Amortized multi-point evaluation of multivariate polynomials

Joris van der Hoeven, Grégoire Lecerf

Amortized multi-point evaluation of multivariate polynomials*†

JORIS VAN DER HOEVEN[ab], GRÉGOIRE LECERF[ac]

a. CNRS, École polytechnique, Institut Polytechnique de Paris
Laboratoire d'informatique de l'École polytechnique (LIX, UMR 7161)
Bâtiment Alan Turing, CS35003
1, rue Honoré d'Estienne d'Orves
91120 Palaiseau, France

b. *Email:* vdhoeven@lix.polytechnique.fr
c. *Email:* lecerf@lix.polytechnique.fr

Preliminary version of December 26, 2021

The evaluation of a polynomial at several points is called the problem of multi-point evaluation. Sometimes, the set of evaluation points is fixed and several polynomials need to be evaluated at this set of points. Several efficient algorithms for this kind of "amortized" multi-point evaluation have been developed recently for the special cases of bivariate polynomials or when the set of evaluation points is generic. In this paper, we extend these results to the evaluation of polynomials in an arbitrary number of variables at an arbitrary set of points. We prove a softly linear complexity bound when the number of variables is fixed.

1. INTRODUCTION

Let $\mathbb{K}$ be an effective field, so that we have algorithms for the field operations. Given a polynomial $P \in \mathbb{K}[X_1, \ldots, X_n]$ and a tuple $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_d) \in (\mathbb{K}^n)^d$ of points, the computation of $P(\boldsymbol{\alpha}) := (P(\boldsymbol{\alpha}_1), \ldots, P(\boldsymbol{\alpha}_d)) \in \mathbb{K}^d$ is called the problem of *multi-point evaluation*.

This problem naturally occurs in several areas of applied algebra. When solving a polynomial system, multi-point evaluation can for instance be used to check whether all points in a given set are indeed solutions of the system. In [16], we have shown that efficient algorithms for multi-point evaluation actually lead to efficient algorithms for polynomial system solving. Bivariate polynomial evaluation has also been used to compute generator matrices of algebraic geometry error correcting codes [20].

In the univariate case when $n = 1$, it is well known that one may use so-called "remainder trees" to compute multi-point evaluations in quasi-optimal time [1, 2, 4, 8, 22]. More precisely, if $\mathsf{M}(d)$ stands for the cost to multiply two univariate polynomials of degree $<d$ (in terms of the number of field operations in $\mathbb{K}$), then the multi-point evaluation of a polynomial of degree $<d$ at d points can be computed in time $O(\mathsf{M}(d) \log d)$. Using a variant of the Schönhage–Strassen algorithm [3, 27, 28], it is well known that $\mathsf{M}(d) = O(d \log d \log\log d)$. If we restrict our attention to fields $\mathbb{K}$ of positive characteristic, then we may take $\mathsf{M}(d) = O(d \log d\, 4^{\log^* d})$ [5] and conjecturally even $\mathsf{M}(d) = O(d \log d)$ [6]. In the remainder of this paper, we make the customary hypothesis that $\mathsf{M}(d)/d$ is a non-decreasing function in d.

*. This paper is part of a project that has received funding from the French "Agence de l'innovation de défense".

†. This article has been written using GNU TEXMACS [10].

Currently, the fastest general purpose algorithm for multivariate multi-point evaluation is based on the "baby-step giant-step" method; see e.g. [12, section 3]. For a fixed dimension n and $D := \prod_{i=1}^{n} (\deg_{X_i} P + 1)$ such that $d = O(D)$, this method requires $\tilde{O}(d^{\omega-1} D)$ operations in $\mathbb{K}$ (e.g. by taking $\mathbb{A} = \mathbb{K}$ and $\mathbb{B} = \mathbb{K}^d$ in [12, Proposition 3.3]). Here $\tilde{O}(\Phi)$ is a common abbreviation for $O(\Phi (\log \Phi)^{O(1)})$, and $\omega > 1.5$ is a constant such that two $\sqrt{n} \times \sqrt{n}$ and $\sqrt{n} \times n$ matrices with coefficients in $\mathbb{K}$ can be multiplied using $O(n^\omega)$ operations in $\mathbb{K}$. The best known theoretical bound is $\omega < 1.629$ [21, Table 2, half of the upper bound for $\omega(2)$] (combined with the tensor permutation lemma [17, Corollary 7]). In the special case when $n = 2$, $d = D$, and $\deg_{X_1} P = \deg_{X_2} P$, Nüsken and Ziegler proved the sharper bound $\tilde{O}(d^{(\omega+1)/2})$ [24, particular case of Result 4].

In 2008, Kedlaya and Umans achieved a major breakthrough [18, 19]. On the one hand, they gave an algorithm of complexity $(d + D)^{1+o(1)}$ for the case when $\mathbb{K}$ has positive characteristic. On the other hand, they gave algorithms for multi-point evaluations over $\mathbb{Z}/r\mathbb{Z}$ with quasi-optimal bit-complexity exponents. Unfortunately, to the best of our knowledge, these algorithms do not seem suitable for practical purposes, as observed in [13, Conclusion]. Even in the case when the dimension n is fixed, we also note that the algorithms by Kedlaya and Umans do not achieve a smoothly linear complexity of the form $\tilde{O}(d + D)$.

Recently, several algorithms have been proposed for multi-point evaluation in the case when $\boldsymbol{\alpha}$ is a fixed tuple of points [15, 23]. These algorithms are *amortized* in the sense that we allow for potentially expensive precomputations as a function of $\boldsymbol{\alpha}$. The algorithms from [15, 23] are both restricted to the case when $\boldsymbol{\alpha}$ is sufficiently generic, whereas [14, 23] focus on the bivariate case $n = 2$. In the present paper, we deal with the general case when both n and $\boldsymbol{\alpha}$ are arbitrary. Our main result is the following:

THEOREM 1.1. *Let $n > 1$ be a fixed dimension and let $\boldsymbol{\alpha} \in (\mathbb{K}^n)^d$ be a fixed set of points. Then, given a polynomial $P \in \mathbb{K}[X_1, \ldots, X_n]$ with $D := \prod_{i=1}^{n} (\deg_{X_i} P + 1) \geqslant d$ and an element of $\mathbb{K}$ of multiplicative order at least $4^n n! (D+1)$, we can compute $P(\boldsymbol{\alpha}) = (P(\boldsymbol{\alpha}_1), \ldots, P(\boldsymbol{\alpha}_d))$ using*

$$O(\mathsf{M}(D) (\log D)^{n+1} + \mathsf{M}(d) (\log d)^{n+2})$$

operations in $\mathbb{K}$.

Remark 1.2. In characteristic zero, any integer different from 0, 1, and -1 has infinite order. For finite fields $\mathbb{F}_q$, working in an algebraic extension $\mathbb{F}_{q^k}$ yields elements of order up to $q^k - 1$. Replacing $\mathbb{F}_q$ by such an extension induces an additional overhead of $\tilde{O}(k)$ in our complexity bound.

As in the generic case from [15], our algorithm heavily relies on the relaxed algorithm for polynomial reduction from [9]. Given polynomials $P, B_1, \ldots, B_\ell \in \mathbb{K}[X_1, \ldots, X_n]$, the reduction algorithm computes $Q_1, \ldots, Q_\ell, R \in \mathbb{K}[X_1, \ldots, X_n]$ with

$$P \;=\; Q_1 B_1 + \cdots + Q_\ell B_\ell + R, \tag{1.1}$$

where R is reduced with respect to some admissible ordering $\prec$ on monomials. The relaxed algorithm from [9] essentially performs this reduction with the same complexity (up to a logarithmic factor) as checking the relation.

For our application to multi-point evaluation, the idea is to pick the polynomials B_i in the vanishing ideal

$$\mathfrak{I}_{\boldsymbol{\alpha}} \;:=\; \{A \in \mathbb{K}[X_1, \ldots, X_n] : A(\boldsymbol{\alpha}) = \mathbf{0}\}$$

of $\boldsymbol{\alpha}$, so $P(\boldsymbol{\alpha}) = R(\boldsymbol{\alpha})$. We may next recursively split-up the tuple of points $\boldsymbol{\alpha}$ into two halves and adopt a similar strategy as in the univariate case.

However, there are several technical problems with this idea. First of all, it would be too costly to work with a full Gröbner basis $\{B_1, \ldots, B_\ell\}$ of $\mathfrak{I}_{\boldsymbol{\alpha}}$, because the mere storage of such a basis generally requires more than $\tilde{O}(d)$ terms. In [15], we solved this issue by working with respect to a so called "axial basis" instead. In the case when $\boldsymbol{\alpha}$ is not generic, we have the additional problem that the shape of a Gröbner basis may become very irregular. This again has the consequence that the mere size of all products $Q_i B_i$ in (1.1) may exceed $\tilde{O}(D)$.

In this paper, we address these problems by introducing the new technique of *quasi-reduction*. The idea is to work *simultaneously* with several orderings on the monomials; in particular, each B_i belongs to a Gröbner basis for $\mathfrak{I}_{\boldsymbol{\alpha}}$ with respect to a different admissible ordering. The result of a quasi-reduction (1.1) is usually not reduced with respect to any usual Gröbner basis of $\mathfrak{I}_{\boldsymbol{\alpha}}$, but we will still be able to control the size of R. Moreover, during the quasi-reduction process, we will be able to ensure that the ratios

$$\deg_{X_1} Q_i : \cdots : \deg_{X_n} Q_i$$

and

$$\deg_{X_1} B_i : \cdots : \deg_{X_n} B_i$$

are similar for every i, which further allows us to control the size of the products $Q_i B_i$.

