Generic bivariate multi-point evaluation, interpolation and modular composition with precomputation

Vincent Neiger, Johan Rosenkilde, Grigory Solomatov

Generic Bivariate Multi-point Evaluation, Interpolation and Modular Composition with Precomputation

Vincent Neiger
Univ. Limoges, CNRS, XLIM, UMR 7252
F-87000 Limoges, France

Johan Rosenkilde
Technical University of Denmark
Kgs. Lyngby, Denmark

Grigory Solomatov
Technical University of Denmark
Kgs. Lyngby, Denmark

ABSTRACT

Suppose $\mathbb{K}$ is a large enough field and $\mathcal{P} \subset \mathbb{K}^2$ is a fixed, generic set of points which is available for precomputation. We introduce a technique called *reshaping* which allows us to design quasi-linear algorithms for both: computing the evaluations of an input polynomial $f \in \mathbb{K}[x, y]$ at all points of $\mathcal{P}$; and computing an interpolant $f \in \mathbb{K}[x, y]$ which takes prescribed values on $\mathcal{P}$ and satisfies an input y-degree bound. Our genericity assumption is explicit and we prove that it holds for most point sets over a large enough field. If $\mathcal{P}$ violates the assumption, our algorithms still work and the performance degrades smoothly according to a distance from being generic. To show that the reshaping technique may have an impact on other related problems, we apply it to modular composition: suppose generic polynomials $M \in \mathbb{K}[x]$ and $A \in \mathbb{K}[x]$ are available for precomputation, then given an input $f \in \mathbb{K}[x, y]$ we show how to compute $f(x, A(x)) \operatorname{rem} M(x)$ in quasi-linear time.

KEYWORDS

Multi-point evaluation, interpolation, modular composition, bivariate polynomials, precomputation.

ACM Reference Format:
Vincent Neiger, Johan Rosenkilde, and Grigory Solomatov. 2020. Generic Bivariate Multi-point Evaluation, Interpolation and Modular Composition with Precomputation. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Athens, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404032

1 INTRODUCTION

Outline. Let $\mathbb{K}$ be an effective field. We consider the three classical problems for bivariate polynomials $\mathbb{K}[x, y]$ mentioned in the title. We assume a model where part of the input is given early as *preinput* which is available for heavier computation, and the primary goal is to keep the complexity of the *online phase*, once the remaining part of the input is given, to a minimum.

Multi-point evaluation (MPE): with preinput a point set $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n \subseteq \mathbb{K}^2$ and input $f \in \mathbb{K}[x, y]$, compute $\big(f(\alpha_i, \beta_i)\big)_{i=1}^n$. We give two algorithms: the first requires pairwise distinct α_i's and has online complexity $\tilde{O}(\deg_x f \deg_y f + n)$ as long as $\mathcal{P}$ is *balanced*, a notion described below; the second accepts repeated x-coordinates with online complexity $\tilde{O}(\deg_x f(\deg_x f + \deg_y f) + n)$ as long as a certain "shearing" of $\mathcal{P}$ is balanced. "*soft-O*" ignores logarithmic terms: $O(f(n)(\log f(n))^c) \subset \tilde{O}(f(n))$ for any $c \in \mathbb{Z}_{\geq 0}$.

Interpolation: with preinput a point set $\mathcal{P}$ as before, and input values $\boldsymbol{\gamma} \in \mathbb{K}^n$, compute $f \in \mathbb{K}[x, y]$ such that $\big(f(\alpha_i, \beta_i)\big)_{i=1}^n = \boldsymbol{\gamma}$, satisfying some constraints on the monomial support. We give an algorithm which preinputs a degree bound d and outputs f such that $\deg_y f < d$ and $\deg_x f \in O(n/d)$. The online complexity is $\tilde{O}(n)$ if $\mathcal{P}$ and a shearing of $\mathcal{P}$ are both balanced; d should exceed the *x-valency of $\mathcal{P}$*, i.e. the maximal number of y-coordinates for any given x-coordinate.

Modular composition: with preinput $M, A \in \mathbb{K}[x]$, we input $f \in \mathbb{K}[x, y]$ and compute $f(x, A) \operatorname{rem} M$. Our algorithm has online complexity $\tilde{O}(\deg_x f \deg_y f + \deg A + \deg M)$, as long as the bivariate ideal $\langle M, y - A \rangle$ is balanced.

We prove that if $\mathcal{P} \subseteq \mathbb{K}^2$ is random of fixed cardinality n, and if $|\mathbb{K}| \gg n^2 \log(n)$ then $\mathcal{P}$ is balanced with high probability. Similarly, if M is square-free and A is uniformly random of degree less than $\deg M$, then $\langle M, y - A \rangle$ is balanced with high probability. Our proof techniques currently do not extend to proving that sheared point sets are balanced. A few trials we conducted suggest that this may often be the case if the x-valency of $\mathcal{P}$ is not too high. The cost of the second MPE algorithm is not symmetric in the x- and y-degree, so whenever $\deg_x f < \deg_y f$ one should consider transposing the input, i.e. evaluating $f(y, x)$ on $\{(\beta_i, \alpha_i)\}_{i=1}^n$. In this case, the balancedness assumption is on the transposed point set.

Our algorithms are deterministic, and once the preinput has been processed, the user knows whether it is balanced and hence whether the algorithms will perform well. Further, the performance of our algorithms deteriorates smoothly with how "unbalanced" the preinput is, in the sense of certain polynomials, which depend only on preinput, having sufficiently well behaved degrees. In a toolbox one might therefore apply our algorithms whenever the preinput turns out to be sufficiently balanced and reverting to other algorithms on very unbalanced preinput.

A typical use of precomputation is if we compute e.g. MPEs on the same point set for many different polynomials. This occurs in coding theory, where bivariate MPE corresponds to the encoding stage of certain families of codes such as some Reed-Muller codes [1, Chap. 5] and some algebraic-geometric codes [14]: here $\mathcal{P}$ is fixed and communication consists of a long series of bivariate MPEs on $\mathcal{P}$. In these applications, $\mathcal{P}$ is often not random but chosen carefully, and so our genericity assumptions might not apply.

Techniques. We introduce a tool we call *reshaping* for achieving the following: given an ideal $I \subseteq \mathbb{K}[x, y]$ and $f \in \mathbb{K}[x, y]$, compute $\hat{f} \in f + I$ with smaller y-degree. For instance in MPE, we let $\Gamma \subset \mathbb{K}[x, y]$ be the ideal of polynomials which vanish on all the points $\mathcal{P}$. Then all elements of $f + \Gamma$ have the same evaluations on $\mathcal{P}$, so we compute a $\hat{f} \in f + \Gamma$ of y-degree 0 (it exists if $\mathcal{P}$ has distinct x-coordinates), and then apply fast univariate MPE.

ISSAC '20, July 20–23, 2020, Athens, Greece

An obvious idea to accomplish this iteratively is to find some $g \in \Gamma$ of lower y-degree than f and whose leading y-term is 1, and then compute $\tilde{f} = f \operatorname{rem} g$. The problem is that the x-degree of $\tilde{f}$ may now be as large as $\deg_x f + (\deg_y f - \deg_y g)\deg_x g$. Our idea is to seek polynomials g that we call *reshapers*, which have the form

$$g = y^{2d/3} - \hat{g}\,,$$

where $\deg_y \hat{g} < d/3$ and $d = \deg_y f + 1$ (for simplicity, here 3 divides d). Writing $f = f_1 y^{2d/3} + f_0$ with $\deg_y f_0 < 2d/3$, then $\tilde{f} = f_1\hat{g} + f_0$ is easy to compute, has y-degree less than $2d/3$, and x-degree only $\deg_x f + \deg_x g$. Repeating such a reduction $O(\log(d))$ times with reshapers of progressively smaller y-degree, we eventually reach y-degree 0.

For efficiency, we therefore need the x-degrees of all these reshapers g to be small. For MPE, stating that $g \in \Gamma$ specifies n linear contraints on the coefficients of $\hat{g}$, so we look for g with about n monomials. Generically, since $\deg_y \hat{g} \approx d/3$, one may expect to find g with $\deg_x g \approx 3n/d$. Informally, $\mathcal{P}$ is *balanced* if all the reshapers needed in the above process satisfy this degree constraint.

Above, we assumed the point set has distinct x-coordinates. To handle repetitions, we shear the points by $(\alpha, \beta) \mapsto (\alpha + \theta\beta, \beta)$, where θ generates an extension field of $\mathbb{K}$ of degree 2. The resulting point set has distinct x-coordinates. This replaces $f(x, y)$ with $f(x - \theta y, y)$, and whenever $\deg_x f < \deg_y f$ we stay within quasi-linear complexity if the sheared point set is balanced.

Previous work. Quasi-linear complexity has been achieved for multivariate MPE and interpolation on special point sets and monomial support: Pan [18] gave an algorithm on grids, and van der Hoeven and Schost [26] (see also [5, Sec. 2]) generalised this to certain types of subsets of grids, constraining both the points and the monomial support. See [26] for references to earlier work on interpolation, not achieving quasi-linear complexity.

In classical univariate modular composition, we are given f, M, A in $\mathbb{K}[x]$ and seek $f(A) \operatorname{rem} M$. Brent and Kung's baby-step giant-step algorithm [2, 19] performs this operation in $\tilde{O}(n^{(\omega+1)/2})$, where ω is the matrix multiplication exponent with best known bound $\omega < 2.373$ [13]. Nüsken and Ziegler [17] extended this to a bivariate f, computing $f(x, A) \operatorname{rem} M$ in complexity $O(\deg_x f (\deg_y f)^{(\omega+1)/2})$, assuming that A and M have degree at most $\deg_x f \deg_y f$. They applied this to solve MPE in the same cost; in this paper, we use essentially the same link between these problems. To the best of our knowledge, this is currently the best known cost bound for these problems, in the algebraic complexity model.

In a breakthough, Kedlaya and Umans [11] achieved "almost linear" time for modular composition and MPE, for specific types of fields $\mathbb{K}$ and in the bit complexity model. For modular composition, the cost is $O(n^{1+\epsilon})$ bit operations for any $\epsilon > 0$, while for MPE it is $O((n + (\deg_x f)^2)^{1+\epsilon})$, assuming $\deg_y f < \deg_x f$ (the algorithm also supports multivariate MPE). Unfortunately, these algorithms have so far resisted attempts at a practical implementation [25].

