Faster multi-point evaluation over any field[*][†]

Joris van der Hoeven[a], Grégoire Lecerf[b]

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

a. Email: vdhoeven@lix.polytechnique.fr
b. Email: lecerf@lix.polytechnique.fr

Preliminary version of November 8, 2024

The evaluation of a polynomial at several points is called the problem of multi-point evaluation. We design slightly new faster deterministic algorithms to solve this problem for an algebraic computational model. For this purpose, we analyze the precomputation costs of recent amortized evaluation algorithms, and then study the complexity of the problem as a function of the number of evaluation points.

Keywords: polynomial, multi-point evaluation, algorithm, complexity

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 $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N) \in (\mathbb{K}^n)^N$ of points, the computation of $P(\boldsymbol{\alpha}) = (P(\alpha_1), \ldots, P(\alpha_N)) \in \mathbb{K}^N$ is called the problem of *multi-point evaluation*. The converse problem is called *interpolation* and takes a candidate support of P as input.

These problems naturally occur in several areas of applied algebra. For instance in [16], we have shown that fast multi-point evaluation leads to fast polynomial system solving. Multi-point evaluation with a large number of variables also leads to fast modular composition [20]. The more specific bivariate case $n = 2$ appears for example in the computation of generator matrices of algebraic geometry error correcting codes [21].

The problem of multi-point evaluation is typically studied in the case when $N \asymp d^n$, where d is the total degree of P. One particularity of this paper is that, besides this classical case, we also study the complexity when N and d^n have different orders of magnitude. Especially in the case when $N \ll d^n$, our recent work [14, 17] on *amortized* multi-point evaluation (when the set of points is fixed) turns out to be very useful; this application was also one of our motivations for the present work.

[*]. Grégoire Lecerf has been supported by the French ANR-22-CE48-0016 NODE project. Joris van der Hoeven has been supported by an ERC-2023-ADG grant for the ODELIX project (number 101142171).

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Funded by the European Union

European Research Council

[†]. This article has been written using GNU TeXmacs [11].

1.1. Main results

In this paper, the number n of variables is always assumed to be fixed, so the constants hidden in the "O" of our complexity bounds depend on n. For complexity analyses, we will only consider algebraic complexity models like computation trees or RAM machines [5]. The time complexity then measures the number of arithmetic operations and zero-tests in $\mathbb{K}$. The *soft-Oh* notation $f(t) = \tilde{O}(g(t))$ is a shorthand for $f(t) = g(t)\,(\log(g(t)))^{O(1)}$; see [9, chapter 25, section 7] for technical details.

The constant ω will denote a real value between 2 and 3 such that two $m \times m$ matrices over a commutative ring can be multiplied with $O(m^{\omega})$ ring operations. The current best known bound is $\omega < 2.371552$ [30]. The constant ω_2 will be a real value between 3 and 4 such that the product of a $m \times m^2$ matrix by a $m^2 \times m$ matrix takes $O(m^{\omega_2})$ operations; one may take $\omega_2 < 3.250385$ [30].

The first line of results, presented in section 2, concerns the derandomization of the Nüsken–Ziegler algorithm [26]. When $n \geqslant 3$ our new deterministic bound coincides with their probabilistic one; see Theorem 2.6. For $n = 2$ our deterministic evaluation in degree d at $O(d^2)$ points costs $\tilde{O}(d^3)$ operations in $\mathbb{K}$, whereas the probabilistic version of [26] takes $\tilde{O}(d^{1+\omega_2/2})$, which tends to $\tilde{O}(d^{2.5})$ when ω_2 tends to the lower bound 3; see Theorems 2.7 and 2.8, and Remark 2.4.

In section 3 we turn to the amortized multi-point evaluation of multivariate polynomials. For such algorithms, the set of evaluation points is fixed, so we may use potentially expensive precomputations as a function of these points. A softly optimal algorithm for amortized multi-point evaluation was first given in [17]. In section 3 we provide a careful analysis of the cost of the precomputations. Building on algorithms from [23], it turns out that this can be done with a complexity exponent below the one of linear algebra: see Lemma 3.6. We next use this to derive our main complexity bounds for non-amortized multi-point evaluation: Theorems 3.9 and 3.13. The latter theorem slightly improves upon the Nüsken–Ziegler algorithm when $n = 3$ or $n \geqslant 7$; see Remark 3.16 for details. We also show that the evaluation at $O((d^n)^{(n-1)/(\omega(n-2)+1)})$ points can be performed in softly linear time, which again improves upon the Nüsken–Ziegler algorithm.

If $n = 2$, then Theorem 3.9 also improves on our deterministic version of the Nüsken–Ziegler algorithm from Theorem 2.7; see Remark 3.10. The comparison with the randomized version is given in Remark 3.11.

In order to design the above deterministic algorithms for any effective field $\mathbb{K}$, we frequently need to assume that the cardinality of $\mathbb{K}$ is sufficiently large or that we explicitly know an element of a sufficiently large multiplicative order. This subtlety only concerns finite fields and is usually addressed by computing over an algebraic extension of $\mathbb{K}$. Since we work over an arbitrary effective field in this paper, we need to deal with this extension issue in detail. In particular, we may not assume that the cardinality or characteristic of $\mathbb{K}$ are known. In Appendix A, we present a general way to address these issues. Our solution can be implemented with programming languages that support generic programming, such as C++, MATHEMAGIX [18], etc.

1.2. Related work

The general problem of multivariate multi-point evaluation is notoriously hard. If $\mathbb{K} = \mathbb{Q}$ or $\mathbb{K}$ is a field of finite characteristic, then theoretical algorithms due to Kedlaya and Umans [20] achieve a complexity exponent $1 + \epsilon$, where $\epsilon > 0$ is a constant that can be taken arbitrarily close to zero. Unfortunately, to our best knowledge, these algorithms do not seem suitable for practical purposes [13, Conclusion]. Recent advances in this vein are to be found in [2, 3].

The best previously known complexity bound for $n=2$ and for general fields and input is due to Nüsken and Ziegler [26]: the evaluation of P at $N=O(\deg_{x_1} P \deg_{x_2} P)$ points can be done using $O(\deg_{x_1} P\,(\deg_{x_2} P)^{\omega_2/2})$ operations in $\mathbb{K}$, assuming suitable coordinates. So, for P of total degree d, the cost is an expected number of $O(d^{\omega_2/2+1})$ operations in $\mathbb{K}$, that equals $O(d^3)$ without fast linear algebra, and that tends to $O(d^{2.5})$ when ω_2 tends to 3. We further mention [19] for efficient algorithms for special sets of points $\boldsymbol{\alpha}$.

