} having degree less than d .

It remains to discuss how to compute μ_a . Here, we follow an algorithm of Shoup (the deterministic version, for f irreducible, is in [69, Theorem 3.4]; the randomized one is in [70, Section 4]). We take a random linear form $\ell : \mathbb{K}[x]/\langle f \rangle \rightarrow \mathbb{K}$ and compute the sequence $(\ell(a^k \bmod f))_{0 \leq k < 2d}$. With high probability, its minimal polynomial is μ_a ; the algorithm verifies whether it is the case, and returns either a correct result or FAIL. In Algorithm **MODULARCOMPOSITION-SMALLMINIMALPOLYNOMIAL**, μ_a is computed using an Extended Euclidean scheme called **MINIMALPOLYNOMIALFORSEQUENCE** [26, Algorithm 12.9].

The following lemma analyses the runtime of this procedure, and the probability of success. As per our convention at the end of Section 2, the “random” linear form ℓ is actually given as an argument, through the vector $(r_0, \dots, r_{n-1}) \in \mathbb{K}^n$ of its coefficients.

LEMMA 3.2. *Given $f \in \mathbb{K}[x]$ of degree n , a in $\mathbb{K}[x]_{<n}$, g in $\mathbb{K}[y]_{<n}$, d in $\{1, \dots, n\}$ and $(r_i)_{0 \leq i < n}$ in \mathbb{K}^n , Algorithm **MODULARCOMPOSITION-SMALLMINIMALPOLYNOMIAL** uses $\tilde{O}(nd^{(\omega_2/2)-1})$ operations in \mathbb{K} and returns either $g(a) \text{ rem } f$ or FAIL. If μ_a has degree at most d , and the entries of $(r_i)_{0 \leq i < n}$ are chosen uniformly and independently from a finite subset S of \mathbb{K} , then with probability at least*

ALGORITHM 3.3 MODULARCOMPOSITION-SMALLMINIMALPOLYNOMIAL($f, a, g, d, (r_i)_{0 \leq i < n}$)

Input: f of degree n in $\mathbb{K}[x]$, a in $\mathbb{K}[x]_{<n}$, g in $\mathbb{K}[y]_{<n}$, d in $\{1, \dots, n\}$, $(r_i)_{0 \leq i < n}$ in \mathbb{K}^n

Output: $g(a) \bmod f$ or FAIL

```

1:  $(v_0, \dots, v_{2d-1}) \leftarrow \text{POWERPROJECTION}(f, a, 2d, (r_i)_{0 \leq i < n})$ 
2:  $\mu \leftarrow \text{MINIMALPOLYNOMIALFORSEQUENCE}(v_0, \dots, v_{2d-1})$ 
3:  $t \leftarrow \text{MODULARCOMPOSITION-BRENTKUNG}(f, a, \mu)$ 
4: if  $t \neq 0$  then return FAIL
5: else return  $\text{MODULARCOMPOSITION-BRENTKUNG}(f, a, g \bmod \mu)$ 

```

$1 - n/\text{card}(S)$ the algorithm returns $g(a) \bmod f$ and computes μ_a as a by-product. If μ_a has degree more than d , the algorithm returns FAIL.

PROOF. For any given a in $\mathbb{K}[x]_{<n}$ and (r_0, \dots, r_{n-1}) , the algorithm computes a polynomial μ and tests whether $\mu(a) \equiv 0 \pmod{f}$; if it is the case, it reduces g modulo μ before doing a modular composition. Hence, the output may only be $g(a) \bmod f$ or FAIL, as claimed; it is FAIL if and only if the value t at Step 3 does not vanish.

By the discussion in Section 3.1.2 and Proposition 3.1, the call to POWERPROJECTION takes $\tilde{O}((1+n/d)d^{\omega_2/2})$ operations in \mathbb{K} ; because we take $d \leq n$, this is $\tilde{O}(nd^{(\omega_2/2)-1})$. Step 2 then computes a nonzero annihilating polynomial of degree at most d in $\tilde{O}(d)$ operations in \mathbb{K} [26, Algorithm 12.9]. The remaining lines call Algorithm MODULARCOMPOSITION-BRENTKUNG with a last argument of degree at most d , so the cost is $\tilde{O}(nd^{(\omega_2/2)-1})$ again; this establishes the claim on the runtime.

Suppose first that μ_a has degree greater than d . Then since $\deg(\mu) \leq d$, $\mu(a) \bmod f$ cannot be zero, so the output is FAIL, as claimed.

Finally, suppose that the minimal polynomial μ_a has degree at most d and that the entries of $(r_i)_{0 \leq i < n}$ are chosen uniformly at random and independently from a set S in \mathbb{K} . With M_a the multiplication matrix by $a \bmod f$ and $\mathbf{1}$ the vector $(1, 0, \dots, 0)$, the sequence $(M_a^k \mathbf{1})_{k \geq 0}$ is $(a^k \bmod f)_{k \geq 0}$ and the sequence $((r_i)^\top M_a^k \mathbf{1})_{k \geq 0}$ is $(\ell(a^k \bmod f))_{k \geq 0}$. Following the probabilistic analysis of Wiedemann's algorithm [46, Lemma 2; 47, Lemma 1], the probability that their minimal polynomials coincide is at least $1 - n/\text{card}(S)$. When this occurs, Step 2 computes μ_a ; the value t at Step 3 is then zero, and the output is $g(a) \bmod f$. \square

The main idea in this algorithm—computing an annihilating polynomial for a and using it to reduce g —is actually at the core of our main algorithm as well. Key differences are that we compute several annihilating polynomials (which we call relations), and use them to reduce g into a bivariate polynomial. We then apply Nüsken and Ziegler's extension of Brent and Kung's algorithm, which we present now.

3.2 Bivariate Composition

Here we describe the Nüsken–Ziegler algorithm for modular composition [63], which computes $g(x, a) \bmod f$ for a bivariate g in $\mathbb{K}[x, y]$.

First, however, we address the following question: given f of degree n in $\mathbb{K}[x]$, a in $\mathbb{K}[x]_{<n}$ and an s -tuple (g_0, \dots, g_{s-1}) in $\mathbb{K}[x, y]_{<(m, r)}^s$, compute all compositions

$$(g_0(x, a) \bmod f, \dots, g_{s-1}(x, a) \bmod f) \in \mathbb{K}[x]^s.$$

The solution designed by Nüsken and Ziegler [63] boils down to a multiplication of polynomial matrices. Writing the polynomials g_i as the rows of their coefficients in y gives an $s \times r$ matrix G whose entries are polynomials in $\mathbb{K}[x]_{<m}$. Writing the powers of $1, a, \dots, a^{r-1} \bmod f$ in a column vector A reduces the simultaneous composition to a matrix–vector product GA . This is turned into

ALGORITHM 3.4 SIMULTANEOUSBIVARIATEMODULARCOMPOSITION($f, a, g_0, \dots, g_{s-1}, m, r$)

Input: f of degree n in $\mathbb{K}[x]$, a in $\mathbb{K}[x]_{<n}$, (g_0, \dots, g_{s-1}) in $\mathbb{K}[x, y]_{<(m, r)}^s$

Output: $(g_0(x, a) \text{ rem } f, \dots, g_{s-1}(x, a) \text{ rem } f)$

- 1: $\hat{a}_0 \leftarrow 1$
- 2: **for** $i = 1, \dots, r-1$ **do** $\hat{a}_i \leftarrow a \cdot \hat{a}_{i-1} \text{ rem } f$ $\triangleright \hat{a}_i = a^i \text{ rem } f$
- 3: $A \leftarrow \text{matrix } ([\hat{a}_i]_{jm}^{m-1})_{\substack{0 \leq i < r \\ 0 \leq j < \lceil n/m \rceil}} \text{ in } \mathbb{K}[x]_{<m}^{r \times \lceil n/m \rceil}$
- 4: $G \leftarrow \text{matrix } (g_{i,j}(x))_{\substack{1 \leq i \leq s \\ 0 \leq j < r}} \text{ in } \mathbb{K}[x]_{<m}^{s \times r}, \text{ where } g_i(x, y) = \sum_j g_{i,j}(x)y^j$
- 5: $B = (B_{i,j})_{\substack{0 \leq i < r \\ 0 \leq j < \lceil n/m \rceil}} \leftarrow GA$
- 6: **for** $i = 0, \dots, s-1$ **do** $b_i \leftarrow (\sum_{0 \leq j < \lceil n/m \rceil} B_{i,j}x^{jm}) \text{ rem } f$
- 7: **return** (b_0, \dots, b_{s-1})

a matrix-matrix product by spreading the coefficients of A as follows. If

$$g_i(x, y) = \sum_{0 \leq j < r} g_{i,j}(x)y^j,$$

then computing the product

$$B = \begin{pmatrix} g_{0,0}(x) & \cdots & g_{0,r-1}(x) \\ \vdots & & \vdots \\ g_{s-1,0}(x) & \cdots & g_{s-1,r-1}(x) \end{pmatrix} \begin{pmatrix} [a^0 \text{ rem } f]_0^{m-1} & \cdots & [a^0 \text{ rem } f]_{(\lceil n/m \rceil - 1)m}^{m-1} \\ \vdots & & \vdots \\ [a^{r-1} \text{ rem } f]_0^{m-1} & \cdots & [a^{r-1} \text{ rem } f]_{(\lceil n/m \rceil - 1)m}^{m-1} \end{pmatrix}$$

yields a matrix whose entry $B_{i,\ell}$ is

$$B_{i,\ell} = \sum_{0 \leq j < r} g_{i,j}(x)[a^j \text{ rem } f]_{\ell m}^{m-1}.$$

Summing the $B_{i,\ell}x^{\ell m}$ modulo f , for $\ell = 0, \dots, \lceil n/m \rceil - 1$, then provides $g_i(x, a) \text{ rem } f$ for $i = 0, \dots, s-1$ at low cost. This is detailed in Algorithm 3.4 and Lemma 3.3.

LEMMA 3.3 ([63, LEMMA 10(iii)]). *Algorithm SIMULTANEOUSBIVARIATEMODULARCOMPOSITION computes $(g_0(x, a) \text{ rem } f, \dots, g_{s-1}(x, a) \text{ rem } f)$. Assuming $s \in \tilde{O}(r)$, it uses $\tilde{O}(c(n, m, r^2)) = \tilde{O}((m + n/r^2)r^{\omega_2})$ operations in \mathbb{K} , with $c(\cdot)$ from Equation (2).*

PROOF. Steps 1 and 2 use $\tilde{O}(rn)$ operations. Similarly, for each $i = 0, \dots, s-1$, Step 6 uses $\lceil n/m \rceil$ additions in $O(m)$ operations each and one reduction in $\tilde{O}(m+n)$ operations. The total cost of Step 6 is thus $\tilde{O}(s(n+m))$.

Steps 3 and 4 do not use any arithmetic operation. The most expensive step is Step 5, the product of an $s \times r$ matrix by an $r \times \lceil n/m \rceil$ matrix, both with entries in $\mathbb{K}[x]_{<m}$. Using the same kind of block decomposition as in Proposition 3.1, this is done using $\lceil \lceil n/m \rceil/r^2 \rceil \in n/(mr^2) + O(1)$ products in sizes $s \times r$ and $r \times r^2$. With the assumption $s \in \tilde{O}(r)$, each of them uses $\tilde{O}(mr^{\omega_2})$ operations in \mathbb{K} , for a total of $\tilde{O}(c(n, m, r^2)) = \tilde{O}((m + n/r^2)r^{\omega_2})$ operations in \mathbb{K} .

The other steps, in $\tilde{O}((r+s)(n+m)) = \tilde{O}(rm+rn)$, are at most of the same order, since $\omega_2 \geq 3$. \square

Algorithm SIMULTANEOUSBIVARIATEMODULARCOMPOSITION is the central step in bivariate composition as showed in Algorithm BIVARIATEMODULARCOMPOSITION, leading to the complexity stated in Proposition 3.4.

ALGORITHM 3.5 BIVARIATEMODULARCOMPOSITION(f, a, g) (Nüsken-Ziegler algorithm [63])

Input: f of degree n in $\mathbb{K}[x]$, a in $\mathbb{K}[x]_{<n}$, g in $\mathbb{K}[x, y]_{<(m, d)}$

Output: $g(x, a) \bmod f$

- 1: $r \leftarrow \lceil d^{1/2} \rceil$, $s \leftarrow \lceil d/r \rceil$
- 2: Write $g(x, y) = g_0(x, y) + g_1(x, y)y^r + \cdots + g_{s-1}(x, y)y^{r(s-1)}$ with $\deg_y(g_i) < r$ for $0 \leq i < s$
- 3: $(b_0, \dots, b_{s-1}) \leftarrow \text{SIMULTANEOUSBIVARIATEMODULARCOMPOSITION}(f, a, g_0, \dots, g_{s-1}, m, r)$
 - ▷ $b_i = g_i(x, a) \bmod f$
- 4: $\hat{a} \leftarrow a^r \bmod f$ ▷ is computed in the previous step
- 5: **return** $b_0 + b_1\hat{a} + \cdots + b_{s-1}\hat{a}^{s-1}$ ▷ Horner evaluation

PROPOSITION 3.4 ([63, THEOREM 9]). *Given $f \in \mathbb{K}[x]$ of degree n , $a \in \mathbb{K}[x]_{<n}$, g in $\mathbb{K}[x, y]_{<(m, d)}$, Algorithm **BIVARIATEMODULARCOMPOSITION** computes $g(x, a) \bmod f$ using $\tilde{O}(c(n, m, d)) = \tilde{O}((m + n/d)d^{\omega_2/2})$ operations in \mathbb{K} , with $c(\cdot)$ from Equation (2).*

PROOF. The correctness of the algorithm is straightforward. For the complexity analysis, we first note that $s \sim r \sim d^{1/2}$. Lemma 3.3 then shows that the complexity of Step 3 is $\tilde{O}(c(n, m, r^2)) = \tilde{O}(c(n, m, d))$. The other task involving arithmetic operations is the final Horner evaluation which costs $\tilde{O}(rn)$. As in the proof of Lemma 3.3, this is smaller than the other part, since $\omega_2 \geq 3$. \square

3.3 Sequence of Truncated Modular Powers

Another key ingredient in our composition algorithm also relies on polynomial matrix multiplication. To our knowledge this is a new algorithm, whose properties are summarized in the next lemma.

LEMMA 3.5. *Given f of degree n in $\mathbb{K}[x]$, (p_0, \dots, p_{r-1}) in $\mathbb{K}[x]_{<n}^r$, (q_0, \dots, q_{s-1}) in $\mathbb{K}[x]_{<n}^s$ and $m \in \mathbb{N}_{>0}$, Algorithm **SIMULTANEOUSTRUNCATEDMODULARMULTIPLICATION** computes the simultaneous truncated modular multiplications*

$$\{[p_i q_j \bmod f]_0^{m-1} \mid 0 \leq i < r, 0 \leq j < s\}.$$

If $s \in \tilde{O}(r)$, it uses $\tilde{O}(c(n, m, r^2)) = \tilde{O}((m + n/r^2)r^{\omega_2})$ operations in \mathbb{K} , with $c(\cdot)$ from Equation (2).

The basic approach to this problem is to first compute all the products $p_i q_j$ modulo f and then truncate the computed polynomials. However, this produces an intermediate result of size nrs , which is $\Theta(nr^2)$ when s is in $\Theta(r)$, and is larger than our target complexity.

Hereafter, we use the reversal of a polynomial $p \in \mathbb{K}[x]$ with respect to $m \in \mathbb{N}$ defined by $\text{rev}(p, m) = x^m p(1/x)$; when $m = \deg(p)$ this is the classical reciprocal of the polynomial p .

PROOF. For all $i < r$ and $j < s$, let $p_i q_j = h_{i,j} f + r_{i,j}$, with $\deg(r_{i,j}) < n$, be the Euclidean division of the product $p_i q_j$ by the polynomial f . The main task of the algorithm is to compute the truncated quotients $[h_{i,j}]_0^{m-1}$ (Steps 1 to 5); from there, the truncated remainders $[r_{i,j}]_0^{m-1}$ are easily obtained (Step 6) at a total cost of $\tilde{O}(mrs)$ operations.

For the efficient computation of the quotients $h_{i,j}$, we rely on the classical approach via reciprocals and power series operations. More specifically we use the identity

$$\text{rev}(h_{i,j}, n-2) = \frac{\text{rev}(p_i, n-1) \text{rev}(q_j, n-1)}{\text{rev}(f, n)} \bmod x^{n-1} = \bar{p}_i \bar{q}_j \bmod x^{n-1},$$

obtained by evaluating $p_i q_j = h_{i,j} f + r_{i,j}$ at $1/x$ and multiplying by $x^{n-2}/f(1/x) = x^{2n-2}/\text{rev}(f, n)$; here we have $\bar{p}_i = \text{rev}(p_i, n-1)$ and $\bar{q}_j = \text{rev}(q_j, n-1)/\text{rev}(f, n) \bmod x^{n-1}$, as in the pseudocode.

ALGORITHM 3.6 SIMULTANEOUS TRUNCATED MODULAR MULTIPLICATION($f, (p_i)_{i < r}, (q_j)_{j < s}, m$)

Input: f of degree n in $\mathbb{K}[x]$, (p_0, \dots, p_{r-1}) in $\mathbb{K}[x]_{\leq n}^r$, (q_0, \dots, q_{s-1}) in $\mathbb{K}[x]_{\leq n}^s$, $m \in \mathbb{N}_{>0}$

Output: $([p_i q_j \bmod f]_0^{m-1})_{\substack{0 \leq i < r \\ 0 \leq j < s}}$

- 1: $(\ell, t) \leftarrow (\text{quotient, remainder})$ in the Euclidean division $n - m - 1 = \ell m + t$ with $\ell = 0$ if $m \geq n$
- 2: **for** $i = 0, \dots, r - 1$ **do** $\bar{p}_i \leftarrow \text{rev}(p_i, n - 1)$
- 3: **for** $j = 0, \dots, s - 1$ **do** $\bar{q}_j \leftarrow \text{power series expansion } \text{rev}(q_j, n - 1) / \text{rev}(f, n) \bmod x^{n-1}$
- 4: Form the matrices

$$P_1 \leftarrow ([\bar{p}_i]_{jm+t}^{m-1})_{\substack{0 \leq i < r \\ 0 \leq j \leq \ell}} \in \mathbb{K}[x]_{\leq m}^{r \times (\ell+1)} \quad P_2 \leftarrow ([\bar{p}_i]_{jm+t}^{m-1})_{\substack{0 \leq i < r \\ 0 \leq j < \ell}} \in \mathbb{K}[x]_{\leq m}^{r \times \ell}$$

(note that P_2 is the $r \times \ell$ left submatrix of P_1), and

$$Q_1 \leftarrow ([\bar{q}_j]_{(\ell-i)m}^{m-1})_{\substack{0 \leq i \leq \ell \\ 0 \leq j < s}} \in \mathbb{K}[x]_{\leq m}^{(\ell+1) \times s} \quad Q_2 \leftarrow ([\bar{q}_j]_{(\ell-i)m}^{m-1})_{\substack{0 \leq i < \ell \\ 0 \leq j < s}} \in \mathbb{K}[x]_{\leq m}^{\ell \times s}$$

- 5: $H \leftarrow [P_1 Q_1]_0^{m-1} + [P_2 Q_2]_m^{m-1} + \left([\bar{p}_i]_0^{t-1} [\bar{q}_j]_{\ell m+1}^{m-1+t-1} \right)_{t=1}^{m-1} \quad \triangleright H = (\bar{h}_{i,j})_{i,j} \text{ is in } \mathbb{K}[x]_{\leq m}^{r \times s}$
- 6: **for** $i = 0, \dots, r - 1$ and $j = 0, \dots, s - 1$ **do** $r_{i,j} \leftarrow (p_i q_j - \text{rev}(\bar{h}_{i,j}, m - 1)f) \bmod x^m$
- 7: **return** $(r_{i,j})_{\substack{0 \leq i < r \\ 0 \leq j < s}}$

The idea of our algorithm is to compute only the last m coefficients of this expansion by means of two polynomial matrix multiplications.

For any $t \in \{0, \dots, m - 1\}$, for any polynomials a, b written as

$$a = [a]_0^{t-1} + x^t \sum_{i \geq 0} a_i x^{im} \text{ with } \deg(a_i) < m, \quad b = \sum_{j \geq 0} b_j x^{jm} \text{ with } \deg(b_j) < m,$$

and for any positive integer ℓ , one has

$$[ab]_{\ell m+t}^{m-1} = \left[\sum_{i+j=\ell} a_i b_j \right]_0^{m-1} + \left[\sum_{i+j=\ell-1} a_i b_j \right]_m^{m-1} + [[a]_0^{t-1} [b]_{\ell m+1}^{m-1+t-1}]_0^{m-1}. \quad (11)$$

(The last summand is a product of small degree polynomials that is 0 when $t = 0$.) We use this formula with ℓ and t as defined in Step 1, so that the left-hand side is $[ab]_{n-m-1}^{m-1}$; applying this to $a = \bar{p}_i$ and $b = \bar{q}_j$ gives $\bar{h}_{i,j} = [\text{rev}(h_{i,j}, n - 2)]_{n-m-1}^{m-1}$, and thus $[\bar{h}_{i,j}]_0^{m-1}$ by reversal.

Since $\ell \sim n/m$, using this formula for a single pair i, j requires $\tilde{O}(n)$ operations in \mathbb{K} and thus is as costly as computing $\bar{p}_i \bar{q}_j \bmod x^n$. In our algorithm, the gain comes from using this formula simultaneously for several products, in which case matrix multiplication helps.

The first multiplication in Step 5 is the matrix product

$$\begin{pmatrix} [\bar{p}_0]_t^{m-1} & \cdots & [\bar{p}_0]_{n-m-1}^{m-1} \\ \vdots & & \vdots \\ [\bar{p}_{r-1}]_t^{m-1} & \cdots & [\bar{p}_{r-1}]_{n-m-1}^{m-1} \end{pmatrix} \begin{pmatrix} [\bar{q}_0]_{\ell m}^{m-1} & \cdots & [\bar{q}_{s-1}]_{\ell m}^{m-1} \\ \vdots & & \vdots \\ [\bar{q}_0]_0^{m-1} & \cdots & [\bar{q}_{s-1}]_0^{m-1} \end{pmatrix}.$$

Its entries are the first summand in Equation (11) for $a = \bar{p}_i$ and $b = \bar{q}_j$, for $0 \leq i < r$ and $0 \leq j < s$. Similarly, the second summand in Equation (11) is obtained from the matrix product

$$\begin{pmatrix} [\bar{p}_0]_t^{m-1} & \cdots & [\bar{p}_0]_{n-2(m-1)}^{m-1} \\ \vdots & & \vdots \\ [\bar{p}_{r-1}]_t^{m-1} & \cdots & [\bar{p}_{r-1}]_{n-2(m-1)}^{m-1} \end{pmatrix} \begin{pmatrix} [\bar{q}_0]_{(\ell-1)m}^{m-1} & \cdots & [\bar{q}_{s-1}]_{(\ell-1)m}^{m-1} \\ \vdots & & \vdots \\ [\bar{q}_0]_0^{m-1} & \cdots & [\bar{q}_{s-1}]_0^{m-1} \end{pmatrix}.$$

ALGORITHM 3.7 TRUNCATEDPOWERS(f, a, b, m, d)

Input: f of degree n in $\mathbb{K}[x]$, a and b in $\mathbb{K}[x]_{<n}$, m and d in $\mathbb{N}_{>0}$

Output: the truncated powers $[ba^k \text{ rem } f]_0^{m-1}$ for $0 \leq k < d$

```

1:  $r \leftarrow \lceil d^{1/2} \rceil$ ;  $s \leftarrow \lceil d/r \rceil$ 
2:  $\hat{a}_0 \leftarrow 1$ ; for  $i = 1, \dots, r$  do  $\hat{a}_i \leftarrow a \cdot \hat{a}_{i-1} \text{ rem } f$  ▷  $\hat{a}_i = a^i \text{ rem } f$ 
3:  $\bar{a}_0 \leftarrow b$ ; for  $j = 1, \dots, s-1$  do  $\bar{a}_j \leftarrow \hat{a}_r \cdot \bar{a}_{j-1} \text{ rem } f$  ▷  $\bar{a}_j = ba^{jr} \text{ rem } f$ 
4:  $(c_{i,j})_{\substack{0 \leq i < r \\ 0 \leq j < s}} \leftarrow$  SIMULTANEOUS TRUNCATED MODULAR MULTIPLICATION( $f, \hat{a}_0, \dots, \hat{a}_{r-1}, \bar{a}_0, \dots, \bar{a}_{s-1}, m$ )
5: for  $i = 0, \dots, r-1$  and  $j = 0, \dots, s-2$  do  $r_{i+rj} \leftarrow c_{i,j}$ 
   for  $i = 0, \dots, d-1 - (s-1)r$  do  $r_{i+r(s-1)} \leftarrow c_{i,s-1}$ 
6: return  $(r_k)_{0 \leq k < d}$ 

```

In terms of complexity, the multiplication P_1Q_1 involves $r \times (\ell + 1)$ and $(\ell + 1) \times s$ matrices, while P_2Q_2 involves $r \times \ell$ and $\ell \times s$ matrices; all four operands have degree less than m .

Since $\ell = \lfloor (n-1)/m \rfloor - 1$, we have $\ell + 1 \leq n/m$, so each matrix product can be done using at most $\lceil n/(mr^2) \rceil \leq n/(mr^2) + 1$ products in sizes $r \times r^2$ and $r^2 \times s$. Since $s \in \tilde{O}(r)$, each of these take $\tilde{O}(mr^{\omega_2})$, for a total cost of $c(n, m, r^2) = \tilde{O}((m + n/r^2)r^{\omega_2})$.

The other operations performed by the algorithm are $O(r)$ power series expansions at precision $n-1$ in $\tilde{O}(n)$ operations each (precisely, one inverse and t multiplications, see Step 3), and $O(r^2)$ power series expansions at precision m in $\tilde{O}(m)$ operations each (precisely, at most $3rs$ multiplications and rs subtractions, see Steps 5 and 6). This amounts to a total of $\tilde{O}(nr + mr^2)$ operations, and can thus be neglected, since $\omega_2 \geq 3$. \square

Using simultaneous truncated modular multiplication combined with a baby steps/giant steps strategy leads to Algorithm **TRUNCATEDPOWERS**, with the following properties.

PROPOSITION 3.6. *Given f in $\mathbb{K}[x]$ of degree n , a and b in $\mathbb{K}[x]_{<n}$, m and d in $\mathbb{N}_{>0}$, Algorithm **TRUNCATEDPOWERS** computes the truncations*

$$[a^k b \text{ rem } f]_0^{m-1}, \quad 0 \leq k < d$$

using $\tilde{O}(c(n, m, d)) = \tilde{O}((m + n/d)d^{\omega_2/2})$ operations in \mathbb{K} , with $c(\cdot)$ from Equation (2).

PROOF. The algorithm computes $1, a, \dots, a^{r-1} \text{ rem } f$ and $b, ba^r, \dots, ba^{(s-1)r} \text{ rem } f$, which costs $\tilde{O}(nr)$ operations in \mathbb{K} since $r \sim s$. From these two sets of polynomials, Algorithm **SIMULTANEOUS TRUNCATED MODULAR MULTIPLICATION** is then used to compute $[ba^k \text{ rem } f]_0^{m-1}$ for $0 \leq k \leq rs-1$ using $\tilde{O}((m + n/r^2)r^{\omega_2})$ operations, by Lemma 3.5; since $\omega_2 \geq 3$, this is larger than $\tilde{O}(nr)$. The choice of s makes $(s-1)r < d \leq rs$, so the output consists of the terms $k = i + rj$ for $j < s-1$ and $i < r$, and for $j = s-1$ and $i < d - (s-1)r \in \{1, \dots, r\}$. \square

Finally, Algorithm **BLOCKTRUNCATEDPOWERS** computes truncations of products of the form $x^i a^k \text{ rem } f$, which are needed in our composition algorithm; here, we assume that $f(0)$ is nonzero (see Remark 3.8).

PROPOSITION 3.7. *Given f in $\mathbb{K}[x]$ of degree n with $f(0) \neq 0$, a in $\mathbb{K}[x]_{<n}$, m and d in $\mathbb{N}_{>0}$, Algorithm **BLOCKTRUNCATEDPOWERS** computes*

$$[x^i a^k \text{ rem } f]_0^{m-1}, \quad 0 \leq i < m, 0 \leq k < d$$

using $\tilde{O}(c(n, m, d)) + O(m^2d) = \tilde{O}((m + n/d)d^{\omega_2/2}) + O(m^2d)$ operations in \mathbb{K} , with $c(\cdot)$ from Equation (2).

ALGORITHM 3.8 BLOCKTRUNCATEDPOWERS(f, a, m, d)

Input: f of degree n in $\mathbb{K}[x]$, with $f_0 = f(0) \neq 0$, a in $\mathbb{K}[x]_{<n}$, m and d in $\mathbb{N}_{>0}$

Output: the truncated powers $[x^i a^k \bmod f]_0^{m-1}$, for $0 \leq i < m$ and $0 \leq k < d$

```

1:  $(r_k)_{0 \leq k < d} \leftarrow \text{TRUNCATEDPOWERS}(f, a, x^{m-1} \bmod f, 2m-1, d)$  ▷  $r_k = [x^{m-1} a^k]_0^{2m-2}$ 
2:  $f_n \leftarrow \text{coeff}(f, n)$  ▷ leading coefficient
3: for  $k = 0, \dots, d-1$  do
4:    $a_{m-1, k} \leftarrow r_k$ 
5:   for  $i = m-1, \dots, 2, 1$  do
6:      $c \leftarrow -a_{i, k}(0)/f_0$ 
7:      $a_{i-1, k} \leftarrow (a_{i, k} + c[f]_0^{m+i-1})/x$  ▷  $a_{i-1, k} = [x^{i-1} a^k \bmod f]_0^{m+i-2}$ 
8: return  $([a_{i, k}]_0^{m-1})_{\substack{0 \leq i < m \\ 0 \leq k < d}}$ 

```

PROOF. Proposition 3.6 shows that the first step computes the sequence $[x^{m-1} a^k]_0^{2m-2}$ for $k = 0, \dots, d-1$ in the announced complexity. The remaining truncations are obtained from the identity

$$[xp \bmod f]_0^j = x[p \bmod f]_0^{j-1} - \frac{p_{n-1}}{f_n} [f]_0^j,$$

for any integer j and polynomial p , where p_{n-1} is the coefficient of degree $n-1$ of $p \bmod f$ and f_n is the coefficient of degree n in f . If we know $[xp \bmod f]_0^j$, we get $-p_{n-1}f_0/f_n$ as its constant coefficient, whence p_{n-1} since $f_0 \neq 0$ and from there $[p \bmod f]_0^{j-1}$ is easily obtained. At iteration k of the loop at Step 3, the truncation $[x^{m-1} a^k \bmod f]_0^{2m-2}$ computed previously is used to deduce all $[x^{m-1-i} a^k \bmod f]_0^{2m-2-i}$ for $1 \leq i < m$ in $O(m^2)$ operations. Thus this loop has a total cost of $O(m^2 d)$ operations. \square

Remark 3.8. The assumption $f(0) \neq 0$ is harmless in the context of modular composition: in the computation of $g(a) \bmod f$, one can rather evaluate $g(y)$ at $a(x+c) \bmod f(x+c)$ for a randomly chosen $c \in \mathbb{K}$, and unshift the result. See Steps 3 and 12 in Algorithm **MODULARCOMPOSITIONBASECASE**.

3.4 Notes

3.4.1 Linear Algebra Interpretation. Representing polynomials by their vector of coefficients leads to viewing the operations performed by Algorithms **BIVARIATEMODULARCOMPOSITION** and **TRUNCATEDPOWERS** as computing the product of special matrices by column vectors. Recall the notation M_a for the $n \times n$ matrix of multiplication by $a \bmod f$ in the basis $(1, x, \dots, x^{n-1})$, and X for the matrix $(I_m \ 0)^\top \in \mathbb{K}^{n \times m}$ with $m \in \{1, \dots, n\}$. Then, Algorithms **BIVARIATEMODULARCOMPOSITION** and **TRUNCATEDPOWERS** correspond, respectively, to multiplication by

$$K_{m, d}^{(a, f)} = (X \quad \dots \quad M_a^{d-1} X) \in \mathbb{K}^{n \times (md)} \quad \text{and} \quad L_{m, d}^{(a, f)} = \begin{pmatrix} X^\top \\ \vdots \\ X^\top M_a^{d-1} \end{pmatrix} \in \mathbb{K}^{(md) \times n}. \quad (12)$$

Indeed, $K_{m, d}^{(a, f)}$ is the matrix of the mapping $\kappa_{m, d}^{(a, f)}$ of bivariate modular composition with bounded degrees, as computed by Algorithm **BIVARIATEMODULARCOMPOSITION**:

$$\begin{aligned} \kappa_{m, d}^{(a, f)} : \mathbb{K}[x, y]_{<(m, d)} &\rightarrow \mathbb{K}[x]_{<n} \\ g(x, y) &\mapsto g(x, a) \bmod f. \end{aligned}$$

On the other hand, $L_{m,d}^{(a,f)}$ represents the mapping $\lambda_{m,d}^{(a,f)}$ that extracts the low-degree part of multiplications by powers of a , as computed by Algorithm [TRUNCATEDPOWERS](#):

$$\begin{aligned}\lambda_{m,d}^{(a,f)} : \mathbb{K}[x]_{<n} &\rightarrow \mathbb{K}[x]_{<m}^d \\ b &\mapsto ([b \text{ rem } f]_0^{m-1}, \dots, [ba^{d-1} \text{ rem } f]_0^{m-1}).\end{aligned}$$

These maps and matrices play an important role in the study of the generic behavior of our algorithm starting from Section 7.

3.4.2 Complexity Equivalence. Proposition 3.4 (Algorithm [BIVARIATEMODULARCOMPOSITION](#)) and Proposition 3.6 (Algorithm [TRUNCATEDPOWERS](#)) give similar complexity bounds for the evaluation of $\kappa_{m,d}^{(a,f)}$ and $\lambda_{m,d}^{(a,f)}$, but the computational equivalence of these problems, possibly up to some conditions, is still unclear to us in general.

However, for $m = 1$, when $f(0) \neq 0$, these two problems are indeed equivalent. This is a consequence of the transposition principle in an indirect way, starting from the equality

$$L_{1,n}^{(a,f)} v_b = (M_b K_{1,n}^{(a,f)})^\top \mathbf{1},$$

where v_b is the vector associated to b , M_b is the matrix of multiplication by $b \bmod f$, and $\mathbf{1}$ is the first canonical vector. First, this equality gives a way to evaluate $\lambda_{1,n}^{(a,f)}$ for the cost of one multiplication by M_b^\top (i.e., $\tilde{O}(n)$ by the transposition principle), plus one multiplication by the transpose of $K_{1,n}^{(a,f)}$, which has the same asymptotic cost as that of $K_{1,n}^{(a,f)}$ itself, by the same principle. Conversely, if $v = M_b^\top \mathbf{1}$, the equality reads $(K_{1,n}^{(a,f)})^\top v = L_{1,n}^{(a,f)} v_b$, so that, again by the transposition principle, the evaluation of $\kappa_{1,n}^{(a,f)}$ reduces to that of $\lambda_{1,n}^{(a,f)}$ provided v_b can be computed from v in low complexity. When $f(0) \neq 0$, this can be done in $\tilde{O}(n)$ by solving a linear system of Hankel type [71, Section 3].

If $f(0) = 0$, it is unclear whether such a reduction holds: in the special case $f = x^n$, the map $\lambda_{1,n}^{(a,f)}$ becomes much simpler, as it simply computes the sequence $a_0^i b_0$ for $0 \leq i < d$, where a_0 and b_0 are the constant coefficients of a and b . This only requires a linear number $O(d)$ of operations. On the other hand, $\kappa_{1,d}$ is the composition of a univariate polynomial $g(y)$ of degree less than d with the power series $a(x)$ and no quasi-linear complexity result is known for this operation.

3.4.3 Transposition of the Nüsken–Ziegler Algorithm. Finally, we discuss a different approach to Algorithm [BLOCKTRUNCATEDPOWERS](#), which actually bypasses Algorithm [TRUNCATEDPOWERS](#) altogether, and uses the transpose of Algorithm [BIVARIATEMODULARCOMPOSITION](#) instead.

Algorithm [BLOCKTRUNCATEDPOWERS](#) computes the $m \times m$ projections $H_k = X^\top M_{a^k} X$, for $k = 0, \dots, d-1$, using the fact that for $k < d$, H_k can be deduced in $O(m^2)$ operations (for loop at Step 5) from the column vector $\bar{X}^\top M_{a^k} u$ of size $2m-1$, where $\bar{X} = (\mathbf{I}_{2m-1} \ 0)^\top \in \mathbb{K}^{(2m-1) \times n}$ and u is the m th column of X , i.e., the m th canonical vector (Step 1). (Here, we have taken $m \leq (n+1)/2$.) Algorithm [TRUNCATEDPOWERS](#) computes the vectors $\bar{X}^\top M_a^k u$ for $0 \leq k < d$ using $\tilde{O}(c(n, m, d))$ operations.

Alternatively, we can consider a recursion similar to the one in the proof of Proposition 3.7, but now for learning a new coefficient of a polynomial rather than a coefficient of a new polynomial. Assuming $f(0) \neq 0$, for a polynomial p one has

$$[xp \text{ rem } f]_{i+1}^0 = [p \text{ rem } f]_i^0 + (c/f_0)[f]_{i+1}^0,$$

where c is the coefficient of degree 0 of $xp \text{ rem } f$: we see that from the row vector $\mathbf{1}^\top M_{x^{-m+1}a^k} \bar{X}$, one can also deduce $H_k = X^\top M_{a^k} X$ using $O(m^2)$ operations.

Now, if we set $v = M_{x^{-m+1}}^\top \mathbf{1}$, computing $v^\top K_{d,2m-1}^{(a,f)}$ precisely gives all vectors $\mathbf{1}^\top M_{x^{-m+1}a^k} \bar{X}$, for $0 \leq k < d$. Since v can be computed in quasi-linear time, the application of the transposition principle to Algorithm **BIVARIATEMODULARCOMPOSITION** shows that these vectors can be computed using $\tilde{O}(c(n, m, d))$ operations. Altogether, this gives an alternative to Algorithm **BLOCKTRUNCATEDPOWERS** with the same asymptotic complexity.

4 MATRICES OF RELATIONS FOR COMPOSITION

The heart of our algorithm for finding $g(a) \bmod f$ is the computation of a *matrix of relations*, which gives a collection of polynomials of small degree in the ideal \mathcal{I} generated by $y - a$ and f in $\mathbb{K}[x, y]$. For a given positive integer m , these polynomials are in the $\mathbb{K}[y]$ -module $\mathcal{M}_m^{(a,f)}$ obtained by degree restriction as $\mathcal{I} \cap \mathbb{K}[x, y]_{<(m, \cdot)}$.

In Section 4.1, we show that the invariant factors of $\mathcal{M}_m^{(a,f)}$ are the m invariant factors of highest degree of the characteristic matrix $yI_n - M_a$, where M_a is the matrix of multiplication by $a \bmod f$. Once a matrix of relations has been obtained, it can be used to perform composition by reducing univariate composition to a small bivariate composition problem; this is described in Section 4.2. Finally, in Section 4.3, the results of this section are applied to the efficient computation of annihilating polynomials for a modulo f .

In all of Section 4, notation such as $\mathcal{M}_m^{(a,f)}$ and $v_m^{(a,f)}$ is shortened into \mathcal{M}_m and v_m , except for the main definitions and statements, as there is no ambiguity as to the dependency on a or f .

4.1 Structure of the Module of Relations

This section introduces the module of relations $\mathcal{M}_m^{(a,f)}$ and relates it to the characteristic matrix.

4.1.1 Definitions.

Relations. We call *relations* the polynomials of the ideal $\mathcal{I} = \langle y - a(x), f(x) \rangle$ of $\mathbb{K}[x, y]$; these are the bivariate polynomials $r(x, y)$ such that $r(x, a) \equiv 0 \bmod f$, i.e., they are algebraic relations satisfied by $a \bmod f$. We are interested in those relations whose x -degree is bounded from above by a given positive integer m . They form the $\mathbb{K}[y]$ -module

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

which is denoted \mathcal{M}_m when a and f are clear from the context.

This is a $\mathbb{K}[y]$ -submodule of $\mathbb{K}[x, y]_{<(m, \cdot)}$, itself a free $\mathbb{K}[y]$ -module with basis $(1, x, \dots, x^{m-1})$; we refer to [23, Part III] for basic notions of module theory and modules over principal ideal domains. As stated in Section 2, we often identify a polynomial $r_0(y) + \dots + r_{m-1}(y)x^{m-1}$ in $\mathbb{K}[x, y]_{<(m, \cdot)}$ with the column vector $(r_0 \ \dots \ r_{m-1})^\top$ in $\mathbb{K}[y]^m$ of its coefficients on that basis. Since $\mathbb{K}[y]$ is a principal ideal domain, \mathcal{M}_m is free as well, and it has rank m since it contains $\mu_a \mathbb{K}[y]^m$, where μ_a is the minimal polynomial of $a \bmod f$.

In terms of ideals, there is a chain of inclusions $\{0\} = \langle \mathcal{M}_0 \rangle \subseteq \dots \subseteq \langle \mathcal{M}_{n+1} \rangle = \mathcal{I}$; the latter identity follows from the fact that $y - a$ and f have x -degree less than $n+1$. Furthermore, $\mathcal{M}_1 \neq \{0\}$ since μ_a belongs to $\mathcal{I} \cap \mathbb{K}[y]$. For small m , the module \mathcal{M}_m may not contain all the information in \mathcal{I} : the inclusion $\langle \mathcal{M}_m \rangle \subset \mathcal{I}$ can be strict.