Our quasi-linear complexities improve upon the above results (including Kedlaya and Umans' ones since quasi-linear compares favorably to almost linear); however we stress that none of the latter have the two constraints of our work: allowing precomputation, and genericity assumption. For modular composition, precomputation on M was suggested in [24] to leverage its factorisation structure. Except for slight benefits of precomputation in Brent and Kung's modular composition (used in the Flint and NTL libraries [8, 22]), we are unaware of previous work focusing on the use of precomputation for MPE, Interpolation, and Modular Composition.

Genericity has recently been used by Villard [27], who showed how to efficiently compute the resultant of two generic bivariate polynomials; a specific case computes, for given univariate M and A, the characteristic polynomial of A in $\mathbb{K}[x]/\langle M\rangle$, with direct links to the modular composition $f(A) \operatorname{rem} M$ [27, 28]. This led to an ongoing work on achieving exponent $(\omega+2)/3$ for modular composition [15]. In that line, the main benefit from genericity is that $\langle M, y - A\rangle$ admits bases formed by m polynomials of y-degree $< m$ and x-degree at most $\deg(M)/m$, for a given parameter $2 \le m \le \deg(M)$. Such a basis is represented as an $m \times m$ matrix over $\mathbb{K}[x]$ with all entries of degree at most $\deg(M)/m$, and one can then rely on fast univariate polynomial matrix algorithms. In this paper, genericity serves a purpose similar to that in [15, 27]: it ensures the existence of such bases for several parameters m, and also of the reshapers g mentioned above; besides we make use of these bases to precompute these reshapers. Whereas an important contribution of [27] is the efficient computation of such bases, here they are only used to find reshapers in the precomputation stage and the speed of computing them is not a main concern. Once the reshapers are known, our algorithms work without requiring any other genericity property.

Organisation. After some preliminaries in Section 2, we describe the reshaping strategy for an arbitrary ideal in Section 3. Then Sections 4 to 6 give algorithms for each of the three problems. We discuss precomputation in Section 7 and genericity in Section 8.

2 PRELIMINARIES

For complexity estimates, we use the algebraic RAM model and count arithmetic operations in $\mathbb{K}$. By $\mathsf{M}(n)$ we denote the cost of multiplying two univariate polynomials over $\mathbb{K}$ of degree at most n; one may take $\mathsf{M}(n) \in O(n \log n \log\log n) \subset \tilde{O}(n)$ [3]. Division with remainder in $\mathbb{K}[x]$ also costs $O(\mathsf{M}(n))$ [30, Thm. 9.6]. When degrees of a polynomial, say $f \in \mathbb{K}[x, y]$, appear in complexity estimates, we abuse notation and let $\deg_x f$ denote $\max(\deg_x f, 1)$.

It is well-known that univariate interpolation and multi-point evaluation can be done in quasi-linear time [30, Cor. 10.8 and 10.12]: given $f \in \mathbb{K}[x]$ and $\alpha_1, \ldots, \alpha_n \in \mathbb{K}$, we may compute $\big(f(\alpha_i)\big)_{i=1}^n$ in time $O(\mathsf{M}(\deg_x f + n)\log n) \subseteq \tilde{O}(\deg_x f + n)$; given $\alpha_1, \ldots, \alpha_n$ and $\beta_1, \ldots, \beta_n$ in $\mathbb{K}$ with the α_i's pairwise distinct, we may compute the unique corresponding interpolant in time $O(\mathsf{M}(n)\log n) \subseteq \tilde{O}(n)$. We will also use the fact that two bivariate $f, g \in \mathbb{K}[x, y]$ can be multiplied in time $O(\mathsf{M}(d_x d_y)) \subset \tilde{O}(d_x d_y)$, where $d_x = \max(\deg_x f, \deg_x g)$ and $d_y = \max(\deg_y f, \deg_y g)$ [30, Cor. 8.28].

For a bivariate polynomial $f = \sum_{i=0}^k f_i(x) y^i \in \mathbb{K}[x, y]$ such that $f_k \neq 0$, we define its y-leading coefficient as $\mathsf{LC}_y(f) = f_k \in \mathbb{K}[x]$.

For our genericity results, we will invoke the following staple:

Lemma 2.1 (DeMillo-Lipton-Schwartz-Zippel [7, 21, 31]). *Let $f \in \mathbb{K}[x_1, \ldots, x_n]$ be non-zero of total degree d, and $\mathcal{T} \subseteq \mathbb{K}$ be finite. For $\alpha_1, \ldots, \alpha_k \in \mathcal{T}$ chosen independently and uniformly at random, the probability that $f(\alpha_1, \ldots, \alpha_k) = 0$ is at most $d/|\mathcal{T}|$.*

For a point set $\mathcal{P} \subseteq \mathbb{K}^2$, the x-valency of $\mathcal{P}$, denoted by $v_x(\mathcal{P})$, is the largest number of y-coordinates for any given x-coordinate, i.e.

$$v_x(\mathcal{P}) = \max_{\alpha \in \mathbb{K}} |\{\beta \in \mathbb{K} \mid (\alpha, \beta) \in \mathcal{P}\}| .$$

When $v_x(\mathcal{P}) = 1$, the x-coordinates of $\mathcal{P}$ are pairwise distinct.

The vanishing ideal of $\mathcal{P}$ is the bivariate ideal

$$\Gamma(\mathcal{P}) = \{f \in \mathbb{K}[x, y] \mid f(\alpha, \beta) = 0 \text{ for all } (\alpha, \beta) \in \mathcal{P}\} ,$$

Hereafter, $\prec_{\text{lex}}$ stands for the lexicographic order on $\mathbb{K}[x, y]$ with $x \prec_{\text{lex}} y$, and $\mathsf{LT}_{\text{lex}}(f)$ is the $\prec_{\text{lex}}$-leading term of $f \in \mathbb{K}[x, y]$. The following is folklore and follows e.g. from [12] and [6, Thm. 3].

Lemma 2.2. *Let $\mathcal{P} \subset \mathbb{K}^2$ be a point set of cardinality n and let $G = \{g_1, \ldots, g_s\}$ be the reduced $\prec_{\text{lex}}$-Gröbner basis of $\Gamma(\mathcal{P})$, ordered by $\prec_{\text{lex}}$. Then $g_1 \in \mathbb{K}[x]$, and g_s is y-monic with $\deg_y g_s = v_x(\mathcal{P})$.*

3 RESHAPE

We first describe our algorithm Reshape which takes $f \in \mathbb{K}[x, y]$ and an ideal I and finds $\hat{f} \in f + I$ whose y-degree is below some target. This will pass through several intermediate elements of $f + I$ of progressively smaller y-degree. This sequence of y-degrees has the following form:

Definition 3.1. We say $\boldsymbol{\eta} = (\eta_i)_{i=0}^k \in \mathbb{Z}_{>0}^{k+1}$ is a (η_0, η_k)-*reshaping sequence* if $\eta_{i-1} > \eta_i \geq \lfloor \frac{2}{3}\eta_{i-1} \rfloor$ for $i = 1, \ldots, k$. For $I \subseteq \mathbb{K}[x, y]$ an ideal and $\boldsymbol{\eta} = (\eta_i)_{i=0}^k$ a reshaping sequence, we say $\boldsymbol{g} = (g_i)_{i=1}^k \in I^k$ is an $\boldsymbol{\eta}$-*reshaper for* I if $g_i = y^{\eta_i} + \hat{g}_i$ where $\deg_y \hat{g}_i \leq 2\eta_i - \eta_{i-1}$, for each $i = 1, \ldots, k$.

Our algorithms are faster with short reshaping sequences, so we should choose $\eta_i \approx \frac{2}{3}\eta_{i-1}$, and hence $2\eta_i - \eta_{i-1} \approx \frac{1}{3}\eta_i$. It is easy to see that for any $a, b \in \mathbb{Z}_{>0}$, there is an (a, b)-reshaping sequence of length less than $\log_{3/2}(a) + 2$. Observe that for any (a, b)-reshaping sequence we have $\eta_i \geq \frac{2}{3}(\eta_{i-1} - 1)$ for $i = 1, \ldots, k$ and therefore

$$2\eta_i - \eta_{i-1} \geq \frac{\eta_{i-1} - 4}{3} \geq \frac{\eta_i}{3} - 1 . \tag{1}$$

By considering the cases $\eta_i \geq 3$ and $\eta_i = 1, 2$, we get $2\eta_i - \eta_{i-1} \geq 0$.

Theorem 3.2. *Algorithm 1 is correct and has complexity*

$$\tilde{O}(\textstyle\sum_{i=i_0}^k \eta_i(\deg_x f + \sum_{j=i_0}^i \deg_x g_j)) \subseteq \tilde{O}(k \deg_y f \deg_x f + k \sum_{i=i_0}^k \eta_i \deg_x g_i) ,$$

for the smallest i_0 such that $\eta_{i_0} \leq \deg_y f$.

Proof. Let $\hat{f}_i, \hat{f}_{i,0}, \hat{f}_{i,1}$ be the values of $\hat{f}, \hat{f}_0, \hat{f}_1$ at the end of iteration i. First, the iterations for $i < i_0$ perform no operation and keep $\hat{f}_i = f$, since $\eta_i > \deg_y \hat{f}_{i-1}$ implies $\hat{f}_{i,1} = 0$ and $\hat{f}_i = \hat{f}_{i-1}$. In particular, if $\eta_i > \deg_y f$ for all i then the algorithm is correct and returns f without using any arithmetic operation. Now for $i \geq i_0$, observe that $\hat{f}_i = \hat{f}_{i,1}\hat{g}_i + \hat{f}_{i,0} = \hat{f}_{i-1} - \hat{f}_{i,1}g_i$; thus in the end $\hat{f} \in f + I$ since each g_i belongs to I. We show the following loop invariants, which imply the degree bounds on the output:

$$\deg_x \hat{f}_i \leq \deg_x f + \textstyle\sum_{j=i_0}^i \deg_x g_j, \text{ and } \deg_y \hat{f}_i < \eta_i .$$

Both are true for $i = i_0 - 1$ (just before the loop, if $i_0 = 1$). For the x-degree, $\hat{f}_i = \hat{f}_{i-1} - \hat{f}_{i,1}g_i$ yields $\deg_x \hat{f}_i \leq \deg_x \hat{f}_{i-1} + \deg_x g_i$, and the loop invariant follows. For the y-degree, by construction

Algorithm 1 Reshape$(f, \boldsymbol{\eta}, \boldsymbol{g})$

Input: A bivariate polynomial $f \in \mathbb{K}[x, y]$; a reshaping sequence $\boldsymbol{\eta} = (\eta_i)_{i=0}^k \in \mathbb{Z}_{>0}^{k+1}$ with $\deg_y f < \eta_0$; an $\boldsymbol{\eta}$-reshaper $\boldsymbol{g} = (g_i)_{i=1}^k \in I^k$ for some ideal $I \subseteq \mathbb{K}[x, y]$.