Another recent result for any field $\mathbb{K}$, and for $n=2$, is due to Neiger, Salvy, Schost, and Villard [25]. Their algorithm is derived from their fast modular composition algorithm. Given f, g, and h in $\mathbb{K}[x]$ of degree $\leqslant D$, they show how to compute the polynomial $f\circ g \operatorname{rem} h$ by a probabilistic algorithm of Las Vegas type using an expected number of

$$\tilde{O}(D^{\kappa}),\text{ where } \kappa := 1+\frac{1}{\frac{1}{\omega-1}+\frac{2}{\omega_2-2}}$$

operations in $\mathbb{K}$; see [25, Theorem 1.1]. Then the bivariate evaluation problem reduces to modular composition in the following way. First, up to a sufficiently generic linear change of the coordinates, their exist $\chi\in\mathbb{K}[x]$ of degree N and $v\in\mathbb{K}[x]$ of degree $<N$ such that $\boldsymbol{\alpha}=\{(a,v(a)):\chi(a)=0\}$, so $P(\boldsymbol{\alpha})$ can easily be recovered from $P(x,v(x)) \operatorname{rem} \chi(x)$. In [25, section 10.3] it is shown that $P(x,v(x)) \operatorname{rem} \chi(x)$ can be computed using $\tilde{O}(N^{\kappa})$ operations in $\mathbb{K}$, whenever $\deg P=O(N^{1/2})$, $\mathbb{K}$ has characteristic 0, and the input is sufficiently generic. Without fast linear algebra, that is when $\omega=3$, one has $\kappa=5/3$. With the best known value for ω, one has $\kappa<1.42945$. If ω and ω_2 tend to their trivial lower bound 2 and 3, then κ tends to $4/3$.

In recent years, softly linear time has been achieved for multi-point evaluation and interpolation when $\boldsymbol{\alpha}$ is a fixed generic tuple of points [15, 24]. These algorithms are *amortized* in the sense that potentially expensive precomputations as a function of $\boldsymbol{\alpha}$ are allowed. When the dimension n is arbitrary but fixed, the amortized algorithms from [15] generalize the classical univariate "divide and conquer" approach, as presented for instance in [9, chapter 10]. The results in [24] restrict to the case $n=2$. They take into account the partial degrees of P and are based on changes of polynomial bases that are similar to the ones of [12, section 6.2].

The article [14] handles arbitrary (*i.e.* possibly non-generic) tuples of evaluation points $\boldsymbol{\alpha}$, while restricting to the amortized bivariate case $n=2$ and $\deg P=O(N^{1/2})$. New techniques for general dimensions n were presented in [17]. In the present paper we analyze the cost of the precomputations needed in [17]. This is key for our improved complexity bounds for non-amortized multi-point evaluation, especially when N is substantially smaller than $(\deg P)^n$.

2. THE NÜSKEN–ZIEGLER ALGORITHM

Throughout this paper, we assume the dimension n to be fixed. So the constants hidden in the $''O''$ of the complexity estimates will depend on n. We recall that $P(\boldsymbol{\alpha})$ denotes the tuple of values $(P(\alpha_1),\ldots,P(\alpha_N))$. The $\mathbb{K}$-vector space of the polynomials of total degree $<d$ in $\mathbb{K}[x]$ will be written $\mathbb{K}[x]_{<d}$. The cardinality of $\boldsymbol{\alpha}$ is written $|\boldsymbol{\alpha}|$.

We denote by $\mathsf{M}(d)$ the time that is needed to compute a product PQ of two polynomials $P, Q \in \mathbb{K}[x]$ of degree $<d$. We make the usual assumptions that $\mathsf{M}(d)/d$ is non-decreasing as a function of d and that $\mathsf{M}(kd) = O(k\,\mathsf{M}(d))$ whenever $k = O(d)$. Using a variant of the Schönhage–Strassen algorithm [6], 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 even take $\mathsf{M}(d) = O(d \log d\, 4^{\log^* d})$ [10].

2.1. Separating forms

A linear form $u \in \mathbb{K}[x_1, \ldots, x_n]$ is said to **separate** the points $\boldsymbol{\alpha}$ if it takes pairwise different values at pairwise different points of $\boldsymbol{\alpha}$. It will often be convenient to split $\boldsymbol{\alpha}$ into non-empty subsequences $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_L$. Then **joined** separating forms for each of the $\boldsymbol{\alpha}_i$ with $i = 1, \ldots, L$ may be computed as follows.

LEMMA 2.1. *Let $\mathcal{S}$ be a set of $> \binom{|\boldsymbol{\alpha}_1|}{2} + \cdots + \binom{|\boldsymbol{\alpha}_L|}{2}$ elements in $\mathbb{K}$. We can compute a joined separating form $x_1 + \lambda_2 x_2 + \cdots + \lambda_n x_n$ for $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_L$ with $(\lambda_2, \ldots, \lambda_n) \in \mathbb{K}^{n-1}$ using*

$$O(|\boldsymbol{\alpha}_1|^2 + \cdots + |\boldsymbol{\alpha}_L|^2)$$

operations in $\mathbb{K}$.

Proof. Recall that $n = O(1)$ is constant. We proceed by induction on n. If $n = 1$ then the proof is immediate. Assume that $n \geqslant 2$ and let

$$\begin{array}{rcl} \pi\colon \quad \mathbb{K}^n & \to & \mathbb{K}^{n-1} \\ (x_1, \ldots, x_n) & \mapsto & (x_2, \ldots, x_n). \end{array}$$

By the induction hypothesis, we may compute $\lambda_3', \ldots, \lambda_n' \in \mathbb{K}^{n-2}$ in time $O(|\boldsymbol{\alpha}_1|^2 + \cdots + |\boldsymbol{\alpha}_L|^2)$, such that $u := x_2 + \lambda_3' x_3 + \cdots + \lambda_n' x_n$ is a joined separating form for $\pi(\boldsymbol{\alpha}_1), \ldots, \pi(\boldsymbol{\alpha}_L)$. Let

$$\mathcal{X} := \left\{ -\frac{x_1(\alpha) - x_1(\alpha')}{u(\pi(\alpha)) - u(\pi(\alpha'))} : 1 \leqslant i \leqslant L, \alpha, \alpha' \in \boldsymbol{\alpha}_i, \pi(\alpha) \neq \pi(\alpha') \right\}$$

with

$$|\mathcal{X}| \leqslant \binom{|\boldsymbol{\alpha}_1|}{2} + \cdots + \binom{|\boldsymbol{\alpha}_L|}{2}.$$