The constant factor in our complexity bound of Theorem 1.1 grows exponentially as a function of n. For the time being, we therefore expect our methods to be of practical use only in low dimensions like $n = 2$ or $n = 3$. Of course, the present paper focuses on worst case bounds for potentially highly non-generic cases. There is hope to drastically lower the constant factors for more common use cases.

One major technical contribution of this paper is the introduction of quasi-reduction and heterogeneous orderings. An interesting question for future work is whether these concepts can be applied to other problems. For instance, consider a zero-dimensional ideal $\mathfrak{I} \subseteq \mathbb{K}[X_1, \ldots, X_n]$ and two Gröbner bases for $\mathfrak{I}$ with respect to different monomial orderings. Each Gröbner basis induces a representation for elements of the quotient algebra $\mathbb{K}[X_1, \ldots, X_n]/\mathfrak{I}$. Does there exist a smoothly linear time algorithm to convert between these two representations?

2. POLYNOMIAL REDUCTION

In this section we recall and extend some results from [9] to the context of this paper.

2.1. Admissible orderings

Let $\mathfrak{M}$ be the set of monomials $X_1^{a_1} \cdots X_n^{a_n}$ with $a_1, \ldots, a_n \in \mathbb{N}$. Any polynomial $P \in \mathbb{K}[X_1, \ldots, X_n]$ can uniquely be written as a linear combination

$$P = \sum_{M \in \mathfrak{M}} P_M M$$

with coefficients P_M in $\mathbb{K}$ and finite *support*

$$\operatorname{supp} P := \{M \in \mathfrak{M} : P_M \neq 0\}.$$

Given a total ordering $\prec$ on $\mathfrak{M}$, the support of any non-zero polynomial P admits a unique maximal element $\operatorname{lm}(P) = \operatorname{lm}_\prec(P) \in \mathfrak{M}$ that is called the *leading monomial* of P; the corresponding coefficient $\operatorname{lc}(P) = \operatorname{lc}_\prec(P) = P_{\operatorname{lm}(P)} \in \mathbb{K}$ is called the *leading coefficient* of P.

A total ordering $\prec$ on $\mathfrak{M}$ is said to be *admissible* if

$$M \mid N \implies M \preccurlyeq N \quad \text{and} \quad M \preccurlyeq N \implies X_i M \preccurlyeq X_i N$$

for all monomials $M, N \in \mathfrak{M}$ and $i \in \{1, \ldots, n\}$. In particular, the lexicographical ordering $\prec_{\text{lex}}$ defined by

$$X_1^{a_1} \cdots X_n^{a_n} \prec_{\text{lex}} X_1^{b_1} \cdots X_n^{b_n} \iff \begin{cases} a_1 < b_1 & \text{or} \\ a_1 = b_1 \wedge a_2 < b_2 & \text{or} \\ \vdots & \\ a_1 = b_1 \wedge \cdots \wedge a_{n-1} = b_{n-1} \wedge a_n < b_n \end{cases}$$

is admissible.

2.2. Sparse polynomial products

A *sparse representation* of a polynomial P in $\mathbb{K}[X_1, \ldots, X_n]$ is a data structure that stores the set of the non-zero terms of P. Each such term is a pair made of a coefficient and a degree vector. In an algebraic complexity model the bit size of the exponents counts for free, so the relevant size of such a polynomial is the cardinality of its support.

Consider two polynomials P and Q of $\mathbb{K}[X_1, \ldots, X_n]$ in sparse representation. An extensive literature exists on the general problem of multiplying P and Q; see [26] for a recent survey. For the purposes of this paper, a superset $\mathcal{S}$ for the support of PQ will always be known. Then we define $\mathsf{SM}(s)$ to be the cost to compute PQ, where s is the maximum of the sizes of $\mathcal{S}$ and the supports of P and Q. We will assume that $\mathsf{SM}(s)/s$ is a non-decreasing function in s. Under suitable assumptions, the following proposition will allow us to take $\mathsf{SM}(s) = O(\mathsf{M}(s) \log s)$ in our multivariate evaluation algorithm.

PROPOSITION 2.1. *Let $\pi_1, \ldots, \pi_n$ be positive integers and let θ in $\mathbb{K}$ be of multiplicative order at least $\pi := \pi_1 \cdots \pi_n$.*

i. The set $\mathcal{D}$ of all products $\theta^{e_1} \theta^{e_2 \pi_1} \cdots \theta^{e_n \pi_1 \cdots \pi_{n-1}}$ for $(e_1, \ldots, e_n) \in \prod_{i=1}^{n} \{0, \ldots, \pi_i - 1\}$ can be computed using $O(\pi)$ operations in $\mathbb{K}$.

ii. Let P and Q be in $\mathbb{K}[X_1, \ldots, X_n]$, in sparse representation. Let $\mathcal{S}$ be a superset of the support of PQ with $\deg_{X_i} M < \pi_i$ for all $M \in \mathcal{S}$ and $i = 1, \ldots, n$. Assume that $\mathcal{D}$ has been precomputed. Then the product PQ can be computed using $O(\mathsf{M}(s) \log s)$ operations in $\mathbb{K}$, where s denotes the maximum of the sizes of $\mathcal{S}$ and the supports of P and Q.

Proof. The first statement is straightforward. The second one is an adapted version of [11, Proposition 6]. □

2.3. Relaxed multivariate series

We assume that the reader is familiar with the technique of relaxed power series evaluations [7], which is an efficient way to solve so-called recursive power series equations. In [9], this technique was generalized to multivariate Laurent series that are expanded according to some admissible ordering $\prec$.

Given a general admissible ordering $\prec$, we know from [25] that there exist real vectors $\lambda_1, \ldots, \lambda_n \in \mathbb{R}^n$, such that

$$X^i \prec X^j \;\Leftrightarrow\; X^{\lambda \cdot i} \prec_{\text{lex}} X^{\lambda \cdot j},$$

where

$$\begin{aligned} X^i &:= X_1^{i_1} \cdots X_n^{i_n} \\ \lambda \cdot i &:= (\lambda_1 \cdot i, \ldots, \lambda_n \cdot i). \end{aligned}$$

For clarity we sometimes denote this ordering $\prec$ by $\prec_\lambda$. In [9], it is always assumed that $\lambda_1,\ldots,\lambda_n \in \mathbb{N}^n$ and $\gcd((\lambda_i)_1,\ldots,(\lambda_i)_n) = 1$ for all i.

Example 2.2. The graded lexicographical ordering $\prec_{\text{glex}}$ is obtained by taking $\lambda_1 = (1,...,1)$, $\lambda_2 = (1,0,\ldots,0)$, $\lambda_3 = (0,1,0\ldots,0,)$, $\ldots$, $\lambda_n = (0,\ldots,0,1,0)$.

Example 2.3. In section 4.1, we will only need orderings for which $\lambda_1 = (2^{a_1},\ldots,2^{a_n})$ for certain $a_1,\ldots,a_n \in \mathbb{Z}$ and $\lambda_2 = (1,0,\ldots,0), \lambda_3 = (0,1,0,\ldots,0),\ldots,\lambda_n = (0,\ldots,0,1,0)$. In fact, we note that $\prec_\lambda = \prec_{c\lambda}$ for any $c \in \mathbb{R}^>$, so modulo the replacement of λ_1 by $2^{-\min(a_1,\ldots,a_n)}\lambda_1$, we may assume without loss of generality that $a_1,\ldots,a_n \in \mathbb{N}$, whenever we wish to apply the theory from [9].

In order to analyze the complexity of relaxed products in this multivariate context, we need to introduce the following quantities for finite subsets $\mathcal{R} \subseteq \mathfrak{M}$:

$$\begin{aligned} \vartheta_i(\mathcal{R}) &= \max\{\lambda_i \cdot e : X^e \in \mathcal{R}\} + 1, \qquad i = 1,\ldots,n \\ \vartheta(\mathcal{R}) &= \vartheta_1(\mathcal{R}) \cdots \vartheta_n(\mathcal{R}). \end{aligned}$$

We also define $\vartheta_{\prec,i}(\mathcal{R}) := \vartheta_i(\mathcal{R})$ and $\vartheta_\prec(\mathcal{R}) := \vartheta(\mathcal{R})$ when we need to emphasize the dependence on $\prec$.