Matrix and basis of relations, determinantal degree. A *matrix of relations* of \mathcal{M}_m is any nonsingular matrix in $\mathbb{K}[y]^{m \times m}$ whose columns are elements of the module \mathcal{M}_m (represented as column vectors). Such a matrix is further called a *basis* of relations if its columns generate \mathcal{M}_m ; all bases of relations of \mathcal{M}_m can be obtained from any single one of them via right multiplication by an unimodular matrix in $\mathbb{K}[y]^{m \times m}$, i.e., a matrix whose determinant is in $\mathbb{K} \setminus \{0\}$. It follows that any matrix of relations of \mathcal{M}_m is a square, nonsingular right multiple of any basis of relations of \mathcal{M}_m , and,

therefore, bases of relations are exactly the matrices of relations whose determinant has minimal degree. This degree is called the *determinantal degree* of the module \mathcal{M}_m .

4.1.2 Relation to Invariant Factors. As a finitely generated module over a principal ideal domain, \mathcal{M}_m has an invariant factor decomposition. The next result shows that these invariant factors can be found in any triangular basis of \mathcal{M}_m , and that the largest of these factors is precisely μ_a , the minimal polynomial of a modulo f . It also relates the degrees of these factors to the quantity $v_m^{(a,f)}$ (written more simply as v_m when context is clear), already highlighted in Equation (8), and which plays an important role in the analysis of our approach.

PROPOSITION 4.1. *Let B be an upper triangular basis of $\mathcal{M}_m^{(a,f)}$ for some $m \geq 1$. Then its diagonal entries are the invariant factors of $\mathcal{M}_m^{(a,f)}$, up to multiplication by nonzero elements of \mathbb{K} . A number $k \leq \min(m, n)$ of these invariant factors are nontrivial, and these nontrivial ones are the k invariant factors of highest degree of the characteristic matrix $yI_n - M_a$, which is a basis of relations of $\mathcal{M}_n^{(a,f)}$. The determinantal degree of $\mathcal{M}_m^{(a,f)}$ is the sum $v_m^{(a,f)}$ of the degrees of these invariant factors, hence it satisfies $\min(m, n) \leq v_m^{(a,f)} \leq n$.*

4.1.3 Proof of Proposition 4.1. Our proof relies on Lazard's structure theorem [54] on lexicographic Gröbner bases in $\mathbb{K}[x, y]$. Here, the *degree* of a zero-dimensional ideal $\mathcal{I} \subset \mathbb{K}[x, y]$ is the dimension of the \mathbb{K} -vector space $\mathbb{K}[x, y]/\mathcal{I}$.

LEMMA 4.2 (LAZARD'S STRUCTURE THEOREM FOR BIVARIATE IDEALS). *Let \mathcal{I} be a zero-dimensional ideal of degree n in $\mathbb{K}[x, y]$. Any minimal Gröbner basis of \mathcal{I} for the $(y < x)$ -lexicographic order has the form $\{r_0(y)h_k(x, y), r_1(y)h_{k-1}(x, y), \dots, r_k(y)h_0(x, y)\}$ for some $k \geq 1$, where*

$$\left\{ \begin{array}{l} r_k = h_k = 1 \\ n \geq \deg(r_0) > \dots > \deg(r_k) = 0 \\ n \geq \deg_x(h_0) > \dots > \deg_x(h_k) = 0 \\ \text{for } 0 \leq i < k, r_i \in \mathbb{K}[y] \text{ is divisible by } r_{i+1} \\ \text{for } 0 \leq i \leq k, h_i \in \mathbb{K}[x, y] \text{ has leading monomial a power of } x. \end{array} \right.$$

PROOF. The form of a minimal Gröbner basis of \mathcal{I} is given by Lazard's result [54, Theorem 1]. The additional assumption that \mathcal{I} is zero-dimensional ensures that this Gröbner basis contains a polynomial whose leading term is a power of y , hence $h_k = 1$, and one whose leading term is a power of x , hence $r_k = 1$. Since \mathcal{I} has degree n , there are precisely n monomials that are not multiples of the leading monomials of $\{r_i h_{k-i} \mid 0 \leq i \leq k\}$. These leading monomials are $\{x^{\deg_x(h_{k-i})} y^{\deg(r_i)} \mid 0 \leq i \leq k\}$, whence the bounds $\deg(r_0) \leq n$ and $\deg_x(h_0) \leq n$. \square

COROLLARY 4.3. *With the same notation, when $\mathcal{I} = \langle f, y - a \rangle$ and $m \geq 1$, a basis of $\mathcal{M}_m^{(a,f)} = \mathcal{I} \cap \mathbb{K}[x, y]_{<(m, \cdot)}$ is given by the first m polynomials in the sequence*

$$(x^j r_0 h_k)_{0 \leq j < \delta_k}, \dots, (x^j r_{k-1} h_1)_{0 \leq j < \delta_1}, (x^j h_0)_{j \geq 0}, \quad (13)$$

where $\delta_i = \deg_x(h_{i-1}) - \deg_x(h_i)$. If $s = \deg_x(h_0) = \delta_1 + \dots + \delta_k$, the nontrivial invariant factors of $\mathcal{M}_m^{(a,f)}$ are the first $\min(m, s)$ polynomials in

$$\underbrace{(r_0, \dots, r_0, \dots)}_{\delta_k}, \underbrace{(r_{k-1}, \dots, r_{k-1})}_{\delta_1}. \quad (14)$$

PROOF. The polynomials in the sequence in Equation (13) form a (nonfinite) Gröbner basis of \mathcal{I} , made of polynomials of x -degree $0, 1, 2, \dots$ respectively [54, Proposition 1]. By design, the first

m elements in this sequence belong to \mathcal{M}_m , and considering their x -degrees shows that they are $\mathbb{K}[y]$ -linearly independent.

Any polynomial $p(x, y) \in \mathcal{M}_m$ is a $\mathbb{K}[y]$ -linear combination of the first m of these polynomials. Indeed, it can be divided by the Gröbner basis with a remainder equal to 0; in view of its degree in x , only these m polynomials are involved in the division. This proves the claim on the basis of \mathcal{M}_m described in Equation (13).

The matrix $T(y) \in \mathbb{K}[y]^{m \times m}$ representing this basis (with basis elements written in columns) is upper triangular, with its first $\min(m, s)$ diagonal entries being the first $\min(m, s)$ polynomials in Equation (14) in this order, and with its remaining diagonal entries being nonzero elements of \mathbb{K} . Furthermore, each of these diagonal entries divides all other entries in the same column, hence the Smith normal form of $T(y)$ has the same diagonal entries as $T(y)$, which proves the claim on the invariant factors of \mathcal{M}_m . \square

PROOF OF PROPOSITION 4.1. Corollary 4.3 implies that the determinantal degree ν_m of \mathcal{M}_m is the sum of the degrees of the elements of the first $\min(m, s)$ elements of Equation (14). It follows that

$$\nu_m \leq \delta_k \deg(r_0) + \cdots + \delta_1 \deg(r_{k-1}) = n,$$

where the last identity comes from considering the \mathbb{K} -vector space dimension of $\mathbb{K}[x, y]/\mathcal{I}$. If $s \leq m$, all the nontrivial invariant factors appear and the bound is reached, while otherwise $m < s$ and $\deg(\det(B))$, being the sum of the degrees of m nonconstant polynomials, is at least m .

If B is a basis of \mathcal{M}_m , then there exists an unimodular matrix $U \in \mathbb{K}[y]^{m \times m}$ such that $UB = T$ with T as in the previous proof. If, moreover, B is upper triangular, then so is U and since $\det(U) \in \mathbb{K} \setminus \{0\}$, the diagonal entries of U belong to $\mathbb{K} \setminus \{0\}$. It follows that B has the same diagonal entries as T up to multiplication by nonzero elements of \mathbb{K} .

The columns of the characteristic matrix $yI_n - M_a$ represent the polynomials $x^k(y - a(x)) \bmod f$ for $0 \leq k < n$, making this matrix a matrix of relations of \mathcal{M}_n . It has determinantal degree $\deg(\chi_a) = n$, which coincides with the determinantal degree of \mathcal{M}_n , by the previous inequalities. Thus $yI_n - M_a$ is actually a basis of \mathcal{M}_n and its invariant factors are given by the previous paragraph. \square

4.1.4 Note. For $m \in \{1, \dots, n\}$, the module of relations \mathcal{M}_m is isomorphic to the module of vector generators for the matrix sequence $\{M_a^k X\}_{k \geq 0}$, where $X = (I_m \ 0)^T \in \mathbb{K}^{n \times m}$ as above (this elementary fact is established within the proof of Lemma 5.2, for instance); the bases of relations are the *minimal generating polynomials* for that sequence [50, 74].

The relation between Coppersmith's block Wiedemann algorithm and invariant factors of a characteristic matrix was described by Kaltofen and Villard: they show that for generic projections V and W in $\mathbb{K}^{n \times \ell}$ and $\mathbb{K}^{n \times m}$, with $\ell \geq m$, the invariant factors of minimal generating polynomial of the sequence $(V^T A^k W)_{k \geq 0}$ are the m invariant factors of largest degree of the characteristic matrix $yI_n - A$ [50, Theorem 2.12]. In our more specific setting, Proposition 4.1 shows that this relation holds when the right projection is the structured matrix X (see also Section 5.1.4).

4.2 Composition using Matrices of Relations

Matrices of relations are used to reduce the univariate problem $g(a) \bmod f$ with $g \in \mathbb{K}[y]$, to a bivariate one with better degree properties, thanks to a matrix division.

4.2.1 Division for Polynomial Matrices. If R is a nonsingular matrix in $\mathbb{K}[y]^{m \times m}$ and v_g is a vector in $\mathbb{K}[y]^m$, then there exist quotient and remainder vectors w and $v_{\tilde{g}}$ such that

$$v_g = R w + v_{\tilde{g}}, \tag{15}$$

ALGORITHM 4.1 BIVARIATEMODULARCOMPOSITIONWITHRELATIONMATRIX(f, a, g, R)

Input: f of degree n in $\mathbb{K}[x]$, a in $\mathbb{K}[x]_{<n}$, g in $\mathbb{K}[x, y]_{<(m,.)}$,
 $R \in \mathbb{K}[y]_{\leq d}^{m \times m}$ a matrix of relations of $\mathcal{M}_m^{(a,f)}$

Output: $g(x, a) \bmod f$

- 1: Write $g(x, y) = g_0(y) + g_1(y)x + \cdots + g_{m-1}(y)x^{m-1}$ and set $v_g \leftarrow (g_0 \cdots g_{m-1})^\top \in \mathbb{K}[y]^m$
- 2: \triangleright Compute $v \in \mathbb{K}[y]^m$ and $r \in \mathbb{K}[y]$ using [80, Algorithm 1]
- $\begin{pmatrix} v \\ r \end{pmatrix} \in \mathbb{K}[y]^{m+1} \leftarrow \text{MINIMALNULLSPACEBASIS}((R - v_g), (d, \dots, d, \deg_y(g)))$
- 3: $v_{\tilde{g}} \leftarrow R(v \bmod r)/r \in \mathbb{K}[y]_{\leq d}^m$ $\triangleright v \bmod r$ is the vector of entry-wise remainders
- 4: $\tilde{g}(x, y) \leftarrow$ the polynomial in $\mathbb{K}[x, y]_{<(m, d)}$ corresponding to $v_{\tilde{g}}$
- 5: **return** **BIVARIATEMODULARCOMPOSITION**(f, a, \tilde{g}) $\triangleright \tilde{g}(x, a) \bmod f$, Algorithm 3.5

and each entry of $v_{\tilde{g}}$ has degree less than that of the corresponding row of R [43, Theorem 6.3-15, p. 389]. The latter reference actually states a stronger condition on $v_{\tilde{g}}$, namely that the matrix fraction $R^{-1}v_{\tilde{g}}$ is *strictly proper* (see Section 5.1.1); this implies the above degree condition [43, Lemma 6.3-10, p. 383], which is sufficient for our needs.

For computing this division, it is customary to use $\mathbb{K}[y]$ -linear system solving. For this, we rely on a kernel basis algorithm [80]: this returns v in $\mathbb{K}[y]^m$ and r in $\mathbb{K}[y]$ such that $R^{-1}v_g = v/r$, with r of minimal degree. From this, the remainder is obtained as $v_{\tilde{g}} = R(v \bmod r)/r$, and here we do not need the quotient vector w .

4.2.2 Composition Algorithm. In the case where $v_g = (g \ 0 \cdots 0)^\top$ and R is a matrix of relations of \mathcal{M}_m of degree at most d , the remainder in the above division is a vector $v_{\tilde{g}}$ of degree less than d whose entries yield $\tilde{g} \in \mathbb{K}[x, y]_{<(m, d)}$ such that $g - \tilde{g} \in \mathcal{M}_m$. Thus, analogously to a reduction modulo a Gröbner basis of the ideal $\mathcal{I} = \langle y - a, f \rangle$, this provides a bivariate polynomial \tilde{g} with smaller degree in y and controlled degree in x , and such that $\tilde{g} - g \in \mathcal{I}$, that is, $\tilde{g}(x, a) \equiv g(a) \bmod f$.

Algorithm **BIVARIATEMODULARCOMPOSITIONWITHRELATIONMATRIX** is given a matrix of relations R of \mathcal{M}_m as a parameter and performs this division; then it completes the composition by evaluating $\tilde{g}(x, a) \bmod f$ using Algorithm **BIVARIATEMODULARCOMPOSITION**. Algorithm **BIVARIATEMODULARCOMPOSITIONWITHRELATIONMATRIX** actually accepts a slightly more general input: g can be a bivariate polynomial with x -degree less than m (however, the rest of the article focuses on the case of g in $\mathbb{K}[y]$ highlighted above). The algorithm accepts g of arbitrary degree in y , but the cost analysis is done under the assumption $\deg_y(g) \in O(n)$.

PROPOSITION 4.4. *Given f in $\mathbb{K}[x]$ of degree n , a in $\mathbb{K}[x]_{<n}$, g in $\mathbb{K}[x, y]_{<(m,.)}$ with $\deg_y(g) = O(n)$ and a matrix of relations R in $\mathbb{K}[y]_{\leq d}^{m \times m}$ of $\mathcal{M}_m^{(a,f)}$, Algorithm **BIVARIATEMODULARCOMPOSITIONWITHRELATIONMATRIX** computes $g(x, a) \bmod f$ using $\tilde{O}(m^\omega(d + n/m) + c(n, m, d))$ operations in \mathbb{K} , with $c(\cdot)$ from Equation (2).*

PROOF. First, Step 2 computes $r \in \mathbb{K}[y]$ and $v = rR^{-1}v_g \in \mathbb{K}[y]^m$ with r of minimal degree. Indeed, since R is nonsingular, the right kernel of $(R - v_g) \in \mathbb{K}[y]^{m \times (m+1)}$ has rank 1. We use [80, Algorithm 1] to compute a basis $(v^\top \ r)^\top$ of this kernel. Thus by construction $Rv = rv_g$ holds, and the fact that $(v^\top \ r)^\top$ generates the kernel ensures that the greatest common divisor of r and all the entries of v is 1, hence the minimality of $\deg(v)$ and $\deg(r)$.

At Step 3 one considers the vector $\bar{v} = v \bmod r \in \mathbb{K}[y]^m$ such that $\deg(\bar{v}) < \deg(r)$ and $v = rw + \bar{v}$ for some $w \in \mathbb{K}[y]^m$. It follows that $v_g = Rv/r = Rw + v_{\tilde{g}}$, where $v_{\tilde{g}} = R\bar{v}/r$ is the vector

computed at Step 3; by construction the i th entry of $v_{\tilde{g}}$ has degree less than that of the i th row of R . In short, Steps 2 and 3 compute a vector $v_{\tilde{g}} \in \mathbb{K}[y]^m$ that has degree less than d and is a remainder of v_g modulo R . Since R is a matrix of relations, the polynomial $\tilde{g}(x, y)$ at Step 4 is such that $\tilde{g}(x, a) \equiv g(x, a) \bmod f$. The correctness follows, since $\tilde{g}(x, a) \bmod f$ is the polynomial returned by `BIVARIATEMODULARCOMPOSITION`(f, a, \tilde{g}) (see Proposition 3.4).

As required by Algorithm 1 of [80], the tuple of integers $(d, \dots, d, \deg_y(g)) \in \mathbb{Z}^{m+1}$ bounds the column degrees of $(R - v_g)$. Then, since the sum of this tuple is $md + \deg_y(g)$, with $\deg_y(g) = O(n)$, Step 2 costs $\tilde{O}(m^\omega(d + n/m))$ operations [80, Theorem 4.1]. The minimality of $\deg(r)$ implies $\deg(r) \leq \deg \det(R) \leq md$, and then v has degree at most $\deg \det(R)R^{-1}v_g \leq (m-1)d + n$ since $\det(R)R^{-1}$ is the transpose of the cofactor matrix of R . Thus, the computation of $\bar{v} = v \bmod r$ in Step 3 uses $\tilde{O}(m(md + n))$ operations, which is smaller than the cost of Step 2. Next, the matrix-vector product $R\bar{v}$ can be performed in $\tilde{O}(m^\omega d)$ operations: write the column \bar{v} of degree $< md$ as m columns of degree $< d$ via y^d -adic expansion; use a matrix-matrix product to left-multiply these columns by R ; finally, recombine the resulting columns into a single column that gives $R\bar{v}$. To obtain $v_{\tilde{g}}$ it remains to divide each entry of $R\bar{v}$ by r , which costs $\tilde{O}(m^2d)$ since $\deg(R\bar{v}) < (m+1)d$. By Proposition 3.4, the call at Step 5 uses $c(n, m, d)$ operations. The cost bound in the Proposition follows. \square

Note. Comparing Proposition 4.4 with Proposition 3.4, note that when $m \sim n^\eta$ and $d \sim n^{1-\eta}$ with η from Equation (3), then the complexity bound of Proposition 4.4 is the same as the one given by the Nüsken–Ziegler algorithm, however, the y -degree of g can now go up to the order of n .

4.3 Annihilating Polynomials Using Matrices of Relations

Our main algorithm requires an annihilating polynomial for a , that is, a polynomial h in $\mathbb{K}[y]$ such that $h(a) \equiv 0 \bmod f$. It can readily be obtained from a matrix of relations.

PROPOSITION 4.5. *Let $R \in \mathbb{K}[y]_{\leq d}^{m \times m}$ be a matrix of relations of $\mathcal{M}_m^{(a,f)}$. Its determinant is a nonzero annihilating polynomial for a modulo f . It has degree at most md in $\mathbb{K}[y]$ and can be computed from R using $\tilde{O}(m^\omega d)$ operations in \mathbb{K} .*

PROOF. As a polynomial combination of relations in \mathcal{M}_m , the entry $(1, 1)$ of the (upper triangular) Hermite normal form of a matrix of relations is a relation in $\langle f, y - a \rangle \cap \mathbb{K}[y]$, so it is a nonzero multiple of the minimal polynomial of a . This implies the same property for the determinant, since it is a multiple of that entry. The bound on the degree of the determinant is straightforward, and the cost bound is from [53, Theorem 1.1]. \square

Note. For the computations of the minimal polynomial and of the characteristic polynomial of a modulo f , see Section 10.1.

5 COMPUTING MATRICES OF RELATIONS

In this section, we give an algorithm computing a matrix of relations. This study may be viewed as a specialization of the formalism developed by Kaltofen and Villard for the block Wiedemann approach (see Sections 1.4.3 and 1.4.4) in terms of manipulations of bivariate polynomials in the ideal generated by $y - a$ and f .

As already done in Section 4, notation such as $\mathcal{M}_m^{(a,f)}$ and $v_m^{(a,f)}$ is shortened into \mathcal{M}_m and v_m in this section, except in the main statements.

In Section 5.1, we show that for $m \in \{1, \dots, n\}$, denominators of irreducible right matrix fraction descriptions of $(yI_n - M_a)^{-1}X$ with $X = (I_m \ 0)^\top \in \mathbb{K}^{n \times m}$ yield bases of $\mathcal{M}_m^{(a,f)}$. For efficiency reasons, a further truncation is required: this leads us to introduce modules $\mathcal{M}_{\ell,m}^{(a,f)}$ whose bases are the denominators of irreducible right matrix fraction descriptions of $Y^\top(yI_n - M_a)^{-1}X$, where $Y^\top = (I_\ell \ 0) \in \mathbb{K}^{\ell \times n}$, with $\ell \in \{1, \dots, n\}$; thus, we use structured left and right block projections. If $\ell = n$, Y is the identity matrix of size n , and we recover $\mathcal{M}_m^{(a,f)}$, but this value is too large for our cost objectives. Instead, we focus on $\ell = m$, and thus $Y = X$.

Section 5.2 describes how a basis of $\mathcal{M}_{\ell,m}^{(a,f)}$ can be reconstructed using *minimal approximant bases* [4, 73], from sufficiently many terms of the power series expansion of the matrix $H = X^\top(yI_n - M_a)^{-1}X$.

This strategy is turned into an algorithm for computing matrices of relations in Section 5.3: the expansion of H is obtained via Algorithm **BLOCKTRUNCATEDPOWERS**, while approximant bases are computed using a matrix Padé version of the Berlekamp–Massey algorithm [4, 31]. The correctness and efficiency of this approach depends on a fundamental condition on M_a , i.e., on f and a (Proposition 5.6, first item). First, it expresses that the left projection does not prevent us from getting the right denominators of $(yI_n - M_a)^{-1}X$ from those of H . It also ensures the existence of matrices of relations of “small” degree ℓ , and, in this way, appropriately limits the number of terms of the expansion of H that are required for the reconstruction. We prove in Section 7 that these properties are satisfied for generic inputs; in Section 8, we further study cases where randomization can ensure such a condition.

Verifying the condition on M_a , or verifying that a certain matrix is a matrix of relations, are expensive tasks: except for some restricted cases, the algorithm of Section 5.3 does not certify that its output is indeed a matrix of relations. As such, this would lead to a Monte Carlo composition algorithm. To achieve Las Vegas composition instead, in Section 5.4, we propose an algorithm which either detects that the output mentioned above is not a matrix of relations, or uses this output to build a certified matrix of relations of slightly larger dimensions.

5.1 Matrices of Relations as Denominators of Matrix Fractions

This section relates denominators of some matrix fractions to bases of the module of relations \mathcal{M}_m and of a truncated version $\mathcal{M}_{\ell,m}$ of it.

5.1.1 Definitions.

Matrix Fractions. We first recall several notions on matrix fractions that can be found in Kailath’s book [43, Chap. 6]. Let N be in $\mathbb{K}[y]^{\ell \times m}$, let $D \in \mathbb{K}[y]^{m \times m}$ be nonsingular, and consider the rational matrix $F = ND^{-1} \in \mathbb{K}(y)^{\ell \times m}$. Then ND^{-1} is called a *right fraction description* of F . Similarly, if $F = \hat{D}^{-1}\hat{N}$, then $\hat{D}^{-1}\hat{N}$ is called a *left fraction description* of F . The right fraction ND^{-1} is said to be *irreducible* if N and D are right coprime, i.e., any right divisor common to N and D is unimodular, or equivalently $UN + VD = I_m$ for some $U \in \mathbb{K}[y]^{m \times \ell}$ and $V \in \mathbb{K}[y]^{m \times m}$ [43, Lemma 6.3.5 p. 379]. The fraction ND^{-1} is said to be *strictly proper* if for each nonzero entry of the rational matrix $F = ND^{-1}$, the degree of the numerator is less than the degree of the denominator. A matrix $F \in \mathbb{K}(y)^{\ell \times m}$ is said to be *describable in degree d* if it admits both a left and a right fraction description with denominators of degree at most d .

Truncated Module of Relations. For efficiency reasons, we consider a $\mathbb{K}[y]$ -module similar to \mathcal{M}_m , but where only the first ℓ coefficients of the polynomials are required to be 0, for some positive

integer ℓ with $\ell \leq n = \deg(f)$. Explicitly, for $\ell, m \in \mathbb{N}_{>0}$ we define the $\mathbb{K}[y]$ -modules

$$\mathcal{M}_{\ell,m}^{(a,f)} = \left\{ r(x,y) \in \mathbb{K}[x,y]_{<(m,\cdot)} \mid [a(x)^k r(x, a(x)) \text{ rem } f]_0^{\ell-1} = 0 \text{ for all } k \geq 0 \right\},$$

together with the usual simplified notation $\mathcal{M}_{\ell,m}$. They satisfy the inclusions $\mathcal{M}_{1,m} \supseteq \mathcal{M}_{2,m} \supseteq \dots \supseteq \mathcal{M}_{n,m} = \mathcal{M}_m$. If $r(x, a(x)) \text{ rem } f = 0$ then $a(x)^k r(x, a(x)) \text{ rem } f = 0$ for all $k \geq 0$, but note that this is no longer true if truncated polynomials are considered. This explains the presence of k th powers of a in the definition of $\mathcal{M}_{\ell,m}^{(a,f)}$, while they are not necessary in the definition of \mathcal{M}_m in Section 4.1.1.

The determinantal degree of $\mathcal{M}_{\ell,m}$ is denoted $\nu_{\ell,m}$. Of particular interest is the case when $\mathcal{M}_{m,m} = \mathcal{M}_m$.

5.1.2 Relation Between Bases of Relations and Denominators of Matrix Fractions.

PROPOSITION 5.1. *For $\ell, m \in \{1, \dots, n\}$, the columns of a matrix $D \in \mathbb{K}[y]^{m \times m}$ form a basis of $\mathcal{M}_{\ell,m}^{(a,f)}$ if and only if D is the denominator of an irreducible right fraction description ND^{-1} of*

$$(\mathbf{I}_\ell \ 0)(y\mathbf{I}_n - M_a)^{-1}X \in \mathbb{K}[y]^{\ell \times m},$$

the denominator of any right fraction description of this matrix is a right multiple of any such basis D .

5.1.3 Proof of Proposition 5.1.

For a matrix of rational functions $F \in \mathbb{K}(y)^{\ell \times m}$, we let

$$\mathcal{D}(F) = \{v \in \mathbb{K}[y]^m \mid Fv \in \mathbb{K}[y]^\ell\}, \quad (16)$$

which is a $\mathbb{K}[y]$ -submodule of $\mathbb{K}[y]^m$ of rank m . Then, we can establish the relation between the module $\mathcal{M}_{\ell,m}$ and the matrix in Proposition 5.1.

LEMMA 5.2. *For ℓ, m in $\{1, \dots, n\}$, one has $\mathcal{M}_{\ell,m} = \mathcal{D}((\mathbf{I}_\ell \ 0)(y\mathbf{I}_n - M_a)^{-1}X)$.*

PROOF. Taking $Y^\top = (\mathbf{I}_\ell \ 0)$, define $H(y) = Y^\top(y\mathbf{I}_n - M_a)^{-1}X$ and $H_k = Y^\top M_a^k X \in \mathbb{K}^{\ell \times m}$, so that, by power series expansion in y^{-1} ,

$$H(y) = \sum_{k \geq 0} H_k y^{-k-1} = \sum_{k \geq 0} Y^\top M_a^k X y^{-k-1}.$$

Let $r(x, y) = \sum_{0 \leq i \leq d} r_i(x)y^i \in \mathbb{K}[x, y]_{<(m,\cdot)}$ be of y -degree d , and let $v_i \in \mathbb{K}^m$ be the coefficient vector of r_i for $i = 0, \dots, d$. Then, for $k \geq 0$,

$$\left[a^k r(x, a) \text{ rem } f \right]_0^{\ell-1} = \left[\sum_{0 \leq i \leq d} a^{k+i} r_i \text{ rem } f \right]_0^{\ell-1} = \sum_{0 \leq i \leq d} \left[a^{k+i} r_i \text{ rem } f \right]_0^{\ell-1}$$

and $\left[a^{k+i} r_i \text{ rem } f \right]_0^{\ell-1}$ has coefficient vector $Y^\top M_a^{k+i} X v_i = H_{k+i} v_i$. Hence, $\left[a^k r(x, a) \text{ rem } f \right]_0^{\ell-1}$ has coefficient vector $H_k v_0 + \dots + H_{k+d} v_d$. Therefore, $r(x, y)$ is in $\mathcal{M}_{\ell,m}$ if and only if

$$H_k v_0 + \dots + H_{k+d} v_d = 0 \quad \text{for all } k \geq 0. \quad (17)$$

On the other hand, defining $v = \sum_{0 \leq i \leq d} v_i y^i$ and setting $H_k = 0$ for $k < 0$, the expansion of Hv at infinity reads

$$Hv = \sum_{k \geq 0} H_k y^{-k-1} \sum_{0 \leq i \leq d} v_i y^i = \sum_{k \geq -d} (H_k v_0 + \dots + H_{k+d} v_d) y^{-k-1}, \quad (18)$$

which implies that Equation (17) holds if and only if Hv has polynomial entries. \square

Proposition 5.1 is then a direct consequence of the following general result on matrix fractions, which is a reformulation of [43, Theorem 6.5-4 and Lemma 6.5-5, p. 441].

LEMMA 5.3. Let $F \in \mathbb{K}(y)^{\ell \times m}$ be a matrix of rational fractions. The columns of $D \in \mathbb{K}[y]^{m \times m}$ form a basis of $\mathcal{D}(F)$ if and only if D is the denominator of an irreducible right fraction description ND^{-1} of F . Besides, the denominator of any right fraction description of F is a right multiple of such a D .

5.1.4 *Notes.* The role of the truncated modules $\mathcal{M}_{\ell, m}$ is to reduce the cost of computations: we decrease the dimension of the relevant matrices using a structured left projection. The more usual approach [50] uses generic projections matrices; our choice here is similar to the one used for the efficient computation of generic resultants [75].

Although not used in this work, genericity on the left is sufficient: if $V \in \mathbb{K}^{n \times \ell}$ is generic with $\ell \in \{m, \dots, n\}$, then one has $\mathcal{M}_m = \mathcal{D}((y\mathbf{I}_n - M_a)^{-1}X) = \mathcal{D}(V^T(y\mathbf{I}_n - M_a)^{-1}X)$. The latter occurs if and only if $\text{rank}(V^T P, V^T P A, V^T P A^2, \dots) = v_m$ for a well chosen full rank matrix $P \in \mathbb{K}^{n \times v_m}$, and a restriction $A \in \mathbb{K}^{v_m \times v_m}$ of M_a to the invariant subspace generated by X [74, Lemma 4.2]. The rank condition is satisfied for a generic projection [74, Corollary 6.4 and its proof].

In terms of generators of matrix sequences, Equation (17) shows that the denominators of Proposition 5.1 are bases of modules of vector generators for the matrix sequence $\{(\mathbf{I}_\ell \ 0)M_a^k X\}_{k \geq 0}$ [50, Lemma 2.8].

5.2 Reconstructing Denominators of Matrix Fractions Via Approximant Bases

Algorithm **BLOCKTRUNCATEDPOWERS** from Section 3.3 allows one to compute a truncated power series expansion of $H(y) = X^T(y\mathbf{I}_n - M_a)^{-1}X$. When the precision of this expansion is sufficient, a basis of $\mathcal{M}_{m, m}$ can be reconstructed.

5.2.1 Definitions.

Weak Popov matrices. Let $P \in \mathbb{K}[y]^{m \times m}$ be a matrix whose column j has degree $d_j \geq 0$. The *(column) leading matrix* of P is the matrix in $\mathbb{K}^{m \times m}$ whose entry (i, j) is the coefficient of degree d_j of the entry (i, j) of P . Then P is said to be *(column) reduced* if its leading matrix is invertible. This is the case if and only if [43, Equation (24), p. 384]

$$\deg \det(P) = d_1 + \dots + d_m. \quad (19)$$

A (column) reduced matrix is in *(column) weak Popov form* if its leading matrix is invertible and upper triangular. Any submodule of $\mathbb{K}[y]^m$ has at least one basis which is in weak Popov form [5, 43].

Approximant bases. Let $F \in \mathbb{K}[[y]]^{m \times k}$ be a matrix of power series and $\sigma \in \mathbb{N}$ be a nonnegative integer. A matrix $P \in \mathbb{K}[y]^{k \times k}$ is an *approximant basis* of F at order σ if its columns form a basis of the $\mathbb{K}[y]$ -module $\{v \in \mathbb{K}[y]^k \mid Fv \equiv 0 \pmod{y^\sigma}\}$, which is free of rank k . This approximant basis is said to be *minimal* if it is reduced. Minimal approximant bases are also called σ -bases, or order bases [4, 73].

5.2.2 *Denominators from Approximant Bases.* We are going to use approximant bases for solving equations of the type of Equation (7). As pointed out in Section 1.5, we use expansions at $y = 0$ rather than infinity (see Remark 5.7).

PROPOSITION 5.4. Let $H \in \mathbb{K}(y)^{m \times m}$ be strictly proper, and δ be the determinantal degree of $\mathcal{D}(H)$ (notation from Equation (16)). Suppose that H has a power series expansion $H = \sum_{k \geq 0} S_k y^k$ at $y = 0$, with $S_k \in \mathbb{K}^{m \times m}$. Let

$$F = \left(\sum_{k=0}^{2d-1} S_k y^k \ - \mathbf{I}_m \right) \in \mathbb{K}[y]^{m \times (2m)},$$

and let

$$P = \begin{pmatrix} D & P_1 \\ N & P_2 \end{pmatrix} \in \mathbb{K}[y]^{(2m) \times (2m)}$$

be an approximant basis at order $2d$ of F in weak Popov form, with each submatrix of size $m \times m$. Then the following properties hold:

- (i) D is weak Popov; $\deg(N) < \deg(D)$; the sum of the degrees of the diagonal entries of D is $\deg \det(D)$ and satisfies $\deg \det(D) \leq \delta$.
- (ii) If $\deg \det(D) = \delta$ and each of the m rightmost columns of P has degree at least $\deg(D)$, then ND^{-1} is an irreducible description of H .
- (iii) If H is describable in degree d , then ND^{-1} is an irreducible description of H such that $\deg(D) \leq d$ and each of the m rightmost columns of P has degree at least $\deg(D)$.

The first item gives general properties of the approximant basis in weak Popov form, whereas Items (ii) and (iii) give sufficient conditions to guarantee it recovers an irreducible fraction description of H .

5.2.3 Proof of Proposition 5.4.

LEMMA 5.5. *Let $P \in \mathbb{K}[y]^{m \times m}$. If P is reduced and $B \in \mathbb{K}[y]^{m \times m}$ is a right multiple $B = PU$ with U nonsingular, then $\deg(P) \leq \deg(B)$. If $P \in \mathbb{K}[y]^{m \times m}$ is weak Popov, with diagonal degrees $d_1, \dots, d_m \in \mathbb{N}$, and $v \in \mathbb{K}[y]^m$ is a nonzero right multiple $v = Pu$ whose bottom-most entry of largest degree is in row i and has degree d , then $d_i \leq d = \max_{1 \leq j \leq m} (\deg(u_j) + d_j)$, where $u = (u_j)_{1 \leq j \leq m}$.*

PROOF. The first claim follows from the predictable degree property [43, Theorem 6.3–13, p. 387], and so does the identity $\deg(v) = d = \max_{1 \leq j \leq m} (\deg(u_j) + d_j)$ since (d_1, \dots, d_m) are also the column degrees of P by definition of a weak Popov form. The inequality $d_i \leq d$ is from [60, Lemma 1.17]. \square

We now prove Proposition 5.4. Consider an irreducible fraction description $QR^{-1} = H$ for some $Q \in \mathbb{K}[y]^{m \times m}$ and some weak Popov $R \in \mathbb{K}[y]^{m \times m}$. Since H is strictly proper we have $\deg(Q) < \deg(R)$ and, more precisely, the i th column of Q has degree less than the i th column of R . Thus the i th column of $(\frac{R}{Q})$ has its bottom-most entry of largest degree in row i ; let d_i be this degree.

In Item (i), the first two claims follow from the definition of P being weak Popov. In particular D is column reduced, hence Equation (19) shows that $\deg \det(D)$ is the sum of column degrees of D , which is also the sum of diagonal degrees of D since D is weak Popov. The identity $(H - I_m)(\frac{R}{Q}) = 0$ implies $F(\frac{R}{Q}) = 0 \bmod y^{2d}$, and therefore $(\frac{R}{Q})$ is a right multiple of P . Hence, by Lemma 5.5, d_i is at least the degree of the i th column of P , which is the degree of the i th column of D ; it follows that $\deg \det(D) \leq d_1 + \dots + d_m$. On the other hand, since R is reduced and since by Lemma 5.3 its columns form a basis of $\mathcal{D}(H)$ we have $d_1 + \dots + d_m = \deg \det(R) = \delta$, proving the last claim of Item (i).

Concerning Item (ii), the assumption $\deg \det(D) = \delta = \deg \det(R)$ implies that the sum of column degrees of D is $d_1 + \dots + d_m$, while as showed above the i th column of D has degree at most d_i . Thus D has the same column degrees (d_1, \dots, d_m) as R . In particular $\deg(D) = \deg(R) > \deg(Q)$. Then, since by assumption the m rightmost columns of P have bottom-most entries of largest degree in rows at least $m + 1$ and of degree at least $\deg(D)$, one can deduce from Lemma 5.5 that $(\frac{R}{Q})$ is a right multiple of the leftmost m columns of P . Indeed, let $u = (\frac{u_1}{u_2}) \in \mathbb{K}[y]^{2m}$, with u_1 and u_2 each of dimension m , such that the i th column of $(\frac{R}{Q})$ is $Pu = (\frac{Du_1 + P_1u_2}{Nu_1 + P_2u_2})$; we want to prove $u_2 = 0$. Using the last identity in Lemma 5.5 on each of the weak Popov matrices P , D , and P_2 , we observe that $\deg(Pu) = \max(\deg(Du_1), \deg(P_2u_2))$; note $\deg(Pu) = d_i \leq \deg(D)$ by construction. On the other hand, since all diagonal degrees of P_2 are at least $\deg(D)$, Lemma 5.5 shows $\deg(D) \leq \deg(P_2u_2)$, provided that $u_2 \neq 0$, which we now assume by contradiction. This implies $\deg(Du_1) \leq \deg(Pu) = \deg(P_2u_2) = \deg(D)$, hence $\max_{1 \leq j \leq m} d_j + \deg(u_{1,j}) \leq \deg(D)$ using Lemma 5.5, where $u_1 = (u_{1,j})_{1 \leq j \leq m}$. Now by definition of weak Popov forms, the j th column of N has degree less than d_j for $1 \leq j \leq m$, hence $\deg(Nu_1) < \max_{1 \leq j \leq m} d_j + \deg(u_{1,j})$. This gives $\deg(Nu_1 + P_2u_2) = \deg(D)$,

which is a contradiction since $Nu_1 + P_2u_2$ is the i th column of Q and has degree strictly less than d_i , itself at most $\deg(D)$. So, $u_2 = 0$. Gathering this over all columns $1 \leq i \leq m$, this means $(\begin{smallmatrix} R \\ Q \end{smallmatrix}) = (\begin{smallmatrix} D \\ N \end{smallmatrix})U = (\begin{smallmatrix} DU \\ NU \end{smallmatrix})$ for some $U \in \mathbb{K}[y]^{m \times m}$, and U is unimodular since R and D are nonsingular with $\deg \det(R) = \deg \det(D)$. Hence $H = QR^{-1} = ND^{-1}$ and the fraction ND^{-1} is irreducible.

The following proof of Item (iii) reflects that of [31, Lemma 3.7]. The assumption implies first the existence of a left fraction $H = \hat{R}^{-1}\hat{Q}$ with $\deg(\hat{Q}) < \deg(\hat{R}) \leq d$, and second the degree bound $\deg(R) \leq d$ thanks to the degree minimality of reduced bases (see Lemma 5.5). The above paragraph shows in particular $\deg(D) \leq \max_i(d_i) = \deg(R) \leq d$.

Now, since $\hat{R}(\sum_{0 \leq k < 2d} S_k y^k) \equiv \hat{Q} \pmod{y^{2d}}$, multiplying on the left by \hat{R} both sides of $F(\begin{smallmatrix} D \\ N \end{smallmatrix}) \equiv 0 \pmod{y^{2d}}$ shows that $\hat{Q}D - \hat{R}N$ is a right multiple of $y^{2d}\hat{R}$. On the other hand, $\hat{Q}D - \hat{R}N$ has degree less than $2d$. Hence it is zero, and $H = \hat{R}^{-1}\hat{Q} = ND^{-1}$. To prove that the latter fraction is irreducible, assume by contradiction that D and N have a nonsingular common right divisor $B \in \mathbb{K}[y]^{m \times m}$, with $\deg \det(B) > 0$. Then $H = (NB^{-1})(DB^{-1})^{-1}$ yields $F(\begin{smallmatrix} DB^{-1} \\ NB^{-1} \end{smallmatrix}) \equiv 0 \pmod{y^{2d}}$, and $P \operatorname{diag}(B^{-1}, I_m)$ is a right multiple of P (since P is a basis): this is impossible since $\deg \det(P \operatorname{diag}(B^{-1}, I_m)) < \deg \det(P)$.

It remains to prove the last degree assertion. By contradiction, assume that P has a column $(\begin{smallmatrix} v_0 \\ v_1 \end{smallmatrix})$ of index larger than m with v_0 and v_1 in $\mathbb{K}[y]^m$ both of degree less than d . Then an argument similar to the one above shows that $\hat{Q}v_0 - \hat{R}v_1 = 0$. Altogether, we obtain a matrix $(\begin{smallmatrix} D & v_0 \\ N & v_1 \end{smallmatrix})$ of rank $m+1$ which is in the right kernel of $(\hat{Q} \quad \hat{R}) \in \mathbb{K}[y]^{m \times (2m)}$ whose rank is m : this is not possible.

5.2.4 Notes. The existence of appropriate left and right descriptions of H was used before for the reconstruction of matrix fractions within the approximant framework [31, Section 3.2]. Our proof is similar to that of [31, Lemma 3.7], with the additional use of the weak Popov form.

Reduced forms were introduced [78] as a way to get a better control over the degrees when computing with polynomial matrices and matrix fractions, see e.g., [43, Lemma 6.3-11, p. 385] for proper fractions, [43, Theorem 6.3-13, p. 387] for a predictable degree property, and [43, Theorem 6.5-10, p. 458] concerning the *minimality* of the column degrees. Weak Popov forms were introduced later [5, 59] (under the name quasi-Popov and up to column permutation) and provide a refined degree control as illustrated by Lemma 5.5.

5.3 Candidate Basis of Relations

Algorithm **CANDIDATEBASIS** takes as input a polynomial $f \in \mathbb{K}[x]$ of degree n with $f(0) \neq 0$, a polynomial $a \in \mathbb{K}[x]_{<n}$ such that $\gcd(a, f) = 1$, and two positive integers $m \leq n$ and d . With this input, it computes an $m \times m$ matrix of degree at most $2d$.