Here $x_1(\alpha)$ stands for the first coordinate of α. If $\lambda_2 \notin \mathcal{X}$, then $x_1 + \lambda_2 u$ is a joined separating form for $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_L$. We may compute $u(\pi(\boldsymbol{\alpha}))$ using $O(|\boldsymbol{\alpha}_1| + \cdots + |\boldsymbol{\alpha}_L|)$ operations in $\mathbb{K}$ and then $\mathcal{X}$ using $O(|\boldsymbol{\alpha}_1|^2 + \cdots + |\boldsymbol{\alpha}_L|^2)$ further operations. We finally pick $\lambda_2 \in \mathcal{S} \setminus \mathcal{X}$ and derive the required joined separating form $x_1 + \lambda_2 u$ (so $\lambda_i = \lambda_2 \lambda_i'$ for $i \geqslant 3$) using $O(n) = O(1)$ operations in $\mathbb{K}$. □

If x_1 is a separating form for $\boldsymbol{\alpha}$, then there exist a monic separable polynomial $\chi(x)$ of degree $\leqslant N$ in $\mathbb{K}[x_1]$ and $v_2, \ldots, v_n$ other polynomials in $\mathbb{K}[x]_{<N}$ such that

$$\boldsymbol{\alpha} = \{(\zeta, v_2(\zeta), \ldots, v_n(\zeta)) : \chi(\zeta) = 0\}.$$

If the points of the sequence $\boldsymbol{\alpha}$ are pairwise distinct, it is well-known that these polynomials can be computed using $O(\mathsf{M}(N) \log N)$ operations in $\mathbb{K}$, thanks to sub-product tree algorithms; see [9, chapter 10] or [4], for instance. Otherwise we will rely on the following lemma.

LEMMA 2.2. *The univariate representation of $\boldsymbol{\alpha}$ can be computed using $O(\mathsf{M}(N)\log^2 N)$ operations in $\mathbb{K}$.*

Proof. We proceed recursively as follows. If $N=1$ then the univariate representation is obtained easily in time $O(1)$. Otherwise, we let $h:=\lceil N/2\rceil$ and we recursively compute the univariate representations $\chi^{(1)}, v_2^{(1)}, \ldots, v_n^{(1)}$ of $\boldsymbol{\alpha}^{(1)}=(\alpha_1,\ldots,\alpha_h)$ and $\chi^{(2)}, v_2^{(2)}, \ldots, v_n^{(2)}$ of $\boldsymbol{\alpha}^{(2)}=(\alpha_{h+1},\ldots,\alpha_N)$. We next compute $\Xi:=\gcd(\chi^{(1)},\chi^{(2)})$, $\hat{\chi}^{(2)}:=\chi^{(2)}/\Xi$, and $v_i^{(2)}:=v_i^{(2)}\operatorname{rem}\hat{\chi}^{(2)}$ for $i=2,\ldots,n$. In this way, the univariate representation of $\boldsymbol{\alpha}$ can be obtained as $\chi:=\chi^{(1)}\hat{\chi}^{(2)}$, and

$$v_i:=(\hat{\chi}^{(2)}(\chi^{(1)})'v_i^{(1)}+\hat{\chi}^{(1)}(\chi^{(2)})'\hat{v}_i^{(2)})\,U\operatorname{rem}\chi$$

for $i=2,\ldots,n$, where U is the inverse of χ' modulo χ. The cost $\mathsf{C}(N)$ of this method satisfies

$$\mathsf{C}(N)=\mathsf{C}(h)+\mathsf{C}(N-h)+O(\mathsf{M}(N)\log N),$$

which yields $\mathsf{C}(N)=O(\mathsf{M}(N)\log^2 N)$. □

2.2. Evaluation when x_1 is a separating form

Let m and l be positive integer parameters such that $m\,l\geqslant d$ and $m\,l=O(d)$. We expand the polynomial $P\in\mathbb{K}[x_1,\ldots,x_n]_{<d}$ to be evaluated as

$$P(x_1,\ldots,x_n)=\sum_{i_2<l,\ldots,i_n<l}P_{(i_2,\ldots,i_n)}(x_1,\ldots,x_n)\,x_2^{m i_2}\cdots x_n^{m i_n}, \tag{2.1}$$

where $\deg_{x_j}P_{(i_2,\ldots,i_n)}<m$ for $i_2<l,\ldots,i_n<l$ and $j\geqslant 2$.

We partition $\boldsymbol{\alpha}$ into subsequences $\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_L$ with $L:=\lceil N/d\rceil$ and $|\boldsymbol{\alpha}_i|\leqslant d$ for $i=1,\ldots,L$. Let

$$\begin{aligned}\sigma_l\colon\quad &\{0,\ldots,l-1\}^{n-1}\rightarrow\{0,\ldots,l^{l-1}-1\}\\ \sigma_m\colon\quad &\{0,\ldots,m-1\}^{n-1}\rightarrow\{0,\ldots,m^{n-1}-1\}\end{aligned}$$

be arbitrary bijections, e.g. $\sigma_l(i_0,\ldots,i_{n-2}):=i_0+i_1\,l+\cdots+i_{n-2}\,l^{n-2}$ and similarly for σ_m.

Algorithm 2.1

Input. $P\in\mathbb{K}[x_1,\ldots,x_n]_{<d}$ and $\boldsymbol{\alpha}\in(\mathbb{K}^n)^N$.

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

Assumption. x_1 is a joined separating form for $\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_L$.

1. If $N\leqslant d^{(n+1)/2}$ then set $m:=\lceil d/L^{1/(n-1)}\rceil$. Otherwise set $m:=\lceil d^{1/2}\rceil$.
   Set $l:=\lceil (d+1)/m\rceil$.
2. For $k=1,\ldots,L$, compute the univariate representation $\chi_k(x_1), v_{k,2}(x_1),\ldots,v_{k,n}(x_1)$ of $\boldsymbol{\alpha}_k$. Note that $\deg\chi_k\leqslant d$ for all k.
3. For $k=1,\ldots,L$ and $j_2=0,\ldots,m-1,\ldots,$ $j_n=0,\ldots,m-1$, compute

$$V_{\sigma_m(j_2,\ldots,j_n),k}:=v_{k,2}^{j_2}\cdots v_{k,n}^{j_n}\operatorname{rem}\chi_k.$$

   We regard V as a $m^{n-1}\times L$ matrix over $\mathbb{K}[x_1]$, with entries of degrees $<d$.

4. For $k=1,\ldots,L$, compute $v_{k,2}^m \operatorname{rem} \chi_k,\ldots,v_{k,n}^m \operatorname{rem} \chi_k$ and then

$$v_{k,2}^{m i_2} \cdots v_{k,n}^{m i_n} \operatorname{rem} \chi_k,$$

for $i_2=0,\ldots,l-1,\ldots,i_n=0,\ldots,l-1$.