Output: a polynomial $\hat{f} \in f + I$ such that $\deg_y \hat{f} < \eta_k$ and $\deg_x \hat{f} \leq \deg_x f + \sum_{i=1}^k \deg_x g_i$.

1: $\hat{f} \leftarrow f$
2: **for** $i = 1, \ldots, k$ **do**
3: Write $g_i = y^{\eta_i} + \hat{g}_i$ where $\deg_y \hat{g}_i \leq 2\eta_i - \eta_{i-1}$
4: Write $\hat{f} = \hat{f}_1 y^{\eta_i} + \hat{f}_0$ where $\deg_y \hat{f}_0 < \eta_i$
5: $\hat{f} \leftarrow \hat{f}_1\hat{g}_i + \hat{f}_0$ ▷ *equivalent to* $\hat{f} \leftarrow \hat{f} - \hat{f}_1 g_i$
6: **return** $\hat{f}$

$\deg_y \hat{f}_{i,0} < \eta_i$ and $\deg_y \hat{f}_{i,1} \leq \deg_y \hat{f}_{i-1} - \eta_i$ hold; the assumption $\deg_y \hat{f}_{i-1} < \eta_{i-1}$ then gives $\deg_y \hat{f}_{i,1}\hat{g}_i < \eta_i$, hence $\deg_y \hat{f}_i < \eta_i$.

For complexity, the only costly step is at Line 5 and for iterations $i \geq i_0$. From the above bound $\deg_y \hat{f}_{i,1}\hat{g}_i < \eta_i$, multiplying $\hat{f}_{i,1}$ and $\hat{g}_i$ costs $O(\mathsf{M}((\deg_x \hat{f}_{i,1} + \deg_x \hat{g}_i)\eta_i))$. Since $\deg_x \hat{g}_i = \deg_x g_i$, since both $\hat{f}_{i,0}$ and $\hat{f}_{i,1}$ have x-degree at most $\deg_x \hat{f}_{i-1}$, and since $\deg_y \hat{f}_{i,0} < \eta_i$, the total cost of the ith iteration is in

$$\tilde{O}((\deg_x \hat{f}_{i-1} + \deg_x \hat{g}_i)\eta_i) \subseteq \tilde{O}((\deg_x f + \textstyle\sum_{j=i_0}^i \deg_x g_j)\eta_i).$$

Summing over all iterations, we get the first complexity bound in the theorem; the second one follows from it, using the fact that $\deg_y f \geq \eta_{i_0} > \eta_{i_0+1} > \ldots > \eta_k$ and $i_0 \geq 1$. □

We now define the balancedness of a point set. In Section 8 we prove that this notion captures the *expected* x-degree of reshapers.

Definition 3.3. Let $\mathcal{P} \subseteq \mathbb{K}^2$ be a point set of cardinality n, and let $\boldsymbol{\eta} = (\eta_i)_{i=0}^k$ be a reshaping sequence. Then $\mathcal{P}$ is $\boldsymbol{\eta}$-*balanced* if there exists an $\boldsymbol{\eta}$-reshaper $\boldsymbol{g} = (g_i)_{i=1}^k \in \mathbb{K}[x, y]^k$ for $\Gamma(\mathcal{P})$ such that $\deg_x g_i \leq \lfloor \frac{n}{2\eta_i - \eta_{i-1} + 1} \rfloor + 1$ for $i = 1, \ldots, k$.

The next bound is often used below for deriving complexity estimates; it follows directly from Eq. (1).

Lemma 3.4. *Let $\boldsymbol{\eta} = (\eta_i)_{i=0}^k$ be a reshaping sequence, $\mathcal{P} \subseteq \mathbb{K}^2$ be an $\boldsymbol{\eta}$-balanced point set of cardinality n, and $\boldsymbol{g} = (g_i)_{i=1}^k$ be an $\boldsymbol{\eta}$-reshaper for $\Gamma(\mathcal{P})$. Then $\sum_{i=i_0}^k \eta_i \deg_x g_i \leq (3n + \eta_{i_0})k$ for $1 \leq i_0 \leq k$.*

We conclude this section with two results about the existence of $\boldsymbol{\eta}$-reshapers for vanishing ideals of point sets.

Lemma 3.5. *Let $\mathcal{P} \subseteq \mathbb{K}^2$ be a point set and $\boldsymbol{\eta} = (\eta_i)_{i=0}^k$ a reshaping sequence. If $v_x(\mathcal{P}) \leq \min_{1 \leq i \leq k}(2\eta_i - \eta_{i-1} + 1)$, then there exists an $\boldsymbol{\eta}$-reshaper $\boldsymbol{g} \in \mathbb{K}[x, y]^k$ for $\Gamma(\mathcal{P})$.*

Proof. By Lemma 2.2, the reduced $\prec_{\text{lex}}$-Gröbner basis G of $\Gamma(\mathcal{P})$ contains a polynomial with $\prec_{\text{lex}}$-leading term $y^{v_x(\mathcal{P})}$. Thus $\deg_y y^\eta \operatorname{rem} G < v_x(\mathcal{P})$ for any η, and setting $g_i = y^{\eta_i} - (y^{\eta_i} \operatorname{rem} G)$ yields an $\boldsymbol{\eta}$-reshaper as long as $v_x(\mathcal{P}) \leq 2\eta_i - \eta_{i-1} + 1$ for all i. □

Corollary 3.6. *Let $\mathcal{P} \subseteq \mathbb{K}^2$ be a point set of cardinality n and $a, b \in \mathbb{Z}_{>0}$ with $n > a > b \geq v_x(\mathcal{P})$. Then there is an (a, b)-reshaping sequence $\boldsymbol{\eta}$ which satisfies the condition of Lemma 3.5 and has length $k \leq \log_{3/2}(a) + 1 \in O(\log(a))$.*

Proof. Let $v = v_x(\mathcal{P}) - 1$ and let $\boldsymbol{\eta}' = (\eta'_0, \dots, \eta'_k)$ be any $(a-v, b-v)$-reshaping sequence with $k \le \log_{3/2}(a-v)+1$. Now let $\boldsymbol{\eta} = (\eta_0, \dots, \eta_k)$ be defined by $\eta_i = \eta'_i + v$ for $i = 0, \dots, k$. Then, $\boldsymbol{\eta}$ is an (a,b)-reshaping sequence. Indeed, clearly the endpoints are correct and $\eta_{i-1} > \eta_i$ for $i = 1, \dots, k$; moreover,

$$\eta_i = \eta'_i + v \ge \lfloor \tfrac{2}{3}\eta'_{i-1} \rfloor + v = \lfloor \tfrac{2}{3}\eta_{i-1} + \tfrac{1}{3}v \rfloor \ge \lfloor \tfrac{2}{3}\eta_{i-1} \rfloor .$$

To conclude, we use $2\eta'_i - \eta'_{i-1} \ge 0$ as mentioned above to observe that $2\eta_i - \eta_{i-1} + 1 = 2\eta'_i - \eta'_{i-1} + v + 1 \ge v + 1 = v_x(\mathcal{P})$. □

4 MULTI-POINT EVALUATION

In this section we use reshaping for MPE with precomputation; i.e. given a point set $\mathcal{P} \subset \mathbb{K}^2$ upon which we are allowed to perform precomputation, and a polynomial $f \in \mathbb{K}[x,y]$ which is assumed to be received at online time, compute $f(P)$ for all $P \in \mathcal{P}$. Algorithm 2 deals with the case $v_x(\mathcal{P}) = 1$, which we reduce to an instance of univariate MPE using Reshape. The cost of Algorithm 2 follows directly from Theorem 3.2 and Lemma 3.4.

Algorithm 2 MPE-DistinctX$_{d,\boldsymbol{\eta},\mathcal{P}}(f)$

Preinput: $d \in \mathbb{Z}_{>0}$; a $(d,1)$-reshaping sequence $\boldsymbol{\eta}$; a point set $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n \subset \mathbb{K}^2$ with the α_i's pairwise distinct.
Precomputation:
a: $\boldsymbol{g} \leftarrow \boldsymbol{\eta}$-reshaper for $\Gamma(\mathcal{P})$
Input: $f \in \mathbb{K}[x,y]$ with $\deg_y f < d$.
Output: $\big(f(\alpha_1,\beta_1), \dots, f(\alpha_n,\beta_n)\big) \in \mathbb{K}^n$.
1: $\hat{f} \leftarrow$ Reshape$(f, \boldsymbol{\eta}, \boldsymbol{g}) \in \mathbb{K}[x]$
2: **return** $\big(\hat{f}(\alpha_1), \dots, \hat{f}(\alpha_n)\big) \in \mathbb{K}^n$ ▷ *univariate MPE*

Theorem 4.1. *Algorithm 2 is correct. If $\mathcal{P}$ is $\boldsymbol{\eta}$-balanced and $\boldsymbol{\eta}$ has length in $O(\log(n))$, the complexity is $\tilde{O}(\deg_x f \deg_y f + n)$.*

Algorithm 2 can easily be extended to the case where $v_x(\mathcal{P}) > 1$ by partitioning $\mathcal{P}$ into $v_x(\mathcal{P})$ many subsets, each having x-valency one. This approach also has quasi-linear complexity in the input size as long as $v_x(\mathcal{P}) \ll n$, or more precisely if $n v_x(\mathcal{P}) \in \tilde{O}(n)$.

When $v_x(\mathcal{P})$ is large, this strategy is costly, and we proceed instead by shearing the point set, as proposed by Nüsken and Ziegler [17], so that the resulting point set has distinct x-coordinates: by taking $\theta \in \mathbb{L} \setminus \mathbb{K}$, where $\mathbb{L}$ is an extension field of $\mathbb{K}$ of degree 2, we apply the map $(\alpha, \beta) \mapsto (\alpha + \theta\beta, \beta)$ to each element of $\mathcal{P}$. The problem then reduces to evaluating $\bar{f} = f(x - \theta y, y)$ at the sheared points. To compute $\bar{f}$, [17] provides an algorithm with complexity $O(\mathsf{M}(d_x(d_x+d_y))\log(d_x))$ using a univariate Taylor shift of f seen as a polynomial in x over the ring $\mathbb{L}[y]$. Algorithm 3 describes an algorithm for this task which improves the cost on the logarithmic level, by using Taylor shifts of the homogeneous components of f.