The algorithm starts by computing a truncated expansion at order $2d$ of $H = X^T(yI_n - M_a)^{-1}X$ at $y = 0$ using Algorithm **BLOCKTRUNCATEDPOWERS**. Then, it computes a $2m \times 2m$ minimal approximant basis as in Proposition 5.4 using the algorithm **PM-BASIS** of [31], and extracts a potential basis of relations. In some cases, we can certify that it is indeed basis of \mathcal{M}_m , but it is not always possible to do so; a flag is returned to indicate this. This certification is actually an optimization, rather than strictly necessary; Section 5.4 discusses this question in more detail.

PROPOSITION 5.6. *Given $f \in \mathbb{K}[x]$ of degree n with $f(0) \neq 0$, $a \in \mathbb{K}[x]_{<n}$ such that $\gcd(a, f) = 1$, and two positive integers $m \leq n$ and d , Algorithm **CANDIDATEBASIS** uses $\tilde{O}(m^\omega d + c(n, m, d))$ operations in \mathbb{K} , with $c(\cdot)$ from Equation (2), and computes a weak Popov matrix $R \in \mathbb{K}[y]_{\leq 2d}^{m \times m}$. The matrix R is a basis of $\mathcal{M}_m^{(a, f)}$ in either of the following cases:*

ALGORITHM 5.1 CANDIDATEBASIS(f, a, m, d)

Input: $f \in \mathbb{K}[x]$ of degree n , with $f(0) \neq 0$, $a \in \mathbb{K}[x]_{<n}$ with $\gcd(a, f) = 1$, $m \leq n$ and d in $\mathbb{N}_{>0}$

Output: a weak Popov matrix $R \in \mathbb{K}[y]_{\leq 2d}^{m \times m}$ and a flag in $\{\text{CERT}, \text{NoCERT}\}$; R is a basis of \mathcal{M}_m in either of the following cases:

- $v_{m,m} = v_m$ and $H = X^T(yI_n - M_a)^{-1}X$ is describable in degree d , in which case $\deg(R) \leq d$
- the flag is CERT, which implies $v_{m,m} = v_m = n$

1: \triangleright Truncated expansion of H : compute $S_k = -X^T M_a^{-k-1} X$ for $0 \leq k < 2d$ using Algorithm 3.8

$(A_{i,k}^*)_{\substack{0 \leq i < m \\ 0 \leq k < 2d+1}} \leftarrow \text{BLOCKTRUNCATEDPOWERS}(f, a^{-1} \bmod f, m, 2d+1)$

$S_{i,k} \in \mathbb{K}^m \leftarrow$ vector of coefficients of $-A_{i,k+1}^* \in \mathbb{K}[x]_{<m}$, for $0 \leq i < m$ and $0 \leq k < 2d$

2: \triangleright Fraction reconstruction: compute approximant basis using algorithm from [31, 42]

$F \in \mathbb{K}[y]_{\leq 2d}^{m \times 2m} \leftarrow (\sum_{0 \leq k < 2d} S_k y^k - I_m)$ where $S_k = (S_{0,k} \ \cdots \ S_{m-1,k}) \in \mathbb{K}^{m \times m}$

$P \in \mathbb{K}[y]_{\leq 2d}^{2m \times 2m} \leftarrow \text{PM-BASIS}(F^T, 2d, 0)^T$, with P in weak Popov form

3: \triangleright Return candidate matrix and result of basic certification

$R \leftarrow P_{1..m, 1..m}$

if the sum of diagonal degrees of R is equal to n

\triangleright Item (ii) of Proposition 5.4

and each of the m rightmost columns of P has degree $\geq \deg(R)$

then return (R, CERT) else return (R, NoCERT)

— The determinantal degree $v_{m,m}^{(a,f)}$ is equal to $v_m^{(a,f)}$ and the fraction $H(y) = X^T(yI_n - M_a)^{-1}X$ is describable in degree d ; in that case we further have $\deg(R) \leq d$; if in addition $v_m^{(a,f)} = n$ then the flag is CERT.

— The flag is CERT, which implies $v_{m,m}^{(a,f)} = v_m^{(a,f)} = n$.

PROOF. Proposition 3.7 shows that Step 1 uses $\tilde{O}(m^2d + c(n, m, d))$ operations to compute the vectors $S_{i,k} \in \mathbb{K}^m$. These vectors are such that the matrices S_k built in Step 2 are $S_k = -X^T M_a^{-k-1} X$; as a result, the matrix $S = \sum_{0 \leq k < 2d} S_k y^k$ considered at Step 2 is the power series expansion of H truncated at order $2d$. Then Step 2 correctly computes a weak Popov approximant basis P for $F = (S - I_m)$ at order $2d$ with $\deg(P) \leq 2d$ using $\tilde{O}(m^\omega d)$ operations [31, Theorem 2.4] [42, Proposition 3.2]. (Note that transposes are used at Step 2 because in References [31, 42] approximant bases are considered row-wise, rather than column-wise here.) The claimed cost bound for Algorithm CANDIDATEBASIS is proved.

For the first item, assume that H is describable in degree d . Then Item (iii) of Proposition 5.4 ensures that R is the denominator of an irreducible right fraction description of H , that $\deg(R) \leq d$, and that each of the m rightmost columns of P has degree at least $\deg(R)$. From Proposition 5.1 we obtain that R is a basis of $\mathcal{M}_{m,m}$, hence a basis of \mathcal{M}_m when $v_{m,m} = v_m$. This also proves the last claim of the item: if $v_{m,m} = v_m = n$, then $\deg \det(R) = n$ and this is the sum of diagonal degrees of R since this matrix is in weak Popov form; hence the flag CERT is returned.

For the second item, assume that the output flag is CERT. Then, the sum of diagonal degrees of R is n ; according to Item (i) of Proposition 5.4, this sum is also $\deg \det(R)$ and is at most δ , the determinantal degree of bases of $\mathcal{D}(H)$. On the other hand, Proposition 5.1 implies that δ is the determinantal degree $v_{m,m}$ of $\mathcal{M}_{m,m}$. Hence $n = \deg \det(R) \leq \delta = v_{m,m}$, from which we deduce $\deg \det(R) = \delta = v_{m,m} = v_m = n$, since $v_{m,m} \leq v_m \leq n$ always holds. Since the output flag is CERT we know in addition that each of the m rightmost columns of P has degree at least $\deg(R)$. Thus, Item (ii) of Proposition 5.4 applies, and R is the denominator of an irreducible right fraction description of H . We conclude as done for the first item that R is a basis of \mathcal{M}_m . \square

Remark 5.7. The assumption that f and a are coprime is used here to ensure that M_a is invertible, so that the expansion $H = \sum_{k \geq 0} S_k y^k = \sum_{k \geq 0} (-X^T M_a^{-k-1} X) y^k$ at $y = 0$ can be used for fraction reconstruction. This is different from what happened in the proof of Proposition 5.1, where we used the expansion at infinity $H = \sum_{k \geq 0} H_k y^{-k-1}$. The latter expansion involves powers of M_a and thus our formalism remains close to that of Reference [50] with Krylov sequences. From an algorithmic point of view, expansions at $y = 0$ allow us to use directly the existing efficient algorithms for matrix fraction reconstruction [31, 42], and exploit their properties.

This assumption on $\gcd(f, a)$ is harmless in our context: in the computation of $g(a) \bmod f$, one can instead evaluate $g(y - c)$ at $y = a + c$ for a randomly chosen $c \in \mathbb{K}$, ensuring $\gcd(a + c, f) = 1$ with good probability. See Step 2 in Algorithm **MODULARCOMPOSITIONBASECASE**.

Notes. For some families of approximation instances, **PM-BASIS** has been used to design faster minimal approximant basis algorithms [42, 79]. Yet, the instances considered here are ones where **PM-BASIS** is the fastest known algorithm.

A candidate matrix of relations in $\mathbb{K}[y]_{\leq 2d}^{m \times m}$ corresponds to m polynomials in $\mathbb{K}[x, y]_{\leq (m, 2d)}$. Using Algorithm **SIMULTANEOUSBIVARIATEMODULARCOMPOSITION** to verify that the evaluations of these polynomials at $a \bmod f$ are zero uses $\tilde{O}(c(n, m, d^2))$ operations in \mathbb{K} , by Lemma 3.3. For the values of m and d used to obtain the exponent $\kappa < 1.43$ in our main algorithm, this is $O(n^{2.55})$, and thus too costly.

5.4 Certified Matrix of Relations

In general, when Algorithm **CANDIDATEBASIS** does not certify its result, we do not know methods to verify that the matrix it returns is a matrix of relations within our complexity bound.

Instead, from a matrix R computed by Algorithm **CANDIDATEBASIS**, Algorithm **MATRIXOFRELATIONS** either detects that it is not a matrix of relations of \mathcal{M}_m , or constructs from R a matrix of relations of $\mathcal{M}_{m'}$ of degree at most $2d$, for some $m' < 2m$. This is the key toward making our modular composition algorithm Las Vegas, rather than Monte Carlo.

To achieve this, instead of evaluating all columns of R at $a \bmod f$, Algorithm **MATRIXOFRELATIONS** evaluates only two polynomials built randomly from these columns (and only one polynomial in the special case $m = 1$), which is within our target complexity using the Nüsken–Ziegler algorithm. If these evaluations are not both zero, then R was not a matrix of relations. Otherwise, the algorithm constructs a Sylvester matrix from these two vectors (see e.g., [26, Section 6.3] for the definition and properties of the Sylvester matrix). When this matrix is nonsingular, it is a matrix of relations of a module $\mathcal{M}_{m'}$ for $m' \leq \max(1, 2(m - 1))$; since m' cannot be much larger than m , this matrix can be used for efficient composition.

PROPOSITION 5.8. *Given $f \in \mathbb{K}[x]$ of degree n with $f(0) \neq 0$, $a \in \mathbb{K}[x]_{<n}$ with $\gcd(a, f) = 1$, two positive integers $m (\leq n)$ and d , and $(r_i)_{3 \leq i \leq m} \in \mathbb{K}^{m-2}$, Algorithm **MATRIXOFRELATIONS** uses $\tilde{O}(m^\omega d + c(n, m, d))$ operations in \mathbb{K} , with $c(\cdot)$ from Equation (2), and returns either FAIL or a matrix of relations $R' \in \mathbb{K}[y]_{\leq 2d}^{m' \times m'}$ of $\mathcal{M}_{m'}^{(a, f)}$ where $m' \leq \max(1, 2(m - 1))$.*

If $v_{m, m}^{(a, f)} = v_m^{(a, f)}$, the fraction $H = X^T (yI_n - M_a)^{-1} X$ is describable in degree d , and (r_3, \dots, r_m) are chosen uniformly and independently at random from a finite subset S of \mathbb{K} , then failure happens with probability at most $(m - 1)/\text{card}(S)$ and in case of success, $\deg(R') \leq d$.

PROOF. If **FLAG** = **CERT** at Step 1, then from the second item of Proposition 5.6 an appropriate matrix of relations is returned. Now assume that **FLAG** = **NoCERT** and Algorithm **MATRIXOFRELATIONS** does not return FAIL. If $m = 1$ then the relation has been checked at Step 2, proving the result. Otherwise, let $R' \in \mathbb{K}[y]_{\leq 2d}^{m' \times m'}$ be the output matrix, which is constructed from the polynomials r, s of x -degree less than m ; in particular, $m' = \deg_x(r) + \deg_x(s) \leq 2(m - 1)$

ALGORITHM 5.2 MATRIXOFRELATIONS($f, a, m, d, (r_i)_{3 \leq i \leq m}$)

Input: $f \in \mathbb{K}[x]$ of degree n , with $f(0) \neq 0$, $a \in \mathbb{K}[x]_{<n}$ with $\gcd(a, f) = 1$,
 $m \leq n$ and d in $\mathbb{N}_{>0}$, $(r_i)_{3 \leq i \leq m} \in \mathbb{K}^{m-2}$

Output: either FAIL or a matrix $R' \in \mathbb{K}[y]_{\leq 2d}^{m' \times m'}$ of relations of $\mathcal{M}_{m'}$ with $m' \leq \max(1, 2(m-1))$

- 1: \triangleright Use Algorithm 5.1 to find a candidate basis of relations
 $(R, \text{FLAG}) \in \mathbb{K}[y]_{\leq 2d}^{m \times m} \times \{\text{CERT}, \text{NOCERT}\} \leftarrow \text{CANDIDATEBASIS}(f, a, m, d)$
if FLAG = CERT **then return** R
- 2: \triangleright Case $m = 1$, check that $R_{1,1} \in \mathbb{K}[y]_{<2d+1}$ annihilates $a \bmod f$
if $m = 1$ **then**
if MODULARCOMPOSITION-BRENTKUNG($f, a, R_{1,1}$) $\neq 0$ **then return** FAIL
else return R
- 3: \triangleright Build candidate relations and verify them
 $r(x, y) \leftarrow R_{*,1}; s(x, y) \leftarrow R_{*,2} + r_3 R_{*,3} + \cdots + r_m R_{*,m}$ \triangleright both in $\mathbb{K}[x, y]_{<(m, 2d+1)}$
if BIVARIATEMODULARCOMPOSITION(f, a, r) $\neq 0$
or BIVARIATEMODULARCOMPOSITION(f, a, s) $\neq 0$
then return FAIL \triangleright Algorithm 3.5
if $m = 2$ **then return** R
- 4: \triangleright Construct and return the Sylvester matrix of f and s , if it is nonsingular
if $\gcd_x(r, s) \neq 1$ **then return** FAIL $\triangleright r$ and s not coprime as elements of $\mathbb{K}(y)[x]$
return the Sylvester matrix of (r, s) as in [26, Section 6.3, Equation (5)], with rows in reversed order, viewing r and s as polynomials in x over $\mathbb{K}[y]$

[26, Section 6.3]. The fact that the test at Step 4 has not failed ensures that r and s are coprime as univariate polynomials in $\mathbb{K}(y)[x]$, and therefore R' is nonsingular [26, Corollary 6.15]. Furthermore, since the tests at Step 3 have not failed, r and s are relations of \mathcal{M}_m . It follows that the columns of R' , which are by construction multiples of r and s in $\mathbb{K}[x, y]$ represented as vectors in $\mathbb{K}[y]^{m'}$, are relations of $\mathcal{M}_{m'}$. Besides, the construction of the Sylvester matrix does not increase the y -degree, hence $\deg(R') \leq \deg(R) \leq 2d$. We have proved the fact that if the output is not FAIL, then it is a matrix of relations of $\mathcal{M}_{m'}$.

For the complexity bound, the cost for finding R is given in Proposition 5.6, while the ones for checking that $R_{1,1}$, r and s are relations are given in Propositions 3.1 and 3.4. As for the gcd test at Step 4, it can be done via the resultant of r and s with respect to x , computed using $\tilde{O}(m^2d)$ operations [67].

It remains to prove the third assertion and the probability bound. Since when $\text{FLAG} = \text{CERT}$ a basis is returned with no randomization, assume $\text{FLAG} = \text{NOCERT}$. The assumptions here and the first item of Proposition 5.6 ensure that R is a basis of \mathcal{M}_m with $\deg(R) \leq d$, hence $\deg(R') \leq d$. In that case failure never occurs at Step 2 for $m = 1$. It never occurs either at Step 3 for $m \geq 2$, and r and s are relations of \mathcal{M}_m . The columns of R represent bivariate polynomials $b_1, \dots, b_m \in \mathbb{K}[x, y]_{<(m, d+1)}$ and we claim that $\gcd_x(b_1, \dots, b_m) = 1$, meaning that there is a $\mathbb{K}[y]$ -linear combination of b_1, \dots, b_m in $\mathbb{K}[y] \setminus \{0\}$. Since R is nonsingular, the first column of a transformation for the (upper triangular) Hermite normal form of R provides such a combination. It follows that FAIL is returned with probability at most $(m-1)/\text{card}(S)$ at Step 4 [26, Theorem 6.46]. \square

Note. The computation of CERT by Algorithm CANDIDATEBASIS is only an optimization. Algorithm MATRIXOFRELATIONS works as it is, even if CERT is never returned. When the candidate matrix R at Step 1 is not a matrix of relations, this is often detected at Step 3, but not always. Even if it is not detected, it suffices to find two coprime polynomials $r(x, y)$ and $s(x, y)$ that are relations

to ensure that Algorithm `MATRIXOFRELATIONS` returns a matrix of relations. For example, it may happen that R is not a matrix of relations but some columns of it still give low-degree relations of \mathcal{M}_m .

6 CHANGE OF BASIS

In this section, we present an algorithm for performing a change of basis in $\mathbb{A} = \mathbb{K}[x]/\langle f \rangle$. This algorithm is used in a randomized manner in Section 8, in order to handle arbitrary inputs with good complexity bounds. Our approach is based on an extension of the approximant bases used in Section 5; we start with necessary definitions.

6.1 Definitions

We use an extension of the forms of polynomial matrices introduced in Section 5.2.1, called *shifted* forms [5, 73]. For a given tuple $t = (t_1, \dots, t_m) \in \mathbb{Z}^m$ and a column vector $v \in \mathbb{K}[y]^m$, the t -shifted degree of v is $\max_{1 \leq i \leq m} (\deg(v_i) + t_i)$. Then, for a matrix $P \in \mathbb{K}[y]^{m \times m}$ whose j th column has t -shifted degree $d_j \in \mathbb{Z}$, the *(column) t -shifted leading matrix* of P is the matrix in $\mathbb{K}^{m \times m}$ whose entry (i, j) is the coefficient of degree $d_j - t_i$ of the entry (i, j) of P . Then P is said to be *t -shifted weak Popov* if this t -shifted leading matrix is invertible and upper triangular.

We also need the corresponding normal form: P is said to be *t -shifted Popov* if it is t -shifted weak Popov and its *row* leading matrix is the identity of $\mathbb{K}^{m \times m}$ [5, 43]. For a given t , any submodule of rank m of $\mathbb{K}[y]^m$ admits a unique basis in t -shifted Popov normal form [5, Theorem 3.7]. By definition, t -shifted Popov matrices are also (nonshifted) *row* reduced; in particular, Hermite normal forms are t -shifted Popov for an appropriate choice of t , hence are *row* reduced.

Row reduced matrices allow for a *division with remainder* with stronger properties than the one for general nonsingular matrices presented in Section 4.2.1; namely, they ensure uniqueness of the remainder. Precisely, if a matrix $P \in \mathbb{K}[y]^{m \times m}$ is row reduced, for any vector $v \in \mathbb{K}[y]^m$ there exists a unique vector $\tilde{v} \in \mathbb{K}[y]^m$ such that $v - \tilde{v}$ is a right multiple of P and the i th entry of \tilde{v} has degree less than the i th row of P [43, Theorem 6.3-15, p. 389].

We also use the fact that, by definition, for any block decomposition $P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}$ of a matrix P , if P is in Hermite (resp. t -shifted Popov) normal form, then:

- P_{11} and P_{22} are in Hermite normal form (resp. in shifted Popov normal form with respect to the corresponding subtuple of t);
- each column of P_{12} (resp. P_{21}) is its own remainder in the division by P_{11} (resp. P_{22}).

Finally, t -shifted forms induce the notion of *t -shifted approximant bases* [5, 42, 79], which are approximant bases (see Section 5.2.1) in t -shifted Popov normal form.

6.2 Inverse Modular Composition and Change of Basis Via Approximant Bases

Let f be in $\mathbb{K}[x]$ of degree n . A core ingredient for the randomization in our composition algorithm is an instance of *inverse modular composition*, which is used to change the basis of $\mathbb{A} = \mathbb{K}[x]/\langle f \rangle$ from $(1, x, \dots, x^{n-1})$ to $(1, \gamma, \dots, \gamma^{n-1}) \bmod f$, for some $\gamma \in \mathbb{K}[x]$ whose minimal polynomial μ_γ modulo f has degree n . This change of basis induces the \mathbb{K} -algebra isomorphism

$$\phi_\gamma : \mathbb{A} \rightarrow \mathbb{K}[y]/\langle \mu_\gamma \rangle, \quad (20)$$

which maps any $u \in \mathbb{A}$ to v such that $v(\gamma) \equiv u \bmod f$. Given a in $\mathbb{K}[x]_{<n}$, this section explains how to compute the unique polynomial representative $\alpha \in \mathbb{K}[y]_{<n}$ of $\phi_\gamma(a \bmod f)$, i.e., the unique $\alpha \in \mathbb{K}[y]_{<n}$ such that $\alpha(\gamma) \bmod f = a$.

Reversing the path followed in our modular composition approach, we first find a bivariate $\tilde{\alpha} \in \mathbb{K}[x, y]$ such that $\tilde{\alpha} - \alpha \in \mathcal{M}_m^{(\gamma, f)}$, hence $\tilde{\alpha}(x, \gamma) \equiv \alpha(\gamma) \bmod f$. Then, the univariate solution α

is recovered from $\tilde{\alpha}$ and a basis of relations R of $\mathcal{M}_m^{(y,f)}$ by reversing the division from Equation (15); this corresponds to a division by the Hermite normal form of R .

Our algorithm for computing $\tilde{\alpha}$ can be seen as a generalization to $m \geq 1$ of Shoup's algorithm for computing α , mentioned in Section 1.4.2. The latter algorithm deals with the case $m = 1$: from the power projections $(\ell(1), \ell(y), \dots, \ell(y^{2n-1}))$ and $(\ell(a), \ell(ya), \dots, \ell(y^{n-1}a))$, it obtains both α and μ_y by solving two Padé approximation problems. In the matrix case $m \geq 1$, Algorithm **CHANGEOFBASIS** computes solutions to equations similar to Equations (7) and (9) given in the introduction. These are matrix generalizations of the Padé approximation problems; their solutions provide, respectively, a basis of relations R of $\mathcal{M}_m^{(y,f)}$ and $\tilde{\alpha}$.

In more details, Steps 2 and 3 first compute the power series expansions involved in Equations (7) and (9), which amounts to a type of generalized power projections. Then both approximation problems are solved at once using shifted approximant bases:

- The choice of the first $2m$ columns of $F = (S \ -I_m \ s)$ at Step 4, which are the same as in Step 2 of Algorithm **CANDIDATEBASIS**, and the use of a corresponding “zero shift” (first $2m$ entries of the tuple t at Step 4), make this equivalent to the computation in Section 5.3 (compare Steps 3 to 5 of Algorithm **CHANGEOFBASIS** to Steps 1 to 3 of Algorithm **CANDIDATEBASIS**). This yields a basis R of $\mathcal{M}_m^{(y,f)}$.
- Equation (9) is solved thanks to an additional series expansion in F (its last column), and the use of a sufficiently large shift (the last entry $2d$ of the tuple t). This yields a bivariate polynomial $\tilde{\alpha}$ that is the remainder of the requested α in the division by R .

Finally, this sought α can be obtained by reversing this division, using a Hermite normal form computation which also provides the minimal polynomial μ_y (Steps 6 and 7).

The assumptions in Algorithm **CHANGEOFBASIS** yield a slightly stronger statement in Proposition 6.1 than in Proposition 5.6 for Algorithm **CANDIDATEBASIS**. Indeed, we suppose that y is such that $\deg(\mu_y) = n$, whereas we make no such assumption in Algorithm **CANDIDATEBASIS**. From the module properties in Proposition 4.1, we deduce that $\deg(\mu_y) = n$ implies $v_m^{(y,f)} = n$, which allows us to certify the basis of relations R when FAIL is not returned.

Algorithm **CHANGEOFBASIS** may still return FAIL; Section 8 shows that when it is called with a random y , then with high probability, it does not fail, at least under some assumptions on f .

PROPOSITION 6.1. *Given $f \in \mathbb{K}[x]$ of degree n with $f(0) \neq 0$, y and a in $\mathbb{K}[x]_{<n}$, $m \leq n$ and d in $\mathbb{N}_{>0}$, Algorithm **CHANGEOFBASIS** uses $\tilde{O}(m^\omega d + c(n, m, d))$ operations in \mathbb{K} , with $c(\cdot)$ from Equation (2), to return either FAIL or (R, μ, α) where $R \in \mathbb{K}[y]_{\leq 2d}^{m \times m}$ is the Popov basis of $\mathcal{M}_m^{(y,f)}$, μ is the minimal polynomial μ_y of $y \bmod f$ and has degree n , and α is the unique polynomial in $\mathbb{K}[y]_{<n}$ such that $\alpha(y) \equiv a \bmod f$.*

If $\gcd(y, f) = 1$, $v_{m,m}^{(y,f)} = v_m^{(y,f)}$, $\deg(\mu_y) = n$ and the fraction $H = X^T(yI_n - M_y)^{-1}X$ is describable in degree d , then the output is not FAIL; in that case we further have $\deg(R) \leq d$.

PROOF. We start by showing that if the algorithm does not fail, then the truncated module $\mathcal{M}_{m,m}^{(y,f)}$ and the module of relations $\mathcal{M}_m^{(y,f)}$ are equal, and R is a basis of this module.

Steps 1 to 4: the approximant basis \tilde{P} . If the test at Step 1 does not fail then the specifications for Steps 2 and 3 are met. At Step 2, Algorithm **TRUNCATEDPOWERS** returns $r_k = [a \ y^{-k-1} \ \text{rem } f]_0^{m-1}$ for $0 \leq k < 2d$ using $\tilde{O}(c(n, m, d))$ operations, according to Proposition 3.6. Thus the coefficient vector $s_k \in \mathbb{K}^m$ of r_k is $X^T M_y^{-k-1} v_a$, where $v_a \in \mathbb{K}^n$ is the coefficient vector of a , so that the polynomial vector $s(y)$ computed at Step 2 is the power series expansion of $-X^T(yI_n - M_y)^{-1}v_a$ truncated

ALGORITHM 6.1 CHANGEOFBASIS(f, γ, a, m, d)

Input: f of degree n in $\mathbb{K}[x]$, with $f(0) \neq 0$, $\gamma \in \mathbb{K}[x]_{<n}$, $a \in \mathbb{K}[x]_{<n}$, $m \leq n$ and d in $\mathbb{N}_{>0}$
Output: either FAIL or (R, μ, α) where $R \in \mathbb{K}[y]_{\leq 2d}^{m \times m}$ is the Popov basis of $\mathcal{M}_m^{(\gamma, f)}$, μ is the minimal polynomial of γ in $\mathbb{K}[x]/\langle f \rangle$ and has degree n , and $\alpha \in \mathbb{K}[y]_{<n}$ with $\alpha(\gamma) \equiv a \pmod{f}$

- 1: **if** $\gcd(\gamma, f) \neq 1$ **then return** FAIL
- 2: \triangleright Truncated expansion of $-X^T(yI_n - M_\gamma)^{-1}v_a$ using Algorithm 3.7, $v_a \in \mathbb{K}^n$ is the coefficient vector of a
 $(r_k)_{0 \leq k < 2d} \leftarrow \text{TRUNCATEDPOWERS}(f, \gamma^{-1} \bmod f, \gamma^{-1}a \bmod f, m, 2d)$
 $s \in \mathbb{K}[y]^m \leftarrow \sum_{0 \leq k < 2d} s_k y^k$ where $s_k \in \mathbb{K}^m$ is the coefficient vector of r_k
- 3: \triangleright Truncated expansion of $X^T(yI_n - M_\gamma)^{-1}X$ using Algorithm 3.8 (analogous to Step 1 of Algorithm 5.1)
 $(\Gamma_{i,k})_{\substack{0 \leq i < m \\ 0 \leq k < 2d+2}} \leftarrow \text{BLOCKTRUNCATEDPOWERS}(f, \gamma^{-1} \bmod f, m, 2(d+1))$
 $S_{i,k} \in \mathbb{K}^m \leftarrow$ vector of coefficients of $-\Gamma_{i,k+1} \in \mathbb{K}[x]_{<m}$, for $0 \leq i < m$ and $0 \leq k < 2d$
 $S \in \mathbb{K}[y]_{\leq 2d}^{m \times m} \leftarrow \sum_{0 \leq k < 2d} S_k y^k$ where $S_k = (S_{0,k} \ \dots \ S_{m-1,k}) \in \mathbb{K}^{m \times m}$
- 4: \triangleright Fraction reconstruction using [31, 42] (analogous to Step 2 of Algorithm 5.1)
 $F \in \mathbb{K}[y]_{\leq 2d}^{m \times (2m+1)} \leftarrow (S(y) \ - I_m \ s(y))$
 $t \in \mathbb{N}^{2m+1} \leftarrow (0, \dots, 0, 2d)$
 $\bar{P} \in \mathbb{K}[y]_{\leq 2d}^{(2m+1) \times (2m+1)} \leftarrow \text{POPOV-PM-BASIS}(F^T, 2d, t)^T$, with \bar{P} in t -shifted Popov normal form
 $P \leftarrow \bar{P}_{1..2m, 1..2m}$
 $v_{\bar{\alpha}} \in \mathbb{K}[y]_{<\deg(R)}^m \leftarrow \bar{P}_{1..m, 2m+1} \quad \triangleright$ represents $\bar{\alpha}(x, y)$, expected to satisfy $\bar{\alpha}(x, \gamma) \equiv a \pmod{f}$
- 5: \triangleright Ensure R is a basis of $\mathcal{M}_m^{(\gamma, f)}$, from Item (ii) of Proposition 5.4 (analogous to Step 3 of Algorithm 5.1)
 $R \leftarrow P_{1..m, 1..m}$
if the sum of diagonal degrees of R is less than n
or among the m rightmost columns of P , one has degree $< \deg(R)$ **then return** FAIL
- 6: \triangleright Compute μ_γ , and ensure it has degree n
 $T \in \mathbb{K}[y]^{m \times m} \leftarrow$ Hermite normal form of R \triangleright using [53, Algorithm 1 and 3]
 $\mu \in \mathbb{K}[y] \leftarrow T_{1,1};$ **if** $\deg(\mu) < n$ **then return** FAIL
- 7: \triangleright Deduce α and return
 $\alpha \in \mathbb{K}[y]_{<n} \leftarrow \bar{\alpha}_1 - (T_{1,2}\bar{\alpha}_2 + \dots + T_{1,m}\bar{\alpha}_m) \text{ rem } \mu$, where $v_{\bar{\alpha}} = (\bar{\alpha}_1 \ \dots \ \bar{\alpha}_m)$
return (R, μ, α)

at order $2d$. From Proposition 3.7, the computation of $S(y)$ at Step 3 uses $\tilde{O}(m^2d + c(n, m, d))$ operations; S is the power series expansion of $X^T(yI_n - M_\gamma)^{-1}X$ truncated at order $2d$.

Step 4 computes the t -shifted Popov approximant basis \bar{P} for $F = (S - I_m s)$ at order $2d$, which uses $\tilde{O}(m^\omega d)$ operations [31, Theorem 2.4; 42, Section 3]. Writing

$$\bar{P} = \begin{pmatrix} P & u \\ z & \lambda \end{pmatrix} \text{ for some } P \in \mathbb{K}[y]_{\leq 2d}^{(2m) \times (2m)}, \lambda \in \mathbb{K}[y]_{\leq 2d}, u \in \mathbb{K}[y]_{\leq 2d}^{2m}, \text{ and } z \in \mathbb{K}[y]_{\leq 2d}^{1 \times (2m)},$$

the fact that \bar{P} is t -shifted Popov and the choice $t = (0, \dots, 0, 2d)$ ensure that P is (nonshifted) Popov, that $\lambda \neq 0$, and that $\deg(z) + 2d < \deg(P) \leq 2d$. The latter degree bound yields $z = 0$, hence P is the Popov approximant basis of $(S - I_m)$ at order $2d$. The fact that \bar{P} is t -shifted Popov also ensures that the (unique) remainder in the division of u by P is u itself, and that the i th entry of u has degree less than the i th diagonal degree of P .

After Step 5, R is a basis of $\mathcal{M}_{m,m}^{(\gamma, f)} = \mathcal{M}_m^{(\gamma, f)}$. Let R be the $m \times m$ leading principal submatrix of P (and of \bar{P}), and let $v_{\bar{\alpha}} \in \mathbb{K}[y]^m$ be the length- m top subvector of u . Similarly to the above, R is (nonshifted) Popov, and the i th entry of $v_{\bar{\alpha}}$ has degree less than the i th diagonal degree of R ; in particular, $\deg(v_{\bar{\alpha}}) < \deg(R)$. Considering $H(y) = X^T(yI_n - M_\gamma)^{-1}X$, recall from Proposition 5.1

that $\mathcal{D}(H) = \mathcal{M}_{m,m}^{(\gamma,f)}$, and recall that $\mathcal{M}_{m,m}^{(\gamma,f)} \supseteq \mathcal{M}_m^{(\gamma,f)}$ with equality if and only if $v_{m,m}^{(\gamma,f)} = v_m^{(\gamma,f)}$. In particular, bases of $\mathcal{D}(H)$ have determinantal degree $v_{m,m}^{(\gamma,f)} \leq v_m^{(\gamma,f)} \leq n$. Applying Item (i) of Proposition 5.4 to H and the approximant basis P shows that the sum of diagonal degrees of R is $\deg \det(R)$, and is at most $v_{m,m}^{(\gamma,f)}$.

As a result, if Step 5 does not return FAIL, then $n \leq \deg \det(R) \leq v_{m,m}^{(\gamma,f)}$, hence $v_{m,m}^{(\gamma,f)} = v_m^{(\gamma,f)} = n$ and $\mathcal{D}(H) = \mathcal{M}_{m,m}^{(\gamma,f)} = \mathcal{M}_m^{(\gamma,f)}$. Furthermore, Item (ii) of Proposition 5.4 shows that R is a basis of $\mathcal{M}_m^{(\gamma,f)}$.

After Step 6, μ is μ_γ and has degree n . Using $\tilde{O}(m^\omega d)$ operations [53], Step 6 finds the Hermite normal form T of R . Since T is a basis of $\mathcal{M}_m^{(\gamma,f)}$ in upper triangular form, Proposition 4.1 states that its first diagonal entry is the minimal polynomial of γ in $\mathbb{K}[x]/\langle f \rangle$. Hence μ computed at Step 6 is this minimal polynomial. It has degree at most n , and the algorithm returns FAIL at this step if and only if $\deg(\mu) < n$.

After Step 6, $v_{\bar{\alpha}}$ represents $\bar{\alpha}(x, y)$ such that $\bar{\alpha}(x, y) \equiv a \pmod{f}$. Let $\bar{\alpha} \in \mathbb{K}[x, y]_{<(m, \deg(R))}$ be the polynomial whose coefficient vector is $v_{\bar{\alpha}}$, that is, $\bar{\alpha} = \bar{\alpha}_1(y) + x\bar{\alpha}_2(y) + \cdots + x^{m-1}\bar{\alpha}_m(y)$ using notation from Step 7.

The fact that $\mu = \mu_\gamma$ has degree n also ensures that there exists a unique $\alpha \in \mathbb{K}[y]_{<n}$ such that $\alpha(y) \equiv a \pmod{f}$. Then let $v_\alpha = (\alpha \ 0 \ \cdots \ 0)^\top \in \mathbb{K}[y]^m$, and let $v_{\tilde{\alpha}} \in \mathbb{K}[y]^m$ be the unique remainder in the division of v_α by R . The entries of $v_{\tilde{\alpha}}$ have degree strictly less than that of the corresponding row of R : the degree of the i th entry of $v_{\tilde{\alpha}}$ is less than the i th diagonal degree of R . We also define $\tilde{\alpha} \in \mathbb{K}[x, y]_{<(m, \deg(R))}$ as the polynomial whose coefficient vector is $v_{\tilde{\alpha}}$; in particular $\tilde{\alpha}(x, y) = \alpha(y) \equiv a \pmod{f}$.

We now show that $\bar{\alpha} = \tilde{\alpha}$, which yields $\bar{\alpha}(x, y) = a \pmod{f}$. By construction, $\tilde{\alpha}(x, y) - a(x)$ is in $\mathcal{M}_n^{(\gamma,f)}$, and since $yI_n - M_\gamma$ is a basis of $\mathcal{M}_n^{(\gamma,f)}$ (see Proposition 4.1) there is a vector $v \in \mathbb{K}[y]^n$ such that $(yI_n - M_\gamma)v = Xv_{\tilde{\alpha}} - v_a$. Applying the predictable degree property [43, Theorem 6.3-13, p. 387] to the column reduced matrix $yI_n - M_\gamma$, all of whose columns have degree 1, we obtain that $\deg(v) + 1 = \deg(Xv_{\tilde{\alpha}} - v_a) = \deg(v_{\tilde{\alpha}})$. Furthermore, from $(yI_n - M_\gamma)v = Xv_{\tilde{\alpha}} - v_a$ we get

$$X^\top(yI_n - M_\gamma)^{-1}Xv_{\tilde{\alpha}} - X^\top v - X^\top(yI_n - M_\gamma)^{-1}v_a = 0,$$

and considering truncated power series it follows that $Fq = (S \ - I_m \ s)q \equiv 0 \pmod{y^{2d}}$, where

$$q = \begin{pmatrix} \tilde{u} \\ 1 \end{pmatrix} \in \mathbb{K}[y]^{2m+1} \quad \text{and} \quad \tilde{u} = \begin{pmatrix} v_{\tilde{\alpha}} \\ X^\top v \end{pmatrix} \in \mathbb{K}[y]^{2m}.$$

Therefore q is a right multiple of the approximant basis $\bar{P} = \begin{pmatrix} P & u \\ 0 & \lambda \end{pmatrix}$, which shows that λ is an element of \mathbb{K} (we had shown $\lambda \neq 0$ in $\mathbb{K}[y]$ at Step 4), hence $\lambda = 1$ as it is a monic polynomial in the Popov form. It follows that $\tilde{u} - u$ is a right multiple of P , and we check finally that the remainder of \tilde{u} in the division by P —which is u by construction—is \tilde{u} itself. Indeed the degree of the i th entry of $v_{\tilde{\alpha}}$ is less than the i th diagonal degree of R , which is the i th diagonal degree of P ; and as seen above all entries of $X^\top v$ have degree at most $\deg(v_{\tilde{\alpha}}) - 1 < \deg(R) - 1$, with $\deg(R)$ being itself at most the i th diagonal degree of P for $m+1 \leq i \leq 2m$. In particular $v_{\tilde{\alpha}} = v_{\bar{\alpha}}$, hence $\bar{\alpha} = \tilde{\alpha}$ and $\bar{\alpha}(x, y) = a \pmod{f}$.

Step 7 computes $\alpha \in \mathbb{K}[x]_{<n}$ such that $\alpha(y) = a \pmod{f}$. Since $\deg(\mu) = n$, the Hermite normal form of R has the shape $T = \begin{pmatrix} \mu & T_{1,*} \\ 0 & I_{m-1} \end{pmatrix}$, with $T_{1,*} = (T_{1,2} \ \cdots \ T_{1,m})$ and $\deg(T_{1,j}) < \deg(\mu) = n$ for $2 \leq j \leq m$. Then, the polynomial $\alpha = \bar{\alpha}_1 - (T_{1,2}\bar{\alpha}_2 + \cdots + T_{1,m}\bar{\alpha}_m) \pmod{\mu}$ constructed at Step 7 has degree less than n and, by construction as well, the vector $v_\alpha = (\alpha \ 0 \ \cdots \ 0)^\top \in \mathbb{K}[y]^m$ is such that $v_{\bar{\alpha}} - v_\alpha$ is a right multiple of T . (In fact, v_α is the unique remainder in the division of $v_{\bar{\alpha}}$ by T .) In

particular, $v_{\bar{\alpha}} - v_\alpha$ is a right multiple of R , meaning that v_α is equal to $v_{\bar{\alpha}}$ modulo relations of $\mathcal{M}_m^{(y,f)}$, which implies $\alpha(y) = \bar{\alpha}(x, y) \equiv a \pmod{f}$. The computation of α costs $\tilde{O}(nm)$ operations in \mathbb{K} .

This concludes the proof of the properties of (R, μ, α) in the case where the algorithm does not return FAIL. Furthermore, adding the above costs yields the cost bound claimed in the lemma, which, therefore, holds in general since the cost can only be smaller when the algorithm returns FAIL.

Proof of the last claim. The assumption $\gcd(y, f) = 1$ ensures that Step 1 does not return FAIL, in which case we have seen that P is a weak Popov approximant basis of $(S - I_m)$ at order $2d$.

From $\deg(\mu_Y) = n$ and Proposition 4.1 we know that $v_m^{(y,f)} = n$, hence with the assumption $v_{m,m}^{(y,f)} = v_m^{(y,f)}$ we have $v_{m,m}^{(y,f)} = v_m^{(y,f)} = n$. Using $\mathcal{D}(H) = \mathcal{M}_{m,m}^{(y,f)} = \mathcal{M}_m^{(y,f)}$, and Item (iii) of Proposition 5.4 thanks to the assumption on $H(y) = X^T(yI_n - M_Y)^{-1}X$, we deduce that the m rightmost columns of P have degree at least $\deg(R)$ and that R is a basis of $\mathcal{M}_m^{(y,f)}$ with $\deg(R) \leq d$. In particular $\deg \det(R) = n$, and it follows that Step 5 does not return FAIL.

Then, the assumption on the degree of the minimal polynomial also ensures, using Proposition 4.1 as above, that the first diagonal entry of the Hermite normal form T of R is μ_Y , and is the polynomial μ computed at Step 6. Therefore, Step 6 does not return FAIL either: we have proved that, under the assumptions $\gcd(y, f) = 1$, $v_{m,m}^{(y,f)} = v_m^{(y,f)}$, $\deg(\mu_Y) = n$, and $H = X^T(yI_n - M_Y)^{-1}X$ is describable in degree d , then the output is not FAIL and $\deg(R) \leq d$. \square

Notes. Shoup's algorithm for computing α in the case $m = 1$ uses only n terms of the sequence $(\ell(\gamma^k a))_{k \geq 0}$, or more generally d terms, where d is a known bound on $\deg(\mu_a)$. Here as well, if one knows that the sought basis of relations satisfies $\deg(R) \leq d$, for example under the conditions of Proposition 6.1 ensuring success, then the algorithm may be modified so as to require only d terms of the expansion of $-X^T(yI_n - M_Y)^{-1}v_a$ instead of $2d$. The vector $v_{\bar{\alpha}}$ would appear in the approximant basis at order d , and from there one would consider a residual approximant problem focusing on obtaining the missing part of R . This is not detailed here, as this would complicate the presentation without bringing an improvement to the asymptotic complexity.