5. For $j_2=0,\ldots,m-1,\ldots,j_n=0,\ldots,m-1$, let $Z_{\sigma_l(i_2,\ldots,i_n),\sigma_m(j_2,\ldots,j_n)}(x_1)\in\mathbb{K}[x_1]$ denote the coefficient $P_{(i_2,\ldots,i_n),(j_2,\ldots,j_n)}$ of $x_2^{j_2}\cdots x_n^{j_n}$ in $P_{(i_2,\ldots,i_n)}\in\mathbb{K}[x_1][x_2,\ldots,x_n]$. We regard Z as a $l^{n-1}\times m^{n-1}$ matrix over $\mathbb{K}[x_1]$, with entries of degrees $\leqslant d$. Compute

$$W := Z\,V.$$

6. For $k=1,\ldots,L$, compute

$$R_k := \sum_{i_2<l,\ldots,i_n<l} W_{\sigma_l(i_2,\ldots,i_n),k}\, v_{k,2}^{m i_2} \cdots v_{k,n}^{m i_n} \operatorname{rem} \chi_k.$$

7. Return the concatenation of the vectors $R_k(x_1(\boldsymbol{\alpha}_k))$ for $k=1,\ldots,L$, where $x_1(\boldsymbol{\alpha}_k)$ stands for the vector with the first coordinates of the points in $\boldsymbol{\alpha}_k$.

PROPOSITION 2.3. *Algorithm 2.1 is correct and takes $\tilde{O}(d^n + N^{\omega-2} d^{n+2-\omega})$ operations in $\mathbb{K}$ if $N \leqslant d^{(n+1)/2}$ and $\tilde{O}(N d^{(n-1)(\omega-1)/2})$ operations otherwise.*

Proof. The assumption on x_1 is needed for step 2. For all $k=1,\ldots,L$ and all points $(a_1,\ldots,a_n)$ in $\boldsymbol{\alpha}_k$ we verify in step 6 that

$$R_k(a_1) = \sum_{i_2<l,\ldots,i_n<l} W_{\sigma_l(i_2,\ldots,i_n),k}(a_1)\, a_2^{m i_2} \cdots a_n^{m i_n}.$$

In step 5 we have

$$\begin{aligned} W_{\sigma_l(i_2,\ldots,i_n),k}(a_1) &= \sum_{j_2<m,\ldots,j_n<m} Z_{\sigma_l(i_2,\ldots,i_n),\sigma_m(j_2,\ldots,j_n)}(a_1)\, V_{\sigma_m(j_2,\ldots,j_n),k}(a_1) \\ &= \sum_{j_2<m,\ldots,j_n<m} P_{(i_2,\ldots,i_n),(j_2,\ldots,j_n)}(a_1)\, a_2^{j_2} \cdots a_n^{j_n} \\ &= P_{(i_2,\ldots,i_n)}(a_1,\ldots,a_n). \end{aligned}$$

From the expansion (2.1), we deduce that $R_k(a_1) = P(a_1,\ldots,a_n)$.

By Lemma 2.2, step 2 takes $\tilde{O}(N)$ operations in $\mathbb{K}$. Steps 3 and 4 take

$$O(m^{n-1}\mathsf{M}(N) + l^{n-1}\mathsf{M}(N)) = \tilde{O}((l^{n-1}+m^{n-1})\,N)$$

operations. Step 6 contributes $O(l^{n-1}\mathsf{M}(N)) = \tilde{O}(l^{n-1}N)$ to the cost. Step 7 performs univariate multi-point evaluations in total time $O(\mathsf{M}(N)\log N)$. For the complexity of step 5 we distinguish the following cases:

- If $N \leqslant d^{(n+1)/2}$, then $L = O(d^{(n-1)/2})$, $m = O(d/L^{1/(n-1)})$, $d^{1/2} = O(m)$, $l = O(L^{1/(n-1)})$, and $l = O(m)$. Consequently, $l^{n-1} = O(\min(l^{n-1}, m^{n-1}, L))$, so the product $Z\,V$ can be split into products of $l^{n-1}\times l^{n-1}$ matrices, and the cost of step 5 is

$$\begin{aligned} O\!\left(\frac{m^{n-1}}{l^{n-1}}\frac{L}{l^{n-1}}(l^{n-1})^{\omega}\mathsf{M}(d)\right) &= O(L m^{n-1} l^{(n-1)(\omega-2)}\mathsf{M}(d)) \\ &= O\!\left(L\frac{d^{n-1}}{L}L^{\omega-2}\mathsf{M}(d)\right) \\ &= \tilde{O}(L^{\omega-2}d^n). \end{aligned}$$

In this case, the total cost of the algorithm is

$$\begin{aligned}\tilde{O}((l^{n-1}+m^{n-1})N+L^{(\omega-2)}d^n) &= \tilde{O}\left(\frac{d^{n-1}}{L}N+\left(\frac{N}{d}+1\right)^{(\omega-2)}d^n\right)\\ &= \tilde{O}(d^n+N^{\omega-2}d^{n+2-\omega}).\end{aligned}$$

- Otherwise, we have $d^{(n+1)/2}<N$, $l\asymp m\asymp d^{1/2}$, and $d^{(n-1)/2}=O(L)$. Consequently, $l^{n-1}=O(\min(l^{n-1},m^{n-1},L))$, the product $Z\,V$ can again be split into products of $l^{n-1}\times l^{n-1}$ matrices, and the cost of step 5 becomes

$$\begin{aligned}O\left(\frac{m^{n-1}}{l^{n-1}}\frac{L}{l^{n-1}}(l^{n-1})^{\omega}\mathsf{M}(d)\right) &= O(L\,l^{(n-1)(\omega-1)}\mathsf{M}(d))\\ &= O\left(\left(\frac{N}{d}+1\right)l^{(n-1)(\omega-1)}\mathsf{M}(d)\right)\\ &= \tilde{O}(Nd^{(n-1)(\omega-1)/2}).\end{aligned}$$

This dominates the total cost of the algorithm since

$$\tilde{O}((m^{n-1}+l^{n-1})N)=\tilde{O}(Nd^{(n-1)/2}). \qquad \square$$

Remark 2.4. The complexity of the product $Z\,V$ in step 5 of Algorithm 2.1 actually depends on the ratios of the dimensions of Z and V. For simplicity, we have reduced this product to several products of $l^{n-1}\times l^{n-1}$ matrices, whence a complexity bound in terms of ω. This choice is slightly sub-optimal if $\omega>2$; see [30]. For instance, if $N=O(d^n)$, then $L=O(d^{n-1})$ and the product $Z\,V$ can be done faster, using only

$$\tilde{O}((d^{(n-1)/2})^{\omega_2}d)=\tilde{O}(d^{(n-1)(\omega_2/2)+1}).$$

operations in $\mathbb{K}$. The product $W:=Z\,V$ still dominates the total cost of Algorithm 2.1.