THEOREM 2.4. [9, Theorem 3] *Given sparse polynomials $P, Q \in \mathbb{K}[X_1,\ldots,X_n]$ and a set $\mathcal{R}$ with $(\operatorname{supp} P)\,(\operatorname{supp} Q) \subseteq \mathcal{R}$, the relaxed product PQ can be computed in time*

$$O(\mathsf{SM}(|\mathcal{R}|)\log(\vartheta(\mathcal{R}))).$$

2.4. Quasi-reduction

Let $\prec$ be a total ordering on $\mathfrak{M}$ and consider a finite family $(B_i)_{i\in I}$ of non-zero polynomials in $\mathbb{K}[X_1,\ldots,X_n]$. Each B_i comes with a distinguished monomial $L_i \in \operatorname{supp} B_i$ (that will play the role of the leading monomial) and we define $T_i := (B_i)_{L_i} L_i$. The family $(B_i)_{i\in I}$ is equipped with a *selection strategy*, that is a function

$$\sigma\colon\ \mathfrak{M} \longrightarrow I.$$

For each $i \in I$, we also assume that we have fixed the set $\mathfrak{R}_i \subseteq L_i\mathfrak{M}$ of monomials that will be selected for reduction with respect to B_i, for $i \in I$. We assume that the $\mathfrak{R}_i$ are pairwise disjoint and we set

$$\mathfrak{R} := \coprod_{i\in I} \mathfrak{R}_i.$$

Note that this corresponds to fixing a selection strategy, although we do *not* require that $\mathfrak{R} \supseteq \bigcup_{i\in I} L_i\mathfrak{M}$ (which explains the terminology "quasi-reduction" below). For our application later in this paper, the complement $\mathfrak{R} \setminus (\bigcup_{i\in I} L_i\mathfrak{M})$ will be a "rather relatively small" finite set.

We say that $P \in \mathbb{K}[X_1,\ldots,X_n]$ is *quasi-reduced* if $\operatorname{supp} P \cap \mathfrak{R} = \emptyset$. We say that a total ordering $\prec$ on $\mathfrak{M}$ is *quasi-admissible* (with respect to our choices of $(B_i)_{i\in I}$, $(L_i)_{i\in I}$ and the selection strategy σ) if $M \,|\, N \Longrightarrow M \preccurlyeq N$ for all $M, N \in \mathfrak{M}$ and if

$$\operatorname{supp}\left(\frac{M}{L_i} B_i\right) \preccurlyeq M$$

for any $i \in I$ and any $M \in \mathfrak{R}_i$.

Polynomials that are not quasi-reduced are said to be *quasi-reducible*. In order to quasi-reduce $P \in \mathbb{K}[X_1,\ldots,X_n]$ with respect to $(B_i)_{i\in I}$ we may use Algorithm 2.1 below. This yields the relation

$$P \;=\; \sum_{i\in I} Q_i B_i + R,$$

such that R is quasi-reduced with respect to $(B_i)_{i\in I}$. We call $((Q_i)_{i\in I}, R)$ the *extended quasi-reduction* of P with respect to $(B_i)_{i\in I}$.

Algorithm 2.1

Input. P and a finite family $(B_i)_{i\in I}$ with a selection strategy σ.

Output. The extended quasi-reduction $((Q_i)_{i\in I}, R)$ of P by $(B_i)_{i\in I}$.

1. Initialize $R := P$ and $Q_i := 0$ for all $i \in I$.
2. While $\operatorname{supp} R \cap \mathfrak{R} = \emptyset$ do:
   a. Let M be the largest monomial of $\operatorname{supp} R \cap \mathfrak{R}$;
   b. Replace $Q_{\sigma(M)}$ by $Q_{\sigma(M)} + \frac{R_M M}{T_{\sigma(M)}}$;
   c. Replace R by $R - \frac{R_M M}{T_{\sigma(M)}} B_{\sigma(M)}$.
3. Return $((Q_i)_{i\in I}, R)$.

By construction, we have

$$\operatorname{supp}(Q_i L_i) \;\subseteq\; \mathfrak{R}_i \tag{2.1}$$

for all $i \in I$.

2.5. Relaxed quasi-reduction

The main contribution of [9] is a relaxed algorithm for the computation of extended reductions. This algorithm generalizes to our setting as follows and provides an alternative for Algorithm 2.1 with a better computational complexity.

For each $i \in \{1,\ldots,|I|+1\}$, let e_i be the i-th canonical basis vector

$$e_i := \big(0, \overset{(i-1)\times}{\ldots}, 0, 1, 0, \ldots, 0\big) \;\in\; \{0,1\}^{|I|+1}.$$

We first define the operator Φ on monomials:

$$\begin{aligned} \Phi\colon\quad \mathfrak{M} &\longrightarrow \mathbb{K}[X_1,\ldots,X_n]^{|I|+1} \\ M &\longmapsto \frac{M}{T_{\sigma(M)}} e_{\sigma(M)}, \text{ if } M \in \mathfrak{R} \\ M &\longmapsto M e_{|I|+1} \text{ otherwise.} \end{aligned}$$

By linearity we next extend Φ to $\mathbb{K}[X_1,\ldots,X_n]$:

$$\Phi\left(\sum_{M\in\mathfrak{M}} P_M M\right) := \sum_{M\in\mathfrak{M}} P_M \Phi(M).$$

Let $B_i^* := B_i - T_i$. By construction, we have

$$((Q_i)_{i\in I}, R) \;=\; \Phi\left(P - \sum_{i\in I} Q_i B_i^*\right),$$

which allows us to regard $((Q_i)_{i\in I}, R)$ as a fixed point of the operator

$$((Q_i)_{i\in I}, R) \;\longmapsto\; \Phi\left(P - \sum_{i\in I} Q_i B_i^*\right).$$

This operator is "recursive" in the sense of [9, section 4.2] with respect to the ordering $\prec$. Consequently $((Q_i)_{i\in I}, R)$ can be computed efficiently using relaxed power series evaluation. The complexity of this relaxed quasi-reduction is stated in Theorem 4.4 below for the specific ordering used by our multi-point evaluation algorithm. This definition and study of this ordering is the purpose of the next section.

3. HETEROGENEOUS ORDERINGS

3.1. Weighted degree orderings

An *admissible weight* is an n-tuple $\boldsymbol{w} = (w_1, \ldots, w_n) \in (\mathbb{R}^>)^n$ with $w_1 \cdots w_n = 1$. Given a monomial $X_1^{e_1} \cdots X_n^{e_n}$, we define its $\boldsymbol{w}$*-degree* by

$$\deg_{\boldsymbol{w}} X_1^{e_1} \cdots X_n^{e_n} \;=\; w_1 e_1 + \cdots + w_n e_n.$$

For a non-zero polynomial $P = \sum_{M\in\mathfrak{M}} P_M M \in \mathbb{K}[X_1, \ldots, X_n]$, we define its $\boldsymbol{w}$*-degree* by

$$\deg_{\boldsymbol{w}} P \;=\; \max\{\deg_{\boldsymbol{w}} M : P_M \neq 0\}.$$

We also define the ordering $\prec_{\boldsymbol{w}}$ on $\mathfrak{M}$ by

$$M \prec_{\boldsymbol{w}} N \iff (\deg_{\boldsymbol{w}} M < \deg_{\boldsymbol{w}} N) \vee (\deg_{\boldsymbol{w}} M = \deg_{\boldsymbol{w}} N \wedge M \prec_{\text{lex}} N).$$

It is easy to check that $\prec_{\boldsymbol{w}}$ is an admissible ordering.

3.2. Simplest elements of ideals

Consider a zero-dimensional ideal $\mathfrak{I}$ of $\mathbb{K}[X_1, \ldots, X_n]$ and let

$$d \;:=\; \dim_{\mathbb{K}} \mathbb{K}[X_1, \ldots, X_n]/\mathfrak{I}.$$

Given an admissible weight $\boldsymbol{w}$, there exists a unique non-zero polynomial $B_{\boldsymbol{w}}$ in the reduced Gröbner basis of $\mathfrak{I}$ whose leading monomial is minimal for $\prec_{\boldsymbol{w}}$ and whose leading coefficient is one. We call $B_{\boldsymbol{w}}$ the $\boldsymbol{w}$*-simplest element* of $\mathfrak{I}$. Note that there are at most d monomials below the leading monomial of $B_{\boldsymbol{w}}$ for $\prec_{\boldsymbol{w}}$.

PROPOSITION 3.1. *Let*

$$\delta \;:=\; \sqrt[n]{n!\,(d+1)}$$

and consider the set $\mathcal{S}_{\boldsymbol{w},\delta}$ of monomials $M \in \mathfrak{M}$ with $\deg_{\boldsymbol{w}} M \leqslant \delta$. Then $|\mathcal{S}_{\boldsymbol{w},\delta}| > d$.