Algorithm 3 ShearPoly(f, a, b)

Input: $f = \sum_{i=0}^{d_x} \sum_{j=0}^{d_y} f_{i,j} x^i y^j \in \mathbb{L}[x,y]$; $a \in \mathbb{L}$ and $b \in \mathbb{L}$.
Output: $f(ax + by, y)$.
1: **for** $t = 0, \dots, d_x + d_y$ **do**
2: $h_t \leftarrow \sum_{i=\max(0,\,t-d_y)}^{\min(t,\,d_x)} f_{i,t-i} z^i \in \mathbb{L}[z]$
3: $s_t \leftarrow h_t(az + b)$ ▷ *Taylor shift*
4: **return** $\sum_{t=0}^{d_x+d_y} y^t s_t(x/y)$

Theorem 4.2. *Algorithm 3 correctly computes $f(ax+by, y)$, which has x-degree at most d_x and y-degree at most $d_x + d_y$, at a cost of $O((d_x+d_y)\mathsf{M}(d_x)\log(d_x)) \subset \tilde{O}(d_x(d_x+d_y))$ operations in $\mathbb{L}$.*

Proof. Observe that $y^t h_t(x/y)$ is the homogeneous component of f of degree t, and in particular $f = \sum_{t=0}^{d_x+d_y} y^t h_t(x/y)$. Thus

$$f(ax+by, y) = \sum_{t=0}^{d_x+d_y} y^t h_t\left(\frac{ax+by}{y}\right) = \sum_{t=0}^{d_x+d_y} y^t s_t(x/y),$$

hence the correctness. The degree bounds on the output are straightforward. As for complexity, only Line 3 uses arithmetic operations. First, scaling $h_t(z) \mapsto h_t(az)$ costs $O(d_x)$ operations in $\mathbb{L}$, since $\deg h_t \le d_x$; then the Taylor shift $h_t(az) \mapsto h_t(az+b)$ costs $O(\mathsf{M}(d_x)\log(d_x))$ operations in $\mathbb{L}$ according to [29, Fact 2.1(iv)]. Summing over the $d_x + d_y$ iterations yields the claimed bound. □

This leads to Algorithm 4, where $\mathcal{P}$ may have repeated α_i's.

Algorithm 4 MPE-Shear$_{d,\boldsymbol{\eta},\mathcal{P}}(f)$

Preinput: an integer $d \in \mathbb{Z}_{>0}$; a $(d,1)$-reshaping sequence $\boldsymbol{\eta}$; a point set $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n \subset \mathbb{K}^2$.
Precomputation:
a: $(\mathbb{L}, \theta) \leftarrow$ degree 2 extension of $\mathbb{K}$, element $\theta \in \mathbb{L} \setminus \mathbb{K}$
b: $\bar{\mathcal{P}} \leftarrow \{(\alpha_i + \theta\beta_i, \beta_i)\}_{i=1}^n \subset \mathbb{L}^2$
c: Do the precomputation of MPE-DistinctX$_{d,\boldsymbol{\eta},\bar{\mathcal{P}}}$
Input: $f \in \mathbb{K}[x,y]$ with $\deg_x f + \deg_y f < d$.
Output: $\big(f(\alpha_1,\beta_1), \dots, f(\alpha_n,\beta_n)\big) \in \mathbb{K}^n$.
1: $\bar{f} \leftarrow$ ShearPoly$(f, 1, -\theta)$ ▷ $\bar{f} = f(x - \theta y, y)$
2: **return** MPE-DistinctX$_{d,\boldsymbol{\eta},\bar{\mathcal{P}}}(\bar{f})$

Theorem 4.3. *Algorithm 4 is correct. If $\bar{\mathcal{P}}$ is $\boldsymbol{\eta}$-balanced and $\boldsymbol{\eta}$ has length in $O(\log(n))$, its complexity is $\tilde{O}(\deg_x f(\deg_x f + \deg_y f) + n)$.*

5 INTERPOLATION

In this section we use reshaping for the interpolation problem in a similar setting: we input a point set $\mathcal{P}$ for precomputation, and input interpolation values at online time. When $\mathcal{P}$ is appropriately balanced, we solve the interpolation problem in quasi-linear time (see Algorithm 5). The strategy is to first shear the point set to have unique y-coordinates and compute $u \in \mathbb{L}[y]$ which interpolates the values on the sheared y-coordinates. Then we reshape this into $r \in \mathbb{L}[x,y]$ with x- and y-degrees roughly $\sqrt{n}$. Shearing back this polynomial to interpolate the original point set is now in quasi-linear time; a last reshaping allows us to meet the target y-degree.

Theorem 5.1. *Algorithm 5 is correct and has complexity*

$$\tilde{O}\Big(k_1 n + k_2\Big(\sqrt{n} + \sum_{j=1}^{k_1} \deg_x g_{1,j}\Big)^2 + \sum_{\ell=1}^{2} k_\ell \sum_{j=1}^{k_\ell} \eta_{\ell,k} \deg_x g_{\ell,j}\Big).$$

If $\bar{\mathcal{P}}$ is $\boldsymbol{\eta}_1$-balanced and $\mathcal{P}$ is $\boldsymbol{\eta}_2$-balanced, and both $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ have length in $O(\log n)$, then the complexity is $\tilde{O}(n)$.

Proof. First note that a reshaping sequence of length $O(\log n)$ and satisfying the preinput constraints exists, due to Corollary 3.6 and the assumption $d \ge v_x(\mathcal{P})$. For correctness, observe that all points in $\bar{\mathcal{P}}$ have pairwise distinct y-coordinates, so computing u makes sense. Viewing u as an element of $\mathbb{L}[x,y]$ with $\deg_x u = 0$, we

Algorithm 5 $\textsc{Interpolate}_{d,\boldsymbol{\eta},\mathcal{P}}(\boldsymbol{\gamma})$

Preinput: an integer $d \in \mathbb{Z}_{>0}$; an (n,d)-reshaping sequence $\boldsymbol{\eta} = (\eta_i)_{i=0}^{k}$ such that $\eta_{k_1} = \lfloor\sqrt{n}\rfloor$ for some k_1; a point set $\mathcal{P} = \{(\alpha_i,\beta_i)\}_{i=1}^{n} \subseteq \mathbb{K}^2$ such that $v_x(\mathcal{P}) \le d \le \lfloor\sqrt{n}\rfloor + 1$ and $v_x(\mathcal{P}) \le \min_{1\le i\le k}(2\eta_i - \eta_{i-1} + 1)$.
Precomputation:
a: $\boldsymbol{\eta}_1 \leftarrow (\eta_i)_{i=0}^{k_1}$ and $\boldsymbol{\eta}_2 \leftarrow (\eta_i)_{i=k_1}^{k}$
b: $(\mathbb{L},\theta) \leftarrow \begin{cases}(\mathbb{K},0) \text{ if } v_y(\mathcal{P}) = 1\\ \text{degree 2 extension of } \mathbb{K}, \theta \in \mathbb{L}\setminus\mathbb{K} \text{ otherwise}\end{cases}$
c: $\bar{\mathcal{P}} \leftarrow \{(\alpha_i,\bar{\beta}_i)\}_{i=1}^{n}$, where $\bar{\beta}_i = \theta\alpha_i + \beta_i$
d: $\boldsymbol{g}_1 \leftarrow \boldsymbol{\eta}_1$-reshaper for $\bar{\mathcal{P}}$
e: $\boldsymbol{g}_2 \leftarrow \boldsymbol{\eta}_2$-reshaper for $\mathcal{P}$
Input: Interpolation values $\boldsymbol{\gamma} = (\gamma_i)_{i=1}^{n} \in \mathbb{K}^n$.
Output: $f \in \mathbb{K}[x,y]$ satisfying $f(\alpha_i,\beta_i) = \gamma_i$ for $i = 1,\dots,n$, $\deg_y f < d$ and $\deg_x f \le \lfloor\sqrt{n}\rfloor + \sum_{g\in\boldsymbol{g}_1\cup\boldsymbol{g}_2}\deg_x g$.
1: $u \in \mathbb{L}[y]$ with $\deg u < n$ and $u(\bar{\beta}_i) = \gamma_i$ for $i = 1,\dots,n$
2: $r \leftarrow \textsc{Reshape}(u,\boldsymbol{\eta}_1,\boldsymbol{g}_1) \in \mathbb{L}[x,y]$
3: $s \leftarrow r(x,\theta x+y)$ ▷ *using* ShearPoly
4: Write $s = s_1 + \theta s_2$, where $s_1, s_2 \in \mathbb{K}[x,y]$
5: **return** $\textsc{Reshape}(s_1,\boldsymbol{\eta}_2,\boldsymbol{g}_2) \in \mathbb{K}[x,y]$

have $u(\alpha_i,\bar{\beta}_i) = \gamma_i$. By Theorem 3.2 then r has the same evaluations and $\deg_y r < \lfloor\sqrt{n}\rfloor$ and $\deg_x r \le \sum_{i=1}^{k_1}\deg_x g_{1,i}$.

Then, in both cases $v_y(\mathcal{P}) = 1$ and $v_y(\mathcal{P}) > 1$, we have

$$\gamma_i = r(\alpha_i,\bar{\beta}_i) = s(\alpha_i,\beta_i) = s_1(\alpha_i,\beta_i) + \theta s_2(\alpha_i,\beta_i)$$

for $i = 1,\dots,n$. Since $s_1, s_2 \in \mathbb{K}[x,y]$ and all γ_i's are in $\mathbb{K}$, we get $s_2(\alpha_i,\beta_i) = 0$ and $s_1(\alpha_i,\beta_i) = \gamma_i$ for $i = 1,\dots,n$. We also then have that $\deg_y s_1 \le \deg_y s < \lfloor\sqrt{n}\rfloor$ and

$$\deg_x s_1 \le \deg_x s \le \deg_y r + \deg_x r \le \lfloor\sqrt{n}\rfloor + \sum_{j=1}^{k_1}\deg_x g_{1,j}\ .$$

Thus, by Theorem 3.2 again, the output f is such that $f(\alpha_i,\beta_i) = \gamma_i$ for $i = 1,\dots,n$, and $\deg_y f < d$, and

$$\deg_x f \le \lfloor\sqrt{n}\rfloor + \sum_{j=1}^{k_1}\deg_x g_{1,j} + \sum_{j=1}^{k_2}\deg_x g_{2,j}\ .$$

The complexity bound gathers the calls to Algorithms 1 and 3, and the relaxed cost assuming balancedness is due to Lemma 3.4. □

6 MODULAR COMPOSITION

We now turn to the following modular composition problem: given $M, A \in \mathbb{K}[x]$ with $n := \deg_x M > \deg_x A$, and $f \in \mathbb{K}[x,y]$, compute

$$f(x, A(x)) \operatorname{rem} M(x) \in \mathbb{K}[x]\ . \tag{2}$$

We consider the variant of the problem where M and A are available for precomputation. Computing (2) is tantamount to computing the unique element of $(f+I)\cap\mathbb{K}[x]$ of degree less than n, for the ideal $I = \langle M, y-A\rangle \subseteq \mathbb{K}[x,y]$. One can thus see this as a reshaping task: given f of some y-degree, reshape it to a polynomial of y-degree 0 while keeping it fixed modulo I: this is formalised as Algorithm 6.