Remark 2.5. In their article [26], Nüsken and Ziegler present Algorithm 2.1 in detail only for the case where $n=2$ and $L=1$; see [26, Theorem 8]. They give the complexity bound in terms of ω_2, recalled in Remark 2.4, but also in terms of the partial degrees in x_1 and x_2. The case where $n\geqslant 3$ is only mentioned in the conclusion of [26].

2.3. Case of at least three variables

In general, the form x_1 does not necessarily separate all the $\boldsymbol{\alpha}_i$ for $i=1,\ldots,L$. In such degenerate situations, one may apply a suitable change of coordinates before using Algorithm 2.1, provided that the cardinality of $\mathbb{K}$ is sufficiently large. In [26, section 6], Nüsken and Ziegler use a randomized method to compute such a change of coordinates. In this section, our first contribution is an alternative deterministic method, that takes advantage of the splitting of $\boldsymbol{\alpha}$ into $\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_L$. Another contribution is the complexity analysis in terms of the number of evaluation points. The following theorem summarizes the cost of our deterministic version of the Nüsken–Ziegler algorithm for $n\geqslant 3$.

THEOREM 2.6. *Let $n \geqslant 3$ be a fixed dimension, let $P \in \mathbb{K}[x_1, \ldots, x_n]$ be of total degree $\leqslant d$, let $\boldsymbol{\alpha} \in (\mathbb{K}^n)^N$, and let $\theta := \log N / \log(d^n)$, that is $N = (d^n)^\theta$. Then $P(\boldsymbol{\alpha})$ can be computed using $\tilde{O}((N + d^n)^{\eta_n(\theta)})$ operations in $\mathbb{K}$, where*

$$\eta_n(\theta) := \begin{cases} 1 \quad \text{if } \theta \leqslant \frac{1}{n} \\ (\omega - 2)\left(\theta - \frac{1}{n}\right) + 1 \quad \text{if } \frac{1}{n} \leqslant \theta \leqslant \frac{1}{2} + \frac{1}{2n} \\ \theta + \left(1 - \frac{1}{n}\right)\frac{\omega - 1}{2} \quad \text{if } \frac{1}{2} + \frac{1}{2n} \leqslant \theta \leqslant 1 \\ 1 + \left(1 - \frac{1}{n}\right)\frac{\omega - 1}{2\theta} \quad \text{if } 1 \leqslant \theta. \end{cases}$$

Proof. Assume first that we are given $\max\left(L\binom{d}{2}, d\right) + 1$ distinct elements in $\mathbb{K}$. By Lemma 2.1, we may compute a joined separating form $x_1 + \lambda_2 x_2 + \cdots + \lambda_n x_n$ for $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_L$ using

$$O(L d^2) = O\left(\left(\frac{N}{d} + 1\right) d^2\right) = O(N d + d^2) = O((d^n)^{\max(1, \theta + 1/n)}) \tag{2.2}$$

operations in $\mathbb{K}$. Let

$$\begin{array}{rccc} A: & \mathbb{K}^n & \rightarrow & \mathbb{K}^n \\ & (x_1, \ldots, x_n) & \mapsto & (x_1 + \lambda_2 x_2 + \cdots + \lambda_n x_n, x_2, \ldots, x_n). \end{array}$$

We replace P by $P \circ A^{-1} := P(x_1 - (\lambda_2 x_2 + \cdots + \lambda_n x_n), x_2, \ldots, x_n)$. This takes $\tilde{O}(d^n)$ operations in $\mathbb{K}$ thanks to [16, Appendix A, Proposition A.5] and the $d + 1$ distinct elements in $\mathbb{K}$. We next replace $\boldsymbol{\alpha}$ by $(A(\alpha) : \alpha \in \boldsymbol{\alpha})$, using $O(N)$ further operations. In this way, we reduce the problem to the case where x_1 separates $\boldsymbol{\alpha}$. It remains to compute $P(\boldsymbol{\alpha})$ via Proposition 2.3.

If $\theta \leqslant 1/n$, that is $N \leqslant d$, then the cost of Algorithm 2.1 is $\tilde{O}(d^n)$. If $\theta \geqslant 1/n$ and $N \leqslant d^{(n+1)/2}$ then $N \leqslant d^{n-1}$ (since $n \geqslant 3$) and the cost of Algorithm 2.1 becomes

$$\tilde{O}(d^n + (d^n)^{(\omega - 2)\theta + 1 + (2 - \omega)/n}) = \tilde{O}((d^n)^{(\omega - 2)(\theta - 1/n) + 1}).$$

This dominates the contribution (2.2), since

$$\theta + \frac{1}{n} \leqslant \frac{1}{2} + \frac{3}{2n} \leqslant (\omega - 2)\left(\theta - \frac{1}{n}\right) + 1.$$

If $N \geqslant d^{(n+1)/2}$ and $N \leqslant d^n$, then the cost of Algorithm 2.1 becomes

$$\tilde{O}(N d^{(n-1)(\omega - 1)/2}) = \tilde{O}((d^n)^{\theta + (1 - 1/n)(\omega - 1)/2}),$$

which again dominates the contribution (2.2). Finally, if $N \geqslant d^n$ then the cost is

$$\tilde{O}(N d^{(n-1)(\omega - 1)/2}) = \tilde{O}(N^{1 + (1 - 1/n)(\omega - 1)/(2\theta)}),$$

still dominating the contribution (2.2).

It remains to deal with the case where $\max\left(L\binom{d}{2}, d\right) + 1$ distinct elements in $\mathbb{K}$ are not given. We appeal to Proposition A.2: the hypotheses are met since the values returned by the present evaluation algorithm are independent of the $\max\left(L\binom{d}{2}, d\right) + 1$ given elements of $\mathbb{K}$ or of an algebraic extension of it. The complexity overhead does not exceed (2.2), up to logarithmic factors. □

Figure 2.1 represents the complexity exponent $\eta_n(\theta)$ introduced in Theorem 2.6.