Proof. Let $\mathcal{E}_{\boldsymbol{w},\delta} \subseteq \mathbb{N}^n$ be the set of exponents of $\mathcal{S}_{\boldsymbol{w},\delta}$, so we have

$$\mathcal{S}_{\boldsymbol{w},\delta} \;=\; \{X_1^{e_1} \cdots X_n^{e_n} : (e_1, \ldots, e_n) \in \mathcal{E}_{\boldsymbol{w},\delta}\}.$$

The set $\mathcal{E}_{\boldsymbol{w},\delta} + [0,1]^n$ in particular contains the simplex

$$\mathcal{T} \;:=\; \{(e_1, \ldots, e_n) \in (\mathbb{R}^{\geqslant})^n : w_1 e_1 + \cdots + w_n e_n \leqslant \delta\},$$

whose volume is given by

$$\operatorname{vol} \mathcal{T} = \frac{\delta^n}{n!\, w_1 \cdots w_n} \;=\; \frac{\delta^n}{n!} \;=\; d+1.$$

Consequently, $|\mathcal{S}_{\boldsymbol{w},d}| = |\mathcal{E}_{\boldsymbol{w},d}| = \operatorname{vol}(\mathcal{E}_{\boldsymbol{w},\delta} + [0,1]^n) \geqslant \operatorname{vol} \mathcal{T} = d+1$. □

COROLLARY 3.2. *For each $i \in \{1, \ldots, n\}$, we have*

$$\deg_{X_i} B_{\boldsymbol{w}} \;\leqslant\; \frac{\delta}{w_i}.$$

Proof. We say that $\mathcal{J} \subseteq \mathfrak{M}$ is an *initial segment* $\mathcal{J}$ for $\prec_w$ if $N \prec_w M \Rightarrow N \in \mathcal{J}$ for all $M \in \mathfrak{I}$ and $N \in \mathfrak{M}$. By definition, the set $\delta_{w,\delta}$ is an initial segment for $\prec_w$ and it contains at least $d+1$ monomials. Consequently, the set of linear combinations of monomials in $\delta_{w,\delta}$ contains at least one non-zero element of $\mathfrak{I}$ and therefore B_w. Now consider a monomial $X_1^{e_1} \cdots X_n^{e_n}$ in $\operatorname{supp} B_w \cap \delta_{w,\delta}$. Then $w_i e_i \leqslant \deg_w X_1^{e_1} \cdots X_n^{e_n} \leqslant \delta$ for $i = 1, \ldots, n$, whence $e_i \leqslant \delta / w_i$. □

3.3. Heterogeneous orderings

Now consider a finite non-empty set $\boldsymbol{W}$ of admissible weights. Given a monomial $M \in \mathfrak{M}$, we define its $\boldsymbol{W}$-*degree* $\deg_{\boldsymbol{W}} M$ by

$$\deg_{\boldsymbol{W}} M \;:=\; \min_{w \in \boldsymbol{W}} \deg_w M.$$

For a non-zero polynomial $P = \sum_{M \in \mathfrak{M}} P_M M \in \mathbb{K}[X_1, \ldots, X_n]$, we define its $\boldsymbol{W}$-*degree* by

$$\deg_{\boldsymbol{W}} P \;:=\; \max \{\deg_{\boldsymbol{W}} M : P_M \neq 0\}.$$

We define the *heterogeneous ordering* $\prec_{\boldsymbol{W}}$ by

$$M \prec_{\boldsymbol{W}} N \iff (\deg_{\boldsymbol{W}} M < \deg_{\boldsymbol{W}} N) \vee (\deg_{\boldsymbol{W}} M = \deg_{\boldsymbol{W}} N \wedge M \prec_{\text{lex}} N).$$

LEMMA 3.3. *The ordering $\prec_{\boldsymbol{W}}$ verifies $M \mid N \Longrightarrow M \prec_{\boldsymbol{W}} N$ for all different $M, N \in \mathfrak{M}$.*

Proof. For all $w \in \boldsymbol{W}$ we have $\deg_w M < \deg_w N$, whence $\deg_{\boldsymbol{W}} M < \deg_{\boldsymbol{W}} N$. □

Example 3.4. Consider $n = 2$, $\boldsymbol{W} = \{(1,1), (2, 1/2)\}$. With $M = X_2^2$ and $N = X_1$ we have

$$\begin{aligned} \deg_{\boldsymbol{W}} M &= \min(2,1) = 1 \\ \deg_{\boldsymbol{W}} N &= \min(1,2) = 1 \\ \deg_{\boldsymbol{W}}(X_1 M) &= \min(3,3) = 3 \\ \deg_{\boldsymbol{W}}(X_1 N) &= \min(2,4) = 2. \end{aligned}$$

Therefore $\deg_{\boldsymbol{W}} M \leqslant \deg_{\boldsymbol{W}} N$ but $\deg_{\boldsymbol{W}}(X_1 M) \not\leqslant \deg_{\boldsymbol{W}}(X_1 N)$, which shows that the ordering $\prec_{\boldsymbol{W}}$ is not admissible.

Given $\boldsymbol{w} \in \boldsymbol{W}$, its associated *cone* $\mathfrak{M}_{\boldsymbol{w}} \subseteq \mathfrak{M}$ is defined by

$$\mathfrak{M}_{\boldsymbol{w}} \;:=\; \{M \in \mathfrak{M} : \deg_{\boldsymbol{W}} M = \deg_{\boldsymbol{w}} M\}.$$

By construction, we have

$$\mathfrak{M} \;=\; \bigcup_{w \in \boldsymbol{W}} \mathfrak{M}_w,$$

but this union is not necessarily disjoint. Given $X_1^{e_1} \cdots X_n^{e_n} \in \mathfrak{M}_{\boldsymbol{w}}$, we note that $X_1^{\lambda e_1} \cdots X_n^{\lambda e_n} \in \mathfrak{M}_{\boldsymbol{w}}$ for any $\lambda \in \mathbb{Q}^>$ with $\lambda e_1, \ldots, \lambda e_n \in \mathbb{N}$; this explains the terminology "cone".

3.4. Quasi-reduction

Let $\mathfrak{I}$ again be a zero-dimensional ideal of $\mathbb{K}[X_1, \ldots, X_n]$ and let $d := \dim_{\mathbb{K}} \mathbb{K}[X_1, \ldots, X_n] / \mathfrak{I}$. Let $\boldsymbol{W}$ be a finite non-empty set of weights that will play the role of the index set I from section 2.4. For $\boldsymbol{w} \in \boldsymbol{W}$, let $B_{\boldsymbol{w}}$ be the $\boldsymbol{w}$-simplest element of $\mathfrak{I}$ and let

$$L_{\boldsymbol{w}} \;:=\; \operatorname{lm}_{\prec_{\boldsymbol{w}}}(B_{\boldsymbol{w}}).$$

We also define

$$\Pi \;:=\; X_1 \cdots X_n.$$

We endow $\boldsymbol{W}$ with the lexicographical ordering $<_{\text{lex}}$ and fix the following selection strategy: for increasing $\boldsymbol{w} \in \boldsymbol{W}$, we set

$$\mathfrak{R}_{\boldsymbol{w}} \;:=\; \{M \in L_{\boldsymbol{w}}\mathfrak{M} : \Pi M \in \mathfrak{M}_{\boldsymbol{w}}\} \setminus \left(\bigcup_{\boldsymbol{w}' <_{\text{lex}} \boldsymbol{w}} \mathfrak{R}_{\boldsymbol{w}'} \right).$$

Now consider the total ordering $\prec^*_{\boldsymbol{W}}$ on $\mathfrak{M}$ that is defined by

$$M \prec^*_{\boldsymbol{W}} N \iff \Pi M \prec_{\boldsymbol{W}} \Pi N,$$

for all $M, N \in \mathfrak{M}$. We call $\prec^*_{\boldsymbol{W}}$ a *shifted heterogeneous ordering*. The shift of all exponents by one (through multiplication by Π) is motivated by the fact that the product of the partial degrees of a monomial in $\Pi\mathfrak{M}$ never vanishes. This will be important in the next section in order to establish certain degree bounds.

PROPOSITION 3.5. *The ordering $\prec^*_{\boldsymbol{W}}$ is quasi-admissible.*

Proof. Consider $\boldsymbol{w} \in \boldsymbol{W}$ and $M \in \mathfrak{R}_{\boldsymbol{w}}$, so that

$$\deg_{\boldsymbol{W}} \Pi M \;=\; \deg_{\boldsymbol{w}} \Pi M.$$

Given $N \in \operatorname{supp} B_{\boldsymbol{w}}$, we need to prove that

$$U \;:=\; \frac{M}{L_{\boldsymbol{w}}} N \;\preccurlyeq^*_{\boldsymbol{W}}\; M.$$

Now $L_{\boldsymbol{w}} = \operatorname{lm}_{\prec_{\boldsymbol{w}}}(B_{\boldsymbol{w}})$ implies $N \preccurlyeq_{\boldsymbol{w}} L_{\boldsymbol{w}}$, whence $\Pi U \preccurlyeq_{\boldsymbol{w}} \Pi M$. In particular, $\deg_{\boldsymbol{w}} \Pi U \leqslant \deg_{\boldsymbol{w}} \Pi M$ and $\Pi U \preccurlyeq_{\text{lex}} \Pi M$ whenever $\deg_{\boldsymbol{w}} \Pi U = \deg_{\boldsymbol{w}} \Pi M$.

If $\deg_{\boldsymbol{W}} \Pi U = \deg_{\boldsymbol{w}} \Pi U$, then it follows that $\deg_{\boldsymbol{W}} \Pi U < \deg_{\boldsymbol{W}} \Pi M$ or $\deg_{\boldsymbol{W}} \Pi U = \deg_{\boldsymbol{W}} \Pi M$ and $\Pi U \preccurlyeq_{\text{lex}} \Pi M$, whence $\Pi U \preccurlyeq_{\boldsymbol{W}} \Pi M$ and $U \preccurlyeq^*_{\boldsymbol{W}} M$. Otherwise,

$$\deg_{\boldsymbol{W}} \Pi U \;=\; \deg_{\boldsymbol{w}'} \Pi U \;<\; \deg_{\boldsymbol{w}} \Pi U \;\leqslant\; \deg_{\boldsymbol{w}} \Pi M \;=\; \deg_{\boldsymbol{W}} \Pi M$$

for some $\boldsymbol{w}' \in \boldsymbol{W} \setminus \{\boldsymbol{w}\}$, whence $\Pi U \prec_{\boldsymbol{W}} \Pi M$ and $U \prec^*_{\boldsymbol{W}} M$. □

4. ON THE COMPLEXITY OF QUASI-REDUCTION

In this section we instantiate the weight family $\boldsymbol{W}$ considered in the previous section, and analyze the complexity of the corresponding quasi-reduction.