Like for point sets above, if $\boldsymbol{\eta} = (\eta_i)_{i=0}^{k}$ is a reshaping sequence, we say that $I = \langle M, y-A\rangle$ is $\boldsymbol{\eta}$-balanced if there exists an $\boldsymbol{\eta}$-reshaper $\boldsymbol{g} = (g_i)_{i=1}^{k}$ for I such that $\deg_x g_i \le \lfloor\frac{n}{2\eta_i - \eta_{i-1}+1}\rfloor + 1$.

Theorem 6.1. *Algorithm 6 is correct. If $\langle M, y-A\rangle$ is $\boldsymbol{\eta}$-balanced and $\boldsymbol{\eta}$ has length in $O(\log(n))$, the complexity is $\tilde{O}(\deg_x f \deg_y f + n)$.*

Algorithm 6 $\textsc{ModComp}_{d,\boldsymbol{\eta},M,A}(f)$

Preinput: $d \in \mathbb{Z}_{>0}$; a $(d,1)$-reshaping sequence $\boldsymbol{\eta}$; polynomials $M, A \in \mathbb{K}[x]$ with $n := \deg_x M > \deg_x A$.
Precomputation:
a: $\boldsymbol{g} \leftarrow \boldsymbol{\eta}$-reshaper for $\langle M, y-A\rangle$
Input: $f \in \mathbb{K}[x,y]$ with $\deg_y f < d$.
Output: $f(x,A) \operatorname{rem} M \in \mathbb{K}[x]$.
1: $\hat{f} \leftarrow \textsc{Reshape}(f,\boldsymbol{\eta},\boldsymbol{g}) \in \mathbb{K}[x]$
2: **return** $\hat{f} \operatorname{rem} M$ ▷ *univariate division with remainder*

7 PRECOMPUTING RESHAPERS

7.1 Reshapers for general ideals

Here we describe Algorithm 7 for precomputing reshapers for any zero-dimensional ideal $I \subseteq \mathbb{K}[x,y]$, given a $\prec_{\text{lex}}$-Gröbner basis of I. It operates through the $\mathbb{K}[x]$-module $I_\delta := \{f \in I \mid \deg_y f < \delta\}$, so we first expound the relation between this and I as a corollary of Lazard's structure theorem on bivariate $\prec_{\text{lex}}$-Gröbner bases [12].

Corollary 7.1. *Let $G = \{b_0,\dots,b_s\} \subset \mathbb{K}[x,y]$ be a minimal $\prec_{\text{lex}}$-Gröbner basis defining an ideal $I = \langle G\rangle$. For $\delta \in \mathbb{Z}_{>0}$, let $I_\delta = \{f \in I \mid \deg_y f < \delta\}$, let $\hat{s} = \max\{i \mid \deg_y b_i < \delta,\ 0 \le i \le s\}$, let $d_i = \deg_y b_i$ for $0 \le i \le \hat{s}$ and $d_{\hat{s}+1} = \delta$. Then I_δ is a $\mathbb{K}[x]$-submodule of $\mathbb{K}[x,y]_{\deg_y < \delta}$ which is free of rank $\delta - d_0$ and admits the basis $\{y^j b_i \mid 0 \le j < d_{i+1} - d_i, 0 \le i \le \hat{s}\}$.*

A proof is given in appendix. We will use the following $\mathbb{K}[x]$-module isomorphism which converts between bivariate polynomials of bounded y-degree and vectors over $\mathbb{K}[x]$: for any $\delta \in \mathbb{Z}_{>0}$,

$$\phi_\delta : f = \textstyle\sum_{j=0}^{\delta-1} f_j(x) y^j \in \mathbb{K}[x,y] \mapsto [f_0,\dots,f_{\delta-1}] \in \mathbb{K}[x]^{1\times\delta}\ .$$

If I is zero-dimensional then in Corollary 7.1 we have $d_0 = 0$ and I_δ has rank δ. Any basis B of I_δ can be represented as a nonsingular matrix $M_B \in \mathbb{K}[x]^{\delta\times\delta}$ whose rows are $\phi_\delta(B)$. Then, $\Delta(I_\delta) := \deg\det(M_B)$ does not depend on the choice of B since all bases of I_δ have the same determinant up to scalar multiplication.

In this section, we use the *Popov form* [20], which can be defined for any matrix and with "shifts"; here we only need the unshifted, nonsingular square case.

Definition 7.2. For any row vector $\boldsymbol{v} \in \mathbb{K}[x]^{1\times\delta}$ its *row degree* denoted $\deg\boldsymbol{v}$ is the maximal degree among its entries. The *pivot* of $\boldsymbol{v}$ is the rightmost entry of $\boldsymbol{v}$ with degree $\deg\boldsymbol{v}$. A nonsingular matrix $P = [p_{ij}] \in \mathbb{K}[x]^{\delta\times\delta}$ is in *Popov form* if p_{ii} is the pivot of the ith row, is monic, and $\deg p_{ii} > \deg p_{ji}$ for any $j \ne i$.

For a (free) $\mathbb{K}[x]$-submodule $\mathcal{M} \subset \mathbb{K}[x]^{1\times\delta}$ of rank δ, we identify a basis of $\mathcal{M}$ as the rows of a nonsingular matrix in $\mathbb{K}[x]^{\delta\times\delta}$. Any such $\mathcal{M}$ has a unique basis $P \in \mathbb{K}[x]^{\delta\times\delta}$ in Popov form, which we call *the Popov basis of* $\mathcal{M}$. It has minimal row degrees in the following sense: if $N \in \mathbb{K}[x]^{\delta\times\delta}$ is another basis of $\mathcal{M}$, there is a bijection ψ from the rows of P to the rows of N such that $\deg\boldsymbol{p} \le \deg\psi(\boldsymbol{p})$ for any row $\boldsymbol{p}$ of P. The Popov basis satisfies $\Delta(\mathcal{M}) = \Delta(P) = |\operatorname{cdeg}(P)|$, using the following notation: the sum of the entries of a tuple $\boldsymbol{t} \in \mathbb{Z}_{\ge 0}^{\delta}$ is denoted $|\boldsymbol{t}|$; the column degree of a matrix $B \in \mathbb{K}[x]^{\delta\times\delta}$ is $\operatorname{cdeg}(B) = (d_i)_{i=1}^{\delta} \in \mathbb{Z}_{\ge 0}^{\delta}$, with d_i the largest degree in the ith column of B (for a zero column, $d_i = 0$).

The next result allows us to compute Popov forms efficiently.

Proposition 7.3 ([16]). *There is an algorithm which inputs a nonsingular matrix $B \in \mathbb{K}[x]^{\delta\times\delta}$ and outputs the Popov basis of the $\mathbb{K}[x]$-row space of B using $\tilde{O}(\delta^{\omega-1}|\mathrm{cdeg}(B)|)$ operations in $\mathbb{K}$, assuming that $\delta \in O(|\mathrm{cdeg}(B)|)$.*

Since Popov forms are "column reduced", they are well suited for matrix division with remainder [10, Thm. 6.3-15]: if $P \in \mathbb{K}[x]^{\delta\times\delta}$ is the Popov basis of $\mathcal{M}$, then for any $\boldsymbol{v} \in \mathbb{K}[x]^{1\times\delta}$ there is a unique $\boldsymbol{u} \in \boldsymbol{v} + \mathcal{M}$ such that $\mathrm{cdeg}(\boldsymbol{u}) < \mathrm{cdeg}(P)$ entrywise; we denote $\boldsymbol{u} = \boldsymbol{v} \operatorname{rem} P$. Furthermore, $\boldsymbol{u}$ has minimal row degree among all vectors in $\boldsymbol{v} + \mathcal{M}$. Such remainders can be computed efficiently:

Proposition 7.4 ([16]). *There is an algorithm which inputs a Popov form $P \in \mathbb{K}[x]^{\delta\times\delta}$ and $\boldsymbol{v} \in \mathbb{K}[x]^{1\times\delta}$ such that $\mathrm{cdeg}(\boldsymbol{v}) < \mathrm{cdeg}(P) + (\Delta(P), \ldots, \Delta(P))$ entrywise, and outputs $\boldsymbol{v} \operatorname{rem} P$ using $\tilde{O}(\delta^{\omega-1}\,\Delta(P))$ operations in $\mathbb{K}$, assuming that $\delta \in O(\Delta(P))$.*

Algorithm 7 ComputeReshaper(G, η, δ)

Input: A reduced $\prec_{\mathrm{lex}}$-Gröbner basis $G = \{b_0, \ldots, b_s\} \subset \mathbb{K}[x, y]$, sorted by increasing y-degree, for a zero-dimensional ideal I (hence $b_0 \in \mathbb{K}[x]$); $\eta, \delta \in \mathbb{Z}_{>0}$ with $\delta < \eta$.
Output: If no polynomial in $y^\eta + I$ has y-degree $< \delta$, "Fail"; otherwise, $g = y^\eta - \hat{g} \in I$ with $\deg_y \hat{g} < \delta$ and $\deg_x \hat{g}$ minimal.

1: $R \leftarrow y^\eta \operatorname{rem} G$
2: **if** $\deg_y R \geq \delta$ **then return** "Fail"
3: $B_\delta \leftarrow$ basis of $I_\delta = \{f \in I \mid \deg_y f < \delta\}$ as in Corollary 7.1
4: $B \in \mathbb{K}[x]^{\delta\times\delta} \leftarrow$ row-wise applying ϕ_δ to elements of B_δ
5: $P \in \mathbb{K}[x]^{\delta\times\delta} \leftarrow$ Popov basis of I_δ from the basis B
6: $\hat{g} \leftarrow -\phi_\delta^{-1}(\phi_\delta(R) \operatorname{rem} P) \in \mathbb{K}[x, y]$
7: **return** $g = y^\eta - \hat{g} \in \mathbb{K}[x, y]$

Theorem 7.5. *Algorithm 7 is correct. Assuming $\eta \in O(\Delta(I_\delta))$, it costs $\tilde{O}(\delta^{\omega-1}\,\Delta(I_\delta) + \eta s \deg_x b_0)$ operations in $\mathbb{K}$.*

Proof. Since G is a $\prec_{\mathrm{lex}}$-Gröbner basis, if $y^\eta + I$ contains a polynomial of y-degree less than δ, then $\deg_y(y^\eta \operatorname{rem} G) \leq \delta$ and the algorithm does not fail at Line 2.