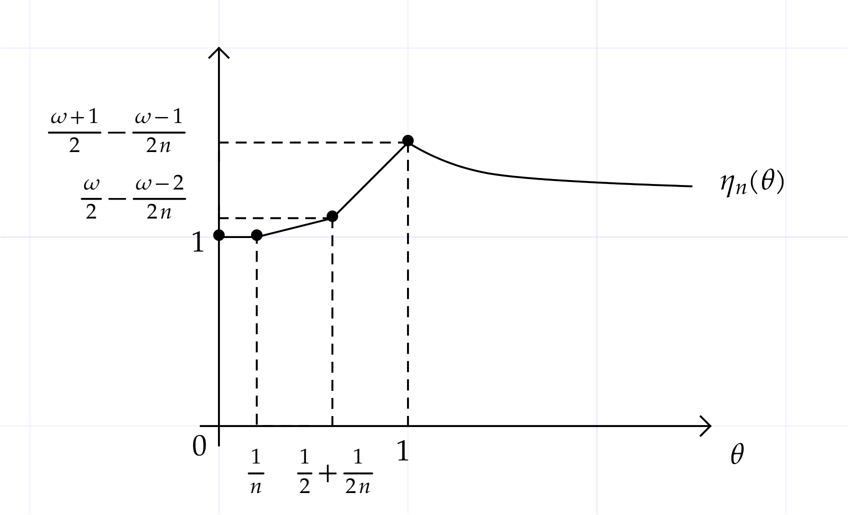

Figure 2.1. Complexity exponent $\eta_n(\theta)$ for the evaluation in degree d at $N=d^{\theta n}$ points when $n\geqslant 3$, via Theorem 2.6.

2.4. Bivariate case

The bivariate case $n=2$ behaves differently from the general one, because the deterministic search of a separating form becomes the bottleneck. The following theorem summarizes the cost of our deterministic bivariate version of the Nüsken–Ziegler algorithm.

THEOREM 2.7. *Let $P\in\mathbb{K}[x_1,x_2]$ be of total degree $\leqslant d$, let $\boldsymbol{\alpha}\in(\mathbb{K}^2)^N$, and let $\theta:=\log N/\log(d^2)$. Then $P(\boldsymbol{\alpha})$ can be computed using $\tilde{O}((N+d^2)^{\eta_2^*(\theta)})$ operations in $\mathbb{K}$, where*

$$\eta_2^*(\theta):=\begin{cases}1 & \text{if } \theta\leqslant\frac{1}{2}\\ \theta+\frac{1}{2} & \text{if } \frac{1}{2}\leqslant\theta\leqslant 1\\ 1+\frac{1}{2\theta} & \text{if } 1\leqslant\theta.\end{cases}$$

Proof. Assume first that we are given $L\binom{d}{2}+1$ elements in $\mathbb{K}$. By Lemma 2.1, a joined separating form $x_1+\lambda_1 x_2$ for $\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_L$ can be obtained in time

$$O(Ld^2)=O(Nd+d^2)=O((d^2)^{\max(1,\theta+1/2)}).\tag{2.3}$$

Applying the linear change of coordinates to P and $\boldsymbol{\alpha}$ takes $\tilde{O}(d^2+N)$ operations: here we may use [1, Lemma 1] to change the variables independently of the cardinality of $\mathbb{K}$.

If $N\leqslant d^{3/2}$, then the cost of Algorithm 2.1 is

$$\tilde{O}(d^2+N^{\omega-2}d^{4-\omega})=\tilde{O}((d^2)^{\max(1,(\omega-2)(\theta-1/2)+1)}),$$

by Proposition 2.3. Since $\omega\leqslant 3$, we also verify that

$$\max(1,(\omega-2)(\theta-1/2)+1)\leqslant\max(1,\theta+1/2).$$

If $N\geqslant d^{3/2}$, then the cost of Algorithm 2.1 is $\tilde{O}(Nd^{(\omega-1)/2})=\tilde{O}(Nd)$, which is again dominated by (2.3), up to logarithmic factors.

Finally, if we are not given $L\binom{d}{2}+1$ elements in $\mathbb{K}$, then we apply Proposition A.2, as in the proof of Theorem 2.6. □

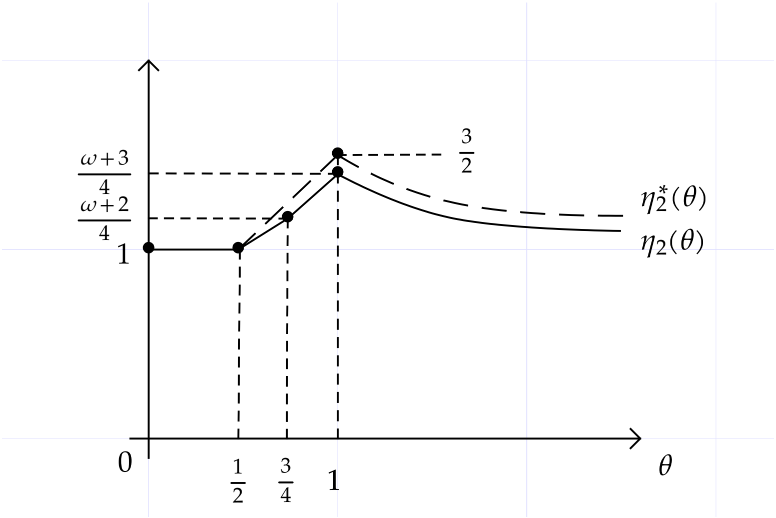

Figure 2.2. Deterministic and probabilistic complexity exponents for bivariate evaluation in degree d at $d^{2\theta}$ points, via Theorems 2.7 and 2.8.

The next theorem rephrases the probabilistic version of Nüsken and Ziegler for two variables and when N varies.

THEOREM 2.8. *Let $P \in \mathbb{K}[x_1, x_2]$ be of total degree $\leqslant d$, let $\boldsymbol{\alpha} \in (\mathbb{K}^2)^N$, let $\theta := \log N / \log(d^2)$, and assume that we are given $2L\binom{d}{2}$ distinct elements in $\mathbb{K}$. Then $P(\boldsymbol{\alpha})$ can be computed by a probabilistic algorithm of Las Vegas type using an expected number of $\tilde{O}((N + d^2)^{\eta_2(\theta)})$ operations in $\mathbb{K}$, where*

$$\eta_2(\theta) := \begin{cases} 1 & \text{if } \theta \leqslant 1/2 \\ (\omega - 2)\left(\theta - \frac{1}{2}\right) + 1 & \text{if } \frac{1}{2} \leqslant \theta \leqslant \frac{3}{4} \\ \theta + \frac{\omega - 1}{4} & \text{if } \frac{3}{4} \leqslant \theta \leqslant 1 \\ 1 + \frac{\omega - 1}{4\theta} & \text{if } 1 \leqslant \theta. \end{cases}$$

Proof. In order to find a joined separating form $x_1 + \lambda_2 x_2$ for $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_L$, it suffices to take a random λ_2 in the given subset of $\mathbb{K}$. The probability of success for each trial is $\geqslant 1/2$; see the proof of Lemma 2.1. So the expected number of trials is $O(1)$. The rest of the proof is adapted from the one of Theorem 2.7. Alternatively, one may adapt the proof of Theorem 2.6, by noting that the cost to compute a separating form is negligible, if we are allowed to use a randomized algorithm. □

Figure 2.2 displays the complexity exponents introduced in Theorem 2.7 and Theorem 2.8. Note that $\eta_2^* = \eta_2$ when $\omega = 3$.