Modular composition and inverse composition are very similar. They both involve the computation of a matrix of relations and use symmetric steps with similar complexities. Indeed, the division with remainder of Section 4.2.1 is used in both algorithms to change between univariate and bivariate representations efficiently. Also, the application of Algorithm **BIVARIATEMODULARCOMPOSITION** at the last step of composition in Algorithm **BIVARIATEMODULARCOMPOSITIONWITHRELATIONMATRIX** is reflected by **TRUNCATEDPOWERS** as starting step of inverse composition in Algorithm **CHANGEOFBASIS**. Both these steps have cost $\tilde{O}(c(n, m, d))$ from Propositions 3.4 and 3.6, respectively (see also Section 3.4.3).

7 THE BLOCK HANKEL MATRIX $H_{m,d}^{(a,f)}$ AND ITS GENERIC PROPERTIES

Matrices of relations are obtained either by Algorithm **MATRIXOFRRELATIONS** directly, or by Algorithm **CHANGEOFBASIS** after a change of basis. In both cases, for the correctness of the computation to be granted via Propositions 5.8 and 6.1, we need $v_{m,m}^{(a,f)}$ and $v_m^{(a,f)}$ to be equal (and, equivalently, $\mathcal{M}_{m,m}^{(a,f)} = \mathcal{M}_m^{(a,f)}$) and the fraction $H(y) = X^T(yI_n - M_a)^{-1}X$ to be describable in degree d , or the same statement with y in place of a . It is thus important to understand when these properties hold.

Recall from Section 3.4.1 the matrices $K_{m,d}^{(a,f)}$ and $L_{m,d}^{(a,f)}$, that are defined for $m \in \{1, \dots, n\}$ by

$$L_{m,d}^{(a,f)} = \begin{pmatrix} X^T \\ \vdots \\ X^T M_a^{d-1} \end{pmatrix} \in \mathbb{K}^{(md) \times n} \quad \text{and} \quad K_{m,d}^{(a,f)} = (X \quad \dots \quad M_a^{d-1} X) \in \mathbb{K}^{n \times (md)}$$

and that correspond to Algorithms **BIVARIATEMODULARCOMPOSITION** and **TRUNCATEDPOWERS** respectively, and also to the maps $\kappa_{m,d}^{(a,f)}$ and $\lambda_{m,d}^{(a,f)}$. Their product forms the *block Hankel matrix*

$$\text{Hk}_{m,d}^{(a,f)} = L_{m,d}^{(a,f)} K_{m,d}^{(a,f)} = \begin{pmatrix} H_0 & H_1 & \dots & H_{d-1} \\ H_1 & \ddots & \ddots & H_d \\ \vdots & \ddots & \ddots & \vdots \\ H_{d-1} & H_d & \dots & H_{2d-2} \end{pmatrix} \in \mathbb{K}^{(md) \times (md)}, \quad (21)$$

with $H_k = X^T M_a^k X$ for k in \mathbb{N} . This matrix, and in particular its rank, is strongly related to the two properties mentioned above [74; 50, p. 97].

The outcomes of this section are the following. For any positive parameters $m \leq n$ and d , as soon as $\text{rank}(\text{Hk}_{m,d}^{(a,f)}) = v_m^{(a,f)}$, then $v_{m,m}^{(a,f)} = v_m^{(a,f)}$ and H is describable in degree d (Section 7.1). This happens in particular when $f(0) \neq 0$, $d \geq \lceil n/m \rceil$ and either $\deg(a) = m$ (Section 7.2) or for a generic choice of a (Section 7.3). Also, for generic choices of the roots of f and of the values of a at these roots, $\text{rank}(\text{Hk}_{m,d}^{(a,f)}) = v_m^{(a,f)}$ as soon as $d \geq \lceil v_m^{(a,f)}/m \rceil$ (Section 7.4). As in previous sections, notation such as $v_m^{(a,f)}$, $\text{Hk}_{m,d}^{(a,f)}$, $\lambda_{m,d}^{(a,f)}$, and so on. is often shortened into v_m , $\text{Hk}_{m,d}$, $\lambda_{m,d}$, and so on.

These results will be used in Section 8 for the analysis of the randomized composition algorithm when f is separable (Section 8.3), or when f is purely inseparable, which includes the case of power series composition (Sections 8.4 and 8.5).

7.1 Relation Between Block Hankel Matrix Rank and Fraction description degree

The key condition to control the degrees of fraction descriptions of $H(y)$ and obtain matrices of relations is the equality

$$\text{rank}(\text{Hk}_{m,d}) = v_m.$$

The special case when $\text{rank}(\text{Hk}_{m,d}) = n$ is common, and appears naturally later on. The proof of the following result relies in an essential manner on Lemma 7.2, which we give next (the references we cite only give a sketch of proof).

PROPOSITION 7.1. *Given $f \in \mathbb{K}[x]$ of degree n , $a \in \mathbb{K}[x]_{<n}$, and positive integers $m \leq n$ and d , the rank of $\text{Hk}_{m,d}^{(a,f)}$ is at most $v_m^{(a,f)}$. In case of equality, we have $v_{m,m}^{(a,f)} = v_m^{(a,f)}$ and $H(y) = X^T(yI_n - M_a)^{-1}X$ is describable in degree d .*

In particular, if $\text{Hk}_{m,d}^{(a,f)}$ has rank n , then $v_{m,m}^{(a,f)} = v_m^{(a,f)} = n$ and $H(y)$ is describable in degree d .

PROOF. Using Proposition 4.1, the inclusion $\mathcal{M}_m \subseteq \mathcal{M}_{m,m}$ implies $v_{m,m} \leq v_m \leq n$, so that by Lemma 7.2 below, we have $\text{rank}(\text{Hk}_{m,d}) \leq v_{m,m} \leq v_m \leq n$. If $\text{Hk}_{m,d}$ has rank v_m , then $v_m = v_{m,m}$, and the claim on H follows again from Lemma 7.2. The case where the rank is n follows similarly. \square

LEMMA 7.2 ([50, SECTION 2.1] AND [75, LEMMA 2.4]). *For positive integers $m \leq n$ and d , the rank of $\text{Hk}_{m,d}^{(a,f)}$ is at most $v_{m,m}^{(a,f)}$, with equality if and only if $H(y) = X^T(yI_n - M_a)^{-1}X$ is describable in degree d .*

PROOF. We denote by $H_k = X^T M_a^k X \in \mathbb{K}^{m \times m}$ the coefficient in the expansion of H at infinity:

$$H(y) = X^T(yI_n - M_a)^{-1}X = \sum_{k \geq 0} H_k y^{-k-1} = \sum_{k \geq 0} X^T M_a^k X y^{-k-1}.$$

To show that the rank is at most $v_{m,m}$ we first note that $\text{Hk}_{m,d}$ is a submatrix of $\text{Hk}_{m,d+1}$ for $d \geq 0$, the sequence $(\text{rank}(\text{Hk}_{m,d}))_{d \geq 0}$ is thus nondecreasing. Since $\mathcal{M}_{m,m}$ is the module of

vector generators for the sequence $\{H_k\}_{k \geq 0}$ (Section 5.1), the minimal generating polynomial $F \in \mathbb{K}[y]^{m \times m}$ in Popov form for that sequence is a basis of $\mathcal{M}_{m,m}$ ([74, Definition 2.5] and [50, Definition 2.3]). It follows that $\deg \det(F) = v_{m,m}$, and [50, Equation (2.6)] shows that for $d \geq n$, the rank of $\text{Hk}_{m,d}$ is $v_{m,m}$. So the first claim is proved.

From Lemma 5.2, F is also a basis of $\mathcal{D}(H)$; we now study the descriptions of H by exploiting identities that we used to prove this lemma. If the rank of $\text{Hk}_{m,d}$ is equal to $v_{m,m}$, then this rank is also that of the infinite matrix corresponding to the system (see also Equation (17))

$$H_k v_0 + \cdots + H_{k+d} v_d = 0 \quad \text{for } k \geq 0, \quad (22)$$

thus a solution to

$$H_k v_0 + \cdots + H_{k+d} v_d = 0 \quad \text{for } 0 \leq k \leq d-1 \quad (23)$$

is also a solution to Equation (22). Since the rank of $\text{Hk}_{m,d}$ is maximal, we also know that the last block column of $\text{Hk}_{m,d+1}$ is a linear combination of the previous ones. This provides with m linearly independent $R_1, \dots, R_m \in \mathbb{K}[y]^m$, of degree d , whose coefficient vectors in y are solutions to Equation (23), hence to Equation (22). Let R be the matrix in $\mathbb{K}[y]^{m \times m}$ whose j th column is R_j . Using Equation (22), we deduce that $HR = Q$ with $Q \in \mathbb{K}[y]^{m \times m}$ (see also Equation (18)). This gives a right fraction description $H = QR^{-1}$ (which may not be irreducible) with denominator of degree d . The same reasoning on the left side gives a left matrix description of degree d , hence H is describable in degree d .

Conversely, a right matrix description $H = QR^{-1}$ with R of degree at most d gives R_j 's whose coefficient vectors are solutions to Equation (22). Since F is a basis of the module of vector generators for $\{H_k\}_{k \geq 0}$, R must be a multiple of F . By minimality F has degree at most d [43, Theorem 6.5-10, p. 458], and using [50, Equation (2.6)] the rank of the infinite block Hankel matrix restricted to its first d block columns is maximal. Starting from a left description, in an analogous way we obtain that the rank restricted to the first d block rows is maximal, which yields that $\text{Hk}_{m,d}$ has rank $\deg \det(F) = v_{m,m}$. \square

7.2 Families with $\text{Hk}_{m,d}^{(a,f)}$ of rank n

A simple condition implies the equality $\text{rank}(\text{Hk}_{m,d}) = n$ of Proposition 7.1.

PROPOSITION 7.3. *Let $f \in \mathbb{K}[x]$ have degree n , let $a \in \mathbb{K}[x]_{<n}$, and let m be a positive integer. If $f(0) \neq 0$ and $\deg(a) = m$ (hence $1 \leq m < n$), then the block Hankel matrix $\text{Hk}_{m,d}^{(a,f)} \in \mathbb{K}^{(md) \times (md)}$ has rank n for all $d \geq \lceil n/m \rceil$.*

The rest of this subsection is devoted to the proof of this result. It is a basis for the genericity result in the next subsection.

PROOF. For a given c in \mathbb{K} , by construction of this block-Hankel matrix, one has

$$\text{Hk}_{m,d}^{(ca,f)} = C \text{Hk}_{m,d}^{(a,f)} C, \text{ where } C = \text{diag}(\underbrace{1, \dots, 1}_{m \text{ times}}, \underbrace{c, \dots, c}_{m \text{ times}}, \underbrace{c^{d-1}, \dots, c^{d-1}}_{m \text{ times}}).$$

It follows that $\text{rank}(\text{Hk}_{m,d}^{(ca,f)}) = \text{rank}(\text{Hk}_{m,d}^{(a,f)})$ for any $c \neq 0$, and therefore in the rest of the proof we can assume that a is *monic* of degree m .

By Equation (21), it is sufficient to show that the mappings $\kappa_{m,d}$ and $\lambda_{m,d}$ associated to $K_{m,d}$ and $L_{m,d}$ are surjective and injective, respectively.

The mapping $\kappa_{m,d}$ is surjective. By assumption, $n \leq md$ so that surjectivity of $\kappa_{m,d}$ is equivalent to the matrix $K_{m,d} \in \mathbb{K}^{n \times (md)}$ from Equation (12) having full row rank n . Indeed, the first n columns of $K_{m,d}$ are the coefficients of the family of polynomials $x^i a^j \bmod f$, for $0 \leq i < m$ and $0 \leq j < d$, with $0 \leq i + jm < n$. Since $\deg(a) = m$, these columns form an upper triangular matrix, with 1's on the diagonal; this proves the claim.

The mapping $\lambda_{m,d}$ is injective. Equivalently, we have to show that $L_{m,d}$ has full column rank n . This follows from the structure of this matrix, seen at the level of polynomials.

LEMMA 7.4. *With the notation and hypotheses of Proposition 7.3, let*

$$p_i = [ax^{n-m+i} \bmod f]_0^{m-1}, \quad i = 0, \dots, m-1.$$

Then,

- (i) if $m \leq n/2$, the m polynomials p_0, \dots, p_{m-1} are linearly independent;
- (ii) if $n/2 < m$, the $n-m$ polynomials p_{2m-n}, \dots, p_{m-1} are linearly independent.

PROOF. The two cases require different proofs, sharing common ingredients. For $i \geq 0$, let $r_i = x^{n+i} \bmod f$. For b in $\mathbb{K}[x]_{<n}$, we then have

$$x^i b \bmod f = [x^i b]_0^{n-1} + \delta_{b,i}, \quad (24)$$

for some $\delta_{b,i}$ in $\text{Span}(r_0, \dots, r_{i-1})$, in particular $\delta_{b,0} = 0$. Applying this to $b = r_0 = x^n \bmod f$ yields $r_i = [x^i r_0]_0^{n-1} + \delta_{r_0,i}$. Taking this relation modulo x^m gives $[r_i]_0^{m-1} = [x^i r_0]_0^{m-1} + \mu_i$, with μ_i in $\text{Span}([r_0]_0^{m-1}, \dots, [r_{i-1}]_0^{m-1})$ for $i > 0$ and $\mu_0 = 0$. By induction on $i \geq 0$, one deduces that

$$\text{Span}([r_0]_0^{m-1}, \dots, [r_i]_0^{m-1}) = \text{Span}([r_0]_0^{m-1}, \dots, [x^i r_0]_0^{m-1}).$$

Writing $f = f_0 + \dots + f_{n-1}x^{n-1} + x^n$, we get $r_0 = -f_0 - f_1 x - \dots - f_{n-1}x^{n-1}$. Since $f_0 \neq 0$ by assumption, $[x^i r_0]_0^{m-1}$ has valuation i for $0 \leq i < m$; this implies that $\text{Span}([r_0]_0^{m-1}, \dots, [r_i]_0^{m-1})$ has dimension $i+1$ for $0 \leq i < m$.

Proof of Item (i). Let $b = ax^{n-m} \bmod f$ in Equation (24). Upon reduction modulo x^m , for $0 \leq i < m$, we obtain the relation $p_i = [x^i b]_0^{m-1} + \mu'_i$, with μ'_i in $\text{Span}([r_0]_0^{m-1}, \dots, [r_{i-1}]_0^{m-1})$.

Since ax^{n-m} is monic of degree n (a has degree m), with valuation at least $n-m \geq m$ (here, $m \leq n/2$), we get $[b]_0^{m-1} = [r_0]_0^{m-1}$, and thus $[x^i b]_0^{m-1} = [x^i r_0]_0^{m-1}$ for $0 \leq i < m$. This gives $p_i = [r_i]_0^{m-1} + \mu'_i - \mu_i$, with $\mu'_i - \mu_i$ in $\text{Span}([r_0]_0^{m-1}, \dots, [r_{i-1}]_0^{m-1})$. In particular, taking all i up to $m-1$, we get the equality $\text{Span}(p_0, \dots, p_{m-1}) = \text{Span}([r_0]_0^{m-1}, \dots, [r_{m-1}]_0^{m-1})$, and we saw that the latter has dimension m . Item (i) is proved.

Proof of Item (ii). Assume that $q = c_m x^m + \dots + c_{n-1} x^{n-1}$ is such that $[aq \bmod f]_0^{m-1} = 0$. We prove that all c_i 's vanish.

We can rewrite $aq \bmod f$ as $x^m b \bmod f$, with $b = a(q/x^m)$; since a has degree m , b is in $\mathbb{K}[x]_{<n}$. Applying Equation (24) to b and $i = m$, our assumption that $[x^m b \bmod f]_0^{m-1} = 0$ implies $[\delta_{b,m}]_0^{m-1} = 0$. Writing $\delta_{b,m} = \sum_{j=0}^{m-1} \bar{\delta}_j r_j$ for some $\bar{\delta}_0, \dots, \bar{\delta}_{m-1} \in \mathbb{K}$, we get $\sum_{j=0}^{m-1} \bar{\delta}_j [r_j]_0^{m-1} = [\delta_{b,m}]_0^{m-1} = 0$. The linear independence of $[r_0]_0^{m-1}, \dots, [r_{m-1}]_0^{m-1}$ ensures $\bar{\delta}_j = 0$ for all j , showing that $\delta_{b,m}$ itself is zero. Hence $x^m b \bmod f = [x^m b]_0^{m-1}$, from which we deduce $x^n [x^m b]_0^{m-1} \bmod f = 0$ using

$$x^m b \bmod f = ([x^m b]_0^{m-1} + x^n [x^m b]_n^{m-1}) \bmod f = [x^m b]_0^{m-1} + (x^n [x^m b]_n^{m-1} \bmod f).$$

Since $x^n [x^m b]_n^{m-1} = x^n [b]_{n-m}^{m-1}$, and since $f_0 \neq 0$ ensures that x is invertible modulo f , it follows that $[b]_{n-m}^{m-1}$ vanishes modulo f , or equivalently that $[b]_{n-m}^{m-1} = 0$. Since a is monic of degree m , and since $n-m < m$, the definition of b then implies that all coefficients c_i 's vanish. Hence, Item (ii) is proved. \square

Let now $v \in \mathbb{K}[x]_{<n}$ be such that

$$[v]_0^{m-1} = [av \text{ rem } f]_0^{m-1} = \cdots = [a^{d-1}v \text{ rem } f]_0^{m-1} = 0.$$

We prove that $\deg(v) < n - mi$ for $i = 0, \dots, d - 1$, by induction. For $d = \lceil n/m \rceil$, this gives $\deg(v) < m$; together with the assumption $[v]_0^{m-1} = 0$, this proves that $v = 0$.

The base case of the induction is for $i = 0$, and $\deg(v) < n$ holds by assumption. If the claim holds for some index $i < d - 1$, since a has degree m , for any w in $\mathbb{K}[x]_{<n}$, the polynomial $[aw \text{ rem } f]_0^{m-1}$ splits into two parts:

$$[aw \text{ rem } f]_0^{m-1} = [a[w]_0^{n-m-1}]_0^{m-1} + [ax^{n-m}[w]_{n-m}^{m-1} \text{ rem } f]_0^{m-1}.$$

Apply this identity with $w = a^i v \text{ rem } f$. Then, both the left-hand side and the first summand vanish: the former because $[a^{i+1}v \text{ rem } f]_0^{m-1} = 0$, the latter because $[a^i v \text{ rem } f]_0^{m-1} = 0$, i.e., $w = a^i v \text{ rem } f$ has valuation at least m . We deduce that $[ax^{n-m}[w]_{n-m}^{m-1} \text{ rem } f]_0^{m-1} = 0$, with $w = a^i v \text{ rem } f$.

- If $m \leq n/2$, the linear independence of the polynomials $p_j = [ax^{n-m+j} \text{ rem } f]_0^{m-1}$, for $j = 0, \dots, m - 1$, then shows that $[w]_{n-m}^{m-1} = [a^i v \text{ rem } f]_{n-m}^{m-1}$ vanishes.
- If $m > n/2$, then the assumption that w has valuation at least m , with thus $m > n - m$, shows that $[w]_{n-m}^{m-1} = x^{2m-n}[w]_m^{n-m-1}$. In this case, the linear independence of the polynomials p_j for $j = 2m - n, \dots, m - 1$ shows that $[w]_{n-m}^{m-1} = 0$.

In other words, in both cases, we have proved that $w = a^i v \text{ rem } f$ has degree less than $n - m$.

On the other hand, the induction assumption that $\deg(v) < n - mi$ implies that $a^i v \text{ rem } f = a^i v$, so the latter has degree less than $n - m$. Since a^i has degree mi , this shows that $\deg(v) < n - m(i + 1)$, as claimed. \square

7.3 Generic Regularity in a and f

In all this document, genericity is understood in the Zariski sense:

Definition 7.5. A property \mathcal{P} of certain parameters (u_1, \dots, u_s) holds for a *generic* choice of (u_1, \dots, u_s) in \mathbb{K}^s if there exists a nonzero polynomial Δ in $\mathbb{K}[\bar{u}_1, \dots, \bar{u}_s]$ (where the \bar{u}_i 's are new indeterminates) such that $\Delta(u_1, \dots, u_s) \neq 0$ implies that $\mathcal{P}(u_1, \dots, u_s)$ holds.

Note that if \mathbb{K} is finite, there may be no choice of the u_i 's in \mathbb{K} for which Δ does not vanish, but such points exist in a finite extension of \mathbb{K} of sufficiently large degree (such as $O(\log(n))$ when the degree of Δ is polynomial in n , as is the case below).

PROPOSITION 7.6. Let f in $\mathbb{K}[x]$ be of degree n and such that $f(0) \neq 0$. For any $m \in \{1, \dots, n\}$ there exists a nonzero polynomial $\Delta_{f,m}$ in $\mathbb{K}[\bar{a}_0, \dots, \bar{a}_{n-1}]$ of degree at most $2n^2/m$ such that for $a = a_0 + \dots + a_{n-1}x^{n-1}$ in $\mathbb{K}[x]_{<n}$, if $\Delta_{f,m}(a_0, \dots, a_{n-1}) \neq 0$ then $\text{Hk}_{m,d}^{(a,f)} \in \mathbb{K}^{(md) \times (md)}$ has rank n for any $d \geq \lceil n/m \rceil$.

7.3.1 Proof of Proposition 7.6.

LEMMA 7.7. Let m, n be positive integers, with $m \in \{1, \dots, n\}$, and let $\bar{f} = \bar{f}_0 + \dots + \bar{f}_{n-1}x^{n-1} + x^n$ and $\bar{a} = \bar{a}_0 + \dots + \bar{a}_{n-1}x^{n-1}$ be polynomials in $\mathbb{Z}[\bar{a}_0, \dots, \bar{a}_{n-1}, \bar{f}_0, \dots, \bar{f}_{n-1}][x]$. Then any n -minor of $\text{Hk}_{m, \lceil n/m \rceil}^{(\bar{a}, \bar{f})}$ has degree at most $2n^2/m$ in $\bar{a}_0, \dots, \bar{a}_{n-1}$ and $2n^2(n-1)/m$ in $\bar{f}_0, \dots, \bar{f}_{n-1}$.

PROOF. The multiplication matrix $M_{\bar{a}}$ can be written as $M_{\bar{a}} = \sum_{k=0}^{n-1} \bar{a}_k M_x^k$, where M_x is the companion matrix of \bar{f} . The entries of $M_{\bar{a}}$, which are the coefficients of $x^k \bar{a} \text{ rem } \bar{f}$ for $k = 0, \dots, n-1$, are therefore polynomials of degree 1 in the coefficients $\bar{a}_0, \dots, \bar{a}_{n-1}$ and at most $n-1$ in the coefficients $\bar{f}_0, \dots, \bar{f}_{n-1}$. In turn, the coefficients of $M_{\bar{a}}^j$ have degree at most j in $\bar{a}_0, \dots, \bar{a}_{n-1}$ and

$j(n-1)$ in $\bar{f}_0, \dots, \bar{f}_{n-1}$. For $0 \leq i, j < \lceil n/m \rceil$, the $m \times m$ block of coordinates (i, j) in $\text{Hk}_{m, \lceil n/m \rceil}^{(\bar{a}, \bar{f})}$ is a submatrix of $M_{\bar{a}}^{i+j}$; it has degree at most $i+j$ in $\bar{a}_0, \dots, \bar{a}_{n-1}$ and $(i+j)(n-1)$ in $\bar{f}_0, \dots, \bar{f}_{n-1}$. As a result, any n -minor of this matrix has degree at most $m\lceil n/m \rceil(\lceil n/m \rceil - 1) \leq 2n^2/m$ in $\bar{a}_0, \dots, \bar{a}_{n-1}$ and $m\lceil n/m \rceil(\lceil n/m \rceil - 1)(n-1) \leq 2n^2(n-1)/m$ in $\bar{f}_0, \dots, \bar{f}_{n-1}$. \square

Take f of degree n with $f(0) \neq 0$. Proposition 7.3 with $a = x^m$ shows that at least one n -minor of $\text{Hk}_{m, \lceil n/m \rceil}^{(x^m, f)}$ is nonzero, so the corresponding n -minor of $\text{Hk}_{m, \lceil n/m \rceil}^{(\bar{a}, f)}$ is not identically zero. We take this minor for $\Delta_{f, m}$, and its degree is then bounded by Lemma 7.7.

7.3.2 Note: Basis of Relations for a Generic a . For any f in $\mathbb{K}[x]$ with $f(0) \neq 0$, and for a generic a in $\mathbb{K}[x]_{<n}$, Proposition 7.6 shows that the rank of $\text{Hk}_{m, d}^{(a, f)}$ is n , with $d = \lceil n/m \rceil$. From Proposition 7.1 we then obtain $v_m^{(a, f)} = v_{m, m}^{(a, f)} = n$ and the describability of H in degree d . Therefore, by Proposition 5.6, Algorithm **CANDIDATEBASIS** returns a basis of $\mathcal{M}_m^{(a, f)}$ and the flag **CERT**.

7.4 Generic Rank for a Separable f

We now study the rank of $\text{Hk}_{m, d}$, for a generic choice of the roots of f , and for a generic choice of the values of a at these roots, subject to certain combinatorial conditions.

7.4.1 Definitions. Consider pairwise distinct ξ_1, \dots, ξ_n in an algebraic closure $\bar{\mathbb{K}}$ of \mathbb{K} . To such points, we associate the polynomial $f = (x - \xi_1) \cdots (x - \xi_n)$. We also consider $a \in \mathbb{K}[x]_{<n}$, and we say that a takes values $\lambda_1, \dots, \lambda_r$ at ξ_1, \dots, ξ_n with multiplicities ℓ_1, \dots, ℓ_r if the following holds:

- $\lambda_1, \dots, \lambda_r$ are pairwise distinct elements in $\bar{\mathbb{K}}$;
- $\ell_1 + \cdots + \ell_r = n$, with all ℓ_i positive integers;
- for $i = 1, \dots, r$, $a(\xi_{\sigma_i+1}) = \cdots = a(\xi_{\sigma_i+\ell_i}) = \lambda_i$, where we write $\sigma_i = \ell_1 + \cdots + \ell_{i-1}$ (the empty sum for $i = 1$ is zero).

In view of our application, we also assume that the ξ_i 's are such that f is in $\mathbb{K}[x]$.

7.4.2 Generic Rank.

PROPOSITION 7.8. Fix positive integers $m \in \{1, \dots, n\}$ and $\ell = (\ell_1, \dots, \ell_r)$ such that $\ell_1 + \cdots + \ell_r = n$. Then, there exists a nonzero polynomial $\Gamma_{\ell, m} \in \mathbb{Z}[\xi_1, \dots, \xi_n, \bar{\lambda}_1, \dots, \bar{\lambda}_r]$ such that the following holds. For pairwise distinct nonzero ξ_1, \dots, ξ_n in $\bar{\mathbb{K}}$ such that $f = c(x - \xi_1) \cdots (x - \xi_n)$ with $c \in \mathbb{K} \setminus \{0\}$ is in $\mathbb{K}[x]$ and for $a \in \mathbb{K}[x]$ that takes values $\lambda_1, \dots, \lambda_r$ at ξ_1, \dots, ξ_n with multiplicities ℓ_1, \dots, ℓ_r , if $\Gamma_{\ell, m}(\xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r)$ is nonzero, then

$$\text{rank}(\text{Hk}_{m, d}^{(a, f)}) = v_m^{(a, f)} \quad \text{for any } d \geq \lceil v_m^{(a, f)} / m \rceil,$$

with in addition the equality

$$v_m^{(a, f)} = \sum_{i=1}^r \min(\ell_i, m).$$

Finally, for any pairwise distinct $\lambda_1, \dots, \lambda_r$, the polynomial $\Gamma_{\ell, m}(\bar{\xi}_1, \dots, \bar{\xi}_n, \lambda_1, \dots, \lambda_r)$ is nonzero and has degree at most $2n^2$.

7.4.3 Proof of Proposition 7.8. The rather long proof is decomposed as follows. First, the expression for the determinantal degree v_m is established. For the proof of the rest of the proposition we exploit the factorization $\text{Hk}_{m, d} = L_{m, d} K_{m, d}$, that is analyzed through a series of lemmas. All along, we use classical linear algebra notions concerning invariant factors and Smith normal forms, and their relation to eigenvalues in the case of diagonalizable matrices; see e.g. [62, Ch. II and III] for more background on these aspects.

The ranks of the matrices $K_{m,d}$ and $L_{m,d}$ are related to that of a simple matrix $\mathcal{P}_{\ell,m,d}$ (see Equation (31)). This leads to the proof that for $d = \lceil v_m/m \rceil$, the rank of $K_{m,d}$ and $L_{m,d}$ is v_m generically. Then we prove that generically, taking any $d_0 \geq \lceil v_m/m \rceil$ is sufficient for studying the rank of $\text{Hk}_{m,d}$. The proof is concluded by establishing that the rank is v_m when d_0 is r , the number of distinct values $a(\xi_k)$'s: for this value of d_0 , we establish that the intersection of the image of $K_{m,d}$ with the kernel of $L_{m,d}$ is reduced to 0. The polynomial $\Gamma_{\ell,m}$ and the degree bounds are derived from the proof.

Determinantal degree $v_m^{(a,f)}$. As in the proposition, let ξ_1, \dots, ξ_n be pairwise distinct in $\overline{\mathbb{K}}$ and let $f = c(x - \xi_1) \cdots (x - \xi_n)$. The Lagrange interpolation polynomials

$$\mathcal{L}_k(x) = \frac{1}{f'(\xi_k)} \frac{f(x)}{x - \xi_k} = \prod_{\ell \neq k} \frac{x - \xi_\ell}{\xi_k - \xi_\ell}, \quad k = 1, \dots, n. \quad (25)$$

form a basis of $\overline{\mathbb{A}} := \overline{\mathbb{K}}[x]/\langle f \rangle$. For any $a \in \overline{\mathbb{A}}$, the matrix of multiplication by a is diagonalizable, its eigenvalues are the values of a at the ξ_j 's, and the Lagrange polynomials are eigenvectors. The characteristic polynomial χ_a of a modulo f is therefore given by

$$\chi_a = \prod_{k=1}^n (y - a(\xi_k)) \in \overline{\mathbb{K}}[y].$$

For $1 \leq i \leq r$, we define $S_i = \{k \in \{1, \dots, n\} \mid a(\xi_k) = \lambda_i\}$ and use that

$$S_i = \{\sigma_i + 1, \dots, \sigma_i + \ell_i\}. \quad (26)$$

With these conventions we have the factorization

$$\chi_a = \prod_{i=1}^r (y - \lambda_i)^{\ell_i},$$

where the factors $(y - \lambda_i)$ are pairwise coprime. The Smith normal form of $yI_n - M_a$ is then known and an explicit expression for the determinantal degree v_m can be given: $yI_n - M_a$ has $\max(\ell_i)$ nontrivial invariant factors; for $1 \leq k \leq \max(\ell_i)$, the k th one is $\prod_{1 \leq i \leq r} (y - \lambda_i)^{\varepsilon_{i,k}}$, where $\varepsilon_{i,k} = 1$ if $k \leq \ell_i$ and 0 otherwise. From there, recalling from Equation (8) that for m in $\{1, \dots, n\}$, v_m is the sum of the degrees of the first m such invariant factors, we have:

$$v_m = \sum_{k=1}^{\min(m, \max(\ell_i))} \text{card}(\{i \mid \ell_i \leq k\}) = \sum_{i=1}^r \min(\ell_i, m). \quad (27)$$

This proves the claim regarding v_m in the proposition (this claim thus holds without further assumption on the ξ_i 's and λ_i 's).

Maximal rank of $\text{Hk}_{m,d}^{(a,f)}$.

LEMMA 7.9. *Let $A \in \mathbb{K}^{n \times n}$ and $m \in \mathbb{N}_{>0}$, and let v be the sum of the degrees of the $\min(m, n)$ highest degree invariant factors of $yI_n - A$. Then for any collection of m vectors $v_1, \dots, v_m \in \mathbb{K}^n$, one has $\dim(\text{Span}(A^i v_j, 0 \leq i, 1 \leq j \leq m)) \leq v$.*

PROOF. We let $\tilde{v} = \dim(\text{Span}(A^i v_j, 0 \leq i, 1 \leq j \leq m))$. For $1 \leq j \leq m$, let $d_j \geq 0$ be the first index such that $A^{d_j} v_j \in \text{Span}(v_j, Av_j, \dots, A^{d_j-1} v_j, \{A^i v_k \mid 0 \leq i, 0 \leq k < j\})$; if $l \geq d_j$ then $A^l v_j$ also belongs to the latter subspace of \mathbb{K}^n , which is therefore stable under left multiplication by A . This holds for any $1 \leq j \leq m$, hence $d_1 + \dots + d_m = \tilde{v}$, and the matrix

$$P_1 = \begin{pmatrix} v_1 & Av_1 & \cdots & A^{d_1-1} v_1 & \cdots & v_m & Av_m & \cdots & A^{d_m-1} v_m \end{pmatrix} \in \mathbb{K}^{n \times \tilde{v}}$$

has rank \tilde{v} and can be completed into a nonsingular matrix $P = (P_1 \ P_2) \in \mathbb{K}^{n \times n}$. By applying the change of basis $P^{-1}AP$ we obtain

$$P^{-1}(yI_n - A)P = \begin{pmatrix} yI_{\tilde{v}} - C & B_1 \\ 0 & yI_{n-\tilde{v}} - B_2 \end{pmatrix} \in \mathbb{K}[y]^{n \times n}, \quad (28)$$

where $C \in \mathbb{K}^{\tilde{v} \times \tilde{v}}$, $B_1 \in \mathbb{K}^{\tilde{v} \times (n-\tilde{v})}$, $B_2 \in \mathbb{K}^{(n-\tilde{v}) \times (n-\tilde{v})}$. Thanks to the form of P_1 , the matrix $C \in \mathbb{K}^{\tilde{v} \times \tilde{v}}$ is block upper triangular with at most m companion blocks C_j of dimensions d_j on the diagonal (there is no block for $d_j = 0$, and at most n of the d_j 's are nonzero). By a unimodular row transformation $U_j \in \mathbb{K}[y]^{d_j \times d_j}$, a matrix $yI_{d_j} - C_j$ can be brought into an upper triangular form $T_j(y) = U_j(y)(yI_{d_j} - C_j)$, which has diagonal entries 1 except for the last entry which is the characteristic polynomial $\chi^{(j)} = y^{d_j} - \chi_{d_j-1}^{(j)}y^{d_j-1} - \dots - \chi_0^{(j)}$ of C_j :

$$\begin{pmatrix} -1 & & & \\ & \ddots & & \\ & & -1 & \\ 1 & y & \dots & y^{d_j-1} \end{pmatrix} \begin{pmatrix} yI_{d_j} - \begin{pmatrix} 1 & & & \chi_0^{(j)} \\ & \ddots & & \chi_1^{(j)} \\ & & \ddots & \vdots \\ & & & 1 & \chi_{d_j-1}^{(j)} \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \chi^{(j)}(y) \end{pmatrix} \in \mathbb{K}[y]^{d_j \times d_j},$$

Therefore Equation (28) can be rewritten as

$$U(y)P^{-1}(yI_n - A)P = \begin{pmatrix} T(y) & \bar{B}_1(y) \\ 0 & yI_{n-\tilde{v}} - B_2 \end{pmatrix} = \begin{pmatrix} I_{\tilde{v}} & \bar{B}_1(y) \\ 0 & yI_{n-\tilde{v}} - B_2 \end{pmatrix} \begin{pmatrix} T(y) & 0 \\ 0 & I_{n-\tilde{v}} \end{pmatrix}, \quad (29)$$

where $U = \text{diag}(U_1, \dots, U_m, I_{n-\tilde{v}})$ is unimodular (with no U_j if $d_j = 0$), and $T \in \mathbb{K}[y]^{\tilde{v} \times \tilde{v}}$ is block upper triangular with diagonal blocks the T_j 's. The matrix T is triangular with 1's on the diagonal except for at most m entries. We deduce that the gcd of the minors of dimension k of T is a unit for $1 \leq k \leq \tilde{v} - m$, and that T has at most m nontrivial invariant factors [62, Ch. II, Equation (13)]. The product of these invariant factors is $\det(T) = \prod_j \chi^{(j)}$, whose degree is $d_1 + \dots + d_m = \tilde{v}$. From the matrix product on the right-hand side of Equation (29), these latter invariant factors divide the m highest degree invariant factors of $yI_n - A$ [62, Theorem II.14]. From the definition of v we obtain $\tilde{v} \leq v$. \square

With $A = M_a$ or M_a^T , and $K_{m,d}, L_{m,d}$ from Equation (12), for any positive integer d , Lemma 7.9 gives

$$\text{rank}(K_{m,d}) \leq v_m, \text{ rank}(L_{m,d}) \leq v_m \text{ and } \text{rank}(\text{Hk}_{m,d}) \leq v_m. \quad (30)$$

Next, we show that the ranks of both $K_{m, \lceil v_m/m \rceil}$ and $L_{m, \lceil v_m/m \rceil}$ are v_m generically.

The relation of $K_{m,d}^{(a,f)}$ and $L_{m,d}^{(a,f)}$ to the matrix $\mathcal{P}_{\ell,m,d}$. For $\ell = (\ell_1, \dots, \ell_r)$, m in $\{1, \dots, n\}$, and a positive integer d , we define the matrix

$$\mathcal{P}_{\ell,m,d} = \begin{pmatrix} V_{\bar{\xi}} & D_{\bar{\lambda}} V_{\bar{\xi}} & \dots & D_{\bar{\lambda}}^{d-1} V_{\bar{\xi}} \end{pmatrix} \in \mathbb{Z}[\bar{\xi}_1, \dots, \bar{\xi}_n, \bar{\lambda}_1, \dots, \bar{\lambda}_r]^{n \times md}, \quad (31)$$

where

$$V_{\bar{\xi}} = \begin{pmatrix} 1 & \bar{\xi}_1 & \dots & \bar{\xi}_1^{m-1} \\ \vdots & \vdots & & \vdots \\ 1 & \bar{\xi}_n & \dots & \bar{\xi}_n^{m-1} \end{pmatrix} \quad \text{and} \quad D_{\bar{\lambda}} = \text{diag}(\underbrace{\bar{\lambda}_1, \dots, \bar{\lambda}_1}_{\ell_1 \text{ times}}, \dots, \underbrace{\bar{\lambda}_r, \dots, \bar{\lambda}_r}_{\ell_r \text{ times}}).$$

The following lemma summarizes the key properties of this matrix in relation with the rank of $K_{m,d}$ and $L_{m,d}$.

LEMMA 7.10. *Let $\ell, \bar{\xi}_1, \dots, \bar{\xi}_n, \bar{\lambda}_1, \dots, \bar{\lambda}_r, f, a$ and m be as in Proposition 7.8, and let d be a positive integer. The following holds:*

- the rank of $K_{m,d}$ is equal to the rank of $\mathcal{P}_{\ell,m,d}(\xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r)$;
- if all ξ_i 's are nonzero, the rank of $L_{m,d}$ is equal to the rank of $\mathcal{P}_{\ell,m,d}(1/\xi_1, \dots, 1/\xi_n, \lambda_1, \dots, \lambda_r)$.

PROOF. We use the same notation

$$\kappa_{m,d} : \overline{\mathbb{K}}[x, y]_{<(m,d)} \rightarrow \overline{\mathbb{K}}[x]/\langle f \rangle \quad \text{and} \quad \lambda_{m,d} : \overline{\mathbb{K}}[x]_{<n} \rightarrow \overline{\mathbb{K}}[x]_{<m}^d$$

for the mappings induced by scalar extension from $\kappa_{m,d}$ and $\lambda_{m,d}$ from Section 3.4.1.

Taking $(x^i y^j)_{0 \leq i < m, 0 \leq j < d}$ for basis of $\overline{\mathbb{K}}[x, y]_{<(m,d)}$ and the Lagrange basis $\mathcal{L}_1, \dots, \mathcal{L}_n$ for $\overline{\mathbb{K}}[x]_{<n}$, the matrix of $\kappa_{m,d}$ is $\mathcal{P}_{\ell,m,d}(\xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r)$. This proves the first point.

To prove the second point, take k in $\{1, \dots, n\}$, and let i in $\{1, \dots, r\}$ be such that $a(\xi_k) = \lambda_i$. The image of the Lagrange polynomial \mathcal{L}_k by $\lambda_{m,d}$ is the polynomial vector

$$\lambda_{m,d}(\mathcal{L}_k) = \left([\mathcal{L}_k]_0^{m-1}, [a \mathcal{L}_k \text{ rem } f]_0^{m-1}, \dots, [a^{d-1} \mathcal{L}_k \text{ rem } f]_0^{m-1} \right) \in \overline{\mathbb{K}}[x]_{<m}^d,$$

and since the Lagrange polynomials are eigenvectors of multiplication by a , we get

$$\begin{aligned} \lambda_{m,d}(\mathcal{L}_k) &= \left([\mathcal{L}_k]_0^{m-1}, [\lambda_i \mathcal{L}_k]_0^{m-1}, \dots, [\lambda_i^{d-1} \mathcal{L}_k]_0^{m-1} \right) \\ &= \left([\mathcal{L}_k]_0^{m-1}, \lambda_i [\mathcal{L}_k]_0^{m-1}, \dots, \lambda_i^{d-1} [\mathcal{L}_k]_0^{m-1} \right). \end{aligned}$$

Let $L' \in \overline{\mathbb{K}}^{(md) \times n}$ be the matrix whose k th column (for $k = 1, \dots, n$) contains the md coefficients of the entries of $\lambda_{m,d}(\mathcal{L}_k)$. This is the matrix of $\lambda_{m,d}$, if we take the Lagrange basis for the domain $\overline{\mathbb{K}}[x]_{<n}$.