4.1. Selecting the weights

Consider a zero-dimensional ideal $\mathfrak{I}$ of $\mathbb{K}[X_1, \ldots, X_n]$ and let $d := \dim_{\mathbb{K}} \mathbb{K}[X_1, \ldots, X_n]/\mathfrak{I}$. In order to devise an efficient algorithm to quasi-reduce polynomials with respect to $\mathfrak{I}$, our next task is to specify a suitable set of weights $\boldsymbol{W}$. We take $\boldsymbol{W} = \boldsymbol{\Omega}_D$, with

$$\boldsymbol{\Omega}_D \;:=\; \{(2^{e_1}, \ldots, 2^{e_n}) : (e_1, \ldots, e_n) \in \mathbb{Z}^n, e_1 + \cdots + e_n = 0, 2^{|e_1|} \leqslant D, \ldots, 2^{|e_n|} \leqslant D\}$$

and where $D \geqslant d$ depends on the degree of the polynomials that we wish to reduce.

LEMMA 4.1. *We have*

$$|\boldsymbol{\Omega}_D| \;\leqslant\; (2\log_2 D + 1)^{n-1}.$$

Proof. Consider $e_1,\dots,e_n\in\mathbb{Z}$ with $e_1+\cdots+e_n=0$ and $2^{|e_1|}\leqslant D,\dots,2^{|e_n|}\leqslant D$. We have $e_i\in\{-\lfloor\log_2 D\rfloor,\dots,\lfloor\log_2 D\rfloor\}$ for $i=1,\dots,n-1$ and e_n is determined uniquely in terms of $e_1,\dots,e_{n-1}$ by $e_n=-(e_1+\cdots+e_{n-1})$. There are at most $(2\lfloor\log_2 D\rfloor+1)^{n-1}$ ways to pick $e_1,\dots,e_n$ in this manner. □

Assume that $\boldsymbol{W}=\boldsymbol{\Omega}_D$ for some $D\geqslant\max(d,4)$. We associate a selection procedure and a quasi-admissible ordering to $\boldsymbol{W}$ as in section 3.4.

LEMMA 4.2. *Let $\boldsymbol{w}=(w_1,\dots,w_n)\in\boldsymbol{W}$ and $M=X_1^{e_1}\cdots X_n^{e_n}\in\mathfrak{M}$ with $\Pi M\in\mathfrak{M}_{\boldsymbol{w}}$ and $(e_1+1)\cdots(e_n+1)\leqslant D$. Then for all $i,j\in\{1,\dots,n\}$, we have*

$$w_i(e_i+1)\ \leqslant\ 2w_j(e_j+1).$$

Proof. Assume the contrary and let i be the index for which $w_i(e_i+1)$ is maximal. Similarly, let j be the index for which $w_j(e_j+1)$ is minimal, so that $w_i(e_i+1)>2w_j(e_j+1)$. Since

$$[w_1(e_1+1)]\cdots[w_n(e_n+1)]=(e_1+1)\cdots(e_n+1)\ \leqslant\ D,$$

we have $w_j(e_j+1)\leqslant\sqrt[n]{D}$. In particular, since $D\geqslant 4$ we have $w_j\leqslant\sqrt[n]{D}\leqslant D/2$.

If $w_i<1$, then there must exist at least one index $k\neq i$ with $w_k\geqslant 2$ and

$$w_i(e_i+1)\ \geqslant\ w_k(e_k+1)\ \geqslant\ 2.$$

Since $e_i+1\leqslant D$, this yields $w_i\geqslant 2/D$. If $w_i\geqslant 1$ then $1/D\leqslant w_i/2\leqslant D$, since $D\geqslant 2$. In both cases we consider the weight $\boldsymbol{w}'=(w_1',\dots,w_n')$ with $w_i'=w_i/2$, $w_j'=2w_j$, and $w_k'=w_k$ for all $k\in\{1,\dots,n\}\setminus\{i,j\}$. By what precedes, we have $\boldsymbol{w}'\in\boldsymbol{\Omega}_D$.

We now verify that

$$\begin{aligned}\deg_{\boldsymbol{w}'}\Pi M-\deg_{\boldsymbol{w}}\Pi M\ &=\ (w_i'-w_i)(e_i+1)+(w_j'-w_j)(e_j+1)\\ &=\ \tfrac{1}{2}(2w_j(e_j+1)-w_i(e_i+1))\\ &<\ 0.\end{aligned}$$

This contradicts our assumption that $\Pi M\in\mathfrak{M}_{\boldsymbol{w}}$, i.e. $\deg_{\boldsymbol{w}}\Pi M=\deg_{\boldsymbol{W}}\Pi M$. □

COROLLARY 4.3. *Let $k\in\{1,\dots,n\}$. With the notations from Lemma 4.2 and*

$$\bar{e}\ :=\ \sqrt[n]{(e_1+1)\cdots(e_n+1)}-1,$$

we have

$$\frac{\bar{e}+1}{2w_k}\ \leqslant\ e_k+1\ \leqslant\ \frac{2(\bar{e}+1)}{w_k}\ \leqslant\ \frac{2\sqrt[n]{D}}{w_k}.$$

Proof. Lemma 4.2 implies

$$(\bar{e}+1)^n\ =\ [w_1(e_1+1)]\cdots[w_n(e_n+1)]\ \leqslant\ (2w_k(e_k+1))^n,$$

but also

$$(w_k(e_k+1))^n\ \leqslant\ [2w_1(e_1+1)]\cdots[2w_n(e_n+1)]\ =\ (2(\bar{e}+1))^n.$$

The result follows by extracting n-th roots. □

4.2. Complexity of quasi-reduction

We define $\mathbb{K}[X_1,\dots,X_n]_D$ to be the set of polynomials $P\in\mathbb{K}[X_1,\dots,X_n]$ such that $(e_1+1)\cdots(e_n+1)\leqslant D$ for all $X_1^{e_1}\cdots X_n^{e_n}\in\operatorname{supp}P$. Given $P\in\mathbb{K}[X_1,\dots,X_n]_D$, let us now examine the complexity of extended quasi-reduction.

THEOREM 4.4. *Given $P \in \mathbb{K}[X_1, \ldots, X_n]_D$ with $D \geqslant d$, we may compute its extended quasi-reduction*

$$P = \sum_{w \in W} Q_w B_w + R \tag{4.1}$$

using

$$O(\mathsf{SM}(D)\,(\log D)^n)$$

operations in $\mathbb{K}$ (for any fixed value of n). In addition, for all $w \in W$, we have

$$\prod_{i=1}^{n} (\deg_{X_i}(Q_w B_w) + 1) \leqslant 2^{n+1} n!\,(D+1).$$

Proof. We solve (4.1) using the relaxed approach from [9], recalled in section 2. However, contrary to the situation in [9], each individual product $Q_w B_w$ is computed in a relaxed manner with respect to a different ordering (namely, $\prec_w$). More precisely, for each individual product $Q_w B_w$, we recall from (2.1) that

$$L_w \operatorname{supp} Q_w \subseteq \mathfrak{R}_w.$$

So we can actually compute $Q_w B_w$ using the relaxed product algorithm from Theorem 2.4 with respect to the ordering $\prec_w$. Here we note that the supports of Q_w and B_w are contained in dense blocks of similar proportions: for $i = 1, \ldots, n$, Corollary 3.2 yields

$$\deg_{X_i} B_w \leqslant \frac{\sqrt[n]{n!\,(d+1)}}{w_i},$$

whereas Corollary 4.3 implies

$$\deg_{X_i} Q_w + 1 \leqslant \frac{2\sqrt[n]{D}}{w_i}.$$

Indeed, for any $M = X_1^{e_1} \cdots X_n^{e_n} \in \operatorname{supp} Q_w$, we have $L_w M \in \mathfrak{R}_w$, whence $\Pi M \in \mathfrak{M}_w$ and $e_i + 1 \leqslant 2\sqrt[n]{D}/w_i$. Let $\mathcal{R}_w$ be the set of monomials $X_1^{k_1} \cdots X_n^{k_n}$ with $k_i \leqslant \deg_{X_i}(B_w Q_w)$ for $i = 1, \ldots, n$, so that $\operatorname{supp}(B_w Q_w) \subseteq \mathcal{R}_w$ and

$$|\mathcal{R}_w| \leqslant \prod_{i=1}^{n} (\deg_{X_i}(B_w Q_w) + 1).$$

Using $w_1 \cdots w_n = 1$, we deduce that

$$\begin{aligned}
\prod_{i=1}^{n} (\deg_{X_i}(B_w Q_w) + 1) &= \prod_{i=1}^{n} (\deg_{X_i} B_w + (\deg_{X_i} Q_w + 1)) \\
&\leqslant \prod_{i=1}^{n} \left(\frac{\sqrt[n]{n!\,(d+1)}}{w_i} + \frac{2\sqrt[n]{D}}{w_i} \right) \\
&= \prod_{i=1}^{n} \left(\sqrt[n]{n!\,(d+1)} + 2\sqrt[n]{D} \right) \\
&\leqslant \prod_{i=1}^{n} 2\sqrt[n]{\max(n!, 2^n)\,(D+1)} \\
&\leqslant 2^n \max(n!, 2^n)\,(D+1) \\
&\leqslant 2^{n+1} n!\,(D+1).
\end{aligned}$$

It particular, as a side remark, we note that the dense multiplication of Q_w and B_w can be done using $\mathsf{M}(|\mathcal{R}_w|) = O(\mathsf{M}(D))$ operations in $\mathbb{K}$.