3. AMORTIZED MULTIVARIATE EVALUATION

In this section, we refine our complexity analysis of the amortized algorithm for the evaluation of multivariate polynomials from [17]. The main novelty is a precise analysis of the cost of the required precomputations. From this, we will be able to derive new complexity bounds for non-amortized multi-point evaluation. Throughout this section $P \in \mathbb{K}[x_1, \ldots, x_n]$ and $\boldsymbol{\alpha} \in (\mathbb{K}^n)^N$ denote the polynomial and the tuple of evaluation points, respectively. We also write I for the ideal of $\mathbb{K}[x_1, \ldots, x_n]$ that consists of all polynomials that vanish at $\boldsymbol{\alpha}$. We recall that the dimension n is fixed.

3.1. Shifted Popov forms

Let us consider a vector

$$\boldsymbol{s} := (s_1, \ldots, s_n) \in \mathbb{Z}^n,$$

called a **shift** for the degrees, the $\boldsymbol{s}$-degree of a row vector $\boldsymbol{a} = (a_1, \ldots, a_n)$ in $\mathbb{K}[x]^n$ is defined as

$$\deg_{\boldsymbol{s}} \boldsymbol{a} := \max (\deg a_1 + s_1, \ldots, \deg a_n + s_n).$$

If $\boldsymbol{a}$ is non-zero, then the **pivot index** of $\boldsymbol{a}$ is the largest index i for which the latter maximum is reached. The entry a_i is called the **pivot**, and its degree is the **pivot degree**. If $\boldsymbol{a}$ is zero, then its pivot index is defined to be zero.

Let M denote a $m \times n$ matrix with entries in $\mathbb{K}[x]$. The matrix M is in **$\boldsymbol{s}$-Popov form** if the following properties are satisfied:

- The positive pivot indices of the rows of M are in increasing order;
- The pivots of the rows of M are monic (*i.e.* have leading coefficient 1);
- The pivots of M have a degree strictly larger than the other elements in their column.

If $m = n$ and M is non-singular, then its pivots are the diagonal elements. In this case, M satisfies the "predictable degree" property:

LEMMA 3.1. *Let M be a non-singular $n \times n$ matrix in Popov form. If $\boldsymbol{b} = (b_1, \ldots, b_n) := \boldsymbol{a} M$ for some row vector $\boldsymbol{a} \in \mathbb{K}[x]^n$, then*

$$\deg_{\boldsymbol{s}} \boldsymbol{b} = \max_{i=1,\ldots,n} (d_i + \deg a_i),$$

where d_i denotes the $\boldsymbol{s}$-degree of the i-th row of M.

Proof. See [22, Theorem 1.1], for instance. □

Given non-constant polynomials $\chi_1, \ldots, \chi_L$ in $\mathbb{K}[x]$, and given a $r \times L$ matrix F with entries $F_{i,j}$ in $\mathbb{K}[x]_{<\deg \chi_j}$, Popov forms will be used to compute the kernel of the map

$$\begin{array}{rrcl} E\colon & \mathbb{K}[x]^r & \longrightarrow & \displaystyle\prod_{j=1,\ldots,L} \mathbb{K}[x] / (\chi_j) \\ & u = (u_i)_{i=1,\ldots,r} & \longmapsto & uF = \left(\displaystyle\sum_{i=1,\ldots,r} F_{i,j} u_i \right)_{j=1,\ldots,L}. \end{array}$$

Since the vectors $(\chi_1, 0, \ldots, 0)$, $(0, \chi_2, 0, \ldots, 0), \ldots,$ $(0, \ldots, 0, \chi_L)$ are a free family in $\ker E$, the kernel of E is a free $\mathbb{K}[x]$-module of rank r.

PROPOSITION 3.2. [23, *Theorem 1.4*] *Given non-constant polynomials $\chi_1, \ldots, \chi_L$ in $\mathbb{K}[x]$ and an $r \times L$ matrix F with entries $F_{i,j}$ in $\mathbb{K}[x]_{<\deg \chi_j}$, there exists a unique $r \times r$ matrix U in $\boldsymbol{s}$-Popov form such that the rows of U are a basis of $\ker E$. If $L = O(r)$ then U can be computed using $\tilde{O}(r^{\omega-1} N)$ operations in $\mathbb{K}$, where $N := \deg \chi_1 + \cdots + \deg \chi_L$.*

3.2. Admissible orderings

Let $\mathfrak{M}$ be the set of monomials $x_1^{e_1} \cdots x_n^{e_n}$ with $e_1, \ldots, e_n \in \mathbb{N}$. 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 $\mathrm{lm}_{\prec}(P) \in \mathfrak{M}$ that is called the **leading monomial** of P; the corresponding coefficient $\mathrm{lc}_{\prec}(P) = P_{\mathrm{lm}(P)} \in \mathbb{K}$ is called the **leading coefficient** of P. A total ordering $\prec$ on $\mathfrak{M}$ is said to be **admissible** if

$$M|N \implies M \preccurlyeq N \quad \text{and} \quad M \preccurlyeq N \implies x_i M \preccurlyeq x_i N$$

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

$$x_1^{e_1} \cdots x_n^{e_n} \prec_{\mathrm{lex}} x_1^{f_1} \cdots x_n^{f_n} \iff \begin{cases} e_1 < f_1 & \text{or} \\ e_1 = f_1 \wedge e_2 < f_2 & \text{or} \\ \vdots & \\ e_1 = f_1 \wedge \cdots \wedge e_{n-1} = f_{n-1} \wedge e_n < f_n & \end{cases}$$

is admissible.