Since all ξ_i 's are nonzero, we get $f(0) \neq 0$, so that f is invertible as a power series. Because the \mathbb{K} -linear transformation $b \in \overline{\mathbb{K}}[x]_{<m} \mapsto [b/f]_0^{m-1}$ is invertible, L' has the same rank as the matrix whose columns are the coefficients of the vectors

$$\left([[\mathcal{L}_k]_0^{m-1}/f]_0^{m-1}, \lambda_i [[\mathcal{L}_k]_0^{m-1}/f]_0^{m-1}, \dots, \lambda_i^{d-1} [[\mathcal{L}_k]_0^{m-1}/f]_0^{m-1} \right),$$

for i and k as above. On the other hand, we have $[[\mathcal{L}_k]_0^{m-1}/f]_0^{m-1} = [\mathcal{L}_k/f]_0^{m-1}$ and

$$\frac{\mathcal{L}_k}{f} = \frac{1}{f'(\xi_k)} \frac{1}{x - \xi_k}.$$

This shows that to determine the rank of L' , we may as well consider the vectors

$$\left(\left[\frac{1}{x - \xi_k} \right]_0^{m-1}, \lambda_i \left[\frac{1}{x - \xi_k} \right]_0^{m-1}, \dots, \lambda_i^{d-1} \left[\frac{1}{x - \xi_k} \right]_0^{m-1} \right).$$

Now, note that

$$\left[\frac{1}{x - \xi_k} \right]_0^{m-1} = -\xi_k \left(1 + \frac{1}{\xi_k} x + \dots + \frac{1}{\xi_k^{m-1}} x^{m-1} \right).$$

Thus, up to the factors $-\xi_k$, taking the md coefficients of these vectors and putting them in columns gives us the transpose of $\mathcal{P}_{\ell,m,d}(1/\xi_1, \dots, 1/\xi_n, \lambda_1, \dots, \lambda_r)$. This proves the rank equality claimed in the second item. \square

The rank of $K_{m,d}^{(a,f)}$ and $L_{m,d}^{(a,f)}$ for $d = \lceil v_m^{(a,f)}/m \rceil$. Together with Lemma 7.10, the next lemma establishes that the generic rank of $K_{m,\lceil v_m/m \rceil}$ and $L_{m,\lceil v_m/m \rceil}$ is v_m . Let $\mathcal{R}_{\ell,m}$ be the $v_m \times v_m$ submatrix of $\mathcal{P}_{\ell,m,\lceil v_m/m \rceil}$ obtained by extracting the first $\min(\ell_i, m)$ rows containing $\bar{\lambda}_i$, for $i = 1, \dots, r$ (see Equation (27)), and the first v_m columns (note that $\mathcal{P}_{\ell,m,\lceil v_m/m \rceil}$ has $m \lceil v_m/m \rceil \geq v_m$ columns).

LEMMA 7.11. For $\ell = (\ell_1, \dots, \ell_r)$, $n = \ell_1 + \dots + \ell_r$ and m in $\{1, \dots, n\}$, and for any pairwise distinct $\lambda_1, \dots, \lambda_r$ in $\overline{\mathbb{K}}^r$, the determinant $w_{\ell, m}(\xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r)$ of the $v_m \times v_m$ matrix $\mathcal{R}_{\ell, m}$ at $\lambda_1, \dots, \lambda_r$ is nonzero.

PROOF. We prove the nonvanishing property by exhibiting a vector $(\xi_1, \dots, \xi_n) \in \overline{\mathbb{K}}^n$ for which the evaluation $w_{\ell, m}(\xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r)$ is not zero. In what follows, for $i = 1, \dots, r$, recall that we write $\sigma_i = \ell_1 + \dots + \ell_{i-1}$, so that the rows involving $\bar{\lambda}_i$ in $\mathcal{P}_{\ell, m, \lceil v_m/m \rceil}$ have indices $\sigma_i+1, \dots, \sigma_i+\ell_i$ (see Equation (26)).

Assume first that m is invertible in \mathbb{K} , and choose δ in $\overline{\mathbb{K}}$ such that $\delta + \lambda_i \neq 0$ for $i = 1, \dots, r$. Then, for all i , the polynomial $x^m - (\delta + \lambda_i)$ is separable, since its discriminant is $m^m(\delta + \lambda_i)^{m-1}$, and we choose $\xi_{\sigma_i+1}, \dots, \xi_{\sigma_i+\min(\ell_i, m)}$ to be pairwise distinct roots of this polynomial in $\overline{\mathbb{K}}$. If $m < \ell_i$, we further take $\xi_{\sigma_i+m+1}, \dots, \xi_{\sigma_i+\ell_i}$ arbitrary in $\overline{\mathbb{K}}$ (note that $w_{\ell, m}$ does not depend on these quantities). Now, for any ξ, λ such that $\xi^m = \delta + \lambda$, and for $j \geq 1$, we have $\lambda^j = \xi^{jm} + \sum_{k=1}^j \binom{j}{k}(-\delta)^k \xi^{(j-k)m}$. Up to invertible linear combinations of its columns, $\mathcal{R}_{\ell, m}(\xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r)$ is thus the Vandermonde matrix at the roots $\xi_{\sigma_i+1}, \dots, \xi_{\sigma_i+\min(\ell_i, m)}$, $i = 1, \dots, r$. Since the λ_i 's are pairwise distinct, all these roots are pairwise distinct too, so the determinant $w_{\ell, m}(\xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r)$ is nonzero.

If m is 0 in \mathbb{K} , then for all i , $x^m + x - \lambda_i$ is separable, since its discriminant is $(-1)^{m(m-1)/2} \neq 0$. Again, choosing distinct roots of these polynomials and performing linear combinations of the columns of $\mathcal{R}_{\ell, m}$ leads to a nonzero Vandermonde determinant. \square

In several steps, we now study the rank of $\text{Hk}_{m, d}^{(a, f)}$ and show it is $v_m^{(a, f)}$ for d large enough. Note that unlike in Section 7.2 where we were working with $v_m = n$, additional ingredients are necessary in order to deduce this rank from those of $K_{m, d}^{(a, f)}$ and $L_{m, d}^{(a, f)}$.

If the rank of $\text{Hk}_{m, d}^{(a, f)}$ is $v_m^{(a, f)}$ for some $d \geq 0$, then it is $v_m^{(a, f)}$ for all $d \geq \lceil v_m^{(a, f)}/m \rceil$.

LEMMA 7.12. Let $\ell, \xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r, a, f$ and m be as in Proposition 7.8. If Hk_{m, d_0} has rank v_m for some $d_0 \geq \lceil v_m/m \rceil$, and if $w_{\ell, m}(\xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r)$ and $w_{\ell, m}(1/\xi_1, \dots, 1/\xi_n, \lambda_1, \dots, \lambda_r)$ from Lemma 7.11 are nonzero, then $\text{Hk}_{m, d}$ has rank v_m for all $d \geq \lceil v_m/m \rceil$.

PROOF. Since $w_{\ell, m}(\xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r)$ is nonzero, $\mathcal{P}_{\ell, m, \lceil v_m/m \rceil}(\xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r)$ has rank at least v_m , and so does $K_{m, \lceil v_m/m \rceil}$ (Lemma 7.10).

As a result, for $d \geq \lceil v_m/m \rceil$, $K_{m, d}$ still has rank exactly v_m (recall that this rank cannot exceed v_m , by Equation (30)). Thus, for such d , there exists a nonsingular $P \in \mathbb{K}^{(md) \times (md)}$ such that $K_{m, d}P = [K_{m, \lceil v_m/m \rceil} \ 0]$, where the zero matrix is $n \times (m(d - \lceil v_m/m \rceil))$. In the same way, since $w_{\ell, m}(1/\xi_1, \dots, 1/\xi_n, \lambda_1, \dots, \lambda_r)$ is nonzero, Lemma 7.10 also implies that $L_{m, \lceil v_m/m \rceil}$ has rank v_m , therefore there exists a nonsingular $Q \in \mathbb{K}^{(md) \times (md)}$ such that $QL_{m, d} = [(L_{m, \lceil v_m/m \rceil})^\top \ 0]^\top$. We obtain

$$Q \text{Hk}_{m, d} P = QL_{m, d}K_{m, d}P = \begin{pmatrix} \text{Hk}_{m, \lceil v_m/m \rceil} & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{K}^{(md) \times (md)},$$

which shows that for $d \geq \lceil v_m/m \rceil$ we have $\text{rank}(\text{Hk}_{m, d}) = \text{rank}(\text{Hk}_{m, \lceil v_m/m \rceil})$. \square

The rank of $\text{Hk}_{m, r}^{(a, f)}$ is $v_m^{(a, f)}$ generically. To establish that the rank of $\text{Hk}_{m, r}$ is v_m for generic choices of ξ_1, \dots, ξ_n , we introduce a decomposition into vector spaces associated to the λ_i 's. We then study these spaces separately; their dimensions are $\min(\ell_i, m)$, respectively, leading as expected to a total dimension $\sum_{i=1}^r \min(\ell_i, m) = v_m$.

This is achieved through a description of the images of the mappings $\kappa_{m,d}$ and $\lambda_{m,d}$ in terms of polynomials. Given positive integers $\ell = (\ell_1, \dots, \ell_r)$ and ξ_1, \dots, ξ_n in $\overline{\mathbb{K}}^n$, define

$$P_{i,j} = \sum_{k \in S_i} \xi_k^j \mathcal{L}_k \in \overline{\mathbb{K}}[x], \quad i = 1, \dots, r, \quad j \geq 0, \quad (32)$$

with the Lagrange polynomials $\mathcal{L}_1, \dots, \mathcal{L}_n$ and the sets S_1, \dots, S_r from Equation (26).

LEMMA 7.13. *Let $\ell, \xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r, a, f$ and m be as in Proposition 7.8 and let d be a positive integer. The image of $\kappa_{m,d}$ lies in the linear span of the v_m linearly independent polynomials $P_{i,j}$ from Equation (32), for $1 \leq i \leq r$ and $0 \leq j < \min(\ell_i, m)$.*

PROOF. Let $V(x, y) = \sum_{j=0}^{m-1} c_j(y)x^j$ belong to $\overline{\mathbb{K}}[x, y]_{<(m,d)}$. Lagrange interpolation gives

$$\kappa_{m,d}(V) = V(x, a) \bmod f = \sum_{k=1}^n V(\xi_k, a(\xi_k)) \mathcal{L}_k.$$

Since $V(\xi_k, a(\xi_k)) = \sum_{j=0}^{m-1} c_j(a(\xi_k)) \xi_k^j$, we deduce

$$\kappa_{m,d}(V) = \sum_{i=1}^r \sum_{j=0}^{m-1} c_j(\lambda_i) P_{i,j}.$$

For $i = 1, \dots, r$, at most ℓ_i of the polynomials $P_{i,j}$, $j = 0, \dots, m-1$, can be linearly independent, since they are all linear combinations of ℓ_i linearly independent \mathcal{L}_k . On the other hand, the polynomials $P_{i,j}$ for $j = 0, \dots, \ell_i - 1$ are linearly independent, due to the linear independence of the polynomials \mathcal{L}_k , and the invertibility of the Vandermonde matrix $[\xi_k^j]_{0 \leq j < \ell_i} \in \overline{\mathbb{K}}^{\ell_i \times \ell_i}$. This proves that the image of $\kappa_{m,d}$ is included in the span of the polynomials $P_{i,j}$, for $i = 1, \dots, r$ and $j = 0, \dots, \min(\ell_i, m) - 1$, as claimed. \square

This polynomial-based interpretation then allows us to use the following decomposition.

LEMMA 7.14. *Let $\ell, \xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r, a, f$ and m be as in Proposition 7.8. The rank of $H_{K_{m,r}}$ is the sum of the dimensions of the vector spaces*

$$V_i = \text{Span}([P_{i,j}]_0^{m-1}, j = 0, \dots, \min(\ell_i, m) - 1) \quad (33)$$

with the polynomials $P_{i,j}$ from Equation (32) for $i = 1, \dots, r$.

PROOF. We first claim that for $d = r$, $K_{m,r}$ has rank v_m , or equivalently (Lemma 7.10) that $\mathcal{P}_{\ell, m, r}$ has rank v_m at $(\xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r)$. Indeed, we can extract from $\mathcal{P}_{\ell, m, r}$ a $v_m \times v_m$ submatrix by keeping the first $\min(\ell_i, m)$ rows indexed by $\bar{\lambda}_i$, for $i = 1, \dots, r$, and the columns containing the monomials $\bar{\lambda}^{i-1}, \dots, \bar{\lambda}^{i-1} \xi^{\min(\ell_i, m)-1}$, for $i = 1, \dots, r$. The columns of this matrix contain the evaluations of the polynomials $x^i y^j$ for $i = 0, \dots, r-1$ and $j = 0, \dots, \min(\ell_{i+1}, m) - 1$ at the points (λ_i, ξ_{ij}) for $i = 1, \dots, r$ and $\xi_{ij} = \xi_{\ell_1+\dots+\ell_{i-1}+j}$ for $j = 1, \dots, \min(\ell_i, m)$. Now, up to linear combinations of its columns, the determinant of this matrix is the same as that of the matrix whose columns evaluate the polynomials

$$Q_{ij}(x, y) := \prod_{h=1}^{i-1} (x - \lambda_h) \prod_{k=1}^{j-1} (y - \xi_{ik}), \quad i = 1, \dots, r, \quad j = 1, \dots, \min(\ell_i, m).$$

The latter matrix is triangular; its diagonal elements are $Q_{ij}(\lambda_i, \xi_{ij}) \neq 0$, showing that the matrix is nonsingular and therefore that $K_{m,r}$ has rank at least v_m . (More general determinant factorizations of this kind are considered by [18], [25, Section 2].) Using Equation (30) we deduce that $K_{m,r}$ has

rank exactly v_m as announced, and from Lemma 7.13, we know that the image of $\kappa_{m,r}$ is the span of the polynomials $P_{i,j}$ defined in that lemma.

It follows that the rank of $\text{Hk}_{m,r}$ is the dimension of the span of the image $\lambda_{m,r}(P_{i,j})$. For $i = 1, \dots, r$, and $j = 0, \dots, \min(\ell_i, m) - 1$,

$$\lambda_{m,r}(P_{i,j}) = ([P_{i,j} \text{ rem } f]_0^{m-1}, [aP_{i,j} \text{ rem } f]_0^{m-1}, \dots, [a^{r-1}P_{i,j} \text{ rem } f]_0^{m-1}),$$

and since the Lagrange polynomials are eigenvectors of multiplication by a , we get

$$\begin{aligned} \lambda_{m,r}(P_{i,j}) &= ([P_{i,j}]_0^{m-1}, [\lambda_i P_{i,j}]_0^{m-1}, \dots, [\lambda_i^{r-1} P_{i,j}]_0^{m-1}) \\ &= ([P_{i,j}]_0^{m-1}, \lambda_i [P_{i,j}]_0^{m-1}, \dots, \lambda_i^{r-1} [P_{i,j}]_0^{m-1}). \end{aligned}$$

Let $v_{i,j} = (0, \dots, 0, [P_{i,j}]_0^{m-1}, 0, \dots, 0)$ be $[P_{i,j}]_0^{m-1}$ times the i th canonical vector in $\overline{\mathbb{K}}[x]_{<m}^r$, seen as row vector. The span of the $\lambda_{m,r}(P_{i,j})$'s multiplied on the right by the inverse of the Vandermonde matrix associated to the λ_i 's is the span of the $v_{i,j}$'s. By grouping the $v_{i,j}$'s for each i , this yields a block-diagonal matrix with blocks that span the spaces V_i of the lemma. The result on the rank follows. \square

The dimensions of the vector spaces from Equation (33) can now be analyzed separately.

LEMMA 7.15. *Fix positive integers $\ell = (\ell_1, \dots, \ell_r)$ such that $\ell_1 + \dots + \ell_r = n$, and m in $\{1, \dots, n\}$. There exists a nonzero polynomial $z_{\ell,m} \in \mathbb{Z}[\xi_1, \dots, \xi_n]$ of degree at most $(n-1)(n-v_m)$ such that if pairwise distinct nonzero ξ_1, \dots, ξ_n do not form a zero of $z_{\ell,m}$, then V_i from Equation (33) has dimension $\min(\ell_i, m)$ for all i .*

PROOF. Take i in $\{1, \dots, r\}$, consider the set of indices $S_i = \{\ell_1 + \dots + \ell_{i-1} + 1, \dots, \ell_1 + \dots + \ell_i\}$ from Equation (26). Let then $A_i = \prod_{k \in S_i} (x - \xi_k)$, $B_i = \prod_{k \notin S_i} (x - \xi_k)$ and $C_i = 1/B_i \bmod A_i$. Note that A_i and B_i have respective degrees ℓ_i and $n - \ell_i$, and that C_i is well defined, since B_i and A_i have no common root.

For k in S_i , by construction, B_i divides the Lagrange polynomial \mathcal{L}_k , with a quotient of degree $n - 1 - \deg(B_i) = \ell_i - 1$. In view of Equation (32), B_i divides $P_{i,j} = \sum_{k \in S_i} \xi_k^j \mathcal{L}_k$, for all $j \geq 0$, and the quotient has degree less than ℓ_i . We now prove that it is actually equal to $x^j C_i \bmod A_i$. Since $f = A_i B_i$, for k in S_i , the Lagrange polynomial $\mathcal{L}_k = f / (f'(\xi_k)(x - \xi_k))$ satisfies

$$\frac{\mathcal{L}_k}{B_i} = \frac{1}{f'(\xi_k)} \frac{f}{B_i(x - \xi_k)} = \frac{1}{f'(\xi_k)} \frac{A_i}{x - \xi_k} = \frac{C_i(\xi_k)}{A'_i(\xi_k)} \frac{A_i}{x - \xi_k}.$$

In particular, for $j \geq 0$, $\xi_k^j \mathcal{L}_k / B_i$ takes the value $\xi_k^j C_i(\xi_k)$ at ξ_k , and 0 at all other roots of A_i . Taking the sum over all k in S_i then proves our claim that $P_{i,j} / B_i = x^j C_i \bmod A_i$. Since ξ_1, \dots, ξ_n are nonzero, $B_i(0)$ as well is nonzero, so B_i is invertible as a power series and $[P_{i,j} / B_i]_0^{m-1} = [[P_{i,j}]_0^{m-1} / B_i]_0^{m-1}$. Thus the truncated polynomials $[P_{i,j}]_0^{m-1}$, for $0 \leq j < \min(\ell_i, m)$, are linearly independent if and only if the truncated polynomials $[x^j C_i \bmod A_i]_0^{m-1}$ are.

When $\ell_i \leq m$, the polynomials $x^j C_i \bmod A_i$ have degree less than m and their linear independence follows from that of the polynomials x^j , $j = 0, \dots, \ell_i - 1$, since C_i is invertible modulo A_i . Thus in this case, we always have $\dim(V_i) = \ell_i = \min(\ell_i, m)$.

When $\ell_i > m$, we are going to prove that the polynomials $[x^j C_i \bmod A_i]_0^{m-1}$, $j = 0, \dots, m-1$, are linearly independent for a generic choice of ξ_1, \dots, ξ_n . To achieve this, define the matrix M_{C_i} whose entry (j, ℓ) is the coefficient of $x^{\ell-1}$ in $x^{j-1} C_i \bmod A_i$ for $j = 1, \dots, \ell_i$ and $\ell = 1, \dots, \ell_i$; this is the multiplication matrix by C_i modulo A_i . We also consider its inverse, the multiplication matrix M_{B_i} by B_i modulo A_i .

For our claim to hold, it is enough to guarantee that the $m \times m$ leading principal minor K_i of M_{C_i} be nonzero. We view this minor as a rational function in $\bar{\xi}_1, \dots, \bar{\xi}_n$: this is done by introducing the polynomials $\bar{A}_i = \prod_{k \in S_i} (x - \bar{\xi}_k)$, $\bar{B}_i = \prod_{k \notin S_i} (x - \bar{\xi}_k)$ and $\bar{C}_i = 1/\bar{B}_i \bmod \bar{A}_i$, all of which are in $\mathbb{Q}(\bar{\xi}_1, \dots, \bar{\xi}_n)[x]$. We can then define the matrices $M_{\bar{C}_i}$ and $M_{\bar{B}_i}$ of multiplication by respectively \bar{C}_i and \bar{B}_i modulo \bar{A}_i , and the $m \times m$ leading principal minor \bar{K}_i of $M_{\bar{C}_i}$. This is a rational function of $\bar{\xi}_1, \dots, \bar{\xi}_n$, whose evaluation at ξ_1, \dots, ξ_n gives the scalar $K_i \in \mathbb{K}$.

Note first that \bar{K}_i is not identically zero: if we evaluate all $\bar{\xi}_g$ at 0, for g in S_i , \bar{A}_i becomes x^{ℓ_i} , and the matrix $M_{\bar{C}_i}$ becomes lower triangular, with $1/\bar{B}_i(0) \neq 0$. It then remains to estimate the degree of a numerator of \bar{K}_i . The Schur complement formula gives $\bar{K}_i = \det(M_{\bar{C}_i})\bar{L}_i$, where \bar{L}_i is the $(\ell_i - m) \times (\ell_i - m)$ lower right minor of the inverse $M_{\bar{B}_i}$ of $M_{\bar{C}_i}$. The determinant of $M_{\bar{C}_i}$ is the resultant of \bar{C}_i and \bar{A}_i , that is, $1/\prod_{g \in S_i, h \notin S_i} (\bar{\xi}_g - \bar{\xi}_h)$. On the other hand, \bar{L}_i is a polynomial in $\mathbb{Z}[\bar{\xi}_1, \dots, \bar{\xi}_n]$ (since \bar{B}_i and \bar{A}_i have coefficients in $\mathbb{Z}[\bar{\xi}_1, \dots, \bar{\xi}_n]$, and \bar{A}_i is monic in x).

For $s \geq 0$, write $x^s \bmod \bar{A}_i = c_{s,0} + \dots + c_{s,\ell_i-1}x^{\ell_i-1}$, for $c_{s,t} \in \mathbb{Z}[\bar{\xi}_1, \dots, \bar{\xi}_n]$. By induction on s , we obtain the bound $\deg(c_{s,t}) \leq s - t$. From this, it follows that all entries of $M_{\bar{B}_i}$ have degree at most $n - 1$, and that \bar{L}_i has degree at most $(n - 1)(\ell_i - m) \leq n\ell_i$. To conclude the proof, we let $z_{\ell,m}$ be the product of the polynomials \bar{L}_i , for i such that $\ell_i > m$. The degree bound follows from remarking that $\sum_{\ell_i > m} (\ell_i - m) = n - v_m$. \square

Genericity polynomials and degree bounds. Until here, the conditions we have seen are the nonvanishing of $w_{\ell,m}(\xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r)$, $w_{\ell,m}(1/\xi_1, \dots, 1/\xi_n, \lambda_1, \dots, \lambda_r)$, and $z_{\ell,m}(\xi_1, \dots, \xi_n)$. When nonzero, the first two quantities allow us to apply Lemma 7.12 and obtain the rank of $\text{Hk}_{m,\lceil n/m \rceil}$ from any Hk_{m,d_0} with $d_0 \geq \lceil v_m/m \rceil$; the third condition $z_{\ell,m}(\xi_1, \dots, \xi_n) \neq 0$ allows us to take $d_0 = r$ thanks to Lemmas 7.14 and 7.15.

The bound on the degree of $w_{\ell,m}$ in $\bar{\xi}_1, \dots, \bar{\xi}_n$ follows from summing the degrees of the columns in $\mathcal{P}_{\ell,m,\lceil v_m/m \rceil}$. Each block of m columns involves degrees $1 + \dots + (m - 1) = m(m - 1)/2$, and we consider $\lceil v_m/m \rceil$ such blocks (the last one may not be complete), for a total of at most $(v_m + m)(m - 1)/2$. Next, consider the term $w_{\ell,m}(1/\bar{\xi}_1, \dots, 1/\bar{\xi}_n, \bar{\lambda}_1, \dots, \bar{\lambda}_r)$, which is not a polynomial in the $\bar{\xi}_i$'s. To estimate the degree of its numerator, observe that it is a $v_m \times v_m$ -minor of the matrix

$$\begin{pmatrix} 1 & \frac{1}{\bar{\xi}_1} & \dots & \frac{1}{\bar{\xi}_1^{m-1}} & \bar{\lambda}_1 & \frac{1}{\bar{\xi}_1} \bar{\lambda}_1 & \dots & \frac{1}{\bar{\xi}_1^{m-1}} \bar{\lambda}_1^{\lceil v_m/m \rceil - 1} \\ & \vdots & & & & & \vdots & \\ 1 & \frac{1}{\bar{\xi}_n} & \dots & \frac{1}{\bar{\xi}_n^{m-1}} & \bar{\lambda}_r & \frac{1}{\bar{\xi}_n} \bar{\lambda}_r & \dots & \frac{1}{\bar{\xi}_n^{m-1}} \bar{\lambda}_r^{\lceil v_m/m \rceil - 1} \end{pmatrix}$$

Factoring out (on the right) the diagonal matrix with diagonal $(1/\bar{\xi}_i^{m-1})_{1 \leq i \leq n}$, we see that the nonvanishing of $w_{\ell,m}(1/\xi_1, \dots, 1/\xi_n, \lambda_1, \dots, \lambda_r)$ is equivalent to the nonvanishing of the corresponding $v_m \times v_m$ -minor $\tilde{w}_{\ell,m}$ in

$$\begin{pmatrix} \bar{\xi}_1^{m-1} & \bar{\xi}_1^{m-2} & \dots & 1 & \bar{\xi}_1^{m-1} \bar{\lambda}_1 & \bar{\xi}_1^{m-2} \bar{\lambda}_1 & \dots & \bar{\lambda}_1^{\lceil v_m/m \rceil - 1} \\ & \vdots & & & & & \vdots & \\ \bar{\xi}_n^{m-1} & \bar{\xi}_n^{m-2} & \dots & 1 & \bar{\xi}_n^{m-1} \bar{\lambda}_r & \bar{\xi}_n^{m-2} \bar{\lambda}_r & \dots & \bar{\lambda}_r^{\lceil v_m/m \rceil - 1} \end{pmatrix}.$$

The degree upper bound for $\tilde{w}_{\ell,m}$ is $(v_m + m)(m - 1)/2$, as for $w_{\ell,m}$.

We then take $\Gamma_{\ell,m} = w_{\ell,m} \tilde{w}_{\ell,m} z_{\ell,m}$ to prove Proposition 7.8. For the degree estimate, note that $(v_m + m)(m - 1) + (n - 1)(n - v_m) \leq 2n^2$. For correctness, take pairwise distinct nonzero ξ_1, \dots, ξ_n in $\overline{\mathbb{K}}$ and let $a \in \mathbb{K}[x]$ take distinct values $\lambda_1, \dots, \lambda_r$ at ξ_1, \dots, ξ_n , with multiplicities ℓ_1, \dots, ℓ_r . As before, we write $f = (x - \xi_1) \cdots (x - \xi_n)$, and we assume that f is in $\mathbb{K}[x]$. Finally, we suppose that $\Gamma_{\ell,m}(\xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r)$ is nonzero. Lemmas 7.14 and 7.15 show that for $d_0 = r$, we have

$\text{rank}(\text{Hk}_{m,r}^{(a,f)}) = v_m$. Since $r \geq \lceil v_m/m \rceil$, by Lemma 7.12, it is then also the case for $\text{Hk}_{m,d}^{(a,f)}$ for all $d \geq \lceil v_m/m \rceil$, as claimed.

The only remaining claim is that for any pairwise distinct $\lambda_1, \dots, \lambda_r, \Gamma_{\ell,m}(\bar{\xi}_1, \dots, \bar{\xi}_n, \lambda_1, \dots, \lambda_r)$ is a nonzero polynomial in $\bar{\xi}_1, \dots, \bar{\xi}_n$. That $z_{\ell,m}$ is nonzero is in Lemma 7.15 (this polynomial does not depend on $\lambda_1, \dots, \lambda_r$); Lemma 7.11 proves that $w_{\ell,m}(\bar{\xi}_1, \dots, \bar{\xi}_n, \lambda_1, \dots, \lambda_r)$ is nonzero. That lemma also implies that $w_{\ell,m}(1/\bar{\xi}_1, \dots, 1/\bar{\xi}_n, \lambda_1, \dots, \lambda_r)$ is nonzero (as a rational function), and as a consequence, this is also the case for $\tilde{w}_{\ell,m}(\bar{\xi}_1, \dots, \bar{\xi}_n, \lambda_1, \dots, \lambda_r)$. The claim for $\Gamma_{\ell,m}$ is thus proved.

8 A RANDOMIZED COMPOSITION ALGORITHM THROUGH CHANGE OF BASIS

In this section, we give the base case of our modular composition algorithm that is used when f is either separable or purely inseparable (which includes the case of power series). The core Algorithm `MODULARCOMPOSITIONBASECASE` is studied in Section 8.1, and a variation for computing annihilating polynomials is given in Section 8.2.

The algorithm of Section 4.2 performs bivariate modular composition within our target complexity bound, assuming the knowledge of a matrix of relations with appropriate dimension and degree. Since such a matrix of relations of $\mathcal{M}_m^{(a,f)}$ may not exist for general a and f , Algorithm `MODULARCOMPOSITIONBASECASE` transports the computation of $g(a)$ in $\mathbb{A} = \mathbb{K}[x]/\langle f \rangle$ to an isomorphic algebra that is expected to be more favorable to the computation.

More precisely, we pick a random $\gamma \in \mathbb{K}[x]_{<n}$; generically, its minimal polynomial $\mu_\gamma \in \mathbb{K}[y]$ has degree n and is also its characteristic polynomial χ_γ , so that the powers of γ generate \mathbb{A} . This induces the \mathbb{K} -algebra isomorphism ϕ_γ of Equation (20); Step 5 of Algorithm `MODULARCOMPOSITIONBASECASE` then computes a polynomial representative α of $\phi_\gamma(a \bmod f)$ using the change of basis algorithm of Section 6. Note that a matrix of relations $R_m^{(\gamma,f)}$ is also obtained at Step 5 in preparation for the final stage. Then, with good probability, the conditions for the efficient computation of a certified matrix of relations $R_m^{(\alpha,\mu_\gamma)}$ of $\mathcal{M}_m^{(\alpha,\mu_\gamma)}$ via the approach of Section 5.4 are fulfilled. Step 8 of Algorithm `MODULARCOMPOSITIONBASECASE` computes this matrix of relations, which then allows us to obtain the polynomial $\beta = g(\alpha) \bmod \mu_\gamma$ at Step 9, as seen in Section 4.2. The solution $b = g(a) \bmod f$ to the initial problem is finally recovered by applying ϕ_γ^{-1} to $\beta \bmod \mu_\gamma$, which amounts to computing $b = \beta(\gamma) \bmod f$. Since we already have $R_m^{(\gamma,f)}$ at our disposal, b is obtained with the algorithm of Section 4.2 as well.

Proposition 8.1 in Section 8.1 shows the correctness of this strategy and bounds its complexity. We then study the probability of success for f separable and f purely inseparable. The main point is to ensure that appropriate matrices of relations $R_m^{(\gamma,f)}$ and $R_m^{(\alpha,\mu_\gamma)}$ are actually available. For Steps 5 and 11 where a random γ is involved, we directly rely on the generic properties of the associated block Hankel matrix $\text{Hk}_{m,\lceil n/m \rceil}^{(\gamma,f)}$ (Proposition 7.6). For the computation of $R_m^{(\alpha,\mu_\gamma)}$ we use the fact that α and μ_γ are sufficiently generic, hence also give access to good properties for the associated block Hankel matrix after the change of basis.

The probability of failure for a general separable f is bounded in Section 8.3. The power series case and, more generally, the case of purely inseparable f are treated in Sections 8.4 and 8.5. For such f , the success of Algorithm `MODULARCOMPOSITIONBASECASE` is proven in Section 8.4 under some assumptions on the valuation of the input polynomial a and the characteristic of \mathbb{K} . Still in the case of f purely inseparable, a complete algorithm is then given in Section 8.5: when the valuation is large (with respect to the target value $m \sim n^\eta$ with η from Equation (3)), then the minimal polynomial of a modulo f has small degree and we use the extension of Shoup's algorithm seen in Section 3.1.3. For fields \mathbb{K} of small characteristic, we adapt Bernstein's composition algorithm for power series [6] to our general context.

ALGORITHM 8.1 MODULARCOMPOSITIONBASECASE(f, a, g, r)

Input: f of degree n in $\mathbb{K}[x]$, $a \in \mathbb{K}[x]_{<n}$, $g \in \mathbb{K}[y]$, $r \in \mathbb{K}^{n+\lceil n^\eta \rceil}$

Output: $b = g(a) \bmod f$ or FAIL

- 1: **if** $n = 1$ **then return** $g(a)$ ▷ $a \in \mathbb{K}$
- 2: $g \leftarrow g(y - r_1)$, $a \leftarrow a(x) + r_1$; **if** $\gcd(a, f) \neq 1$ **then return** FAIL
- 3: $f \leftarrow f(x + r_2)$; $a \leftarrow a(x + r_2)$; **if** $f(0) = 0$ **then return** FAIL
- 4: $m \leftarrow \lceil n^\eta \rceil$ ▷ *With η from Equation (3)*
- 5: **Change of basis:** compute a polynomial α such that $\alpha \equiv \phi_Y(a \bmod f) \pmod{\mu_Y}$
▷ *Getting a basis of relations $R^{(Y,f)}$ and the minimal polynomial μ_Y of $Y \bmod f$*
- 6: $\gamma \leftarrow r_3 + r_4x + \cdots + r_{n+2}x^{n-1}$
 $(R^{(Y,f)}, \mu_Y, \alpha)$ ← ▷ **CHANGEOFBASIS**($f, \gamma, a, m, \lceil n/m \rceil$)
- 7: **if** this call returned FAIL **then return** FAIL
- 8: **if** $\mu_Y(0) = 0$ **then return** FAIL
- 9: substitute “ y ” by “ x ” in μ_Y and α , which are then in $\mathbb{K}[x]$
▷ *Compute a matrix of relations for (α, μ_Y)*
- 10: $R^{(\alpha, \mu_Y)} \leftarrow \text{MATRIXOFRELATIONS}(\mu_Y, \alpha, m, \lceil n/m \rceil, (r_{n+i})_{3 \leq i \leq m})$ ▷ *Algorithm 5.2*
- 11: **if** this call returned FAIL **then return** FAIL
- 12: **Bivariate modular composition in the new basis:** $\beta \equiv g(\alpha) \pmod{\mu_Y}$ ▷ *Algorithm 4.1*
- 13: $\beta \leftarrow \text{BIVARIATEMODULARCOMPOSITIONWITHRELATIONMATRIX}(\mu_Y, \alpha, g, R^{(\alpha, \mu_Y)})$
- 14: substitute “ x ” by “ y ” in β , which is then in $\mathbb{K}[y]$
- 15: **Inverse change of basis:** $b \equiv \phi^{-1}(\beta \bmod \mu_Y) \pmod{f}$
 $b \leftarrow \text{BIVARIATEMODULARCOMPOSITIONWITHRELATIONMATRIX}(f, \gamma, \beta, R^{(Y,f)})$ ▷ *Algorithm 4.1*
- 16: **return** $b(x - r_2)$

8.1 Randomized Composition

The procedure is detailed in Algorithm **MODULARCOMPOSITIONBASECASE**. It uses $n+m$ parameters from \mathbb{K} that are available as a sequence r of length $n+m$. The coefficients of the random polynomial γ are given as part of the input as r_i , for $3 \leq i \leq n+2$; we require further parameters in order to reduce to the case where $f(0) \neq 0$ and $\gcd(a, f) = 1$ (Remarks 3.8 and 5.7), and for the random column combination performed by Algorithm **MATRIXOFRELATIONS**.

The parameter m could be taken arbitrarily in $\{1, \dots, n\}$, but we choose the specific value $m = \lceil n^\eta \rceil$, with η from Equation (3), as this choice minimizes the overall cost. The following proposition describes the output of the procedure; the probability of failure is bounded in Sections 8.3 and 8.4.

PROPOSITION 8.1. *Given $f \in \mathbb{K}[x]$ of degree n , $a \in \mathbb{K}[x]_{<n}$, $g \in \mathbb{K}[y]$ with $\deg(g) = O(n)$ and $r \in \mathbb{K}^{n+m}$ with $m = \lceil n^\eta \rceil$ and η from Equation (3), Algorithm **MODULARCOMPOSITIONBASECASE** returns either $g(a) \bmod f$ or FAIL; it uses $\tilde{O}(n^\kappa)$ operations in \mathbb{K} , with $\kappa < 1.43$ as in Equation (1).*

PROOF. If $n = 1$ then as a has degree 0, the result is $g(a) \in \mathbb{K}$ and the algorithm is correct. The rest of the proof assumes $n > 1$.

Steps 2 and 3 ensure that $\gcd(a, f) = 1$ and $f(0) \neq 0$. This does not impact the complexity, as shifting a polynomial of degree $O(n)$ can be achieved in $\tilde{O}(n)$ arithmetic operations [9, Chap. 1, Pb. 3.5]. The same observation applies to the last step.

At Step 5, if Algorithm **CHANGEOFBASIS** does not return FAIL then by Proposition 6.1 the matrix $R^{(Y,f)}$ is a basis of relations of $\mathcal{M}_m^{(Y,f)}$, $\mu_Y = \chi_Y$, and $\alpha(Y) \equiv a \bmod f$. It follows that $\mu_\alpha = \mu_a$ since the quotient algebras are isomorphic, and $\mu_\alpha(0) = \mu_a(0)$ implies $\gcd(\alpha, \mu_Y) = \gcd(a, f) = 1$. If the

ALGORITHM 8.2 ANNIHILATINGPOLYNOMIAL(f, a, r)

Input: f of degree n in $\mathbb{K}[x]$, $a \in \mathbb{K}[x]_{<n}$, $r \in \mathbb{K}^{n+\lceil n^\eta \rceil}$

Output: μ nonzero in $\mathbb{K}[y]_{\leq 4n}$ such that $\mu(a) \equiv 0 \pmod{f}$ or FAIL

- 1: **if** $n = 1$ **then return** $y - a$ $\triangleright a \in \mathbb{K}$
- 2: \triangleright Compute a matrix of relations $R^{(\alpha, \mu_Y)}$ for (α, μ_Y) with $\alpha = \phi_Y(a)$
execute Steps 3 to 8 of Algorithm 8.1
- 3: **if** FAIL has been returned by one of these steps **then return** FAIL $\triangleright [53, \text{Algorithm 2}]$
- 4: $\mu \leftarrow \det(R^{(\alpha, \mu_Y)})$
- 5: **return** μ

test at Step 6 does not fail then the specifications for the call to Algorithm **MATRIXOFRRELATIONS** are met; from Proposition 5.8, if Step 8 does not return FAIL then the matrix $R^{(\alpha, \mu_Y)}$ is a matrix of relations in $\langle \mu_Y, y - \alpha \rangle$. Both these matrices of relations have dimension at most $2(m - 1)$, and degree at most $2\lceil n/m \rceil$; they are obtained in $\tilde{O}(m^\omega d + c(n, m, d)) = \tilde{O}(m^\omega d + md^{\omega/2})$ operations, with $d = \lceil n/m \rceil$. This is $\tilde{O}(n^\kappa)$ arithmetic operations, according to Equations (1) and (3) and the choice of m at Step 4.

The variable substitutions at Steps 7 and 10 are harmless; they make notation match with that in Algorithms **MATRIXOFRRELATIONS** and **BIVARIATEMODULARCOMPOSITIONWITHRELATIONMATRIX**.

At Step 9, within the same complexity bound as above by Proposition 4.4, β is computed such that $\beta \equiv g(\alpha) \pmod{\mu_Y}$ (these polynomials are temporarily in x). After the substitution of Step 10 the latter relation implies the existence of a polynomial $h \in \mathbb{K}[y]$ such that

$$\beta(y) = g(\alpha(y)) + h(y)\mu_Y(y).$$

Since $\mu_Y(y) \equiv 0 \pmod{f}$, evaluating this identity at $y = \gamma$ results in $b = \beta(\gamma) = g(a) \pmod{f}$ at Step 11. \square

8.2 Randomized Annihilating Polynomial

If the choice of γ ensures that the isomorphism ϕ_Y is well defined (the powers of γ generate \mathbb{A}), then a univariate polynomial μ over \mathbb{K} is such that $\mu(a) \equiv 0 \pmod{f}$ if and only $\mu(\alpha) \equiv 0 \pmod{\mu_Y}$. Since Algorithm **MODULARCOMPOSITIONBASECASE** computes a matrix of relations in $\langle \mu_Y, y - \alpha \rangle$ at Step 8, an algorithm for computing such a μ follows from the results of Section 4.3.

COROLLARY 8.2. *Given $f \in \mathbb{K}[x]$ of degree n , $a \in \mathbb{K}[x]_{<n}$ and $r \in \mathbb{K}^{n+m}$ with $m = \lceil n^\eta \rceil$ and η from Equation (3), Algorithm **ANNIHILATINGPOLYNOMIAL** returns either FAIL or a nonzero $\mu \in \mathbb{K}[y]_{\leq 4n}$ such that $\mu(a) \equiv 0 \pmod{f}$; it uses $\tilde{O}(n^\kappa)$ operations in \mathbb{K} , with $\kappa < 1.43$ as in Equation (1).*

PROOF. If $n = 1$ then as $a \in \mathbb{K}$, $\mu = y - a$ is such that $\mu(a) = 0$ and the algorithm is correct. Now assume that $n > 1$. The annihilating polynomials are left unchanged by the substitution $x \leftarrow x + r_2$. As in the proof of Proposition 8.1, if failure does not occur then Step 8 computes a matrix of relations of $M_{m'}^{(\alpha, \mu_Y)}$, for some $m' \leq 2(m - 1)$, within the claimed complexity bound; this matrix has degree at most $2\lceil n/m \rceil$. Then Proposition 4.5 shows that μ annihilates $\alpha \pmod{\mu_Y}$ and thus $a \pmod{f}$, and that it has degree $\deg(\mu) \leq 4(m - 1)\lceil n/m \rceil$. This is at most $4n$ when $m \leq \sqrt{n}$, which is the case when $m = \lceil n^\eta \rceil$ with η as in Equation (3). The complexity then follows from the proof of Proposition 8.1 and Proposition 4.5 again. \square

8.3 Success of Randomization for Separable f

The probabilistic properties of the previous algorithms in the separable case are summarized in the following.