As to the relaxed product of $Q_{\boldsymbol{w}}$ and $B_{\boldsymbol{w}}$, we first note that $\prec_{\boldsymbol{w}} = \prec_\lambda$, with the notation from Example 2.3, where $(2^{a_1}, \ldots, 2^{a_n}) = 2^b\,(w_1, \ldots, w_n)$ with $a_1, \ldots, a_n \in \mathbb{N}$ and $2^b \leqslant D$. Consequently,

$$\begin{aligned}\vartheta_{\prec_{\boldsymbol{w}},1}(\mathcal{R}_{\boldsymbol{w}}) &= \sum_{1\leqslant i\leqslant n} 2^{a_i} O\left(\frac{\sqrt[n]{D}}{w_i}\right) = O(D^{1+1/n})\\ \vartheta_{\prec_{\boldsymbol{w}},i}(\mathcal{R}_{\boldsymbol{w}}) &= O\left(\frac{\sqrt[n]{D}}{w_i}\right), \qquad i = 2, \ldots, n\end{aligned}$$

and

$$\vartheta_{\prec_{\boldsymbol{w}}} = O\left(D^{1+1/n} \prod_{i=2}^{n} \frac{\sqrt[n]{D}}{w_i}\right) = O\left(w_1 D^2 \prod_{i=1}^{n} \frac{1}{w_i}\right) = O(D^3).$$

By Theorem 2.4, we may thus compute the relaxed product of $Q_{\boldsymbol{w}}$ and $B_{\boldsymbol{w}}$ in time

$$O(\mathsf{SM}(|\mathcal{R}_{\boldsymbol{w}}|) \log(\vartheta_{\prec_{\boldsymbol{w}}}(\mathcal{R}_{\boldsymbol{w}}))) = O(\mathsf{SM}(D) \log D).$$

We need to do $|\boldsymbol{W}|$ such relaxed multiplications. Now $|\boldsymbol{W}| = O((\log D)^{n-1})$ by Lemma 4.1, so the complete computation takes $O(\mathsf{SM}(D)\,(\log D)^n)$ operations in $\mathbb{K}$. □

4.3. A degree bound for reduced polynomials

We finish this section with a bound for the size of reduced polynomials.

PROPOSITION 4.5. *Consider a monomial $M = X_1^{e_1} \cdots X_n^{e_n}$ with $e_i \geqslant 1$ for $i = 1, \ldots, n$. If M is reduced with respect to $(B_{\boldsymbol{w}})_{\boldsymbol{w}\in\boldsymbol{W}}$, then*

$$(e_1 + 1) \cdots (e_n + 1) \leqslant 4^n n!\,(d + 1).$$

Proof. Assume for contradiction that there exists an $M = X_1^{e_1} \cdots X_n^{e_n} \in (\Pi\mathfrak{M}) \setminus \mathfrak{R}$ with

$$(e_1 + 1) \cdots (e_n + 1) > 4^n n!\,(d + 1).$$

Setting

$$\bar{e} := \sqrt[n]{(e_1 + 1) \cdots (e_n + 1)} - 1,$$

our assumption can be rewritten as

$$\bar{e} + 1 > 4\sqrt[n]{n!\,(d + 1)}.$$

Let $\boldsymbol{w} \in \boldsymbol{W}$ be such that

$$\deg_{\boldsymbol{W}} \Pi M = \deg_{\boldsymbol{w}} \Pi M.$$

For $i = 1, \ldots, n$, Corollary 4.3 implies

$$\tfrac{1}{2}(\bar{e} + 1) \leqslant w_i\,(e_i + 1) \leqslant 2\,(\bar{e} + 1).$$

Using Corollary 3.2 and our assumption that $e_i \geqslant 1$, this yields

$$\deg_{X_i} B_{\boldsymbol{w}} \leqslant \frac{\sqrt[n]{n!\,(d+1)}}{w_i} \leqslant \frac{2\,(e_i+1)\sqrt[n]{n!\,(d+1)}}{\bar{e}+1} \leqslant \frac{e_i+1}{2} \leqslant e_i.$$

This shows that $L_{\boldsymbol{w}}$ divides M. In combination with our assumption that $\deg_{\boldsymbol{W}} \Pi M = \deg_{\boldsymbol{w}} \Pi M$, this means that $M \in \mathfrak{R}_{\boldsymbol{w}}$, a contradiction. □

5. REDUCTION OF THE BORDER

One limitation of Proposition 4.5 is that it does not apply to monomials $X_1^{e_1} \cdots X_n^{e_n}$ such that $e_i = 0$ for some index $i \in \{1,\ldots,n\}$. In this section, we show how to treat such lower dimensional "border" monomials by adapting our reduction process. For any fixed subset $S \subseteq \{X_1, \ldots, X_n\}$, we will use the fact that the reduction process from the previous subsections can be applied to polynomials in the variables from S and the ideal $\mathfrak{I} \cap \mathbb{K}[S]$. We next combine the reduction processes for all such subsets $S \subseteq \{X_1, \ldots, X_n\}$.

5.1. Generalized weights

Given a finite subset $S \subseteq \{1, \ldots, n\}$, we write

$$\begin{aligned}
\mathfrak{M}_{|S} &:= \{X_1^{e_1} \cdots X_n^{e_n} \in \mathfrak{M} : i \notin S \Longrightarrow e_i = 0\} \\
\mathbb{K}[X_1, \ldots, X_n]_{|S} &:= \{P \in \mathbb{K}[X_1, \ldots, X_n] : \operatorname{supp} P \subseteq \mathfrak{M}_{|S}\} \\
\mathfrak{I}_{|S} &:= \mathfrak{I} \cap \mathbb{K}[X_1, \ldots, X_n]_{|S} \\
\Pi_{|S} &:= \textstyle\prod_{i \in S} X_i.
\end{aligned}$$

We note that $\mathfrak{I}_{|S}$ is an ideal of $\mathbb{K}[X_1, \ldots, X_n]_{|S}$ with $\dim_{\mathbb{K}} \mathbb{K}[X_1, \ldots, X_n]_{|S} / \mathfrak{I}_{|S} \leqslant d$. Given $P \in \mathbb{K}[X_1, \ldots, X_n]$, we define

$$A_P := \{i \in \{1, \ldots, n\} : \deg_{X_i} P > 0\}$$

to be the set of *active* coordinates of P. For any two subsets $S, T \subseteq \{1, \ldots, n\}$, we define

$$S < T \iff (\exists i \in T \setminus S,\ S \cap \{1, \ldots, i-1\} = T \cap \{1, \ldots, i-1\})$$

and note that this defines a total ordering on the set of subsets of $\{1, \ldots, n\}$.

A *generalized weight* is a tuple $\boldsymbol{w} \in (\mathbb{R}^> \cup \{\bot\})^n$ and we call

$$A_{\boldsymbol{w}} := \{i \in \{1, \ldots, n\} : w_i \neq \bot\}$$

its set of *active* coordinates. We say that $\boldsymbol{w}$ is *admissible* if $\prod_{i \in A_{\boldsymbol{w}}} w_i = 1$. For any monomial $M = X_1^{e_1} \cdots X_n^{e_n} \in \mathfrak{M}_{|A_{\boldsymbol{w}}}$, we define its $\boldsymbol{w}$-degree by

$$\deg_{\boldsymbol{w}} X_1^{e_1} \cdots X_n^{e_n} := \sum_{i \in A_{\boldsymbol{w}}} w_i e_i.$$

The $\boldsymbol{w}$-degree induces an ordering on $\mathfrak{M}_{|A_{\boldsymbol{w}}}$ by setting

$$M \prec_{\boldsymbol{w}} N \iff \deg_{\boldsymbol{w}} M < \deg_{\boldsymbol{w}} N$$

for all $M, N \in \mathfrak{M}_{|A_{\boldsymbol{w}}}$. We may naturally extend the notions of $\boldsymbol{w}$-degree, leading monomials, etc. to polynomials in $\mathbb{K}[X_1, \ldots, X_n]_{|A_{\boldsymbol{w}}}$. We also define the $\boldsymbol{w}$-simplest element of $\mathfrak{I}_{|A_{\boldsymbol{w}}}$ to be the unique monic polynomial $B_{\boldsymbol{w}} \in \mathfrak{I}_{|A_{\boldsymbol{w}}}$ whose leading monomial is minimal for $\prec_{\boldsymbol{w}}$.