For correctness of the output, observe that $y^\eta - R \in I$ so satisfactory $g = y^\eta - \tilde{g}$ all have $\tilde{g} \in R + I_\delta$. Now, $\hat{g}$ of Line 6 is clearly in $R + I_\delta$ since P is the Popov basis of I_δ, but also $\hat{g}$ has minimal x-degree in the coset $R + I_\delta$. Hence among all g of the correct form, the algorithm returns that of minimal x-degree.

For complexity, work is done in Lines 1, 5 and 6. Since G is reduced, $\deg_x b_0 > \ldots > \deg_x b_s$. Therefore the diagonal entries in B are dominant in their columns and $|\mathrm{cdeg}\, B| = \Delta(B) = \Delta(P) = \Delta(I_\delta)$. For Line 1, we use the algorithm of [23] with cost $\tilde{O}(\eta s \deg_x b_0)$, see Lemma A.2. Line 5 costs $\tilde{O}(\delta^{\omega-1}|\mathrm{cdeg}\, B|)$ by Proposition 7.3 and Line 6 costs $\tilde{O}(\delta^{\omega-1}\,\Delta(P))$ since $\deg_x R < \deg_x b_0 < \Delta(P)$. □

7.2 Reshapers for the considered problems

We turn to obtaining the reduced $\prec_{\mathrm{lex}}$-Gröbner basis of $\Gamma(\mathcal{P})$. We will consider the $\mathbb{K}[x]$-submodule $\Gamma_m(\mathcal{P}) = \Gamma(\mathcal{P}) \cap \mathbb{K}[x, y]_{\deg_y < m}$ which by Lemma 2.2 and Corollary 7.1 is free and of rank m. To obtain a $\prec_{\mathrm{lex}}$-Gröbner basis, our approach is to first compute the Hermite basis of $\Gamma_m(\mathcal{P})$. This is the unique basis whose corresponding matrix $H \subset \mathbb{K}[x]^{m\times m}$ is lower triangular, with each diagonal entry monic and strictly dominating the degrees in its column.

Lemma 7.6. *For any point set $\mathcal{P} \subseteq \mathbb{K}^2$ and any $m > \nu_x(\mathcal{P})$, we have $\Gamma(\mathcal{P}) = \langle \Gamma_m(\mathcal{P}) \rangle$ and $\Delta(\Gamma_m(\mathcal{P})) = |\mathcal{P}|$.*

Proof. By Lemma 2.2 the elements of the reduced $\prec_{\mathrm{lex}}$-Gröbner basis of $\Gamma(\mathcal{P})$ have y-degree at most $\nu_x(\mathcal{P})$, implying the first claim. Further, this means the quotient $\mathbb{K}[x, y]/\Gamma(\mathcal{P})$ is isomorphic to the quotient of modules $\mathbb{K}[x, y]_{\deg_y < m}/\Gamma_m(\mathcal{P})$. It is a basic property of zero-dimensional varieties that the $\mathbb{K}$-dimension of the former is the number of points in $\mathcal{P}$, which is hence also the $\mathbb{K}$-dimension of the latter. This dimension is $\Delta(\Gamma_m(\mathcal{P}))$ by [16, Lem. 2.3]. □

Proposition 7.7. *There is an algorithm which inputs $\mathcal{P} \subset \mathbb{K}^2$ and outputs the reduced $\prec_{\mathrm{lex}}$-Gröbner basis of $\Gamma(\mathcal{P})$ and has complexity $\tilde{O}(\nu_x(\mathcal{P})^{\omega-1}|\mathcal{P}|)$.*

Proof. Let $\Gamma = \Gamma(\mathcal{P})$, $\Gamma_m = \Gamma_m(\mathcal{P})$, and $m = \nu_x(\mathcal{P}) + 1$. We first compute the Hermite basis H of $\Gamma_m(\mathcal{P})$ in time $\tilde{O}(m^{\omega-1}|\mathcal{P}|)$ using (a special case of) [9, Thm. 1.5], in which taking $\boldsymbol{s} = (0, n, \ldots, (m-1)n)$ ensures that the $\boldsymbol{s}$-Popov basis P of Γ_m is the Hermite basis.

Let $G = \{g_0, \ldots, g_{m-1}\} \subset \mathbb{K}[x, y]$ be given as the ϕ_m^{-1}-image of the rows of H. By Lemma 7.6 and since H is lower triangular, G is a $\prec_{\mathrm{lex}}$-Gröbner basis of Γ but not necessarily minimal. Construct $G' \subseteq G$ from G by excluding the elements $g \in G$ such that there is $g' \in G$ with $\deg_y g' < \deg_y g$ and $\deg_x(\mathrm{LC}_y(g')) \leq \deg_x(\mathrm{LC}_y(g))$, i.e. $\mathrm{LT}_{\mathrm{lex}}(g')$ divides $\mathrm{LT}_{\mathrm{lex}}(g)$. This makes G' a minimal $\prec_{\mathrm{lex}}$-Gröbner basis of Γ [4, Lem. 3 of Chap. 2 §7], and we claim it is the reduced one. Indeed, since H is in Hermite form, the selection criteria for G' ensures that for any $g \neq g'$ in G' and any term $x^i y^j$ in g', we have $i < \deg_x(\mathrm{LT}_{\mathrm{lex}}(g))$ or $j < \deg_y g$, and hence G' is reduced. Obtaining G' from H costs no arithmetic operations. □

Corollary 7.8. *Given a point set $\mathcal{P} \subseteq \mathbb{K}^2$ of cardinality n and a reshaping sequence $\boldsymbol{\eta} = (\eta_i)_{i=0}^k$ with $n \geq \eta_k$ and satisfying the condition of Lemma 3.5, then we can determine if $\mathcal{P}$ is $\boldsymbol{\eta}$-balanced and compute an $\boldsymbol{\eta}$-reshaper $\boldsymbol{g} = (g_i)_{i=1}^k$ for $\mathcal{P}$ where each element has minimal possible x-degree in complexity $\tilde{O}(k\eta_0^{\omega-1}n + \eta_0 \nu_x nk)$.*

Proof. By Proposition 7.7, computing a reduced $\prec_{\mathrm{lex}}$-Gröbner basis $G = (b_i)_{i=0}^{\nu_x}$ of $\Gamma(\mathcal{P})$ costs $\tilde{O}(\nu_x^{\omega-1}n) \subset \tilde{O}(\eta_0^{\omega-1}n)$. We then run Algorithm 7 on input $\eta = \eta_i$ and $\delta_i = 2\eta_i - \eta_{i-1} + 1 > \nu_x$ for $i = 1, \ldots, k$. Lemma 7.6 ensures $\Delta(\Gamma_\delta(\mathcal{P})) = n$ for any $\delta > \nu_x$, thus the cost of each call to Algorithm 7 becomes $\tilde{O}(\eta_0^{\omega-1}n + \eta_0 \nu_x n)$. □

Corollary 7.9. *Given $M, A \in \mathbb{K}[x]$ with $n := \deg M > \deg A$ and a reshaping sequence $\boldsymbol{\eta} = (\eta_i)_{i=0}^k$ with $n \geq \eta_k$, then we can determine if $I := \langle M, y - A \rangle$ is $\boldsymbol{\eta}$-balanced and compute an $\boldsymbol{\eta}$-reshaper $\boldsymbol{g} = (g_i)_{i=1}^k$ for $\mathcal{P}$ where each element has minimal possible x-degree in complexity $\tilde{O}(k\eta_0^{\omega-1}n)$.*

Proof. For any δ, and using the notation of Algorithm 7, the basis B of I_δ is lower triangular with diagonal entries $(M, 1, \ldots, 1)$. Hence $\Delta(B) = \Delta(I_\delta) = n$. Using $s = 1$ and $\deg_x b_0 = \deg_x M = n$, the cost follows from Theorem 7.5. □

8 GENERICITY

Now we show that on random input our algorithms usually have quasi-linear complexity, i.e. that random point sets are balanced and that $\langle M, y - A \rangle$ is balanced for random univariate A, M.

Lemma 8.1. *Let $\alpha_1, \ldots, \alpha_n \in \mathbb{K}$ be distinct, let $y_1, \ldots, y_n$ be new indeterminates, and consider for $s \in \mathbb{Z}_{>0}$ the matrix*

$$A_s = \left[V_s \mid DV_s \mid \ldots \mid D^{m-1}V_s\right] \in \mathbb{K}[y_1, \ldots, y_n]^{n \times ms} \quad (3)$$

where D is the diagonal matrix with entries $(y_1, \ldots, y_n)$, and $V_s = [\alpha_i^{j-1}]_{1 \le i \le n, 1 \le j \le s} \in \mathbb{K}^{n \times s}$. Then A_s has rank $\min(n, ms)$.

Proof. Note that by rank of a matrix over $\mathbb{K}[y_1, \ldots, y_n]$, we mean the rank of that matrix seen as over the field of fractions $\mathbb{K}(y_1, \ldots, y_n)$. If we specialise y_i to α_i^s for $i = 1, \ldots, n$, we obtain the Vandermonde matrix $\hat{A}_s = [\alpha_i^{j-1}]_{1 \le i \le n, 1 \le j \le ms} \in \mathbb{K}^{n \times ms}$ of the points $\alpha_1, \ldots, \alpha_n$. Since these points are distinct, $\hat{A}_s$ has full rank $\min(n, ms)$. Hence A_s must also have full rank. □

The columns of A_s can be identified to monomials $x^i y^j$ with $i < s$ and $j < m$. In particular, if $p \in \Gamma(\mathcal{P})$ is a bivariate polynomial with x-degree less than s and y-degree less than m which vanishes on a point set $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n \subset \mathbb{K}^2$ with distinct α_i's, then the coefficients of p form a vector in the right kernel of the matrix $\hat{A}_s = (A_s)_{|y_i \to \beta_i} \in \mathbb{K}^{n \times ms}$ specializing y_i to β_i.

The next lemma determines the exact row degrees of the Popov basis $P \in \mathbb{K}[x]^{m \times m}$ of $\phi_m(\Gamma_m(\mathcal{P}))$ for a "random" point set $\mathcal{P}$, where $\Gamma_m(\mathcal{P}) = \Gamma(\mathcal{P}) \cap \mathbb{K}[x, y]_{\deg_y < m}$ as in Section 7.2.