PROPOSITION 8.3. *Let a, f be polynomials in $\mathbb{K}[x]$ and g be in $\mathbb{K}[y]$, with f separable of degree n and $\deg(a) < n$. If $r_1, \dots, r_{n+\lceil n \rceil} \in \mathbb{K}$ are chosen uniformly and independently from a finite subset S of \mathbb{K} , then Algorithms **MODULARCOMPOSITIONBASECASE** and **ANNIHILATINGPOLYNOMIAL** return FAIL with probability at most $6n^2/\text{card}(S)$.*

PROOF. The success of modular composition in Algorithm **MODULARCOMPOSITIONBASECASE** and of the computation of an annihilating polynomial in Algorithm **ANNIHILATINGPOLYNOMIAL** relies on: finding good shifts r_1 and r_2 in the first two steps; a choice of γ such that $\mu_\gamma(0) \neq 0$ and μ_γ has degree n ; the availability of matrices of relations $R^{(\gamma, f)}$ and $R^{(\alpha, \mu_\gamma)}$. The probability estimate is obtained by showing the existence of polynomials whose zero sets contain the values of the parameters r_i where these properties do not hold. The probability of avoiding these zero sets is then handled by the Schwartz–Zippel lemma. In what follows, as in the algorithm, we write $m = \lceil n \rceil$.

(1) *A value of r_1 such that $\gcd(a + r_1, f) \neq 1$.* The resultant of $a(x) + r_1$ and $f(x)$ is nonzero of degree n in r_1 . Bad choices thus occur with probability at most $n/\text{card}(S)$.

(2) *A value of r_2 such that $f(0) \neq 0$ after the shift “ $x \leftarrow x + r_2$ ”.* The same reasoning as above applies to the coefficient of degree zero of $f(x + r_2)$.

The next properties all concern the same parameters (r_3, \dots, r_{n+m}) , so their failures are not independent events, and their joint probability is bounded using a product of polynomials encoding each of them. Below, we write $\bar{\gamma} = \bar{\gamma}_0 + \dots + \bar{\gamma}_{n-1}x^{n-1}$, with the $\bar{\gamma}_i$ ’s new indeterminates, and consider polynomials in $\bar{\mathbb{K}}[\bar{\gamma}_0, \dots, \bar{\gamma}_{n-1}]$ to quantify probabilities of failure.

(3) *The constant coefficient $\mu_\gamma(0)$ is not 0.* Write $f = c(x - \varphi_1) \dots (x - \varphi_n)$, for pairwise distinct φ_i in $\bar{\mathbb{K}}$ and $c \in \mathbb{K} \setminus \{0\}$. The roots of μ_γ are the values $\bar{\gamma}(\varphi_i)$, so $\mu_\gamma(0)$ being nonzero is equivalent to $\gcd(\gamma, f)$ being trivial. Thus, we let $\Delta_0 \in \mathbb{K}[\bar{\gamma}_0, \dots, \bar{\gamma}_{n-1}]$ be the resultant of $\bar{\gamma}$ and f . This polynomial has degree n , and choosing $\gamma = 1$ shows that it is not identically zero.

(4) *The minimal polynomial μ_γ has degree n .* For any $\gamma = \gamma_0 + \dots + \gamma_{n-1}x^{n-1}$ in $\mathbb{K}[x]_{<n}$, the characteristic polynomial $\chi_\gamma \in \mathbb{K}[y]$ of $\gamma \bmod f$ factors over $\bar{\mathbb{K}}[y]$ as $\chi_\gamma = \prod_{i=1}^n (y - \xi_i)$, where $\xi_i = \gamma(\varphi_i)$ for all i . We can thus let $\Delta_1 \in \bar{\mathbb{K}}[\bar{\gamma}_0, \dots, \bar{\gamma}_{n-1}]$ be the product $\prod_{1 \leq i < j \leq n} (\bar{\gamma}(\varphi_i) - \bar{\gamma}(\varphi_j))$. This is a polynomial of degree $n(n-1)/2$, and the previous discussion shows that $\Delta_1(\bar{\gamma}_0, \dots, \bar{\gamma}_{n-1}) \neq 0$ implies that χ_γ is separable. In that case, since the n distinct roots of χ_γ must be roots of μ_γ , we have $\chi_\gamma = \mu_\gamma$. Finally, the polynomial Δ_1 itself is nonzero since its value at $(0, 1, 0, \dots, 0)$, i.e. at $\gamma = x$, is not zero.

(5) *The computation of $R^{(\gamma, f)}$ does not fail.* Since $f(0) \neq 0$, Proposition 7.6 shows that the associated block Hankel matrix $\text{Hk}_{m, \lceil n/m \rceil}^{(\gamma, f)}$ has rank n as soon as the coefficients of γ avoid the zero set of a polynomial $\Delta_{f, m}$ of degree at most $2n^2/m$.

When this condition holds, Proposition 7.1 shows that the matrix fraction $X^T(yI_n - M_\gamma)^{-1}X$ is describable in degree $\lceil n/m \rceil$, and that $v_m^{(\gamma, f)} = v_{m, m}^{(\gamma, f)}$. Since we also have $\gcd(\gamma, f) = 1$ by the item above, and since the minimal polynomial μ_γ of $\gamma \bmod f$ has degree n , Proposition 6.1 concludes that the computation of $R^{(\gamma, f)}$ is successful.

(6) *The rank of $\text{Hk}_{m, d}^{(\alpha, \mu_\gamma)}$ is equal to $v_m^{(\alpha, \mu_\gamma)}$ for $d \geq \lceil v_m^{(\alpha, \mu_\gamma)}/m \rceil$.* When the previous properties are all satisfied, there exists a \mathbb{K} -algebra isomorphism $\phi_\gamma : \mathbb{K}[x]/\langle f \rangle \rightarrow \mathbb{K}[y]/\langle \mu_\gamma \rangle$ that maps a to α such that $\alpha(\gamma) \equiv a \bmod f$. Up to changing the indices of the roots φ_i , we can assume that a takes values $\lambda_1, \dots, \lambda_r$ at $\varphi_1, \dots, \varphi_n$ with multiplicities ℓ_1, \dots, ℓ_r , for some positive integers ℓ_1, \dots, ℓ_r , and pairwise distinct $\lambda_1, \dots, \lambda_r$ in $\bar{\mathbb{K}}$ (as in Section 7.4, the φ_i ’s are assumed to be ordered such that

$a(\varphi_1) = \dots = a(\varphi_{\ell_1}) = \lambda_1$, etc). Then, since $\xi_i = \gamma(\varphi_i)$ for all i , the relation $\alpha(\gamma) \equiv a \bmod f$ implies that $\alpha(\xi_i) = a(\varphi_i)$ for all i , so that α takes the values $\lambda_1, \dots, \lambda_r$ at ξ_1, \dots, ξ_n with multiplicities ℓ_1, \dots, ℓ_r .

The assumptions of Proposition 7.8 are satisfied. If $\Gamma_{\ell, m} \in \mathbb{Z}[\bar{\xi}_1, \dots, \bar{\xi}_n, \bar{\lambda}_1, \dots, \bar{\lambda}_r]$ is the polynomial defined in that proposition, then when $\Gamma_{\ell, m}(\xi_1, \dots, \xi_n, \lambda_1, \dots, \lambda_r)$ is nonzero, the rank of $\text{Hk}_{m, d}^{(\alpha, \mu_\gamma)}$ is $v_m^{(\alpha, \mu_\gamma)}$ for $d \geq \lceil v_m^{(\alpha, \mu_\gamma)} / m \rceil$.

The relevant polynomial is thus $\Delta_3 = \Gamma_{\ell, m}(\bar{\gamma}(\varphi_1), \dots, \bar{\gamma}(\varphi_n), \lambda_1, \dots, \lambda_r) \in \bar{\mathbb{K}}[\bar{\gamma}_0, \dots, \bar{\gamma}_{n-1}]$. Proposition 7.8 states that $\Gamma_{\ell, m}(\bar{\gamma}_0, \dots, \bar{\gamma}_{n-1}, \lambda_1, \dots, \lambda_r)$ is nonzero of degree at most $2n^2$; this is thus also the case for Δ_3 , since the transformation $(\bar{\gamma}_0, \dots, \bar{\gamma}_{n-1}) \mapsto (\bar{\gamma}(\varphi_1), \dots, \bar{\gamma}(\varphi_n))$ is linear and invertible (its matrix is the Vandermonde matrix at $\varphi_1, \dots, \varphi_n$).

(7) *The computation of $R^{(\alpha, \mu_\gamma)}$ does not fail.* When the previous properties are all satisfied, Proposition 7.1 applies with $a = \alpha$ and $f = \mu_\gamma$ and shows that $v_{m, m}^{(\alpha, \mu_\gamma)} = v_m^{(\alpha, \mu_\gamma)}$ and $X^T(yI_n - M_\alpha)^{-1}X$ is describable in degree $\lceil v_m^{(\alpha, \mu_\gamma)} / m \rceil$, where M_α is the multiplication matrix of α modulo μ_γ . Since $\mu_\gamma(0) \neq 0$ and $\gcd(\alpha, \mu_\gamma) = \gcd(a, f) = 1$, the assumptions of Proposition 5.8 are satisfied for the successful computation of $R^{(\alpha, \mu_\gamma)}$ (Algorithm **MATRIXOFRELATIONS**) with a probability of failure depending on the choices of $(r_{n+3}, \dots, r_{n+m})$ and bounded by $(m-1)/\text{card}(S)$.

Case $n = 1$. In that situation steps, (5)–(7) above simplify. Since its top left corner is the identity matrix, the rank of the block-Hankel matrix is at least 1, which is equal to n , and thus (5)–(7) succeed with probability 1 in that case.

Probability bounds. The polynomial $\Delta_0 \Delta_1 \Delta_{f, m} \Delta_3 \in \bar{\mathbb{K}}[\bar{\gamma}_0, \dots, \bar{\gamma}_{n-1}]$ is nonzero and has degree at most

$$d_{n, m} = n + \frac{n(n-1)}{2} + \frac{2n^2}{m} + 2n^2.$$

A choice of (r_3, \dots, r_{n+2}) that avoids its zero set ensures that the properties (3)–(6) hold. The other probabilities have been discussed in steps (1), (2) and (7) above. In summary, the probability of success is at least

$$\begin{cases} \left(1 - \frac{1}{\text{card}(S)}\right)^3 \geq 1 - \frac{3}{\text{card}(S)} & \text{if } n = 1, \\ \left(1 - \frac{n}{\text{card}(S)}\right)^2 \left(1 - \frac{d_{n, m}}{\text{card}(S)}\right) \left(1 - \frac{m-1}{\text{card}(S)}\right) \geq 1 - \frac{2n+d_{n, m}+m-1}{\text{card}(S)} & \text{otherwise.} \end{cases}$$

In the second expression, dividing the numerator of the fraction for $n \geq 2$ by n^2 gives

$$\frac{5}{2} + \frac{5}{2n} + \frac{2}{m} + \frac{m-1}{n^2},$$

which decreases as a function of n for $n \geq 0$ and, for fixed n , decreases as a function of m for $m \leq n$. Thus it reaches its maximum at $m = 1, n = 2$, where its value is $23/4 < 6$, proving the probability bound for $n \geq 2$, while $3 < 6$ deals with the case $n = 1$.

The assertion for Algorithm **ANNIHILATINGPOLYNOMIAL** follows: Step 2 apart, it fails in the same cases as Algorithm **MODULARCOMPOSITIONBASECASE**. \square

8.4 Success of Randomization for f Purely Inseparable: Small Valuation

Definition 8.4. A degree n polynomial f in $\mathbb{K}[x]$ is *purely inseparable* if it has only one root in an algebraic closure $\bar{\mathbb{K}}$, so that it factors as $f = (x - \xi)^n$ in $\bar{\mathbb{K}}[x]$; if n is a unit in \mathbb{K} , ξ itself is in \mathbb{K} .

In this section, we study the probabilistic aspects of Algorithm **MODULARCOMPOSITIONBASECASE** for such polynomials. If $a = a_0 + a_v(x - \xi)^v + a_{v+1}(x -$

$\xi)^{v+1} + \dots + a_{n-1}(x - \xi)^{n-1}$, with $a_0 = a(\xi)$ and $a_v \neq 0$, then the valuation $v = \text{val}_\xi(a - a(\xi))$ is the order of vanishing of $a - a(\xi)$ at $x = \xi$. For the moment, we work under two additional assumptions on this valuation: it is not 0 in \mathbb{K} , and it is at most the value chosen for m (which is $\lceil n^\eta \rceil$ in the algorithm, for the target complexity bound). The other cases are discussed in the next section.

PROPOSITION 8.5. *Let a, f be polynomials in $\mathbb{K}[x]$ and g be in $\mathbb{K}[y]$, with $f = (x - \xi)^n \in \mathbb{K}[x]$ where $\xi \in \overline{\mathbb{K}}$, and $\deg(a) < n$. Let p be the characteristic of \mathbb{K} . Suppose that $v = \text{val}_\xi(a - a(\xi))$ satisfies the following inequalities, with η as in Equation (3):*

$$v \leq \lceil n^\eta \rceil, \quad p = 0 \text{ or } v < p.$$

*Take $r_1 = 0$ if $\gcd(a, f) = 1$ and $r_1 = 1$ otherwise, $r_2 = 0$ if $f \neq x^n$ and $r_2 = 1$ otherwise. If $r_3, \dots, r_{n+\lceil n^\eta \rceil}$ are chosen uniformly and independently from a finite subset S of \mathbb{K} , then Algorithms **MODULARCOMPOSITIONBASECASE** and **ANNIHILATINGPOLYNOMIAL** return FAIL with probability at most $2n^4/\text{card}(S)$*

PROOF. The proof follows the same steps as in Section 8.3. As before, we write $m = \lceil n^\eta \rceil$.

(1), (2) *Values of r_1 and r_2 .* The choice of r_1 gives $\gcd(a + r_1, f) = 1$, and r_2 modifies the constant coefficient of f if necessary. The first two steps of Algorithm **MODULARCOMPOSITIONBASECASE** therefore provide polynomials that satisfy $\gcd(a, f) = 1$, $f(0) \neq 0$, and $v = \text{val}_\xi(a - a(\xi)) \leq m$.

(3) *The constant coefficient $\mu_\gamma(0)$ is not 0.* For any $\gamma = \gamma_0 + \dots + \gamma_{n-1}x^{n-1}$, the roots of the characteristic polynomial χ_γ of γ modulo f are the values taken by γ at the roots of f , counted with multiplicities. Since $f = (x - \xi)^n$ over $\overline{\mathbb{K}}[x]$, this implies that $\chi_\gamma = (y - \gamma(\xi))^n$. The minimal polynomial μ_γ then admits a similar factorization as $(y - \gamma(\xi))^c$, for some positive c .

Set $\Delta_0(\bar{\gamma}_0, \dots, \bar{\gamma}_n) = \sum_{i=0}^{n-1} \bar{\gamma}_i \xi^i$; this is a (nonzero) polynomial of degree 1 which is such that $\Delta_0(\gamma_0, \dots, \gamma_{n-1}) = \gamma(\xi)$, so the nonvanishing of this quantity gives the same property for $\mu_\gamma(0)$.

(4) *The minimal polynomial μ_γ has degree n .* Consider now $\Delta_1(\bar{\gamma}_0, \dots, \bar{\gamma}_{n-1}) = \sum_{i=1}^{n-1} i \bar{\gamma}_i \xi^{i-1}$, which is also a nonzero polynomial of degree 1. It is such that $\Delta_1(\gamma_0, \dots, \gamma_{n-1}) = \gamma'(\xi)$, so the nonvanishing of this quantity implies that $\text{val}_\xi(\gamma - \gamma(\xi)) = 1$. This implies that the powers $1, \gamma - \gamma(\xi), (\gamma - \gamma(\xi))^2, \dots, (\gamma - \gamma(\xi))^{n-1} \bmod (x - \xi)^n$ have respective valuations $0, 1, \dots, n-1$ at ξ , and thus are linearly independent. It follows that the minimal polynomial of $\gamma - \gamma(\xi)$ has degree n , and the same then holds for γ itself.

(5) *The computation of $R^{(\gamma, f)}$ does not fail.* Here, the argument of the previous section applies verbatim and relies on a polynomial $\Delta_{f, m}$ of degree at most $2n^2/m$.

(6) *The rank of $\text{Hk}_{m, d}^{(\alpha, \mu_\gamma)}$ is equal to n for $d \geq \lceil n/m \rceil$.* This step is the difficult one in the proof; note that the statement slightly deviates from the one in the separable case in the definition of the threshold degree $\lceil n/m \rceil$.

The result is obtained by bounding the degree of the numerator of a nonzero $n \times n$ minor of $\text{Hk}_{m, \lceil n/m \rceil}^{(\alpha, \mu_\gamma)}$, seen as a polynomial in $\gamma_0, \dots, \gamma_{n-1}$. We first show the existence of $\gamma \in \mathbb{K}[x]_{<n}$ and $\alpha \in \mathbb{K}[y]_{<n}$ such that the block Hankel matrix $\text{Hk}_{m, \lceil n/m \rceil}^{(\alpha, \mu_\gamma)}$ has rank n . This implies the existence of a nonzero $n \times n$ minor of this matrix; the degree of this minor as a polynomial in the coefficients of α and μ_γ is controlled by Lemma 7.7. These, in turn, are related to the coefficients of γ , using its explicit form for μ_γ and a linear system for the coefficients of α .

(6a) *Generic behavior.* We start by proving the existence of α of degree m in $\overline{\mathbb{K}}[y]$ and γ in $\overline{\mathbb{K}}[x]_{<n}$ such that we have $\alpha(\gamma) \equiv a \bmod f$, $\gamma(\xi) \neq 0$ and $\gamma'(\xi) \neq 0$.

Write $a = a_0 + a_v(x - \xi)^v + \cdots + a_{n-1}(x - \xi)^{n-1}$, with $a_0 = a(\xi)$ and, by definition of $v = \text{val}_\xi(a - a(\xi))$, $a_v \neq 0$ and $v > 0$. Since we also assume that the characteristic p of \mathbb{K} is either zero, or greater than v , this means in particular that v is a unit in \mathbb{K} . Let

$$\tilde{a}(x) = \frac{a - a_0}{a_v(x - \xi)^v} = 1 + \sum_{1 \leq i < n-v} \tilde{a}_i(x - \xi)^i,$$

with coefficients $\tilde{a}_i = a_{i+v}/a_v$.

- If $v = m$, we define $\tilde{a}(y) = y^v = y^m$. Since $v \neq 0$ in \mathbb{K} , $\tilde{a}'(1) \neq 0$ and Newton iteration guarantees the existence of a unique $\tilde{y} = 1 + \sum_{1 \leq i < n} \tilde{y}_i(x - \xi)^i$ such that $\tilde{y}^v \equiv \tilde{a} \pmod{f}$.
- If $v < m$, we define $\tilde{a}(y) = y^v + y^m$. This time, we let \tilde{y} be the unique polynomial of the form $\tilde{y} = 1 + \sum_{1 \leq i < n} \tilde{y}_i(x - \xi)^i$ such that $\tilde{y}^v + (x - \xi)^{m-v}\tilde{y}^m \equiv \tilde{a} \pmod{f}$. As previously, existence follows from Newton iteration, using the assumption $v \neq 0$ in \mathbb{K} .

In both cases, we set $\gamma = 1 + (x - \xi)\tilde{y} \pmod{f} \in \overline{\mathbb{K}}[x]$ and $\alpha = a_0 + a_v\tilde{a}(y - 1) \in \overline{\mathbb{K}}[y]$. We can then verify that all requirements $\alpha(\gamma) \equiv a \pmod{f}$, $\gamma(\xi) \neq 0$ and $\gamma'(\xi) \neq 0$ are satisfied.

Since μ_Y then has degree n , and since $\mu_Y(0) = (-\gamma(\xi))^n$ is nonzero, Proposition 7.3 shows that $\text{Hk}_{m, d}^{(\alpha, \mu_Y)}$ has rank n for $d \geq \lceil n/m \rceil$.

(6b) *A polynomial in $\mathbb{K}[\bar{a}_0, \dots, \bar{a}_{n-1}, \bar{f}_0, \dots, \bar{f}_{n-1}]$.* The existence of γ and α implies that of a nonzero $n \times n$ minor δ of $\text{Hk}_{m, \lceil n/m \rceil}^{(\alpha, \chi_Y)}$. Let then $\Delta \in \mathbb{K}[\bar{a}_0, \dots, \bar{a}_{n-1}, \bar{f}_0, \dots, \bar{f}_{n-1}]$ be the corresponding minor of $\text{Hk}_{m, \lceil n/m \rceil}^{(\bar{a}, \bar{f})}$, where $\bar{a} = \bar{a}_0 + \cdots + \bar{a}_{n-1}x^{n-1}$ and $\bar{f} = \bar{f}_0 + \cdots + \bar{f}_{n-1}x^{n-1} + x^n$ are polynomials whose coefficients are indeterminates. Lemma 7.7 shows that this is a polynomial of degree at most $2n^2/m$ in $\bar{a}_0, \dots, \bar{a}_{n-1}$ and $2n^2(n-1)/m$ in $\bar{f}_0, \dots, \bar{f}_{n-1}$.

(6c) *The rational functions $\bar{\alpha}_0, \dots, \bar{\alpha}_{n-1}$.* Next, with $\bar{y} = \bar{y}_0 + \cdots + \bar{y}_{n-1}x^{n-1}$ a polynomial whose coefficients are indeterminates, we consider $\bar{\alpha}$ such that $\bar{\alpha}(\bar{y}) \equiv a \pmod{f}$. The coefficients of $\bar{\alpha}$ are given as solutions of the linear system $\bar{\alpha}(\bar{y}) \equiv a \pmod{f}$, thus they are rational functions $\bar{\alpha}_0, \dots, \bar{\alpha}_{n-1}$ in $\mathbb{K}(\bar{y}_0, \dots, \bar{y}_{n-1})$. In this paragraph, we bound the degrees of their numerators and denominators in $\mathbb{K}[\bar{y}_0, \dots, \bar{y}_{n-1}]$, using power series inversion and composition.

We first consider the solution u to $u(\bar{y}) \equiv x - \xi \pmod{(x - \xi)^n}$, or equivalently $u(\bar{y}) \equiv x \pmod{x^n}$, with $\bar{y} = \bar{y}(x + \xi)$. We write $\bar{y} = \bar{y}_0 + \bar{y}_1x + \cdots + \bar{y}_{n-1}x^{n-1}$, where the coefficients $\bar{y}_0, \dots, \bar{y}_{n-1}$ are linear in $\bar{y}_0, \dots, \bar{y}_{n-1}$, with in particular $\bar{y}_0 = \Delta_0$ and $\bar{y}_1 = \Delta_1$. We can then write $u = \sum_{j=1}^{n-1} u_j(y - \Delta_0)^j$, where for $j \geq 1$, the coefficient u_j is a rational function in $\bar{y}_1, \dots, \bar{y}_{n-1}$, with numerator of degree $j-1$ in $\bar{y}_1, \dots, \bar{y}_{n-1}$ and denominator Δ_1^{2j-1} . More generally, for $i \geq 1$, the power u^i has valuation i , and for $j \geq i$, the coefficient of $(y - \Delta_0)^j$ in it is a rational function with numerator of degree $j-i$ in $\bar{y}_1, \dots, \bar{y}_{n-1}$ and denominator Δ_1^{2j-i} .

It follows that if we write $a = a_0 + a_v(x - \xi)^v + \cdots + a_{n-1}(x - \xi)^{n-1}$, then the solution $\bar{\alpha}$ to the equation $\bar{\alpha}(\bar{y}) \equiv a \pmod{f}$ is given by $\bar{\alpha} = a_0 + a_v u^v + \cdots + a_{n-1} u^{n-1} \pmod{(y - \Delta_0)^n}$. Once we rewrite $\bar{\alpha}$ as $\bar{\alpha}_0 + \cdots + \bar{\alpha}_{n-1} y^{n-1}$, we see that the coefficients $\bar{\alpha}_0, \dots, \bar{\alpha}_{n-1}$ are rational functions with numerator of degree at most $2n-3$ in $\bar{y}_0, \dots, \bar{y}_{n-1}$, and denominator Δ_1^{2n-3} .

(6d) *The polynomial Δ_2 .* We now evaluate the indeterminates \bar{a}_i and \bar{f}_i in the minor Δ of (6b) at the coefficients of $\bar{\alpha}$ and $\chi_{\bar{Y}} = (y - \Delta_0)^n$, respectively. Write $(y - \Delta_0)^n$ as $\bar{q}_0 + \cdots + \bar{q}_{n-1} y^{n-1} + y^n$, so that $\bar{q}_i = \binom{n}{i}(-\Delta_0)^{n-i}$ for all i . It follows that $\Delta(\bar{\alpha}_0, \dots, \bar{\alpha}_{n-1}, \bar{q}_0, \dots, \bar{q}_{n-1})$ is a rational function in the indeterminates $\bar{y}_0, \dots, \bar{y}_{n-1}$, which can be written as

$$\Delta(\bar{\alpha}_0, \dots, \bar{\alpha}_{n-1}, \bar{q}_0, \dots, \bar{q}_{n-1}) = \frac{\Delta_2(\bar{y}_0, \dots, \bar{y}_{n-1})}{\Delta_1(\bar{y}_0, \dots, \bar{y}_{n-1})^\epsilon}, \quad (34)$$

for some polynomial Δ_2 of degree at most

$$\frac{2n^2}{m}(2n-3) + \frac{2n^2(n-1)}{m}(n-1) = \frac{2n^2(n^2-2)}{m},$$

and for some integer exponent $\epsilon \leq 2n^2(2n-3)/m$.

Consider again the polynomials γ and α in (6a), and their coefficients $\gamma_0, \dots, \gamma_{n-1}$ and $\alpha_0, \dots, \alpha_{n-1}$ (with actually $\alpha_{m+1} = \dots = \alpha_{n-1} = 0$). We saw that γ satisfies $\Delta_1(\gamma_0, \dots, \gamma_{n-1}) = \gamma'(\xi) \neq 0$, which implies that the rational functions $\bar{\alpha}_0, \dots, \bar{\alpha}_{n-1}$ are well defined at $\gamma_0, \dots, \gamma_{n-1}$ and take $\alpha_0, \dots, \alpha_{n-1}$ for values there. This implies that the nonzero minor δ is $\delta = \Delta_2(\gamma_0, \dots, \gamma_{n-1})/\Delta_1(\gamma_0, \dots, \gamma_{n-1})^\epsilon$, and in particular that Δ_2 is a nonzero polynomial.

Probability bounds. The end of the proof is as in the previous section. The polynomial $\Delta_0\Delta_1\Delta_{f,m}\Delta_2$ in $\bar{\mathbb{K}}[\bar{\gamma}_0, \dots, \bar{\gamma}_{n-1}]$ has degree at most

$$1 + 1 + \frac{2n^2}{m} + \frac{2n^2(n^2-2)}{m} = \frac{2(n^4-n^2+m)}{m};$$

we can now readily verify that a choice of (r_3, \dots, r_{n+2}) that avoids its zeros ensures that properties (3)–(6) hold. For (3)–(5), this follows immediately from the definitions.

To see that (6) holds, that is, that $\text{Hk}_{m,d}^{(\alpha, \mu_\gamma)}$ has rank n for $d \geq \lceil n/m \rceil$, recall that the algorithm constructs $\gamma = r_3 + r_4x + \dots + r_{n+2}x^{n+1}$. Properties (3)–(4) show that μ_γ has degree n , and that its constant coefficient is nonzero. Since in particular $\Delta_1(r_3, \dots, r_{n+2}) \neq 0$, we deduce that the rational functions $\bar{\alpha}_0, \dots, \bar{\alpha}_{n-1}$ of (6c) are well defined at (r_3, \dots, r_{n+2}) , and that they give the coefficients of the unique polynomial α such that $\alpha(\gamma) \equiv a \pmod{f}$. Since $\Delta_2(r_3, \dots, r_{n+2}) \neq 0$, it follows from Equation (34) that $\text{Hk}_{m, \lceil n/m \rceil}^{(\alpha, \mu_\gamma)}$ has rank n (and thus similarly for $\text{Hk}_{m, d}^{(\alpha, \mu_\gamma)}$, for $d \geq \lceil n/m \rceil$).

The other probabilities have been discussed in step (1)–(2) above and in step (7) of the previous section. Altogether, this gives a probability of success at least

$$\left(1 - \frac{2(n^4-n^2+m)/m}{\text{card}(S)}\right) \left(1 - \frac{m-1}{\text{card}(S)}\right) \geq 1 - \frac{2(n^4-n^2+m)/m + m-1}{\text{card}(S)}.$$

Dividing the numerator of the last fraction by n^4/m gives

$$2 - \frac{2n^2 - m^2 - m}{n^4} \leq 2,$$

where the last inequality comes from $m \leq n$. □

8.4.1 Note. In Proposition 8.5, the role of the condition on the valuation being nonzero in \mathbb{K} is shown by the following example. Take a field \mathbb{K} of characteristic 2, $n = 6$, $m = 3$, $f = (x-1)^6$ and $a = (x-1)^2$. Then for any $\gamma \in \mathbb{K}[x]_{<6}$, the four polynomials $(1, a, \gamma^2, a\gamma^2)$ rem f belong to the vector space generated by $(1, x^2, x^4)$ and are therefore linearly dependent. Using the expression of M_α from Equation (10), we see that this implies that the block Krylov matrix $K_{m, n/m}^{(\alpha, \mu_\gamma)}$ of Equation (12) is singular, and thus so is $\text{Hk}_{m, n/m}^{(\alpha, \mu_\gamma)}$ regardless of the choice of γ .

A more general version of this counterexample when \mathbb{K} has characteristic $p > 0$ is obtained with $m = p+1$, $d = p$, $n = md$, and $\text{val}_\xi(a) = p$.

8.5 Complete Algorithm for f Purely Inseparable

We now extend Proposition 8.5 in order to cover all cases of composition modulo a purely inseparable polynomial f .

If p is the characteristic of \mathbb{K} , any purely inseparable f can be written as $f(x) = (x^{p^e} - c)^\ell$ with c in \mathbb{K} and e, ℓ in \mathbb{N} such that p does not divide ℓ , and $e = 0$ if $p = 0$ [29]; in particular the

degree n of f is equal to $p^e\ell$. We assume that the parameters e, ℓ and c are known, since this is the case when our algorithms have to handle this situation; indeed in the next section we introduce separable factorization techniques that allow us to compute them.

8.5.1 Large Valuation. If $v = \text{val}_\xi(a(x) - a(\xi))$ satisfies $v > \lceil n^\eta \rceil$, the minimal polynomial of a in $\mathbb{K}[x]/\langle f \rangle$ factors over $\overline{\mathbb{K}}$ as $\mu_a(y) = (y - a(\xi))^\delta$, with $\delta = \lceil n/v \rceil \leq \lceil n^{1-\eta} \rceil$. Since the latter degree is small compared with n , this case is handled efficiently by Algorithm [MODULARCOMPOSITION-SMALLMINIMALPOLYNOMIAL](#) from Section 3.1.

8.5.2 Small Characteristic. In the case $0 < p \leq \lceil n^\eta \rceil$, our algorithm is based on Bernstein's composition algorithm for power series [6], which we adapt to work modulo $f(x) = (x^{p^e} - c)^\ell$. See also [34, Algorithm 3.1] for another extension of Bernstein's result, which is, however, not sufficient to reach our target cost for the specific kind of modulus we work with.

If $e = 0, p^e = 1$ and we are working modulo $f = (x - c)^\ell$, with $\ell = n$. In this case, to compute $b = g(a) \text{ rem } f$, we write $\tilde{a}(x) = a(x + c)$, we compute $\tilde{b} = g(\tilde{a}) \text{ rem } x^\ell$, then we obtain b as $\tilde{b}(x - c)$. The bottleneck is the computation of $g(\tilde{a}) \text{ rem } x^\ell$, which can be done in $\tilde{O}(p\ell)$ operations in \mathbb{K} using Bernstein's algorithm (in Algorithm [COMPOSITIONMODULOINSEPARABLE-SMALLCHARACTERISTIC](#), that algorithm is called [POWERSERIESCOMPOSITION-SMALLCHARACTERISTIC](#)).

Suppose now that $e \geq 1$. Write $g = \sum_{i=0}^{p-1} g_i(y^p)y^i$, with $g_i \in \mathbb{K}[y]$ of degree less than $p^{e-1}\ell$. Write also $a(x) = \sum_{i=0}^{n-1} a_i x^i$, and let $\bar{a}(x) = \sum_{i=0}^{n-1} a_i^p x^i$, so that $a^p(x) = \bar{a}(x^p)$. It follows that

$$g(a) \text{ rem } (x^{p^e} - c)^\ell = \sum_{i=0}^{p-1} \bar{g}_i a^i \text{ rem } (x^{p^e} - c)^\ell$$

where, for all $0 \leq i \leq p-1$,

$$\bar{g}_i(x) = g_i(a^p(x)) \text{ rem } (x^{p^e} - c)^\ell = g_i(\bar{a}(x^p)) \text{ rem } (x^{p^e} - c)^\ell.$$

If we define $h_i = g_i(\bar{a}) \text{ rem } (x^{p^{e-1}} - c)^\ell$, it follows that $\bar{g}_i = h_i(x^p)$, so that

$$g(a) \text{ rem } (x^{p^e} - c)^\ell = \sum_{i=0}^{p-1} h_i(x^p) a^i \text{ rem } (x^{p^e} - c)^\ell.$$

The following lemma summarizes the cost of this procedure.

LEMMA 8.6. *For a field \mathbb{K} of characteristic $p > 0$, given a purely inseparable polynomial $f = (x^{p^e} - c)^\ell$ of degree $n = \ell p^e$, $a \in \mathbb{K}[x]_{<n}$ and $g \in \mathbb{K}[y]_{<n}$, Algorithm [COMPOSITIONMODULOINSEPARABLE-SMALLCHARACTERISTIC](#) returns $g(a) \text{ rem } f$ and uses $\tilde{O}(pn)$ operations in \mathbb{K} .*

PROOF. Correctness follows from the previous description. For the runtime analysis when $e = 0$, the result is Bernstein's. For $e > 0$, apart from the p recursive calls, Step 8 takes $\tilde{O}(n)$ operations (we raise all coefficients of a to the power $p \leq n$), Step 11 takes $\tilde{O}(n)$ operations by repeated squaring, and Step 12 takes $\tilde{O}(pn)$ operations, using Horner's rule. Remembering that $n = \ell p^e$, we deduce that the runtime $T(e, p, \ell)$ satisfies $T(e, p, \ell) = pT(e-1, p, \ell) + \tilde{O}(p^{e+1}\ell)$ and $T(0, p, \ell) \in \tilde{O}(p\ell)$. This resolves to $T(e, p, \ell) \in \tilde{O}(p^{e+1}\ell)$, which is $\tilde{O}(pn)$. \square

8.5.3 Main Algorithm. Combining the previous results gives Algorithm [COMPOSITIONMODULOINSEPARABLE](#). It first tests whether the characteristic of \mathbb{K} is small enough for Algorithm [COMPOSITIONMODULOINSEPARABLE-SMALLCHARACTERISTIC](#) to run within our prescribed runtime. Otherwise, rather than computing the valuation v , it simply calls Algorithm [MODULARCOMPOSITION-SMALLMINIMALPOLYNOMIAL](#); in case of failure, it falls back on

ALGORITHM 8.3 COMPOSITIONMODULOINSEPARABLE-SMALLCHARACTERISTIC(c, e, ℓ, a, g)

Input: \mathbb{K} has characteristic $p > 0$,
 c in \mathbb{K} , e in \mathbb{N} and ℓ in $\mathbb{N}_{>0}$ such that $f = (x^{p^e} - c)^\ell$ has degree $n = \ell p^e$,
 a in $\mathbb{K}[x]_{<n}$, g in $\mathbb{K}[y]_{<n}$

Output: $g(a) \bmod f$

- 1: **if** $e = 0$ **then**
- 2: $a \leftarrow a(x + c)$
- 3: $b \leftarrow \text{POWERSERIESCOMPOSITION-SMALLCHARACTERISTIC}(x^n, a, g)$ ▷ [6, Section 2]
- 4: **return** $b(x - c)$
- 5: **else**
- 6: Write $g = g_0(y^p) + \dots + g_{p-1}(y^p)y^{p-1}$
- 7: Write $a = a_0 + \dots + a_{n-1}x^{n-1}$
- 8: $\bar{a} \leftarrow a_0^p + \dots + a_{n-1}^p x^{n-1}$
- 9: **for** $i = 0, \dots, p - 1$ **do**
- 10: $h_i \leftarrow \text{COMPOSITIONMODULOINSEPARABLE-SMALLCHARACTERISTIC}(c, e - 1, \ell, \bar{a}, g_i)$
- 11: $f \leftarrow (x^{p^e} - c)^\ell$
- 12: **return** $h_0(x^p) + \dots + h_{p-1}(x^p)a^{p-1} \bmod f$

ALGORITHM 8.4 COMPOSITIONMODULOINSEPARABLE(c, e, ℓ, a, g, r)

Input: c in \mathbb{K} , e in \mathbb{N} and ℓ in $\mathbb{N}_{>0}$ such that $f = (x^{p^e} - c)^\ell$ has degree $n = \ell p^e$, where p is the characteristic of \mathbb{K} ,
 a in $\mathbb{K}[x]_{<n}$, g in $\mathbb{K}[y]_{<n}$, $r \in \mathbb{K}^{n+\lceil n^\eta \rceil}$ with η from Equation (3)

Output: $b = g(a) \bmod f$ or FAIL

- 1: $n \leftarrow \ell p^e$
- 2: **if** $0 < p \leq \lceil n^\eta \rceil$ **then**
- 3: **return** $\text{COMPOSITIONMODULOINSEPARABLE-SMALLCHARACTERISTIC}(c, e, \ell, a, g)$
- 4: $f \leftarrow (x^{p^e} - c)^\ell$
- 5: $b \leftarrow \text{MODULARCOMPOSITION-SMALLMINIMALPOLYNOMIAL}(f, a, g, \lceil n^{1-\eta} \rceil, (r_i)_{0 \leq i < n})$
- 6: **if** $b \neq \text{FAIL}$ **then return** b
- 7: **if** $\gcd(a, f) = 1$ **then** $r_1 = 0$ **else** $r_1 = 1$; **if** $c \neq 0$ **then** $r_2 = 0$ **else** $r_2 = 1$
- 8: **return** $\text{MODULARCOMPOSITIONBASECASE}(f, a, g, r)$ ▷ Proposition 8.5

Algorithm **MODULARCOMPOSITIONBASECASE**. As previously, the algorithm takes as input a vector r that plays the role of random parameters.

PROPOSITION 8.7. *For a field \mathbb{K} of characteristic p , given c, e, ℓ such that $f = (x^{p^e} - c)^\ell$ is purely inseparable of degree $n = \ell p^e$ ($e = 0$ if $p = 0$), $a \in \mathbb{K}[x]_{<n}$, $g \in \mathbb{K}[y]_{<n}$ and $r \in \mathbb{K}^{n+m}$ with $m = \lceil n^\eta \rceil$ and η from Equation (3), Algorithm **COMPOSITIONMODULOINSEPARABLE** uses $\tilde{O}(n^\kappa)$ operations in \mathbb{K} , with $\kappa < 1.43$ as in Equation (1), and returns either $g(a) \bmod f$, or FAIL.*

If the entries of r are chosen uniformly and independently from a finite subset S of \mathbb{K} , then the algorithm returns $g(a) \bmod f$ with probability at least $1 - 2n^4/\text{card}(S)$.

PROOF. We first analyze the runtime. For a small characteristic $0 < p \leq \lceil n^\eta \rceil$, then Algorithm **COMPOSITIONMODULOINSEPARABLE-SMALLCHARACTERISTIC** has cost $\tilde{O}(pn)$ by Lemma 8.6, which is thus $\tilde{O}(n^{1+\eta}) \in \tilde{O}(n^{1+(\omega-1)\eta}) = \tilde{O}(n^\kappa)$ from Equation (3). Computing f takes time $\tilde{O}(n)$ by repeated squaring. By Lemma 3.2, the call to Algorithm **MODULARCOMPOSITION-**

SMALLMINIMALPOLYNOMIAL uses

$$\tilde{O}\left(n^{1+(1-\eta)(\omega_2/2-1)}\right) = \tilde{O}\left(n^{\eta+(1-\eta)(\omega_2/2)}\right) = \tilde{O}(n^\kappa)$$

operations in \mathbb{K} , and by Proposition 8.1, it is also the case for Algorithm **MODULARCOMPOSITIONBASECASE**. The specifications of the subroutines imply that the output can be either $g(a) \bmod f$ or FAIL, so only the probability analysis remains.

If $0 < p \leq \lceil n^\eta \rceil$, Lemma 8.6 shows that the output is $g(a) \bmod f$; hence, we may now assume that $p > \lceil n^\eta \rceil$, or $p = 0$. Let $\xi = c^{1/p^e} \in \mathbb{K}$, so that $f = (x - \xi)^n$ in $\mathbb{K}[x]$; let further v be the valuation of $a - a(\xi)$ at ξ . The minimal polynomial of a modulo f has degree $\delta = \lceil n/v \rceil$.

Suppose first that $v \leq \lceil n^\eta \rceil$, so that $\delta \geq \lceil n^1/\lceil n^\eta \rceil \rceil$. The value b computed at Step 4 is either $g(a) \bmod f$, or FAIL; let π be the probability of the former (for instance, by Lemma 3.2, $\pi = 0$ if $\delta > \lceil n^{1-\eta} \rceil$). If FAIL is returned at Step 4, then we enter Step 6. At this stage, we have inequalities $v \leq \lceil n^\eta \rceil < p$, or $v \leq \lceil n^\eta \rceil$ and $p = 0$, so by Proposition 8.5 the call to Algorithm **MODULARCOMPOSITIONBASECASE** returns $g(a) \bmod f$ with probability at least $1 - 2n^4/\text{card}(S)$. Overall, the probability of returning $g(a) \bmod f$ in this case is at least $\pi + (1 - \pi)(1 - 2n^4/\text{card}(S))$, which is at least $1 - 2n^4/\text{card}(S)$.