Given a set $\boldsymbol{W}$ of generalized weights and $S \subseteq \{1, \ldots, n\}$, we define

$$\boldsymbol{W}_{|S} := \{\boldsymbol{w} \in \boldsymbol{W} : A_{\boldsymbol{w}} = S\}.$$

If $\boldsymbol{W}_{|S}$ is non-empty, then we define

$$\begin{aligned}
M \prec_{\boldsymbol{W}_{|S}} N &\iff (\deg_{\boldsymbol{W}_{|S}} M < \deg_{\boldsymbol{W}_{|S}} N) \vee (\deg_{\boldsymbol{W}_{|S}} M = \deg_{\boldsymbol{W}_{|S}} N \wedge M \prec_{\text{lex}} N) \\
M \prec^*_{\boldsymbol{W}_{|S}} N &\iff \Pi_{|S} M \prec_{\boldsymbol{W}_{|S}} \Pi_{|S} N,
\end{aligned}$$

for all $M, N \in \mathfrak{M}_{|S}$.

5.2. Quasi-reduction for generalized weights

We next extend the definitions from section 3.4 to the case when $\boldsymbol{W}$ is a set of generalized weights with $\boldsymbol{W}_{|S} \neq \emptyset$ for every subset $S \subseteq \{1,\ldots,n\}$. For each $\boldsymbol{w} \in \boldsymbol{W}$, we take $B_{\boldsymbol{w}}$ to be the $\boldsymbol{w}$-simplest element of $\mathfrak{I}_{|A_{\boldsymbol{w}}}$ and let

$$L_{\boldsymbol{w}} \;:=\; \operatorname{lm}_{\prec_{\boldsymbol{w}}}(B_{\boldsymbol{w}}).$$

We will sometimes write $B_{\boldsymbol{w},\mathfrak{I}}$ instead of $B_{\boldsymbol{w}}$ when we need to emphasize the dependence on $\mathfrak{I}$. After declaring that $\bot < \mathbb{R}^>$, the set $\boldsymbol{W}$ can still be endowed with the lexicographical ordering. For increasing $\boldsymbol{w} \in \boldsymbol{W}$, we now set

$$\mathfrak{R}_{\boldsymbol{w}} \;:=\; \{M \in L_{\boldsymbol{w}}\, \mathfrak{M}_{|A_{\boldsymbol{w}}} : \Pi_{|A_{\boldsymbol{w}}} M \in \mathfrak{M}_{\boldsymbol{w}}, A_M = A_{\boldsymbol{w}}\} \setminus \left(\bigcup_{\boldsymbol{w}' <_{\text{lex}} \boldsymbol{w}} \mathfrak{R}_{\boldsymbol{w}'} \right).$$

Sometimes, we will write $\mathfrak{R}_{\boldsymbol{w},\boldsymbol{W}}$ instead of $\mathfrak{R}_{\boldsymbol{w}}$ when we need to emphasize the dependence on $\boldsymbol{W}$. Finally, we define our total ordering $\prec^*_{\boldsymbol{W}}$ on $\mathfrak{M}$ by

$$M \prec^*_{\boldsymbol{W}} N \;\Longleftrightarrow\; A_M < A_N \text{ or } (A_M = A_N \text{ and } M \prec^*_{\boldsymbol{W}_{|A_M}} N)$$

for all $M, N \in \mathfrak{M}$.

PROPOSITION 5.1. *The ordering $\prec^*_{\boldsymbol{W}}$ is quasi-admissible.*

Proof. It is straightforward to check that $\prec^*_{\boldsymbol{W}}$ is indeed a total ordering such that

$$M \,|\, N \;\Longrightarrow\; M \prec^*_{\boldsymbol{W}} N$$

for all $M, N \in \mathfrak{M}$, by extending Lemma 3.3. Now consider $\boldsymbol{w} \in \boldsymbol{W}$ and

$$M \;\in\; \mathfrak{R}_{\boldsymbol{w}} \;\subseteq\; \mathbb{K}[X_1,\ldots,X_s]_{|A_{\boldsymbol{w}}}.$$

Given $N \in \operatorname{supp} B_{\boldsymbol{w}} \subseteq \mathfrak{M}_{|A_{\boldsymbol{w}}}$, Proposition 3.5 implies

$$\frac{M}{L_{\boldsymbol{w}}} N \;\preccurlyeq^*_{\boldsymbol{W}_{|A_{\boldsymbol{w}}}}\; M.$$

If $A_{(M/L_{\boldsymbol{w}})N} = A_M = A_{\boldsymbol{w}}$, then this yields

$$\frac{M}{L_{\boldsymbol{w}}} N \;\preccurlyeq^*_{\boldsymbol{W}}\; M.$$

Otherwise, we have $A_{(M/L_{\boldsymbol{w}})N} \subsetneq A_M = A_{\boldsymbol{w}}$, so $A_{(M/L_{\boldsymbol{w}})N} < A_M$, with the same conclusion. □

5.3. Complexity analysis

Given a general weight $\boldsymbol{w} \in (\mathbb{R}^>)^n$ and a subset $E \subseteq \{1,\ldots,n\}$ such that $w_i = 1$ for all $i \in E$, we define $\boldsymbol{w}_{\setminus E} \in (\mathbb{R}^> \cup \{\bot\})^n$ to be the generalized weight $\boldsymbol{w}'$ with $w'_i = w_i$ if $i \notin E$ and $w'_i = \bot$ if $i \in E$. If $\boldsymbol{w}$ is admissible, then we note that $\boldsymbol{w}_{\setminus E}$ is again admissible. Given $D \geqslant d$, we define

$$\boldsymbol{\Omega}^{\#}_D \;:=\; \{\boldsymbol{w}_{\setminus E} : \boldsymbol{w} \in \boldsymbol{\Omega}_D, E \subseteq \{1,\ldots,n\}, (\forall i \in E, w_i = 1)\}.$$

Note that Lemma 4.1 directly implies

$$|\boldsymbol{\Omega}^{\#}_D| \;\leqslant\; 2^n |\boldsymbol{\Omega}_D| \;\leqslant\; 2^n\, (2\log_2 D + 1)^{n-1} \;=\; O((\log D)^{n-1}).$$

The complexity bound from Theorem 4.4 also still holds in our new setting:

THEOREM 5.2. *Let $\boldsymbol{W} = \boldsymbol{\Omega}^{\#}_D$. Given $P \in \mathbb{K}[X_1,\ldots,X_n]_D$, we may compute an extended quasi-reduction (4.1) using*

$$O(\mathsf{SM}(D)\,(\log D)^n)$$

operations in $\mathbb{K}$. *In addition, for all* $w \in \boldsymbol{W}$, *we have*

$$\prod_{i=1}^{n} (\deg_{X_i} Q_w B_w + 1) \leqslant 2^{n+1} n! (D+1). \tag{5.1}$$

Proof. For each $w \in \boldsymbol{W}$, we have

$$L_w \operatorname{supp} Q_w \subseteq \mathfrak{R}_w \subseteq \mathfrak{R}_{w, W_{|A_w}}.$$

Using the same arguments as in the proof of Theorem 4.4, we obtain that the relaxed product $Q_w B_w$ can be computed in time $O(\mathsf{SM}(D) \log D)$. Since there are $|\boldsymbol{\Omega}^{\#}_D| = O((\log D)^{n-1})$ such products to be computed, the complexity bound follows. The inequality (5.1) is also obtained in a similar way as for Theorem 4.4. □

This time, we have the following unconditional version of Proposition 4.5.

PROPOSITION 5.3. *Let* $M = X_1^{e_1} \cdots X_n^{e_n}$ *be a reduced monomial with respect to* B. *Then*

$$(e_1+1) \cdots (e_n+1) \leqslant 4^n n! (d+1). \tag{5.2}$$

Proof. The monomial M is reduced with respect to B if and only if M is reduced with respect to $(B_w)_{w \in W_{|A_M}}$. By Proposition 4.5, this implies

$$(e_1+1) \cdots (e_n+1) \leqslant 4^{\tilde{n}} \tilde{n}! (\tilde{d}+1),$$

where $\tilde{n} := |A_M| \leqslant n$ and $\tilde{d} := \dim_{\mathbb{K}} \mathbb{K}[X_1, \ldots, X_n]_{|A_M} / \mathfrak{I}_{|A_M} \leqslant d$. □

6. MULTI-POINT EVALUATION

Let $\boldsymbol{\alpha} \in (\mathbb{K}^n)^d$ be a tuple of pairwise distinct points and consider the problem of fast multi-point evaluation of a polynomial $P \in \mathbb{K}[X_1, \ldots, X_n]_D$ at $\boldsymbol{\alpha}$. For simplicity of the exposition, it is convenient to restrict ourselves to the case when $d = 2^\ell$ is a power of two (without loss of generality, we may reduce to this case by adding extra points). We define

$$\mathfrak{I}_{\boldsymbol{\alpha}} := \{P \in \mathbb{K}[X_1, \ldots, X_n] : P(\boldsymbol{\alpha}) = \boldsymbol{0}\}$$

to be the vanishing ideal of $\boldsymbol{\alpha}$. We assume that we precomputed $(B_{w, \mathfrak{I}_{\boldsymbol{\alpha}}})_{w \in \boldsymbol{\Omega}^{\#}_D}$. For any $m \in \{1, 2, \ldots, 2^{\ell-1}\}$ and $i \in \{0, \ldots, d/m - 1\}$, we also assume that we precomputed $(B_{w, \mathfrak{I}_{\boldsymbol{\alpha}'}})_{w \in \boldsymbol{\Omega}^{\#}_{D'}}$, where $\boldsymbol{\alpha}' = (\boldsymbol{\alpha}_{im+1}, \ldots, \boldsymbol{\alpha}_{im+m}) \in (\mathbb{K}^n)^m$ and $D' = 4^n n! (2m+1)$.