Lemma 8.2. *Let $\alpha_1, \ldots, \alpha_n \in \mathbb{K}$ be distinct, let $\mathcal{T} \subseteq \mathbb{K}$ be a finite subset, and let $\lambda : \mathbb{K}^n \to \mathbb{K}^n$ be an affine map. For $\gamma_1, \ldots, \gamma_n \in \mathcal{T}$ chosen independently and uniformly at random, set $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n$ where $(\beta_1, \ldots, \beta_n) = \lambda(\gamma_1, \ldots, \gamma_n)$. Let $m \in \mathbb{Z}$ with $\nu_x(\mathcal{P}) < m \le n$ and let $(d, t) = \text{QUO\_REM}(n, m)$. With probability at least $1 - 2nm/|\mathcal{T}|$, the Popov basis $P \in \mathbb{K}[x]^{m \times m}$ of $\phi_m(\Gamma_m(\mathcal{P}))$ has exactly $m - t$ rows of degree d and t rows of degree $d+1$ and in particular $\deg_x P \le d+1$.*

Proof. Let $p_1, \ldots, p_m \in \mathbb{K}[x, y]$ be the polynomials defined by the rows of P. Lemma 2.2 shows $\Delta(P) = n = \sum_{i=1}^m \deg_x p_i$.

For any $s \in \mathbb{Z}_{>0}$, let $A_s \in \mathbb{K}[y_1, \ldots, y_n]^{n \times ms}$ be as in Lemma 8.1, hence $\text{rank}(A_s) = \min(n, ms)$. Let $\hat{A}_s = (A_s)_{|y_i \to \beta_i} \in \mathbb{K}^{n \times ms}$. Taking $s = d$, as mentioned above, if $\deg_x p_i < d$ for some i, then the coefficient vector of p_i is in the right kernel of $\hat{A}_d$, and so $\text{rank}(\hat{A}_d) < \text{rank}(A_d) = md \le n$. Thus, letting $M \in \mathbb{K}[y_1, \ldots, y_n]$ be a non-zero $md \times md$ minor of A_d then $M(\beta_1, \ldots, \beta_n) = M(\lambda(\gamma_1, \ldots, \gamma_n)) = 0$; M has degree at most $m-1$ in each variable, so the total degree of M is less than nm, and since λ is affine the composition $M(\lambda(z_1, \ldots, z_n))$ also has total degree less than nm. Then, by Lemma 2.1 the probability that $M(\lambda(\gamma_1, \ldots, \gamma_n)) = 0$ is at most $nm/|\mathcal{T}|$.

Assume now that all rows of P have degree at least d. For each i such that $\deg_x p_i = d$, the coefficients of p_i form a vector in the right kernel of $\hat{A}_{d+1} \in \mathbb{K}^{n \times m(d+1)}$. By Lemma 8.1, A_{d+1} has a right kernel (over the fractions) of dimension $m(d+1) - n = m - t$. Since the rows of P are linearly independent over $\mathbb{K}[x]$, and therefore also over $\mathbb{K}$, whenever $\text{rank}(\hat{A}_{d+1}) = \text{rank}(A_{d+1})$ at most $m - t$ rows of P have x-degree d. We thus consider $N \in \mathbb{K}[y_1, \ldots, y_n]$ a non-zero $n \times n$ minor of A_{d+1}. Again N has total degree less than nm and the probability that $N(\beta_1, \ldots, \beta_n) = N(\lambda(\gamma_1, \ldots, \gamma_n)) = 0$ is at most $nm/|\mathcal{T}|$, bounding the probability that $\text{rank}(\hat{A}_{d+1}) < \text{rank}(A_{d+1})$.

Hence, with probability at least $1 - 2nd/|\mathcal{T}|$, P has all rows of degree at least d and j rows of degree exactly d with $j \le m - t$. Each of the remaining $m - j$ rows has degree at least $d + 1$, while their degrees must sum to $n - jd = md + t - jd = (m-j)d + t \le (m-j)(d+1)$. Hence each of them has degree exactly $d + 1$. □

Algorithm 7 for computing reshapers outputs a $g = y^\eta - \hat{g}$ with $\deg_y \hat{g} < \delta$ satisfying $\deg_x \hat{g} \le \deg_x P$, where P is the Popov basis of $\Gamma_\delta(\mathcal{P})$. Lemma 8.2 states that generically we can expect $\deg_x P \le \lfloor \frac{n}{\delta} \rfloor + 1$, and so when $\delta = 2\eta_i - \eta_{i-1} + 1$ in a reshaping sequence, this matches the definition of $\boldsymbol{\eta}$-balanced.

Corollary 8.3. *Let $\alpha_1, \ldots, \alpha_n \in \mathbb{K}$ be distinct, let $\mathcal{T} \subseteq \mathbb{K}$ a finite subset, and let $\lambda : \mathbb{K}^n \to \mathbb{K}^n$ be an affine map. For $\gamma_1, \ldots, \gamma_n \in \mathcal{T}$ chosen independently and uniformly at random, set $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n$ where $(\beta_1, \ldots, \beta_n) = \lambda(\gamma_1, \ldots, \gamma_n)$. Let $\boldsymbol{\eta} = (\eta_i)_{i=0}^k$ be a reshaping sequence with $\eta_0 \le n$ and satisfying the constraint of Lemma 3.5. Then $\mathcal{P}$ is $\boldsymbol{\eta}$-balanced with probability at least $1 - n^2k/|\mathcal{T}|$.*

The above proposition directly applies to both our MPE and interpolation algorithms on random point sets with unique x-coordinates. Note that in the case of interpolation, where the point set is sheared if its y-valency is greater than one, the property of being $\boldsymbol{\eta}$-balanced is not inherited a priori by the sheared point set. The probability of being $\boldsymbol{\eta}$-balanced, however, is preserved, since the shearing acts as an affine transformation on the y-coordinates. There are many formulations depending on the type of randomness one needs over the point sets; the following is a simple example over finite fields:

Corollary 8.4. *Let $d, n \in \mathbb{Z}_{>0}$ with $d \le n$ and $\mathbb{F}_q$ be a finite field with q elements, and let $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n \subseteq \mathbb{F}_q^2$ be chosen uniformly at random among point sets with cardinality n. Then with probability of at least $\left(1 - \frac{n^2}{q}\right)\left(1 - \frac{3n^2(\log_{3/2}(n)+1)}{q}\right)$ over the choice of $\mathcal{P}$ the following two problems can be solved with cost $\tilde{O}(n)$:*

1. *Input polynomial $f \in \mathbb{F}_q[x, y]$ such that $\deg_x f < n/d$ and $\deg_y f < d$, and output $(f(\alpha_i, \beta_i))_{i=1}^n \in \mathbb{F}_q^n$.*
2. *Input interpolation values $\boldsymbol{\gamma} = (\gamma_i)_{i=1}^n \in \mathbb{F}_q^n$, and output $f \in \mathbb{F}_q[x, y]$ satisfying $f(\alpha_i, \beta_i) = \gamma_i$ for $i = 1, \ldots, n$, as well as $\deg_y f < d$ and $\deg_x f \le cn$ for some constant c which depends only on n and d.*

Proof sketch. The probability simply bounds the probability that $\mathcal{P}$ has unique x-coordinates *and* that it is balanced in all the necessary ways. By Corollary 3.6 there is an appropriate reshaping sequence of length at most $\log_{3/2}(n) + 2$. □

We do not make a claim about the genericity in Algorithm 4: due to the shearing in that algorithm, the arguments of this section do not immediately apply. Lastly, we turn to modular composition.

Theorem 8.5. *Let $M \in \mathbb{K}[x]$ be square-free of degree n and let $\boldsymbol{\eta}$ be a $(d, 1)$-reshaping sequence of length k with $0 < d \le n$. Let $\mathcal{T} \subseteq \mathbb{K}$ be a finite subset, and let $A = \sum_{i=0}^{n-1} a_i x^{i-1} \in \mathbb{K}[x]$ where $a_0, \ldots, a_{n-1}$ are chosen independently and uniformly at random from $\mathcal{T}$. Then $\langle M, y - A \rangle$ is $\boldsymbol{\eta}$-balanced with probability at least $1 - n^2k/|\mathcal{T}|$.*

Proof. Let $\mathbb{L}$ be the splitting field of M, so $M = \prod_{i=1}^n (x - \alpha_i)$ for some pairwise distinct $\alpha_1, \ldots, \alpha_n \in \mathbb{L}$. Define the stochastic variables $\beta_i = A(\alpha_i)$ for $i = 1, \ldots, n$; the map $\lambda(a_0, \ldots, a_{n-1}) = (\beta_1, \ldots, \beta_n)$ is $\mathbb{L}$-linear. Consider $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n \subseteq \mathbb{L}^2$. Then Corollary 8.3 implies that $\mathcal{P}$ is $\boldsymbol{\eta}$-balanced with probability at least $1 - \frac{n^2k}{|\mathcal{T}|}$. In this case, for each i there exists $g_i = y^{\eta_i} + \hat{g}_i \in I_{\mathbb{L}}$ where

$\deg_y \hat{g}_i < 2\eta_i - \eta_{i-1}$ and $\deg_x \hat{g}_i \leq \lfloor \frac{n}{2\eta_i - \eta_{i-1}+1} \rfloor + 1$, and where $I_{\mathbb{L}} = \langle M, y - A \rangle \otimes_{\mathbb{K}} \mathbb{L}$. Let $\{1, \zeta, \ldots, \zeta^{s-1}\} \subset \mathbb{L}$ be a basis of $\mathbb{L} : \mathbb{K}$ and write $g_i = g_{i,0} + \zeta g_{i,1} + \ldots + \zeta^{s-1} g_{i,s-1}$ with $g_{i,j} \in \mathbb{K}[x, y]$. Then $g_i \in I_{\mathbb{L}}$ implies that $g_{i,0} \in I$, and by the shape of g_i then $g_{i,0} = y^{\eta_i} + \hat{g}_{i,0}$ where the x- and y-degree of $\hat{g}_{i,0}$ satisfy the same bounds as $\hat{g}_i$. Then the tuple $\boldsymbol{g}_0 = (g_{1,0}, \ldots, g_{k,0}) \in \mathbb{K}[x, y]^k$ forms a balanced $\boldsymbol{\eta}$-reshaper for I. □

REFERENCES

[1] E. F. Assmus and J. D. Key. 1992. *Designs and Their Codes.* Cambridge University Press. https://doi.org/10.1017/CBO9781316529836

[2] R. P. Brent and H. T. Kung. 1978. Fast Algorithms for Manipulating Formal Power Series. *J. ACM* 25, 4 (1978), 581–595. https://doi.org/10.1145/322092.322099

[3] D. G. Cantor and E. Kaltofen. 1991. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica* 28, 7 (1991), 693–701. https://doi.org/10.1007/BF01178683

[4] D. A. Cox, J. Little, and D. O'Shea. 2015. *Ideals, Varieties, and Algorithms* (4th ed.). Springer. https://doi.org/10.1007/978-3-319-16721-3

[5] N. Coxon. 2018. Fast systematic encoding of multiplicity codes. *J. Symb. Comput.* (2018). https://doi.org/10.1016/j.jsc.2018.08.005

[6] X. Dahan. 2009. Size of Coefficients of Lexicographical Gröbner Bases: The Zero-Dimensional, Radical and Bivariate Case. In *Proceedings ISSAC 2009.* 119–126. https://doi.org/10.1145/1576702.1576721

[7] R. A. DeMillo and R. J. Lipton. 1978. A Probabilistic Remark on Algebraic Program Testing. *Inf. Process. Lett.* 7, 4 (1978), 193–195. https://doi.org/10.1016/0020-0190(78)90067-4

[8] W. Hart, F. Johansson, and S. Pancratz. 2015. FLINT: Fast Library for Number Theory. Version 2.5.2, http://flintlib.org.