Suppose, on the other hand, that $v > \lceil n^\eta \rceil$, so that we have in particular $v \geq n^\eta$, and thus $\delta = \lceil n/v \rceil \leq \lceil n^{1-\eta} \rceil$. By Lemma 3.2, b computed at Step 4 is $g(a) \bmod f$ with probability at least $1 - n/\text{card}(S)$. If it is not the case, the algorithm enters Algorithm **MODULARCOMPOSITIONBASECASE**, which computes $g(a) \bmod f$ with a certain probability $\pi' \geq 0$. Overall, we return $g(a) \bmod f$ with probability at least $1 - n/\text{card}(S) + \pi' \geq 1 - n/\text{card}(S)$. \square

9 ALGORITHM FOR GENERAL f

We now present our Las Vegas Algorithm **MODULARCOMPOSITION** that computes $g(a) \bmod f$ for arbitrary input g, a, f . The analysis of this algorithm in Section 9.5 proves Theorem 1.1.

The starting point is the *separable decomposition* of f (Section 9.1), a generalization of square-free decomposition from fields of characteristic zero to arbitrary base fields. This yields a partial factorization $f = f_1 \cdots f_s$ into pairwise coprime factors. The algorithm then proceeds by computing $g(a)$ modulo each of these factors and the final result is obtained by Chinese remaindering in quasi-linear complexity [26, Section 10.3]. If p is the characteristic of \mathbb{K} then the factors f_i of the separable decomposition of f are the form $h_i(x^{p^{e_i}})^{\ell_i}$ (or more simply $h_i(x)^{\ell_i}$ when $p = 0$), with integers e_i, ℓ_i and separable $h_i \in \mathbb{K}[x]$. Composition modulo such an f_i is achieved via a \mathbb{K} -algebra isomorphism

$$\Psi_i : \mathbb{A}_i = \mathbb{K}[x]/\langle f_i(x) \rangle \rightarrow \mathbb{B}_i = \mathbb{K}[\theta, z]/\langle h_i(\theta), (z^{p^{e_i}} - \theta)^{\ell_i} \rangle$$

that maps x to z (Proposition 9.6). If \mathbb{L}_i denotes $\mathbb{K}[\theta]/\langle h_i(\theta) \rangle$, then, as a \mathbb{K} -vector space, $\mathbb{B}_i \simeq \mathbb{L}_i[z]/\langle (z^{p^{e_i}} - \bar{\theta}_i)^{\ell_i} \rangle$ with $\bar{\theta}_i$ the class of θ in \mathbb{L}_i . The computation of $g(a) \bmod f_i$ over \mathbb{K} is thus mapped to the composition

$$g(A_i) \bmod (z^{p^{e_i}} - \bar{\theta}_i)^{\ell_i}$$

over \mathbb{L}_i , with $A_i = \Psi_i(a \bmod f_i)$ and modulo the purely inseparable $(z^{p^{e_i}} - \bar{\theta}_i)^{\ell_i}$. In order to perform this last composition efficiently, it is also necessary to decrease the degree of g by first reducing g modulo the characteristic polynomial of A_i in $\mathbb{L}_i[z]/\langle (z^{p^{e_i}} - \bar{\theta}_i)^{\ell_i} \rangle$. We call *reduction* of g that step of the process (Proposition 9.8). It produces a representative of $G_i \in \mathbb{L}_i[y]$ such that $B_i = g(A_i) \in \mathbb{B}_i$ is obtained through the univariate modular composition

$$G_i(A_i) \bmod (z^{p^{e_i}} - \bar{\theta}_i)^{\ell_i},$$

which is computed with coefficients in \mathbb{L}_i . Finally, the class $g(a) \bmod f_i \in \mathbb{A}$ is recovered as $\Psi_i^{-1}(B_i)$. In practice, the algorithms working with elements of \mathbb{L}_i use polynomial representatives in $\mathbb{K}[\theta]_{<\deg(f_i)}$, that are the canonical lifts of their class.

The idea of using these homomorphisms was introduced by van der Hoeven and Lecerf in the case $e_i = 0$ [34]; it is extended to the general case in Sections 9.3 and 9.4. We keep their terminology, calling *untangling* an algorithm that computes the map Ψ_i and *tangling*, one which computes the reverse map. Both these operations can be performed efficiently (Section 9.3).

The univariate modular composition in $\mathbb{L}_i[z]$ modulo the purely inseparable polynomial $(z^{p^{e_i}} - \theta)^{\ell_i}$ can be achieved by Algorithm **COMPOSITIONMODULOINSEPARABLE** of Section 8.5 when \mathbb{L}_i is a field. In general, however, \mathbb{L}_i is a *product of fields*. In Section 9.2, the extension of the scope of our algorithms to this setting is obtained using a paradigm also due to van der Hoeven and Lecerf called *directed evaluation* [37].

Conventions. For h of degree d in $\mathbb{K}[\theta]$ and f in $\mathbb{K}[\theta, z]$, monic of degree n in z , and for any P in $\mathbb{K}[\theta, z]$, we denote by $P \bmod \langle h, f \rangle \in \mathbb{K}[\theta, z]_{<(d, n)}$ the polynomial obtained by reducing P first by f , then by h (this is the normal form of P modulo $\langle h, f \rangle$, if we see the latter as a Gröbner basis for the lexicographic order induced by $\theta < z$). Thus $P \bmod \langle h, f \rangle$ is a canonical lift of the class of P in $\mathbb{K}[\theta, z]/\langle h, f \rangle$. If $P \in \mathbb{K}[\theta, z]$, we use the notation $\bar{P}(z)$ to denote the class (projection) of P in $\mathbb{L}[z]$, where \mathbb{L} will be clear from the context.

9.1 Separable Decomposition

Let p be the characteristic of the field \mathbb{K} and let f in $\mathbb{K}[x]$ be of degree n . The *separable decomposition* of f is the set

$$\mathcal{S} = \{(h_1, e_1, \ell_1), \dots, (h_s, e_s, \ell_s)\}, \quad \text{with } h_i \in \mathbb{K}[x] \text{ and } e_i, \ell_i \in \mathbb{N} \text{ for all } i,$$

that satisfies the following properties, where we write $f_i = h_i(x^{p^{e_i}})^{\ell_i}$:

- (1) $f = cf_1 \cdots f_s$ with $c \in \mathbb{K} \setminus \{0\}$;
- (2) for all $i \neq j$ in $\{1, \dots, s\}$, f_i and f_j are coprime;
- (3) for all i in $\{1, \dots, s\}$, $h_i \in \mathbb{K}[x]$ is separable, monic and of positive degree d_i ;
- (4) for all i in $\{1, \dots, s\}$, $e_i = 0$ (if $p = 0$) or e_i is in \mathbb{N} (if $p > 0$);
- (5) for all i in $\{1, \dots, s\}$, ℓ_i is not divisible by p ;
- (6) for all $i \neq j$ in $\{1, \dots, s\}$, $(e_i, \ell_i) \neq (e_j, \ell_j)$.

The separable decomposition of f can be computed in $\tilde{O}(n)$ operations in \mathbb{K} using an algorithm due to Lecerf [57]. The special case when $p = 0$ recovers the more classical *square-free* factorization.

9.2 Composition over Products of Fields, Modulo Purely Inseparable f

Let h be separable of degree d in $\mathbb{K}[\theta]$, and consider f of the form $f = (z^{p^e} - c(\theta))^{\ell} \in \mathbb{K}[\theta, z]$, for integers $e \in \mathbb{N}$ and $\ell \in \mathbb{N}_{>0}$, where p is the characteristic of \mathbb{K} . Given A in $\mathbb{K}[\theta, z]_{<(d, n)}$ and G in $\mathbb{K}[\theta, y]_{<(d, n)}$, with $n = \deg_z(f) = \ell p^e$, we consider here the computation of $B = G(\theta, A) \bmod \langle h, f \rangle$.

This question is mapped to a univariate composition problem with coefficients in $\mathbb{L} = \mathbb{K}[\theta]/\langle h \rangle$: if we let \bar{A} , \bar{G} , \bar{B} and \bar{c} be the projections of respectively A , G , B and c in $\mathbb{L}[z]$, $\mathbb{L}[y]$, $\mathbb{L}[z]$ and \mathbb{L} (the degree constraints show that \bar{A} , \bar{G} , \bar{B} can be obtained without any calculation from A , G , B , and conversely), then $\bar{B} = \bar{G}(\bar{A}) \bmod (z^{p^e} - \bar{c})^{\ell}$ as an equality in $\mathbb{L}[z]$.

When h is irreducible, so that \mathbb{L} is a field, the algorithm of Section 8.5 applies over \mathbb{L} ; as reported in Proposition 8.7, if $n = \deg(f) = \ell p^e$, the runtime is $\tilde{O}(dn^{\kappa})$ operations in \mathbb{K} , coming from $\tilde{O}((\ell p^e)^{\kappa}) = \tilde{O}(n^{\kappa})$ times a factor in $\tilde{O}(d)$ for the cost of arithmetic operations in \mathbb{L} . However, we only assume h separable, so that \mathbb{L} is a *product of fields*. The key difference is the presence of

zero-divisors in \mathbb{L} : a nonzero element of \mathbb{L} is not necessarily invertible. Since the procedures in Section 8.5 use zero-tests and divisions, their direct application is not possible.

9.2.1 Directed Evaluation. The technique of *directed evaluation*, due to van der Hoeven and Lecerf [37], is an efficient version of the classical *dynamic evaluation* process [21].

In dynamic evaluation, prior to each zero-test or inversion, say by a quantity $q \in \mathbb{L}$, the computation of $h_1 = \gcd(q, h)$ gives the factorization $h = h_1 h_2$. Since h is separable, h_1 and h_2 are coprime, and \mathbb{L} can be decomposed as the product $\mathbb{L}_1 \times \mathbb{L}_2$, with $q = 0$ in $\mathbb{L}_1 = \mathbb{K}[\theta]/\langle h_1 \rangle$ and q invertible in $\mathbb{L}_2 = \mathbb{K}[\theta]/\langle h_2 \rangle$. Under the dynamic evaluation paradigm, the calculation can then be continued in two branches, working modulo h_1 and h_2 separately.

In directed evaluation, the idea is rather to run the entire program in a unique branch, then to apply the process recursively in residual branches after reduction of input data modulo the corresponding polynomial. We do not detail the underlying techniques, for which we refer to Sections 3 and 4 of [37], and simply apply their *panoramic evaluation* procedure [37, Algorithm 2]. It takes as input a computation tree \mathcal{T} over \mathbb{K} (see Section 2), a defining separable polynomial h of degree d for \mathbb{L} , and $\lambda = (\lambda_1, \dots, \lambda_s)$ in $\mathbb{K}[\theta]_{<d}^s$ (representing an input to \mathcal{T} in \mathbb{L}^s); it then returns a *panoramic value*, defined as follows.

Definition 9.1 ([37, Definition 1 and Lemma 2]). Given an input $(h, \lambda, \mathcal{T})$ as above, a *panoramic value* of \mathcal{T} at λ is a set of pairs $\{(h_1, \varepsilon_1), \dots, (h_t, \varepsilon_t)\}$, where

- h_1, \dots, h_t are polynomials in $\mathbb{K}[\theta]$ that satisfy $h = h_1 \cdots h_t$ (thus $\mathbb{L} \simeq \mathbb{L}_1 \times \cdots \times \mathbb{L}_t$, with $\mathbb{L}_i = \mathbb{K}[\theta]/\langle h_i \rangle$);
- for all i , ε_i is in $\mathbb{K}[\theta]_{<d_i}^{\ell_i}$ (representing an output in $\mathbb{L}_i^{\ell_i}$), with $d_i = \deg(h_i)$ and ℓ_i in \mathbb{N} ;
- for all $1 \leq i \leq t$, let $h_{i,1}, \dots, h_{i,k_i}$ be the factorization of h_i into irreducibles. For $1 \leq j \leq k_i$, let $\mathbb{L}_{i,j}$ be the field $\mathbb{K}[\theta]/\langle h_{i,j} \rangle$, and denote by $\pi_{i,j} : \mathbb{K}[\theta] \rightarrow \mathbb{L}_{i,j}$ the canonical projection $a \mapsto a \bmod h_{i,j}$ (the notation carries over to vectors over $\mathbb{K}[\theta]$). Then \mathcal{T} is supposed to be evaluable at $\pi_{i,j}(\lambda) \in \mathbb{L}_{i,j}^s$ for all i, j , and $\pi_{i,j}(\varepsilon_i) \in \mathbb{L}_{i,j}^{\ell_i}$ is the result of evaluating \mathcal{T} (seen as a computation tree over $\mathbb{L}_{i,j}$) at $\pi_{i,j}(\lambda)$, using the same branch of \mathcal{T} for all j .

The application of this method requires that one uses computation trees as the underlying computational model, which is the case here (Section 2). Crucially, the cost overhead is then $\tilde{O}(d)$ [37, Theorem 1], i.e., similar (up to logarithmic factors) to the one incurred if h were irreducible.

9.2.2 Algorithm. With Algorithm **COMPOSITIONMODULOINSEPARABLE-PRODUCTOFFIELDS** we apply panoramic evaluation (called **PANORAMIC** in our pseudocode) to Algorithm **COMPOSITIONMODULOINSEPARABLE** for modular composition over \mathbb{K} . Note that in addition to field elements, the latter algorithm also takes two integers e, ℓ as input. Panoramic evaluation can still be used in this context, since each choice of the parameters e, ℓ corresponds to a computation tree, to which the techniques described above apply. This yields a factorization of h , and performs the compositions modulo the corresponding factors; the final result is then reconstructed using Chinese remaindering.

PROPOSITION 9.2. For a field \mathbb{K} of characteristic p , given $h \in \mathbb{K}[\theta]$ separable of degree d , c in $\mathbb{K}[\theta]_{<d}$, integers e in \mathbb{N} and ℓ in $\mathbb{N}_{>0}$, A in $\mathbb{K}[\theta, z]_{<(d,n)}$, G in $\mathbb{K}[\theta, y]_{<(d,n)}$, r in $\mathbb{K}^{n+\lceil n^{\eta} \rceil}$ with $n = \ell p^e$ and η from Equation (3), Algorithm **COMPOSITIONMODULOINSEPARABLE-PRODUCTOFFIELDS** uses $\tilde{O}(d(\ell p^e)^{\kappa}) = \tilde{O}(dn^{\kappa})$ operations in \mathbb{K} , with $\kappa < 1.43$ as in Equation (1).

It returns either $G(\theta, A) \bmod \langle h, f \rangle \in \mathbb{K}[\theta, z]_{<(d,n)}$ or FAIL, with $f = (z^{p^e} - c)^{\ell}$. If the entries of r are chosen uniformly and independently from a finite subset S of \mathbb{K} , then the algorithm returns $G(\theta, A) \bmod \langle h, f \rangle$ with probability at least $1 - 2dn^4/\text{card}(S)$.

ALGORITHM 9.1 COMPOSITIONMODULOINSEPARABLE-PRODUCTOFFIELDS(h, c, e, ℓ, A, G, r)

Input: h separable of degree d in $\mathbb{K}[\theta]$,

c in $\mathbb{K}[\theta]_{<d}$, e in \mathbb{N} and ℓ in $\mathbb{N}_{>0}$ such that $f = (z^{p^e} - c)^\ell$ has degree $n = \ell p^e$,

$A \in \mathbb{K}[\theta, z]_{<(d,n)}$, $G \in \mathbb{K}[\theta, y]_{<(d,n)}$, $r \in \mathbb{K}^{n+\lceil n^\eta \rceil}$

Output: $B = G(\theta, A) \bmod \langle h, f \rangle$, or FAIL

- 1: \triangleright *Splitting $\mathbb{L} \simeq \mathbb{K}[\theta]/\langle h_1 \rangle \times \cdots \times \mathbb{K}[\theta]/\langle h_t \rangle$ and reductions of B , accordingly, using [37, Algorithm 2]*
 $\{(h_1, B_1), \dots, (h_t, B_t)\} \leftarrow \text{PANORAMIC}(\text{COMPOSITIONMODULOINSEPARABLE}, h, c, e, \ell, A, G, r)$
- 2: **if** any of the B_i 's equals FAIL **then return** FAIL
- 3: **return** CHINESEREMAINDERING($(B_1, \dots, B_t), (h_1, \dots, h_t)$)

PROOF. Combined with our Proposition 8.7, Theorem 1 in [37] gives the runtime estimate. In the pseudocode, the output of the panoramic evaluation is written as $\{(h_1, B_1), \dots, (h_t, B_t)\}$, where $h_1 \cdots h_t$ is a factorization of h (not necessarily into irreducibles), and for all i , either $B_i \in \mathbb{K}[\theta, z]_{<(d_i, n)}$ with $d_i = \deg(h_i)$, or $B_i = \text{FAIL}$. At the level of computation trees, a flag such as FAIL is obtained by setting a dedicated output value to 1 (and 0 otherwise); call flag_i this value, for $1 \leq i \leq t$. If $\text{flag}_i = 1$ (failure), we set $B_i = 0$ by convention, so in the rest of this proof, B_i is an element of $\mathbb{K}[\theta, z]$ for all i .

We use the following notation: for $1 \leq i \leq t$, the irreducible factors of h_i are written $h_{i,1}, \dots, h_{i,k_i}$. For $1 \leq j \leq k_i$, we then define $\bar{c}_{i,j}, \bar{A}_{i,j}, \bar{G}_{i,j}$ by taking c, A, G modulo $h_{i,j}$ and seeing them over the field $\mathbb{L}_{i,j} = \mathbb{K}[\theta]/\langle h_{i,j} \rangle$, so $\bar{c}_{i,j}$ is in $\mathbb{L}_{i,j}, \bar{A}_{i,j}$ in $\mathbb{L}_{i,j}[z]$ and $\bar{G}_{i,j}$ in $\mathbb{L}_{i,j}[y]$. The elements in the vector r are already in \mathbb{K} , and thus in $\mathbb{L}_{i,j}$. Finally, we let $\bar{B}_{i,j}$ be the polynomial obtained by taking $B_i \in \mathbb{K}[\theta, z]$ and projecting it to $\mathbb{L}_{i,j}[z]$ through reduction modulo $h_{i,j}$, and we set $\text{flag}_{i,j} = \text{flag}_i$ (recall that $\text{flag}_i \in \mathbb{K}$ is either 0 or 1).

Then, from Definition 9.1, the key property of the output of the first step is that for all indices i, j , $\text{flag}_{i,j}$ and $\bar{B}_{i,j}$ are the result of calling Algorithm COMPOSITIONMODULOINSEPARABLE on input $\bar{c}_{i,j}, e, \ell, \bar{A}_{i,j}, \bar{G}_{i,j}, r$ over the field $\mathbb{L}_{i,j}$. This implies, in particular, that our algorithm returns FAIL if and only if the computation fails over one of the fields $\mathbb{L}_{i,j}$.

To quantify the probability of this event, we apply Proposition 8.7 over all fields $\mathbb{L}_{i,j}$. For any given i, j , Proposition 8.7 shows that $\text{flag}_{i,j} = 1$ occurs with probability at most $2n^4/\text{card}(S)$. Since there are at most d such indices i, j , the probability that this happens for at least one pair of indices is at most $2dn^4/\text{card}(S)$. Assume none of the $\text{flag}_{i,j}$'s is 1, so that the algorithm does not return FAIL. Then, for all i, j , $\bar{B}_{i,j} \in \mathbb{L}_{i,j}[z]_{<n}$ is equal to $\bar{G}_{i,j}(\bar{A}_{i,j}) \bmod (z^{p^e} - \bar{c}_{i,j})^\ell$. In terms of bivariate polynomials, the Chinese Remainder Theorem then implies that for all i , B_i itself is equal to $G(\theta, A) \bmod \langle h_i, (z^{p^e} - c)^\ell \rangle \in \mathbb{K}[\theta, z]_{<(d_i, n)}$. In the last step of the algorithm, we further apply the Chinese Remainder Theorem coefficient-wise to the B_i 's with respect to z ; this gives us $G(\theta, A) \bmod \langle h, (z^{p^e} - c)^\ell \rangle$ as a polynomial in $\mathbb{K}[\theta, z]_{<(d, n)}$. The cost of this last step is in $\tilde{O}(d\ell p^e)$, so the proof is complete. \square

The complexity bound $\tilde{O}(dn^\kappa)$ in Proposition 9.2 indicates that the overhead coming from operations modulo $h(\theta)$ is $\tilde{O}(d)$, as pointed out previously.

9.3 Untangling and Tangling

In this subsection, we give the main tools (tangling, untangling and bivariate reduction) that are needed for reducing composition modulo powers of separable polynomials to the situation of the previous subsection. The central results are due to van der Hoeven and Lecerf [34] with $f = h(x)^\ell$ and h separable (Sections 9.3.1 and 9.3.2). We slightly generalize them to the case $f = h(x^{p^e})^\ell$ with $e > 0$ (Sections 9.3.3 and 9.3.4).

9.3.1 *Tangling and Untangling.* The starting point is the following observation.

LEMMA 9.3 ([34, SECTION 4.2]). *For h of degree d in $\mathbb{K}[x]$ and for a positive integer ℓ , there exists a \mathbb{K} -algebra homomorphism*

$$\begin{aligned}\psi_{h,\ell} : \mathbb{K}[x]/\langle h(x)^\ell \rangle &\rightarrow \mathbb{K}[\theta, z]/\langle h(\theta), (z - \theta)^\ell \rangle \\ x &\mapsto z.\end{aligned}$$

If moreover h is separable then $\psi_{h,\ell}$ is an isomorphism.

This homomorphism is a variant of the homomorphism $\pi_{h,\ell}$ considered by van der Hoeven and Lecerf, that maps $u \in \mathbb{K}[x]/\langle h(x)^\ell \rangle$ to $u(z + \theta) \in \mathbb{K}[\theta, z]/\langle h(\theta), z^\ell \rangle$. The morphism $\psi_{h,\ell}$ is obtained by composing $\pi_{h,\ell}$ with a translation $z \mapsto z - \theta$. It turns out that $\psi_{h,\ell}$ is more convenient than $\pi_{h,\ell}$ for our generalization in Section 9.3.3. van der Hoeven and Lecerf call **UNTANGLING**(h, ℓ, u) the algorithm which implements $\pi_{h,\ell}$; we use this terminology for the algorithm that implements $\psi_{h,\ell}$: given u in $\mathbb{K}[x]_{<d\ell}$, it computes $U \in \mathbb{K}[\theta, z]_{<(d,\ell)}$ such that $U = u(z) \bmod \langle h(\theta), (z - \theta)^\ell \rangle$. When h is separable, the inverse operation is called **TANGLING**(h, ℓ, U). Again, we use their terminology for the inverse of $\psi_{h,\ell}$.

LEMMA 9.4. *UNTANGLING and TANGLING (when defined) take $\tilde{O}(d\ell)$ operations in \mathbb{K} .*

PROOF. This is mostly in Reference [34]. First, it is easy to check that the algorithms 4.3 and 4.5 and the proofs of Proposition 4.6 and 4.10 of that reference do not make use of the separability of h . Next, translation can be performed in quasi-linear complexity over an arbitrary ring [28, Theorem. 4.5], so that the complexity estimate is unchanged for our variant of these algorithms. \square

9.3.2 *Bivariate Reduction.* The computation of the composition $g(a) \bmod h(x)^\ell$ for a separable h reduces to computing $\psi_{h,\ell}^{-1}(g(\psi_{h,\ell}(a \bmod h(x)^\ell)))$, where the inner composition is performed as a univariate composition in $\mathbb{L}[z]$ modulo $(z - \bar{\theta})^\ell$, with $\mathbb{L} = \mathbb{K}[\theta]/\langle h \rangle$.

In order to make use of the algorithms of the previous sections to perform this composition, it is necessary to first reduce the degree of g . Denote by A the canonical lift of $\psi_{h,\ell}(a \bmod h(x)^\ell)$, and by \bar{A} its projection in $\mathbb{L}[z]$. The idea is to reduce g modulo the characteristic polynomial $(y - \bar{A}(\bar{\theta}))^\ell \in \mathbb{L}[y]$ of $\bar{A}(z)$ modulo $(z - \bar{\theta})^\ell$.

This is achieved in two steps. For h of degree d , we let $\alpha \in \mathbb{K}[\theta]_{<d}$ be the canonical lift of $\bar{A}(\bar{\theta}) \in \mathbb{L}$. First, one computes the canonical lift of $\psi_{\mu,\ell}(g \bmod \mu^\ell)$, where μ is an annihilating polynomial of $\alpha \bmod h$. This produces $\tilde{G}(z, y) \in \mathbb{K}[z, y]_{<(\deg \mu, \ell)}$ such that

$$\tilde{G}(z, y) = \sum_{i=0}^{\ell-1} \tilde{G}_i(z) y^i = g(y) + \tilde{U}(z, y) \mu(z) + \tilde{V}(z, y) (y - z)^\ell$$

for some polynomials \tilde{U}, \tilde{V} in $\mathbb{K}[z, y]$.

Next, in view of $\mu(\alpha) \equiv 0 \bmod h$, a modular composition of each of the ℓ coefficients of this polynomial \tilde{G} in y with $\alpha(\theta)$ modulo $h(\theta)$ gives $G(\theta, y) \in \mathbb{K}[z, y]_{<(\deg \mu, \ell)}$ such that

$$G(\theta, y) = g(y) + U(\theta, y) h(\theta) + V(\theta, y) (y - \alpha(\theta))^\ell, \quad (35)$$

for some polynomials U, V in $\mathbb{K}[\theta, y]$. Equation (35) may also be read as $\bar{G}(\bar{A}) = g(\bar{A}) \bmod (z - \bar{\theta})^\ell$ over \mathbb{L} .

These two steps are detailed in Algorithm **BIVARIATEREDUCTION** below and correspond to Steps (2)-(4) of [34, Algorithm 4.2]. The runtime and probability analyses are new; they are based on the results of the previous sections.

ALGORITHM 9.2 BIVARIATEREDUCTION(h, ℓ, α, g, r)

Input: h separable, monic, of degree d in $\mathbb{K}[\theta]$, ℓ in $\mathbb{N}_{>0}$, α in $\mathbb{K}[\theta]_{<d}$, g in $\mathbb{K}[y]$, r in $\mathbb{K}^{d+\lceil d^\eta \rceil}$

Output: $G(\theta, y) = g(y) \text{ rem } \langle h(\theta), (y - \alpha(\theta))^\ell \rangle \in \mathbb{K}[\theta, y]_{<(d, \ell)}$, or FAIL

- 1: \triangleright Either $\mu = \text{FAIL}$, or μ is nonzero in $\mathbb{K}[y]_{\leq 4d}$ and $\mu(\alpha) \equiv 0 \pmod{h}$
- $\mu \leftarrow \text{ANNIHILATINGPOLYNOMIAL}(h, \alpha, r)$ \triangleright Algorithm 8.2
- if** $\mu = \text{FAIL}$ **then return** FAIL
- 2: $\tilde{G} \leftarrow \text{UNTANGLING}(\mu, \ell, g \text{ rem } \mu^\ell)$ $\triangleright \tilde{G}(y, z) \in \mathbb{K}[y, z]_{<(\deg(\mu), \ell)}$, Lemma 9.4
- 3: Write $\tilde{G} = \sum_{0 \leq i < \ell} \tilde{G}_i(y) y^i$ $\triangleright \tilde{G}_i \in \mathbb{K}[y]_{<\deg(\mu)}$
- 4: **for** $i = 0, \dots, \ell - 1$ **do**
 - $G_i \leftarrow \text{MODULARCOMPOSITIONBASECASE}(h, \alpha, \tilde{G}_i, r)$ $\triangleright G_i = \tilde{G}_i(\alpha) \text{ rem } h$ or FAIL, Algorithm 8.1
 - if** $G_i = \text{FAIL}$ **then return** FAIL
- 5: $G \leftarrow \sum_{0 \leq i < \ell} G_i y^i$ $\triangleright G$ is in $\mathbb{K}[\theta, y]_{<(d, \ell)}$
- 6: **return** G

LEMMA 9.5. Given h in $\mathbb{K}[\theta]$ monic, separable and of degree d , α in $\mathbb{K}[\theta]_{<d}$, g in $\mathbb{K}[y]$, r in $\mathbb{K}^{d+\lceil d^\eta \rceil}$ with η from Equation (3), and ℓ in $\mathbb{N}_{>0}$, Algorithm BIVARIATEREDUCTION uses $\tilde{O}(\deg(g) + d^\kappa \ell)$ operations in \mathbb{K} with $\kappa < 1.43$ as in Equation (1), and returns either $g \text{ rem } \langle h(\theta), (y - \alpha(\theta))^\ell \rangle$ or FAIL. If the entries of r are chosen uniformly and independently from a finite subset S of \mathbb{K} , then the algorithm returns $g \text{ rem } \langle h, (y - \alpha)^\ell \rangle$ with probability at least $1 - 6(\ell + 1)d^2/\text{card}(S)$.

PROOF. The reduction of $g \pmod{\mu^\ell}$ is justified by the fact that $\mu(\alpha)^\ell = 0 \pmod{h^\ell}$. The correction of the rest of the algorithm when Step 6 is reached follows from the discussion above.

Since h is separable, Proposition 8.3 applies; it shows that the first step computes an annihilating polynomial for α modulo h with probability at least $1 - 6d^2/\text{card}(S)$. It also shows that each call to Algorithm MODULARCOMPOSITIONBASECASE succeeds with at least the same probability. Altogether, the probability of success of the whole algorithm is thus at least $1 - 6(\ell + 1)d^2/\text{card}(S)$.

By Corollary 8.2, the first step uses $\tilde{O}(d^\kappa)$ operations in \mathbb{K} . Since $\deg(\mu)$ is in $O(d)$, computing $g \text{ rem } \mu^\ell$ takes $\tilde{O}(\deg(g) + d\ell)$ operations in \mathbb{K} , and Lemma 9.4 shows that deducing \tilde{G} takes a further $\tilde{O}(d\ell)$ cost. Finally, by Proposition 8.1, each pass in the loop at Step 4 takes $\tilde{O}(d^\kappa)$ operations, so that the overall runtime is $\tilde{O}(\deg(g) + d^\kappa \ell)$. \square

9.3.3 *General Tangling and Untangling.* In fields of positive characteristic, the isomorphism of Lemma 9.3 and the complexity of its realization generalize as follows.

PROPOSITION 9.6. Let $f = h(x^{p^e})^\ell$ be of degree n , with h of degree d in $\mathbb{K}[x]$, and \mathbb{K} of characteristic p ($e = 0$ if $p = 0$). There exists a \mathbb{K} -algebra homomorphism

$$\begin{aligned} \Psi_{h, \ell} : \mathbb{K}[x]/\langle f \rangle &\rightarrow \mathbb{K}[\theta, z]/\langle h(\theta), (z^{p^e} - \theta)^\ell \rangle \\ x &\mapsto z. \end{aligned}$$

If moreover h is separable then $\Psi_{h, \ell}$ is an isomorphism. Applying $\Psi_{h, \ell}$ or its inverse when the latter is defined takes quasi-linear time $\tilde{O}(n) = \tilde{O}(d\ell p^e)$ over \mathbb{K} .

PROOF. Write $\mathbb{A} = \mathbb{K}[x]/\langle f \rangle$ and $\mathbb{B} = \mathbb{K}[\theta, z]/\langle h(\theta), (z^{p^e} - \theta)^\ell \rangle$. When h is separable, we prove that the minimal polynomial of z in the \mathbb{K} -algebra \mathbb{B} is f . This implies that \mathbb{A} is \mathbb{K} -isomorphic (as a \mathbb{K} -algebra) to the subalgebra of \mathbb{B} generated by z . Since \mathbb{B} has \mathbb{K} -dimension $n = \deg(f)$, this subalgebra is \mathbb{B} itself, and the first claim will follow.

To determine the minimal polynomial of z , we can work in $\bar{\mathbb{B}} = \bar{\mathbb{K}}[\theta, z]/\langle h(\theta), (z^{p^e} - \theta)^\ell \rangle$, where $\bar{\mathbb{K}}$ is an algebraic closure of \mathbb{K} . If we let ξ_1, \dots, ξ_d be the roots of h in $\bar{\mathbb{K}}$ (which are pairwise

distinct), then $\overline{\mathbb{B}}$ is isomorphic, as a $\overline{\mathbb{K}}$ -algebra, to the product

$$\overline{\mathbb{K}}[\theta, z]/\langle \theta - \xi_1, (z^{p^e} - \xi_1)^\ell \rangle \times \cdots \times \overline{\mathbb{K}}[\theta, z]/\langle \theta - \xi_d, (z^{p^e} - \xi_d)^\ell \rangle.$$

The minimal polynomial of z in the i th factor above is $\mu_i = (x^{p^e} - \xi_i)^\ell$ for $1 \leq i \leq d$. These polynomials are pairwise coprime: since $t \mapsto t^{p^e}$ is a bijection in $\overline{\mathbb{K}}$, μ_i has a unique root in $\overline{\mathbb{K}}$, which is the p^e -th root of ξ_i , and these roots are pairwise distinct, since the ξ_i 's are. As a result, the minimal polynomial of z in $\overline{\mathbb{B}}$, or equivalently in \mathbb{B} , is the product $\mu_1 \cdots \mu_d = f$.

For the second claim, we take a in $\mathbb{K}[x]$ of degree less than n , and write it as $a = \sum_{0 \leq i < p^e} a_i (x^{p^e}) x^i$, with all a_i 's of degree less than $n/p^e = d\ell$. Then,

$$\begin{aligned} \Psi_{h, \ell}(a \bmod f) &\equiv \sum_{0 \leq i < p^e} a_i (z^{p^e}) z^i \pmod{\langle h(\theta), (z^{p^e} - \theta)^\ell \rangle}, \\ &\equiv \sum_{0 \leq i < p^e} \tilde{A}_i(\theta, z^{p^e}) z^i \pmod{\langle h(\theta), (z^{p^e} - \theta)^\ell \rangle}, \end{aligned} \quad (36)$$

where $\tilde{A}_i(\theta, z) = a_i(z) \bmod \langle h(\theta), (z - \theta)^\ell \rangle$ is in $\mathbb{K}[\theta, z]_{<(d, \ell)}$; these degree bounds show that the expression in Equation (36) is indeed reduced modulo $\langle f(\theta), (z^{p^e} - \theta)^\ell \rangle$. Each $\tilde{A}_i = \psi_{h, \ell}(a_i)$ can be computed in time $\tilde{O}(d\ell)$ by Lemma 9.4, so that one application of $\Psi_{h, \ell}$ takes $\tilde{O}(d\ell p^e) = \tilde{O}(n)$ operations in \mathbb{K} , as claimed.

Conversely, any element B in $\mathbb{K}[\theta, z]_{<(d, \ell p^e)}$ can be written as in Equation (36), for some \tilde{B}_i 's in $\mathbb{K}[\theta, z]_{<(d, \ell)}$. Applying $\psi_{h, \ell}^{-1}$ to each of them allows us to recover $b = \Psi_{h, \ell}^{-1}(B)$, by reversing the steps above. The cost analysis is similar to the one for $\Psi_{h, \ell}$. \square

We call **UNTANGLING-GENERAL**(h, e, ℓ, a) the algorithm outlined in this proof that applies $\Psi_{h, \ell}$ to (the class modulo f of) $a \in \mathbb{K}[x]_{<n}$, and returns the canonical lift of $\Psi_{h, \ell}(a \bmod f)$ to $\mathbb{K}[\theta, z]_{<(d, \ell p^e)}$; equivalently, $A(\theta, z) = a(z) \bmod \langle h(\theta), (z^{p^e} - \theta)^\ell \rangle$. For B in $\mathbb{K}[\theta, z]_{<(d, \ell p^e)}$, the inverse operation is written **TANGLING-GENERAL**(h, e, ℓ, B).

9.3.4 Main Reduction. A more general form of bivariate reduction is needed in Section 9.4. With h of degree d as before, given g in $\mathbb{K}[y]$ and now a bivariate A in $\mathbb{K}[\theta, z]_{<(d, \ell p^e)}$, the aim is to reduce the degree of g before performing the composition in $\mathbb{L}[z]$ modulo $(z^{p^e} - \bar{\theta})^\ell$ with $\mathbb{L} = \mathbb{K}[\theta]/\langle h \rangle$. Denoting by \bar{A} the projection of A in $\mathbb{L}[z]$, the idea is to compute $\bar{G} = g \bmod \chi_{\bar{A}}$ in $\mathbb{L}[z]$, where $\chi_{\bar{A}} \in \mathbb{L}[y]$ is the characteristic polynomial of $\bar{A} \in \mathbb{L}[z]$ in the extension $\mathbb{L} \rightarrow \mathbb{L}[z]/\langle (z^{p^e} - \bar{\theta})^\ell \rangle$. Thus, $\bar{G} \in \mathbb{L}[y]$ has degree less than ℓp^e ; its canonical lift $G \in \mathbb{K}[\theta, y]_{<(d, \ell p^e)}$ is the output.

The computation of $g \bmod \chi_{\bar{A}}$ is made easy by an explicit formula for the characteristic polynomial $\chi_{\bar{A}}$. In the following lemma, we let $\sigma : \mathbb{L} \rightarrow \mathbb{L}$ be the p^e -th-power operator; we write the image of $\Lambda \in \mathbb{L}$ as Λ^σ . This notation is extended to the coefficient-wise action on polynomial rings over \mathbb{L} .

LEMMA 9.7. *The characteristic polynomial of \bar{A} relative to the extension $\mathbb{L} \rightarrow \mathbb{L}[z]/\langle (z^{p^e} - \bar{\theta})^\ell \rangle$ is $\chi_{\bar{A}} = (y^{p^e} - \bar{\alpha})^\ell \in \mathbb{L}[y]$, where $\bar{\alpha} = \bar{A}^\sigma(\bar{\theta}) \in \mathbb{L}$.*

PROOF. The characteristic polynomial $\chi_{\bar{A}}$ can be computed relative to the extension $\mathbb{L}^* \rightarrow \mathbb{L}^*[z]/\langle (z^{p^e} - \bar{\theta})^\ell \rangle$, where we set $\mathbb{L}^* = \mathbb{L}[w]/\langle w^{p^e} - \bar{\theta} \rangle$. In $\mathbb{L}^*[z]$, we have the factorization

$$(z^{p^e} - \bar{\theta})^\ell = (z^{p^e} - w^{p^e})^\ell = (z - w)^{\ell p^e},$$

so the characteristic polynomial of \bar{A} in $\mathbb{L}^*[z]/\langle (z^{p^e} - \bar{\theta})^\ell \rangle$ is

$$(y - \bar{A}(w))^{\ell p^e} = (y^{p^e} - \bar{A}(w)^{p^e})^\ell = (y^{p^e} - \bar{A}^\sigma(\bar{\theta}))^\ell. \quad \square$$

The reduction of g by this characteristic polynomial is described in Algorithm **MAINREDUCTION**. First, the canonical lift $\alpha \in \mathbb{K}[\theta]_{<d}$ of $\bar{\alpha} \in \mathbb{L}$ from Lemma 9.7 is computed. Next, in

ALGORITHM 9.3 MAINREDUCTION(h, e, ℓ, A, g, r)

Input: h separable, monic, of degree d in $\mathbb{K}[x]$, e in \mathbb{N} , ℓ in $\mathbb{N}_{>0}$, A in $\mathbb{K}[\theta, z]_{<(d, \ell p^e)}$, g in $\mathbb{K}[y]$, r in $\mathbb{K}^{d+\lceil d^n \rceil}$

Output: $G \in \mathbb{K}[\theta, y]_{<(d, \ell p^e)}$ such that $G(\theta, A) \equiv g(A) \pmod{\langle h(\theta), (z^{p^e} - \theta)^\ell \rangle}$, or FAIL

- 1: Write $A = \sum_{0 \leq i < \ell p^e} A_i z^i$ $\triangleright A_i \in \mathbb{K}[\theta]_{<d}$
- 2: *Compute α s.t. the characteristic polynomial of \bar{A} is $(y^{p^e} - \alpha)^\ell$ (see Lemma 9.7)* $\triangleright \alpha \in \mathbb{K}[\theta]_{<d}$
 $\alpha \leftarrow \sum_{0 \leq i < \ell p^e} A_i p^e \theta^i$; $\alpha \leftarrow \alpha \bmod h$
- 3: Write $g = \sum_{0 \leq i < p^e} g_i (y^{p^e})^i$ $\triangleright \deg(g_i) \leq \deg(g)/p^e$
- 4: **for** $i = 0, \dots, p^e - 1$ **do**
 $G_i \leftarrow \text{BIVARIATEREDUCTION}(h, \ell, \alpha, g_i, r)$ $\triangleright G_i \in \mathbb{K}[\theta, y]_{<(d, \ell)}$
- 5: $G \leftarrow \sum_{0 \leq i < p^e} G_i (\theta, y^{p^e})^i$ $\triangleright G \in \mathbb{K}[\theta, y]_{<(d, \ell p^e)}$
- 6: **return** G

Step 3, the polynomial g is rewritten as a polynomial in y of degree less than p^e , with coefficients $g_i(y^{p^e})$. Each of these polynomials $g_i(y)$ can then be reduced modulo $\langle h, (y - \alpha)^\ell \rangle$ by Algorithm **BIVARIATEREDUCTION**, producing a polynomial $G_i(\theta, y) \equiv g_i(y) \pmod{\langle h, (y - \alpha)^\ell \rangle}$, whence $G_i(\bar{\theta}, y^{p^e}) \equiv g_i(y^{p^e}) \pmod{\chi_{\bar{A}}}$. Recombining these coefficients yields $G(\theta, y)$ such that $G(\bar{\theta}, y) \equiv g(y) \pmod{\chi_{\bar{A}}}$. Finally, since $\chi_{\bar{A}}(\bar{A}) \equiv 0$ in $\mathbb{L}[z]/\langle (z^{p^e} - \theta)^\ell \rangle$, it follows that $G(\theta, A) \equiv g(A) \pmod{\langle h(\theta), (z^{p^e} - \theta)^\ell \rangle}$.

PROPOSITION 9.8. *Given h separable, monic, of degree d in $\mathbb{K}[x]$, e in \mathbb{N} , ℓ in $\mathbb{N}_{>0}$, A in $\mathbb{K}[\theta, z]_{<(d, \ell p^e)}$, g in $\mathbb{K}[y]$, and r in $\mathbb{K}^{d+\lceil d^n \rceil}$, Algorithm **MAINREDUCTION** uses $\tilde{O}(\deg(g) + n^\kappa)$ operations in \mathbb{K} , with $n = d\ell p^e$ and $\kappa < 1.43$ as in Equation (1). It returns $G \in \mathbb{K}[\theta, y]_{<(d, \ell p^e)}$ such that $G(\theta, A) \equiv g(A) \pmod{\langle h(\theta), (z^{p^e} - \theta)^\ell \rangle}$, or FAIL.*

If the entries of r are chosen uniformly and independently from a finite subset S of \mathbb{K} , then the algorithm returns G with probability at least $1 - 6(\ell + 1)d^2p^e/\text{card}(S)$.