We are now ready to state our main algorithm.

Algorithm 6.1

Input. $\boldsymbol{\alpha} \in (\mathbb{K}^n)^d$ with $d = 2^\ell$ and $P \in \mathbb{K}[X_1, \ldots, X_n]_D$.

Output. $P(\boldsymbol{\alpha})$.

Note. We assume the precomputations that are stated above.

1. If $\ell = 0$, then return the naive evaluation $P(\boldsymbol{\alpha})$.
2. Compute the quasi-reduction R of P w.r.t. $(B_{w, \mathfrak{I}_{\boldsymbol{\alpha}}})_{w \in \boldsymbol{\Omega}^{\#}_D}$.
3. Recursively apply the algorithm to $\boldsymbol{\alpha}_{1, n/2}$ and R.
4. Recursively apply the algorithm to $\boldsymbol{\alpha}_{n/2+1, n}$ and R.
5. Return the concatenations of the results of the recursive evaluations.

THEOREM 6.1. *Algorithm 6.1 is correct and runs in time*

$$O(\mathsf{SM}(D)\,(\log D)^n + \mathsf{SM}(d)\,(\log d)^{n+1}).$$

Proof. The algorithm is clearly correct if $\ell = 0$. For any recursive call of the algorithm with arguments $\boldsymbol{\alpha}' \in (\mathbb{K}^n)^m$ and P', Proposition 5.3 ensures that we indeed have $P \in \mathbb{K}[X_1, \ldots, X_n]_{D'}$ with $D' = 4^n\, n!\,(2\,m+1)$. The correctness of the general case easily follows from this.

As to the complexity bound, let us first assume that $D \leqslant 4^n\, n!\,(2\,d+1)$. Then the same condition is satisfied for all recursive calls. Now the computation of R takes

$$O(\mathsf{SM}(D)\,(\log D)^n) = O(\mathsf{SM}(d)\,(\log d)^n)$$

operations in $\mathbb{K}$, by Theorem 5.2. Hence, the total execution time $\mathsf{T}(d)$ satisfies

$$\mathsf{T}(d) \;\leqslant\; 2\,\mathsf{T}\left(\tfrac{d}{2}\right) + O(\mathsf{SM}(d)\,(\log d)^n).$$

By unrolling this recurrence inequality, it follows that $\mathsf{T}(d) = O(\mathsf{SM}(d)\,(\log d)^{n+1})$. If $D > 4^n\, n!\,(2\,d+1)$, then the computation of R at the top-level requires $O(\mathsf{SM}(D)\,(\log D)^n)$ operations in $\mathbb{K}$ and we have just shown that the complexity of all recursive computations is $O(\mathsf{SM}(d)\,(\log d)^{n+1})$. □

Proof of Theorem 1.1. The proof follows from Theorem 6.1 and Proposition 2.1, by analyzing the cardinalities of the supports involved in the quasi-reductions.

Let us revisit the quasi-reduction computed in step 2 of Algorithm 6.1. From (5.1) in Theorem 5.2 and (5.2) in Proposition 5.3, it follows that we may apply Proposition 2.1 for $\pi = 4^n\, n!\,(D+1)$. For the recursive quasi-reductions, we may use even smaller values for π. This justifies that we may indeed take $\mathsf{SM}(s) = O(\mathsf{M}(s) \log s)$ in Theorem 6.1. □

BIBLIOGRAPHY

[1] A. Borodin and R. T. Moenck. Fast modular transforms. *J. Comput. System Sci.*, 8:366–386, 1974.

[2] A. Bostan, G. Lecerf, and É. Schost. Tellegen's principle into practice. In *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, ISSAC '03, pages 37–44. New York, NY, USA, 2003. ACM.

[3] D. G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Inform.*, 28:693–701, 1991.

[4] C. M. Fiduccia. Polynomial evaluation via the division algorithm: the fast Fourier transform revisited. In *Proceedings of the Fourth Annual ACM Symposium on Theory of Computing*, STOC '72, pages 88–93. New York, NY, USA, 1972. ACM.

[5] D. Harvey and J. van der Hoeven. Faster polynomial multiplication over finite fields using cyclotomic coefficient rings. *J. Complexity*, 54:101404, 2019.

[6] D. Harvey and J. van der Hoeven. Polynomial multiplication over finite fields in time $O(n \log n)$. Technical Report, HAL, 2019. https://hal.archives-ouvertes.fr/hal-02070816, accepted for publication in J. ACM.

[7] J. van der Hoeven. Relax, but don't be too lazy. *J. Symbolic Comput.*, 34:479–542, 2002.

[8] J. van der Hoeven. Faster Chinese remaindering. Technical Report, HAL, 2016. https://hal.archives-ouvertes.fr/hal-01403810.

[9] J. van der Hoeven. On the complexity of multivariate polynomial division. In I. S. Kotsireas and E. Martínez-Moro, editors, *Applications of Computer Algebra. Kalamata, Greece, July 20–23, 2015*, volume 198 of *Springer Proceedings in Mathematics & Statistics*, pages 447–458. Cham, 2017. Springer International Publishing.

[10] J. van der Hoeven. *The Jolly Writer. Your Guide to GNU TeXmacs*. Scypress, 2020.

[11] J. van der Hoeven and G. Lecerf. On the bit-complexity of sparse polynomial multiplication. *J. Symbolic Comput.*, 50:227–254, 2013.

[12] J. van der Hoeven and G. Lecerf. Accelerated tower arithmetic. *J. Complexity*, 55:101402, 2019.
[13] J. van der Hoeven and G. Lecerf. Fast multivariate multi-point evaluation revisited. *J. Complexity*, 56:101405, 2020.
[14] J. van der Hoeven and G. Lecerf. Amortized bivariate multi-point evaluation. In M. Mezzarobba, editor, *Proceedings of the 2021 International Symposium on Symbolic and Algebraic Computation*, ISSAC '21, pages 179–185. New York, NY, USA, 2021. ACM.
[15] J. van der Hoeven and G. Lecerf. Fast amortized multi-point evaluation. *J. Complexity*, 67:101574, 2021.
[16] J. van der Hoeven and G. Lecerf. On the complexity exponent of polynomial system solving. *Found. Comput. Math.*, 21:1–57, 2021.
[17] J. Hopcroft and J. Musinski. Duality applied to the complexity of matrix multiplication and other bilinear forms. *SIAM J. Comput.*, 2(3):159–173, 1973.
[18] K. S. Kedlaya and C. Umans. Fast modular composition in any characteristic. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*, pages 146–155. Los Alamitos, CA, USA, 2008. IEEE Computer Society.
[19] K. S. Kedlaya and C. Umans. Fast polynomial factorization and modular composition. *SIAM J. Comput.*, 40(6):1767–1802, 2011.
[20] D. Le Brigand and J.-J. Risler. Algorithme de Brill–Noether et codes de Goppa. *Bulletin de la société mathématique de France*, 116(2):231–253, 1988.
[21] F. Le Gall and F. Urrutia. Improved rectangular matrix multiplication using powers of the Coppersmith–Winograd tensor. In A. Czumaj, editor, *Proceedings of the 2018 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1029–1046. Philadelphia, PA, USA, 2018. SIAM.
[22] R. T. Moenck and A. Borodin. Fast modular transforms via division. In *13th Annual Symposium on Switching and Automata Theory*, pages 90–96. USA, 1972. IEEE.
[23] V. Neiger, J. Rosenkilde, and G. Solomatov. Generic bivariate multi-point evaluation, interpolation and modular composition with precomputation. In A. Mantzaflaris, editor, *Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation*, ISSAC '20, pages 388–395. New York, NY, USA, 2020. ACM.
[24] M. Nüsken and M. Ziegler. Fast multipoint evaluation of bivariate polynomials. In S. Albers and T. Radzik, editors, *Algorithms – ESA 2004. 12th Annual European Symposium, Bergen, Norway, September 14-17, 2004*, volume 3221 of *Lect. Notes Comput. Sci.*, pages 544–555. Springer Berlin Heidelberg, 2004.
[25] L. Robbiano. Term orderings on the polynomial ring. In B. F. Caviness, editor, *EUROCAL '85. European Conference on Computer Algebra. Linz, Austria, April 1-3, 1985. Proceedings. Volume 2: Research Contributions*, volume 204 of *Lect. Notes Comput. Sci.*, pages 513–517. Springer-Verlag Berlin Heidelberg, 1985.
[26] D. S. Roche. What can (and can't) we do with sparse polynomials? In C. Arreche, editor, *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*, ISSAC '18, pages 25–30. New York, NY, USA, 2018. ACM.
[27] A. Schönhage. Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2. *Acta Inform.*, 7:395–398, 1977.
[28] A. Schönhage and V. Strassen. Schnelle Multiplikation großer Zahlen. *Computing*, 7:281–292, 1971.