[9] C.-P. Jeannerod, V. Neiger, É. Schost, and G. Villard. 2016. Fast Computation of Minimal Interpolation Bases in Popov Form for Arbitrary Shifts. In *Proceedings ISSAC 2016.* 295–302. https://doi.org/10.1145/2930889.2930928

[10] T Kailath. 1980. *Linear Systems.* Prentice-Hall.

[11] K. Kedlaya and C. Umans. 2011. Fast Polynomial Factorization and Modular Composition. *SIAM J. Comput.* 40, 6 (Jan. 2011), 1767–1802. https://doi.org/10.1137/08073408X

[12] D. Lazard. 1985. Ideal bases and primary decomposition: case of two variables. *J. Symb. Comput.* 1, 3 (1985). https://doi.org/10.1016/S0747-7171(85)80035-3

[13] F. Le Gall. 2014. Powers of tensors and fast matrix multiplication. In *Proceedings ISSAC 2014.* ACM, 296–303. https://doi.org/10.1145/2608628.2608664

[14] S. Miura. 1993. Algebraic geometric codes on certain plane curves. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)* 76, 12 (1993), 1–13. https://doi.org/10.1002/ecjc.4430761201

[15] V. Neiger, B. Salvy, É. Schost, and G. Villard. 2020. Faster modular composition (work in progress).

[16] V. Neiger and T. X. Vu. 2017. Computing Canonical Bases of Modules of Univariate Relations. In *Proceedings ISSAC 2017.* https://doi.org/10.1145/3087604.3087656

[17] M. Nüsken and M. Ziegler. 2004. Fast Multipoint Evaluation of Bivariate Polynomials. In *Proceedings ESA 2004.* https://doi.org/10.1007/978-3-540-30140-0_49

[18] V. Y. Pan. 1994. Simple Multivariate Polynomial Multiplication. *J. Symb. Comput.* 18, 3 (1994), 183–186. https://doi.org/10.1006/jsco.1994.1042

[19] M. S. Paterson and L. J. Stockmeyer. 1973. On the number of nonscalar multiplications necessary to evaluate polynomials. *SIAM J. Comput.* 2, 1 (1973), 60–66. https://doi.org/10.1137/0202007

[20] V Popov. 1970. Some properties of the control systems with irreducible matrix-transfer functions. In *Seminar on Diff. Eq. and Dyn. Sys., II.* 169–180. https://doi.org/10.1007/BFb0059934

[21] J. T. Schwartz. 1980. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. ACM* 27, 4 (1980), 701–717. https://doi.org/10.1145/322217.322225

[22] V. Shoup. 2020. NTL: A Library for doing Number Theory, version 11.4.3. `http://www.shoup.net`.

[23] J. van der Hoeven. 2015. On the complexity of multivariate polynomial division. In *Proceedings ACA 2015.* 447–458. https://doi.org/10.1007/978-3-319-56932-1_28

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

[25] J. van der Hoeven and G. Lecerf. 2019. Fast multivariate multi-point evaluation revisited. *J. Complexity* (2019). https://doi.org/10.1016/j.jco.2019.04.001

[26] J. van der Hoeven and É. Schost. 2013. Multi-point evaluation in higher dimensions. *Appl. Algebra Eng. Commun. Comput.* 24, 1 (2013), 37–52. https://doi.org/10.1007/s00200-012-0179-3

[27] G. Villard. 2018. On computing the resultant of generic bivariate polynomials. In *Proceedings ISSAC 2018.* 391–398. https://doi.org/10.1145/3208976.3209020

[28] G. Villard. 2018. On computing the resultant of generic bivariate polynomials. Presentation at ISSAC 2018. http://www.issac-conference.org/2018/slides/villard-computingresultant.pdf

[29] J. von zur Gathen. 1990. Functional decomposition of polynomials: the tame case. *J. Symb. Comput.* 9, 3 (1990). https://doi.org/10.1016/S0747-7171(08)80014-4

[30] J. von zur Gathen and J. Gerhard. 2013. *Modern Computer Algebra* (3rd ed.). Cambridge University Press. https://doi.org/10.1017/CBO9781139856065

[31] R. Zippel. 1979. Probabilistic algorithms for sparse polynomials. In *Proceedings EUROSAM'79.* 216–226. https://doi.org/10.1007/3-540-09519-5_73

APPENDIX

Corollary A.1 (of [12]). *Let $G = \{b_0, \ldots, b_s\} \subset \mathbb{K}[x, y]$ be a minimal $\prec_{\text{lex}}$-Gröbner basis, sorted according to $\prec_{\text{lex}}$. Then*

(1) $\deg_y b_0 < \ldots < \deg_y b_s$*; and*
(2) $\mathsf{LC}_y(b_s) \mid \mathsf{LC}_y(b_{s-1}) \mid \cdots \mid \mathsf{LC}_y(b_0)$.

Proof of Corollary 7.1. Since I is an ideal of $\mathbb{K}[x, y]$ and $I_\delta = I \cap \mathbb{K}[x, y]_{\deg_y < \delta}$, then I_δ is a $\mathbb{K}[x]$-submodule of $\mathbb{K}[x, y]_{\deg_y < \delta}$. Let $\mathcal{B}$ denote the (claimed) basis in the corollary. Clearly $\mathcal{B} \subseteq I_\delta$, and the elements of $\mathcal{B}$ all have different y-degree and so are $\mathbb{K}[x]$-linearly independent. Also $|\mathcal{B}| = \delta - d_0$, so if $\mathcal{B}$ generates I_δ then it is a basis of it and the rank of I_δ is $\delta - d_0$. It remains to show that $\mathcal{B}$ generates I_δ, so take some $f \in I_\delta$. Since $f \in I$ the multivariate division algorithm using G and the order $\prec_{\text{lex}}$ results in $q_0, \ldots, q_s \in \mathbb{K}[x, y]$ such that $f = q_0 b_0 + \ldots + q_s b_s$ with $\deg_y q_i \leq \deg_y f - \deg_y b_i$. Since $\deg_y f < \delta$ this means $q_{\hat{s}+1} = \ldots = q_s = 0$. Say that in each iteration of the division algorithm, we use the greatest index i for which $\mathsf{LT}_{\text{lex}}(b_i)$ divides the leading term of the current remainder. Thus no term of $q_i b_i$ is divisible by $\mathsf{LT}_{\text{lex}}(b_{i+1})$ for any $i < s$. But by Corollary A.1 then $\mathsf{LC}_y(b_{i+1})$ divides $\mathsf{LC}_y(b_i)$, and so if $\deg_y(q_i b_i) \geq \deg_y b_{i+1}$ then $\mathsf{LT}_{\text{lex}}(b_{i+1}) \mid \mathsf{LT}_{\text{lex}}(q_i b_i)$. Consequently $\deg_y q_i < \deg_y b_{i+1} - \deg_y b_i$, and therefore f is a $\mathbb{K}[x]$-linear combination of the elements of $\mathcal{B}$. □

Lemma A.2. *There is an algorithm which inputs a $\prec_{\text{lex}}$-Gröbner basis $G = [b_0, \ldots, b_s] \subseteq \mathbb{K}[x, y]$ with $\deg_y b_0 = 0$, and a polynomial $f \in \mathbb{K}[x, y]$, and outputs $f \operatorname{rem} G$ in time $\tilde{O}(|G| d_x(\deg_y f))$, where $d_x = \max(\deg_x f, \deg_x b_0)$.*

Proof. This is a special case of [23]: the multivariate division algorithm computes $q_0, \ldots, q_s, R \in \mathbb{K}[x, y]$ such that $f = q_0 b_0 + \ldots + q_s b_s + R$ with $R = f \operatorname{rem} G$, and the cost of the algorithm can be bounded as

$$\textstyle\sum_{i=0}^{s} \deg_x^\circ(q_i b_i) \deg_y^\circ(q_i b_i) + \deg_x^\circ(R) \deg_y^\circ(R)\,,$$

where $\deg_x^\circ(\cdot)$ denotes an a priori upper bound on the x-degree, and similarly for $\deg_y^\circ(\cdot)$. Since G is a $\prec_{\text{lex}}$-Gröbner basis, then $\deg_y^\circ(q_i b_i) \leq \deg_y f$ and $\deg_y^\circ(R) \leq \deg_y f$. For the x-degrees, note that in an iteration of the division algorithm where $b_i, i > 0$ is used, then $\deg_x \tilde{R} < \deg_x b_0$, where $\tilde{R}$ is the current remainder, since otherwise the algorithm would have reduced by b_0 as $\deg_y b_0 = 0$. Hence $\deg_x(q_i) \leq \deg_x(q_i \mathsf{LT}_{\text{lex}}(b_i)) < \deg_x b_0$ and so $\deg_x^\circ(q_i b_i) \leq 2 \deg_x b_0$. Similarly, $\deg_x^\circ(R) < \deg_x b_0$. Left is only $\deg_x^\circ(q_0 b_0)$: since $q_0 b_0 = f - q_1 b_1 - \ldots - q_s b_s - R$, then $\deg_x(q_0 b_0) \leq \max(\deg_x f,\ 2 \deg_x b_0)$. □