PROOF. The correction of the algorithm when it does not return FAIL follows from the discussion above.

Working coefficient-wise, since $e = O(\log(p^e))$ the computation of α at Step 2 takes $\tilde{O}(\ell p^e)$ operations on polynomials modulo h of degree d , so $\tilde{O}(n)$ operations in \mathbb{K} ; reducing it modulo h has the same complexity bound. The cost is thus governed by the loop, which uses $\tilde{O}(\deg(g) + d^\kappa \ell p^e) = \tilde{O}(\deg(g) + (n/d)d^\kappa)$ operations by Lemma 9.5. The latter also allows us to quantify the probability of success: each of the p^e calls to Algorithm **BIVARIATEREDUCTION** succeeds with probability at least $1 - 6(\ell + 1)d^2/\text{card}(S)$. \square

9.4 Composition Modulo Powers

We now consider $f = h(x^{p^e})^\ell$, with h separable of degree d and integers e, ℓ , with ℓ positive and not divisible by p (and $e = 0$ if $p = 0$); the degree of f is $n = d\ell p^e$. Algorithm **MODULARCOMPOSITIONMODULOPOWER** computes $g(a) \bmod f$, extending to $e \neq 0$ the approach of van der Hoeven and Lecerf [34] outlined in Section 9.3.2.

We first compute $A(\theta, z) = a(z) \bmod \langle h(\theta), (z^{p^e} - \theta)^\ell \rangle$; this is done using the general untangling operation of Section 9.3.3. The reduction of the degree of g is done by Algorithm **MAINREDUCTION**, giving G in $\mathbb{K}[\theta, y]_{<(d, \ell p^e)}$, such that $G(\theta, A) \equiv g(A) \pmod{\langle h(\theta), (z^{p^e} - \theta)^\ell \rangle}$; the construction of A then implies $G(\theta, A) \equiv g(a(z)) \pmod{\langle h(\theta), (z^{p^e} - \theta)^\ell \rangle}$. The quantity $B = G(\theta, A) \bmod \langle h(\theta), (z^{p^e} - \theta)^\ell \rangle$ is obtained by Algorithm **COMPOSITIONMODULOINSEPARABLE**.

ALGORITHM 9.4 MODULARCOMPOSITIONMODULOPOWER(h, e, ℓ, a, g, r)

Input: h separable, monic, of degree d in $\mathbb{K}[x]$, e in \mathbb{N} , ℓ in $\mathbb{N}_{>0}$, such that $f = h(x^{p^e})^\ell$ has degree $n = d\ell p^e$, a in $\mathbb{K}[x]_{<n}$, g in $\mathbb{K}[y]$, r in $\mathbb{K}^{\rho+\lceil\rho^\eta\rceil}$ where $\rho = \max(d, n/d)$

Output: $b = g(a) \bmod f$ or FAIL

- 1: \triangleright Conversion of $a \in \mathbb{K}[x]$ to a bivariate polynomial (Proposition 9.6)
 $A \leftarrow \text{UNTANGLING-GENERAL}(h, e, \ell, a)$ $\triangleright A \in \mathbb{K}[\theta, z]_{<(d, \ell p^e)}$
- 2: \triangleright Reduction of g modulo the characteristic polynomial of \bar{A} (Proposition 9.8)
 $G \leftarrow \text{MAINREDUCTION}(h, e, \ell, A, g, (r_k)_{0 \leq k < d + \lceil d^\eta \rceil})$ $\triangleright G \in \mathbb{K}[\theta, y]_{<(d, \ell p^e)}$
if $G = \text{FAIL}$ **then return** FAIL
- 3: \triangleright Modular composition, $B = G(\theta, A) \bmod \langle h(\theta), (z^{p^e} - \theta)^\ell \rangle \in \mathbb{K}[\theta, z]_{<(d, \ell p^e)}$ or FAIL
 $B \leftarrow \text{COMPOSITIONMODULOUNSEPARABLE-PRODUCTOFFIELDS}(h, \theta, e, \ell, A, G, (r_k)_{0 \leq k < \frac{n}{d} + \lceil \frac{n}{d} \rceil^\eta})$
if $B = \text{FAIL}$ **then return** FAIL
- 4: \triangleright Recovery of b over \mathbb{K} (Proposition 9.6)
 $b \leftarrow \text{TANGLING-GENERAL}(h, e, \ell, B)$
- 5: **return** b

PRODUCTOFFIELDS of Section 9.2.2. We finally apply the general tangling procedure of Section 9.3.3 to B ; since tangling is a \mathbb{K} -algebra isomorphism, the outcome is $b = g(a) \bmod h(x^{p^e})^\ell$.

PROPOSITION 9.9. *For a field \mathbb{K} of characteristic p , given h separable, monic and of degree d in $\mathbb{K}[x]$, integers e in \mathbb{N} and ℓ in $\mathbb{N}_{>0}$, a in $\mathbb{K}[x]_{<n}$, g in $\mathbb{K}[y]$, r in $\mathbb{K}^{\rho+\lceil\rho^\eta\rceil}$, with $n = d\ell p^e$, $\rho = \max(d, \ell p^e)$ and η from Equation (3), Algorithm **MODULARCOMPOSITIONMODULOPOWER** uses $\tilde{O}(\deg(g) + n^\kappa)$ operations in \mathbb{K} , with $\kappa < 1.43$ as in Equation (1), and returns $g(a) \bmod h(x^{p^e})^\ell$ or FAIL.*

If the entries of r are chosen uniformly and independently from a finite subset S of \mathbb{K} , then the algorithm returns $g(a) \bmod h(x^{p^e})^\ell$ with probability at least $1 - (2n^4 + 12n^2)/\text{card}(S)$.

PROOF. That the output of the algorithm is $g(a) \bmod h(x^{p^e})^\ell$ or FAIL follows from the previous discussion. By Proposition 9.6, with $n = d\ell p^e$, the first and last step both take $\tilde{O}(n)$ operations in \mathbb{K} . Proposition 9.8 shows that Step 2 takes $\tilde{O}(\deg(g) + n^\kappa)$ operations in \mathbb{K} . Finally, Proposition 9.2 shows that Step 3 takes $\tilde{O}(d(\ell p^e)^\kappa) = \tilde{O}(d(n/d)^\kappa)$ operations in \mathbb{K} , so the runtime estimate is proved.

The steps that may output FAIL are the computation of G at Step 2 and that of B at Step 3. By Proposition 9.8, the former happens with probability at most $6(\ell+1)d^2p^e/\text{card}(S) \leq 12n^2/\text{card}(S)$; by Proposition 9.2, the latter happens with probability at most $2d(\ell p^e)^4/\text{card}(S) \leq 2n^4/\text{card}(S)$. \square

9.5 Main Algorithm and its Analysis

We can now give Algorithm **MODULARCOMPOSITION** performing modular composition with general polynomials, and prove Theorem 1.1.

The separable decomposition $f_1 \cdots f_s$ of f allows us to reduce the problem to compositions modulo the f_i 's, which are powers of polynomials as in Section 9.4. The polynomials a and g are first reduced so that the compositions modulo the f_i 's are called with inputs of appropriate degrees, then the result $b = g(a) \bmod f$ is recovered using Chinese remaindering. The number of random elements in \mathbb{K} we use is an *a priori* bound that can be refined if the separable decomposition of f is known.

PROOF OF THEOREM 1.1. First we prove correctness. Suppose that none of the subroutines returns FAIL; we show that the output is $g(a) \bmod f$.

Using the same notation for p^e th powering as in Lemma 9.7, at the i th pass in the loop at Step 4, the polynomial μ_i satisfies $\mu_i(\alpha_i) \equiv 0 \pmod{h_i}$, with $\alpha_i = a_i^\sigma$ (that is, the coefficients

ALGORITHM 9.5 MODULARCOMPOSITION(f, a, g, r)

Input: f of degree n in $\mathbb{K}[x]$, a in $\mathbb{K}[x]_{<n}$, g in $\mathbb{K}[y]_{<n}$, $r \in \mathbb{K}^{n+\lceil n^\eta \rceil}$

Output: $b = g(a) \bmod f$ or FAIL

```

1: ▷ Decomposition of  $f$  [57, Algorithm 3]
    $(h_1, e_1, \ell_1), \dots, (h_s, e_s, \ell_s) \leftarrow \text{SEPARABLEDECOMPOSITION}(f)$       ▷  $h_i$  monic of degree  $d_i$  in  $\mathbb{K}[x]$ 
2:  $(f_1, \dots, f_s) \leftarrow (h_1(x^{p^{e_1}})^{\ell_1}, \dots, h_s(x^{p^{e_s}})^{\ell_s})$       ▷  $f_i$  of degree  $n_i = d_i \ell_i p^{e_i}$  in  $\mathbb{K}[x]$ 
3: ▷ Degree reduction,  $\deg(a_i) < n_i$ 
    $(a_1, \dots, a_s) \leftarrow (a \bmod f_1, \dots, a \bmod f_s)$ 
4: ▷ Annihilating polynomials of the  $a_i$  modulo  $f_i$ 
   for  $i = 1, \dots, s$  do
     Write  $a_i = \sum_{0 \leq k < n_i} a_{i,k} x^k$ 
      $\alpha_i \leftarrow \sum_{0 \leq k < n_i} a_{i,k} p^{e_i} x^k$ ;  $\alpha_i \leftarrow \alpha_i \bmod h_i$ 
      $\mu_i \leftarrow \text{ANNIHILATINGPOLYNOMIAL}(h_i, \alpha_i, (r_k)_{0 \leq k < d_i + \lceil d_i^\eta \rceil})$  ▷  $\mu_i(\alpha_i) \equiv 0 \pmod{h_i}$ ,  $\deg(\mu_i) \leq 4d_i$ 
     if  $\mu_i = \text{FAIL}$  then return FAIL
      $\chi_i \leftarrow \mu_i(y^{p^{e_i}})^{\ell_i}$       ▷  $\chi_i(a_i) \equiv 0 \pmod{f_i}$ ,  $\deg(\chi_i) \leq 4n_i$ 
5: ▷ Degree reduction,  $\deg(g_i) < 4n_i$ 
    $(g_1, \dots, g_s) \leftarrow (g \bmod \chi_1, \dots, g \bmod \chi_s)$ 
6: ▷ Modular compositions, either  $b_i \equiv g(a) \bmod f_i$  or FAIL
   for  $i = 1, \dots, s$  do
      $\rho_i \leftarrow \max(d_i, n_i/d_i)$ 
      $b_i \leftarrow \text{MODULARCOMPOSITIONMODULOPOWER}(h_i, e_i, \ell_i, a_i, g_i, (r_k)_{0 \leq k < \rho_i + \lceil \rho_i^\eta \rceil})$ 
     if  $b_i = \text{FAIL}$  then return FAIL
7: return CHINESEREMAINDER(( $b_1, \dots, b_s$ ), ( $f_1, \dots, f_s$ ))

```

of α_i are the p^{e_i} th powers of those of a_i). Raising this equality to the power ℓ_i gives $\mu_i^{\ell_i}(\alpha_i) \equiv 0 \pmod{h_i^{\ell_i}}$. Evaluation at $y^{p^{e_i}}$ using the facts that $\alpha_i(y^{p^{e_i}}) = a_i p^{e_i}$ and $\chi_i = \mu_i(y^{p^{e_i}})^{\ell_i}$ finally gives $\chi_i(a_i) \equiv 0 \pmod{f_i}$. The degree bound $\deg(\mu_i) \leq 4d_i$ follows from the specifications of Algorithm **ANNIHILATINGPOLYNOMIAL**, and the degree bound for χ_i follows.

In the second for loop at Step 6, b_i satisfies $b_i \equiv g_i(a_i) \bmod h_i(x^{p^{e_i}})^{\ell_i} \equiv g_i(a_i) \bmod f_i$. Since $g_i = g \bmod \chi_i$, and χ_i cancels a_i modulo f_i , b_i is also equal to $g(a_i) \bmod f_i$, and thus to $g(a) \bmod f_i$. It follows that the return value, obtained by Chinese remaindering, is indeed $g(a) \bmod f$.

Next, we bound the overall cost. The call to **SEPARABLEDECOMPOSITION**(f) takes $\tilde{O}(n)$ operations in \mathbb{K} [57, Proposition 5]. Using repeated squaring, the polynomials f_1, \dots, f_s can be computed in quasi-linear time as well, and the same holds for the remainders a_1, \dots, a_s .

Consider a fixed index i in the loop at Step 4, and denote $d_i \ell_i p^{e_i}$ by n_i . Working coefficient-wise, computing $\alpha_i = a_i^\sigma$ takes $\tilde{O}(n_i)$ operations since $e_i = O(\log(n_i))$, and reducing it modulo h_i has the same complexity bound. By Proposition 8.1, Algorithm **ANNIHILATINGPOLYNOMIAL** uses $\tilde{O}(d_i^\kappa)$ operations in \mathbb{K} . If it does not fail, χ_i is then deduced in $\tilde{O}(n_i)$ operations again, hence the cost of the loop is $\tilde{O}(n^\kappa)$.

When Step 5 is reached, since all χ_i 's have respective degrees at most $4n_i$, fast multiple remaindering gives the polynomials g_i in $\tilde{O}(n)$ operations, with $\deg(g_i) < 4n_i$. Then, by Proposition 9.9, each call to Algorithm **MODULARCOMPOSITIONMODULOPOWER** uses $\tilde{O}(n_i^\kappa)$ operations in \mathbb{K} , so their total cost is $\tilde{O}(n^\kappa)$ again. Finally, the cost of the last step (if reached) is $\tilde{O}(n)$. Altogether, the cost is $\tilde{O}(n^\kappa)$, as claimed.

It remains to discuss the probability of failure. By Proposition 8.3, the i th call to Algorithm **ANNIHILATINGPOLYNOMIAL** fails with probability at most $6d_i^2/\text{card}(S)$; hence, the probability

that we successfully exit the first for loop is at least $1 - 6n^2/\text{card}(S)$. Then, by Proposition 9.9, the i th call to Algorithm **MODULARCOMPOSITIONMODULOPOWER** fails with probability at most $(2n_i^4 + 12n_i^2)/\text{card}(S)$, so the probability that we successfully exit the second for loop is at least $1 - (2n^4 + 12n^2)/\text{card}(S)$. Altogether this gives a failure probability of at most $(2n^4 + 18n^2)/\text{card}(S)$. \square

10 APPLICATIONS

We now list several variants of the modular composition problem and related ones and sketch how the algorithms presented above can improve the best known complexity.

10.1 Annihilating Polynomials

10.1.1 Annihilating Polynomial. A by-product of Algorithm **MODULARCOMPOSITION** is a Las Vegas algorithm that takes $\tilde{O}(n^\kappa)$ (κ from Theorem 1.1) arithmetic operations for computing an annihilating polynomial for a of degree at most $4n$.

Indeed, with the notation of the algorithm, for all $1 \leq i \leq s$, since $\chi_i(a_i) \equiv 0 \pmod{f_i}$ we have $\chi_i(a) = r_i f_i$ for some $r_i \in \mathbb{K}[x]$. Hence $\prod_{i=1}^s \chi_i$ is an annihilating polynomial for a modulo $f = \prod_{i=1}^s f_i$, whose degree is at most $4 \sum_{i=1}^s n_i = 4n$.

10.1.2 Minimal and Characteristic Polynomial. In general, our knowledge of the minimal and characteristic polynomial depends on whether we have a certified basis of relations.

PROPOSITION 10.1. *Let $R \in \mathbb{K}[y]_{\leq 2d}^{m \times m}$ be the matrix produced by Algorithm **CANDIDATEBASIS**. If R is a basis of $\mathcal{M}_m^{(a,f)}$, then the first m invariant factors of $yI_n - M_a$, hence in particular the minimal polynomial $\mu_a \in \mathbb{K}[y]$ of a modulo f , can be computed in $\tilde{O}(m^\omega d)$ operations in \mathbb{K} . If furthermore **CERT** is returned (implying that R is a basis of $\mathcal{M}_m^{(a,f)}$), then the product of these invariant factors gives the characteristic polynomial $\chi_a \in \mathbb{K}[y]$ of a modulo f .*

PROOF. If R is a basis of \mathcal{M}_m , Proposition 4.1 shows that the Hermite normal form of R is a triangular basis of \mathcal{M}_m whose diagonal entries are the first invariant factors $\sigma_1, \dots, \sigma_m$ of $yI_n - M_a$; in particular $\mu_a = \sigma_1$. If **CERT** is returned, then R is a basis of \mathcal{M}_m and $v_m = n$ (Proposition 5.6). Hence $\deg \det(R) = n$ and all the invariant factors are known; the characteristic polynomial is their product. The Hermite normal form of R can be computed in $\tilde{O}(m^\omega d)$ operations [53, Theorem 1.2]. \square

One case of certification of the minimal polynomial is when **CERT** is returned by Algorithm **CANDIDATEBASIS**, which occurs in particular for any $f \in \mathbb{K}[x]$ with $f(0) \neq 0$ and a generic a in $\mathbb{K}[x]_{<n}$ (see Section 7.3.2). Using Proposition 5.6 and a shift as in Remark 3.8, this establishes the complexity bound $\tilde{O}(n^\kappa)$ for computing a basis of relations and the minimal polynomial in the case of a generic $a \in \mathbb{K}[x]_{<n}$.

Under the assumptions of Proposition 8.3 with the additional hypothesis $v_m^{(a,f)} = n$ for $m = \lceil n^\eta \rceil$, a call to Algorithm **CANDIDATEBASIS** instead of a call to Algorithm **MATRIXOFRELATIONS** in Algorithm **MODULARCOMPOSITIONBASECASE**, leads to a *certified* basis of relations of $\mathcal{M}_m^{(\alpha, \mu_Y)}$ with good probability (use Proposition 5.6 instead of Proposition 5.8 in the proof of Proposition 8.3). From Proposition 10.1, this also allows one to compute and certify the minimal and characteristic polynomials in time $\tilde{O}(n^\kappa)$ when f is separable and $v_{\lceil n^\eta \rceil}^{(a,f)} = n$.

The latter can be extended to the case f irreducible and separable since then the minimal polynomial μ_a must be irreducible as well, and therefore $yI_n - M_a$ has r nontrivial invariant factors all equal to μ_a . If for $m = \lceil n^\eta \rceil$ the minimal polynomial satisfies $\delta = \deg(\mu_a) \geq n/m$, then $r \leq m$ and $v_m^{(a,f)} = n$, hence the above certification when f is separable leads to the minimal polynomial. The

low degree case $\delta < n/m$ can be treated directly using Lemma 3.2, allowing to compute μ_a in time $\tilde{O}(n\delta^{(\omega_2/2)-1})$, which is $\tilde{O}(n^\kappa)$ since $\delta < \lceil n^{1-\eta} \rceil$.

However, a matrix R returned by Algorithm `CANDIDATEBASIS` might not be a basis of $\mathcal{M}_m^{(\alpha, \mu_Y)}$: without an efficient certification of this property, Proposition 10.1 only gives a minimal polynomial algorithm of the Monte Carlo kind. Proceeding as done above, with a call to Algorithm `CANDIDATEBASIS` instead of a call to Algorithm `MATRIXOFRRELATIONS` in Algorithm `MODULARCOMPOSITIONBASECASE`, a Monte Carlo minimal polynomial algorithm in $\tilde{O}(n^\kappa)$ can be derived under the assumptions of Proposition 8.3.

10.2 Power Series Reversion and Power Series Equations

In this subsection, the characteristic of \mathbb{K} is 0.

For a given $a \in \mathbb{K}[x]$ with $a(0) = 0$ and $a'(0) \neq 0$, power series reversion (or functional inversion) asks for a power series $g \in \mathbb{K}[[x]]$ such that

$$a(g) = g(a) \equiv x \pmod{x^n}.$$

By Newton's iteration, a composition algorithm in $\tilde{O}(n^c)$ operations for some $c > 1$ induces a reversion algorithm in $\tilde{O}(n^c)$ operations as well [17]. Thus, we get a Las Vegas algorithm for power series reversion in $\tilde{O}(n^\kappa)$ operations in \mathbb{K} . Note that the converse reduction, from reversion to composition, also holds in this situation [17].

The approach for reversion extends partially to the resolution of a class of power series equations. The aim is to solve an equation

$$g(x, y) = b \pmod{x^n} \tag{37}$$

for $y \in \mathbb{K}[[x]]_{<n}$, when $g \in \mathbb{K}[[x]][y]$ satisfies $g(0, 0) = b(0)$ and its partial derivative with respect to y is not 0 at $(0, 0)$.

By Proposition 8.7, Algorithm `COMPOSITIONMODULOINSEPARABLE` computes a composition $g(x, a)$ in $\tilde{O}(n^\kappa)$ operations for g in $\mathbb{K}[x, y]_{<(m, n)}$ with $m = O(n^\eta)$ and η from Equation (3). Together with Newton's iteration, this gives a Las Vegas algorithm solving Equation (37) in $\tilde{O}(n^\kappa)$ operations for $g \in \mathbb{K}[x, y]_{<(n^\kappa, n)}$. Reversion is the special case with $b = x$ and $\deg_x(g) = 0$.

Note. It is known that the complexity of composition of power series (in terms of nonscalar operations) is essentially that of computing the coefficient of x^{n-1} of $g(a)$ [68]. By contrast, computing the coefficient of x^{n-1} in the reverse of a costs only $\tilde{O}(n)$ arithmetic operations [17].

10.3 Bivariate Composition

In this subsection, the characteristic of \mathbb{K} is 0.

Brent and Kung gave an algorithm that computes

$$g(a, b) \pmod{x^n}$$

for $g \in \mathbb{K}[x, y]_{<(n, n)}$ and truncated power series $a, b \in \mathbb{K}[[x]]$ in only $\tilde{O}(n^2)$ operations [16]. This is quasi-optimal, since the number of coefficients of g is $\Theta(n^2)$ in general. In the simple situation where $a(0) = 0$ and $a'(0) = 1$, the algorithm is as follows:

- (1) by power series reversion, compute $s(x)$ such that $a(s) = s(a) \equiv x \pmod{x^n}$;
- (2) by univariate composition, compute $c = b(s) \pmod{x^n}$;
- (3) by uni-bivariate composition, compute $d = g(x, c) \pmod{x^n}$;
- (4) by univariate composition, compute $d(a) \pmod{x^n}$.

The complexity is dominated by the uni-bivariate composition in Step (3), which can be performed by Horner evaluation in $\tilde{O}(n^2)$ operations.

We obtain a Las Vegas algorithm with a complexity reduced to $\tilde{O}(n^\kappa)$ when $g \in \mathbb{K}[x, y]_{<(n^\kappa, n)}$, where the uni-bivariate composition is done in $\tilde{O}(n^\kappa)$ as discussed in the case of power series equations, and all the other steps are univariate compositions that are also performed in $\tilde{O}(n^\kappa)$ by our algorithm.

This method extends to the computation of

$$g(a, b) \text{ rem } f$$

with f of degree n in $\mathbb{K}[x]$, and a, b in $\mathbb{K}[x]_{<n}$. The algorithm becomes

- (1) compute an annihilating polynomial χ of a modulo f ;
- (2) by inverse modular composition, compute c such that $c(a) \equiv b \pmod{f}$;
- (3) by uni-bivariate composition, compute $d = g(x, c) \text{ rem } \chi$;
- (4) by univariate composition, compute $d(a) \text{ rem } f$.

At least for generic a , this is again a Las Vegas algorithm in $\tilde{O}(n^\kappa)$ operations when $g \in \mathbb{K}[x, y]_{<(n^\kappa, n)}$.

ACKNOWLEDGEMENTS

We thank an anonymous reviewer for an extremely detailed list of comments and very useful suggestions that helped us improve the quality of the manuscript.

REFERENCES

- [1] Simon Abelard, Alain Couvreur, and Grégoire Lecerf. 2020. Sub-quadratic time for Riemann-Roch spaces: Case of smooth divisors over nodal plane projective curves. In *Proc. ISSAC*. ACM Press, 14–21. DOI: <https://doi.org/10.1145/3373207.3404053>
- [2] Simon Abelard, Alain Couvreur, and Grégoire Lecerf. 2021. *Efficient Computation of Riemann-Roch Spaces for Plane Curves with Ordinary Singularities*. HAL Report hal-03110135. Retrieved from <https://hal.archives-ouvertes.fr/hal-03110135>
- [3] Josh Alman and Virginia Vassilevska Williams. 2021. A refined laser method and faster matrix multiplication. In *Proc. SODA*. SIAM, 522–539. DOI: <https://doi.org/10.1137/1.9781611976465.32>
- [4] Bernhard Beckermann and George Labahn. 1994. A uniform approach for the fast computation of matrix-type Padé approximants. *SIAM J. Matrix Analysis and Applications* 26, 3 (1994), 804–823. DOI: <https://doi.org/10.1137/S0895479892230031>
- [5] Bernhard Beckermann, George Labahn, and Gilles Villard. 1999. Shifted normal forms of polynomial matrices. In *Proc. ISSAC*. ACM Press, 189–196. DOI: <https://doi.org/10.1145/309831.309929>
- [6] Daniel J. Bernstein. 1998. Composing power series over a finite ring in essentially linear time. *J. Symb. Comput.* 26, 3 (1998), 339–341. DOI: <https://doi.org/10.1006/jsco.1998.0216>
- [7] Vishwas Bhargava, Sumanta Ghosh, Zeyu Guo, Mrinal Kumar, and Chris Umans. 2022. Fast multivariate multipoint evaluation over all finite fields. In *Proc. FOCS*. IEEE, 221–232. DOI: <https://doi.org/10.1109/FOCS54457.2022.00028>
- [8] Vishwas Bhargava, Sumanta Ghosh, Mrinal Kumar, and Chandra Kanta Mohapatra. 2022. Fast, algebraic multivariate multipoint evaluation in small characteristic and applications. In *Proc. STOC*. ACM Press, 403–415. DOI: <https://doi.org/10.1145/3519935.3519968>
- [9] Dario Bini and Victor Y. Pan. 1994. *Polynomial and Matrix Computations*. Birkhäuser. DOI: <https://doi.org/10.1007/978-1-4612-0265-3>
- [10] Alin Bostan, Frédéric Chyzak, Marc Giusti, Romain Lebreton, Grégoire Lecerf, Bruno Salvy, and Éric Schost. 2017. *Algorithmes Efficaces en Calcul Formel*. In French. Edited by the authors. Retrieved from <https://hal.archives-ouvertes.fr/AEFC/>
- [11] Alin Bostan, Frédéric Chyzak, François Ollivier, Bruno Salvy, Éric Schost, and Alexandre Sedoglavic. 2007. Fast computation of power series solutions of systems of differential equations. In *Proc. SODA*. SIAM, 1012–1021. Retrieved from <https://dl.acm.org/doi/10.5555/1283383.1283492>
- [12] Alin Bostan, Philippe Flajolet, Bruno Salvy, and Éric Schost. 2006. Fast computation of special resultants. *J. Symb. Comput.* 41, 1 (2006), 1–29. DOI: <https://doi.org/10.1016/j.jsc.2005.07.001>
- [13] Alin Bostan, Grégoire Lecerf, and Éric Schost. 2003. Tellegen’s principle into practice. In *Proc. ISSAC*. ACM Press, 37–44. DOI: <https://doi.org/10.1145/860854.860870>

- [14] Alin Bostan, Bruno Salvy, and Éric Schost. 2008. Power series composition and change of basis. In *Proc. ISSAC*. ACM Press, 269–276. DOI : <https://doi.org/10.1145/1390768.1390806>
- [15] Richard P. Brent, Fred G. Gustavson, and David Y. Y. Yun. 1980. Fast solution of toeplitz systems of equations and computation of padé approximants. *J. Algorithms* 1, 3 (1980), 259–295. DOI : [https://doi.org/10.1016/0196-6774\(80\)90013-9](https://doi.org/10.1016/0196-6774(80)90013-9)
- [16] Richard P. Brent and H. T. Kung. 1977. Fast algorithms for composition and reversion of multivariate power series. In *Proc. Conference on Theoretical Computer Science (Waterloo ON, August 15–17, 1977)*. University of Waterloo, 149–158.
- [17] Richard P. Brent and H. T. Kung. 1978. Fast algorithms for manipulating formal power series. *J. ACM* 25, 4 (1978), 581–595. DOI : <https://doi.org/10.1145/322092.322099>
- [18] Marshall W. Buck, Raymond A. Coley, and David P. Robbins. 1992. A generalized vandermonde determinant. *J. Algebraic Combin.* 1, 2 (1992), 105–109. DOI : <https://doi.org/10.1023/A:102246801917>
- [19] Peter Bürgisser, Michael Clausen, and Mohammad Amin Shokrollahi. 1997. *Algebraic Complexity Theory*. Grundlehren der mathematischen Wissenschaften, Vol. 315. Springer. DOI : <https://doi.org/10.1007/978-3-662-03338-8>
- [20] Don Coppersmith. 1994. Solving homogeneous linear equations over GF(2) via block wiedemann algorithm. *Math. Comput.* 62, 205 (1994), 333–350. DOI : <https://doi.org/10.2307/2153413>
- [21] Jean Della Dora, Claire Discrecenzo, and Dominique Duval. 1985. About a new method for computing in algebraic number fields. In *EUROCAL '85 (LNCS)*, Vol. 204. Springer, 289–290. DOI : https://doi.org/10.1007/3-540-15984-3_279
- [22] Ran Duan, Hongxun Wu, and Renfei Zhou. 2023. Faster matrix multiplication via asymmetric hashing. In *Proc. FOCS*. IEEE, 2129–2138. DOI : <https://doi.org/10.1109/FOCS57990.2023.00130>
- [23] David S. Dummit and Richard M. Foote. 2003. *Abstract Algebra*. John Wiley & Sons.
- [24] Wayne Eberly, Mark Giesbrecht, Pascal Giorgi, Arne Storjohann, and Gilles Villard. 2007. Faster inversion and other black box matrix computation using efficient block projections. In *Proc. ISSAC*. ACM Press, 143–150. DOI : <https://doi.org/10.1145/1277548.1277569>
- [25] Mariano Gasca and José J. Martínez. 1987. On the computation of multivariate confluent Vandermonde determinants and its applications. In *Proc. Mathematics of Surfaces II*. Vol. 11. Oxford Univ. Press, 101–114.
- [26] Joachim von zur Gathen and Jürgen Gerhard. 1999. *Modern Computer Algebra*. Third edition 2013. Cambridge University Press. DOI : <https://doi.org/10.1017/CBO9781139856065>
- [27] Joachim von zur Gathen and Victor Shoup. 1992. Computing frobenius maps and factoring polynomials. *Comput. Complex.* 2 (1992), 187–224. DOI : <https://doi.org/10.1007/BF01272074>
- [28] Jürgen Gerhard. 2004. *Modular Algorithms in Symbolic Summation and Symbolic Integration*. Springer. DOI : <https://doi.org/10.1007/b104035>
- [29] Patrizia Gianni and Barry Trager. 1996. Square-free algorithms in positive characteristic. *Appl. Algebra Eng. Commun. Comput.* 7, 1 (1996), 1–14. DOI : <https://doi.org/10.1007/BF01613611>
- [30] Mark Giesbrecht, Armin Jamshidpey, and Éric Schost. 2021. Subquadratic-time algorithms for normal bases. *Comput. Complex.* 35, 1 (2021). DOI : <https://doi.org/10.1007/s00037-020-00204-9>
- [31] Pascal Giorgi, Claude-Pierre Jeannerod, and Gilles Villard. 2003. On the complexity of polynomial matrix computations. In *Proc. ISSAC*. ACM Press, 135–142. DOI : <https://doi.org/10.1145/860854.860889>
- [32] Joris van der Hoeven. 2002. Relax, but don't be too lazy. *J. Symb. Comput.* 34, 6 (2002), 479–542. DOI : <https://doi.org/10.1006/jsc.2002.0562>
- [33] Joris van der Hoeven and Robin Larrieu. 2019. Fast Gröbner basis computation and polynomial reduction for generic bivariate ideals. *Appl. Algebr. Eng. Comm.* 30, 6 (2019), 509–539. DOI : <https://doi.org/10.1007/s00200-019-00389-9>
- [34] Joris van der Hoeven and Grégoire Lecerf. 2017. Composition modulo powers of polynomials. In *Proc. ISSAC*. ACM Press, 445–452. DOI : <https://doi.org/10.1145/3087604.3087634>
- [35] Joris van der Hoeven and Grégoire Lecerf. 2018. Modular composition via factorization. *J. Complexity* 48 (2018), 36–68. DOI : <https://doi.org/10.1016/j.jco.2018.05.002>
- [36] Joris van der Hoeven and Grégoire Lecerf. 2019. Accelerated tower arithmetic. *J. Complexity* 55 (2019), 101402. DOI : <https://doi.org/10.1016/j.jco.2019.03.002>
- [37] Joris van der Hoeven and Grégoire Lecerf. 2020. Directed evaluation. *J. Complexity* 60 (2020), 101498. DOI : <https://doi.org/10.1016/j.jco.2020.101498>
- [38] Joris van der Hoeven and Grégoire Lecerf. 2021. Amortized bivariate multi-point evaluation. In *Proc. ISSAC*. ACM Press, 179–185. DOI : <https://doi.org/10.1145/3452143.3465531>
- [39] Joris van der Hoeven and Grégoire Lecerf. 2021. Fast amortized multi-point evaluation. *J. Complexity* 67 (2021), 101574. DOI : <https://doi.org/10.1016/j.jco.2021.101574>
- [40] Joris van der Hoeven and Grégoire Lecerf. 2021. Fast computation of generic bivariate resultants. *J. Complexity* 62 (2021), 101499. DOI : <https://doi.org/10.1016/j.jco.2020.101499>
- [41] Xiohan Huang and Victor Y. Pan. 1998. Fast rectangular matrix multiplication and applications. *J. Complexity* 14 (1998), 257–299. DOI : <https://doi.org/10.1006/jcom.1998.0476>

[42] Claude-Pierre Jeannerod, Vincent Neiger, and Gilles Villard. 2020. Fast computation of approximant bases in canonical form. *J. Symb. Comput.* 98 (2020), 192–224. DOI: <https://doi.org/10.1016/j.jsc.2019.07.011>

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

[44] Erich Kaltofen. 1992. On computing determinants without divisions. In *Proc. ISSAC*. ACM Press, 342–349. DOI: <https://doi.org/10.1145/143242.143350>

[45] Erich Kaltofen. 2000. Challenges of symbolic computation: My favorite open problems. *J. Symb. Comput.* 29, 6 (2000), 891–919. DOI: <https://doi.org/10.1006/jsco.2000.0370>

[46] Erich Kaltofen and Victor Y. Pan. 1991. Processor efficient parallel solution of linear systems over an abstract field. In *Proc. SPAA*. ACM, 180–191. DOI: <https://doi.org/10.1145/113379.113396>

[47] Erich Kaltofen and B. David Saunders. 1991. On wiedemann’s method of solving sparse linear systems. In *AAECC-9 (LNCS)*, Vol. 539. Springer Verlag, 29–38. DOI: https://doi.org/10.1007/3-540-54522-0_93

[48] Erich Kaltofen and Victor Shoup. 1997. Fast polynomial factorization over high algebraic extensions of finite fields. In *Proc. ISSAC*. ACM Press, 184–188. DOI: <https://doi.org/10.1145/258726.258777>

[49] Erich Kaltofen and Victor Shoup. 1998. Subquadratic-time factoring of polynomials over finite fields. *Math. Comp.* 67, 233 (1998), 1179–1197. DOI: <https://doi.org/10.1090/S0025-5718-98-00944-2>

[50] Erich Kaltofen and Gilles Villard. 2005. On the complexity of computing determinants. *Comput. Complex.* 13, 3 (2005), 91–130. DOI: <https://doi.org/10.1007/s00037-004-0185-3>

[51] Erich Kaltofen and George Yiuhasz. 2013. On the matrix berlekamp-massey algorithm. *ACM Trans. Algorithms* 9, 4 (2013), 33:1–33:24. DOI: <https://doi.org/10.1145/2500122>

[52] Kiran S. Kedlaya and Christopher Umans. 2011. Fast polynomial factorization and modular composition. *SIAM J. on Computing* 40, 6 (2011), 1767–1802. DOI: <https://doi.org/10.1137/08073408X>

[53] 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: <https://doi.org/10.1016/j.jco.2017.03.003>

[54] Daniel Lazard. 1985. Ideal bases and primary decomposition: case of two variables. *J. Symb. Comput.* 1, 3 (1985), 261–270. DOI: [https://doi.org/10.1016/S0747-7171\(85\)80035-3](https://doi.org/10.1016/S0747-7171(85)80035-3)

[55] François Le Gall. 2023. *Faster Rectangular Matrix Multiplication by Combination Loss Analysis*. arXiv:2307.06535. Retrieved from <https://arxiv.org/abs/cs/2307.06535>

[56] François Le Gall and Florent Urrutia. 2018. Improved rectangular matrix multiplication using powers of the Coppersmith-Winograd tensor. In *Proc. SODA*. SIAM, 1029–1046. DOI: <https://doi.org/10.1137/1.9781611975031.67>

[57] Grégoire Lecerf. 2008. Fast separable factorization and applications. *Appl. Algebra Eng. Commun. Comput.* 19, 2 (2008), 135–160. DOI: <https://doi.org/10.1007/s00200-008-0062-4>

[58] John D. Lipson. 1976. Newton’s method: A great algebraic algorithm. In *Proc. SYMSAC*. ACM Press, 260–270. DOI: <https://doi.org/10.1145/800205.806344>

[59] Thom Mulders and Arne Storjohann. 2003. On lattice reduction for polynomial matrices. *J. Symb. Comput.* 35, 4 (2003), 377–401. DOI: [https://doi.org/10.1016/S0747-7171\(02\)00139-6](https://doi.org/10.1016/S0747-7171(02)00139-6)

[60] Vincent Neiger. 2016. *Bases of relations in one or several variables: fast algorithms and applications*. Ph.D. Dissertation. École Normale Supérieure de Lyon. Retrieved from <https://tel.archives-ouvertes.fr/tel-01431413/>

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

[62] Morris Newman. 1972. *Integral Matrices*. Academic Press. First edition.

[63] Michael Nüsken and Martin Ziegler. 2004. Fast multipoint evaluation of bivariate polynomials. In *Algorithms—ESA 2004*. Springer, Berlin, 544–555. DOI: https://doi.org/10.1007/978-3-540-30140-0_49

[64] Michael Paterson and Larry J. Stockmeyer. 1973. On the number of nonscalar multiplications necessary to evaluate polynomials. *SIAM J. Comput.* 2, 1 (1973), 60–66. DOI: <https://doi.org/10.1137/0202007>

[65] Adrien Poteaux and É. Schost. 2013. Modular composition modulo triangular sets and applications. *Comput. Complex.* 22, 3 (2013), 463–516. DOI: <https://doi.org/10.1007/s00037-013-0063-y>

[66] Adrien Poteaux and Éric Schost. 2013. On the complexity of computing with zero-dimensional triangular sets. *J. Symb. Comput.* 50 (2013), 110–138. DOI: <https://doi.org/10.1016/j.jsc.2012.05.008>

[67] Daniel Reischert. 1997. Asymptotically fast computation of subresultants. In *Proc. ISSAC*. ACM Press, 233–240. DOI: <https://doi.org/10.1145/258726.258792>

[68] Peter Ritzmann. 1986. A fast numerical algorithm for the composition of power series with complex coefficients. *Theoret. Comput. Sci.* 44, 1 (1986), 1–16. DOI: [https://doi.org/10.1016/0304-3975\(86\)90107-6](https://doi.org/10.1016/0304-3975(86)90107-6)

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

[70] Victor Shoup. 1995. A new polynomial factorization algorithm and its implementation. *J. Symb. Comput.* 20, 4 (1995), 363–397. DOI: <https://doi.org/10.1006/jsco.1995.1055>

- [71] Victor Shoup. 1999. Efficient computation of minimal polynomials in algebraic extensions of finite fields. In *Proc. ISSAC*. ACM Press, 53–58. DOI : <https://doi.org/10.1145/309831.309859>
- [72] Christopher Umans. 2008. Fast polynomial factorization and modular composition in small characteristic. In *Proc. STOC*. ACM Press, 481–490. DOI : <https://doi.org/10.1145/1374376.1374445>
- [73] Marc Van Barel and Adhemar Bultheel. 1992. A general module theoretic framework for vector M-Padé and matrix rational interpolation. *Numer. Algorithms* 3 (1992), 451–462. DOI : <https://doi.org/10.1007/BF02141952>
- [74] Gilles Villard. 1997. *A study of Coppersmith's Block Wiedemann Algorithm using Matrix Polynomials*. RR 975 IM IMAG. Retrieved from <http://perso.ens-lyon.fr/gilles.villard/BIBLIOGRAPHIE/PDF/rr0497.pdf>
- [75] Gilles Villard. 2018. On computing the resultant of generic bivariate polynomials. In *Proc. ISSAC*. ACM Press, 391–398. DOI : <https://doi.org/10.1145/3208976.3209020>
- [76] Douglas H. Wiedemann. 1986. Solving sparse linear equations over finite fields. *IEEE Trans. Information Theory* 32, 1 (1986), 54–62. DOI : <https://doi.org/10.1109/TIT.1986.1057137>
- [77] 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 : <https://doi.org/10.1137/1.9781611977912.134>
- [78] William A. Wolovich. 1974. *Linear Multivariable Systems*. Applied Mathematical Sciences, Vol. 11. Springer-Verlag New-York. DOI : <https://doi.org/10.1007/978-1-4612-6392-0>
- [79] W. Zhou and G. Labahn. 2012. Efficient algorithms for order basis computation. *J. Symb. Comput.* 47, 7 (2012), 793–819. DOI : <https://doi.org/10.1016/j.jsc.2011.12.009>
- [80] Wei Zhou, George Labahn, and Arne Storjohann. 2012. Computing minimal nullspace bases. In *Proc. ISSAC*. ACM Press, 366–373. DOI : <https://doi.org/10.1145/2442829.2442881>

Received 25 October 2021; revised 28 August 2023; accepted 6 December 2023