3.3. Minimal polynomials

In the rest of the paper, an **admissible weight** will be a n-tuple

$$\boldsymbol{w} = (w_1, \ldots, w_n) \in \{2^i : i \in \mathbb{Z}\}^n$$

with

$$w_1 \cdots w_n = 1. \tag{3.1}$$

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

$$\deg_{\boldsymbol{w}} P := \max_{M \in \mathfrak{M}} \{\deg_{\boldsymbol{w}} M : P_M \neq 0\}.$$

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

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

It is easy to check that $\prec_{\boldsymbol{w}}$ is an admissible ordering. Given an admissible weight $\boldsymbol{w}$, there exists a unique non-zero polynomial $B_{\boldsymbol{w}}$ in the reduced Gröbner basis of 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 I. Note that there are at most N monomials below the leading monomial of $B_{\boldsymbol{w}}$ for $\prec_{\boldsymbol{w}}$. From [17, Corollary 1], we know that

$$\deg_{x_i} B_{\boldsymbol{w}} \leqslant \frac{\sqrt[n]{n!\,(N+1)}}{w_i}, \tag{3.2}$$

for $i = 1, \ldots, n$. The main aim of this section is an efficient algorithm for the computation of $B_{\boldsymbol{w}}$. For this purpose, we may assume without loss of generality that we ordered the coordinates such that $w_1 \leqslant \cdots \leqslant w_n$. By (3.1), this yields $w_1 \leqslant 1 \leqslant w_2 \cdots w_n$. Using also (3.2), we get

$$\prod_{2 \leqslant i \leqslant n} \deg_{x_i} B_{\boldsymbol{w}} \leqslant \frac{1}{w_2 \cdots w_n} \prod_{2 \leqslant i \leqslant n} \sqrt[n]{n!\,(N+1)} \leqslant (n!\,(N+1))^{(n-1)/n}.$$

It follows that the set

$$\mathcal{B} := \left\{ x_2^{e_2} \cdots x_n^{e_n} : 0 \leqslant e_i \leqslant \frac{\sqrt[n]{n!\,(N+1)}}{w_i}, i = 2, \ldots, n \right\}$$

of monomials in $x_2, \ldots, x_n$ has cardinality

$$r \leqslant 2^{n-1} (n!\,(N+1))^{(n-1)/n} = O((N+1)^{(n-1)/n}). \tag{3.3}$$

We sort the monomials of $\mathcal{B}$ according to the ordering $\prec_{\mathcal{B}}$ for which

$$x_2^{e_2} \cdots x_n^{e_n} \prec_{\mathcal{B}} x_2^{f_2} \cdots x_n^{f_n}$$

if and only if

$$x_1^{e_1} \cdots x_n^{e_n} \prec_{\mathrm{lex}} x_1^{f_1} \cdots x_n^{f_n},$$

where

$$\begin{aligned} e_1 &:= 0 \\ f_1 &:= w_1^{-1} (w_2 e_2 + \cdots + w_n e_n - (w_2 f_2 + \cdots + w_n f_n)). \end{aligned}$$

For this choice of e_1 and f_1, we note that

$$w_1 e_1 + \cdots + w_n e_n = w_1 f_1 + \cdots + w_n f_n.$$

Let

$$b_1 \prec_{\mathcal{B}} b_2 \prec_{\mathcal{B}} \cdots \prec_{\mathcal{B}} b_r$$

denote the monomials of $\mathcal{B}$ in increasing order.

As in the previous sections, the sequence of points $\boldsymbol{\alpha}$ will be split into subsequences $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_L$ of cardinality $\leqslant M$ such that $M \leqslant N$ and $LM = O(N)$. More precisely, until section 3.6, we take

$$M := \lfloor N^{1/n} \rfloor \text{ and } L := \lceil N/M \rceil, \tag{3.4}$$

so that $L = O(N^{(n-1)/n})$.

LEMMA 3.3. *Let M and L be as in (3.4) and assume that $x_1, \ldots, x_n$ are joined separating forms for $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_L$. Then we can compute B_w using*

$$\tilde{O}(N^{1+(\omega-1)(n-1)/n})$$

operations in $\mathbb{K}$.

Proof. Without loss of generality, we may order the coordinates such that $w_1 \leqslant \cdots \leqslant w_n$, after which we may use the above notation. We write $\mathbb{K}[x_1][x_2, \ldots, x_n]_{\mathcal{B}}$ for the set of polynomials over $\mathbb{K}[x_1]$ with support in $\mathcal{B}$. Let U be the Popov form of the matrix representing the kernel of the projection

$$\begin{aligned} E\colon \quad \mathbb{K}[x_1][x_2, \ldots, x_n]_{\mathcal{B}} &\longrightarrow \mathbb{K}[x_1, \ldots, x_n]/I \\ P &\longmapsto P \bmod I \end{aligned}$$

in the basis $b_1, \ldots, b_r$ and for the shift vector

$$\boldsymbol{s} := (s_1, \ldots, s_r),$$

where $s_i := w_1^{-1} \deg_w(b_i) \in \mathbb{N}$. Regarding B_w also as a $\mathbb{K}[x_1]$-vector in $\mathbb{K}[x_1][x_2, \ldots, x_n]_{\mathcal{B}}$, we have

$$\deg_w B_w = w_1 \deg_{\boldsymbol{s}} B_w. \tag{3.5}$$

Since $\mathbb{K}[x_1]$ is principal, $\ker E$ is a free $\mathbb{K}[x_1]$-module. Let $\chi(x_1)$ be the minimal polynomial of x_1 in I. Since $\chi(x_1)\, b_1, \ldots, \chi(x_1)\, b_r$ is a free family of $\ker E$, the rank of $\ker E$ is r. Consequently, U is a non-singular $r \times r$ matrix.

Every row U_i of U with $i=1,\ldots,r$ can also be regarded as a polynomial in $\mathbb{K}[x_1][x_2,\ldots,x_n]_{\mathcal{B}}$, which belongs to I. We have

$$\deg_w U_i = w_1 \deg_s U_i. \tag{3.6}$$

By construction, there exist $a_1,\ldots,a_r$ in $\mathbb{K}[x_1]$ such that

$$B_w = a_1 U_1 + \cdots + a_r U_r.$$

By Lemma 3.1, (3.5), and (3.6), we have

$$\deg_w B_w = \max_{i=1,\ldots,r} (\deg_w U_i + w_1 \deg a_i).$$

Let $\mathcal{S}$ be the set of row indices i such that $\deg_w U_i$ is minimal, that is

$$\mathcal{S} := \left\{ i \in \{1,\ldots,r\} : \deg_w U_i = \min_{j=1,\ldots,r} \deg_w U_j \right\}.$$

By the minimality of B_w and since the U_i belong to I, the polynomials a_i with $i \notin \mathcal{S}$ must be zero and the others must be in $\mathbb{K}$. Consequently, $B_w = \sum_{i\in\mathcal{S}} a_i U_j$. By definition (see section 3.1), the pivot index of

$$U_i(x_1,\ldots,x_n) = \sum_{1\leqslant j\leqslant r} U_{i,j}(x_1)\, b_j$$

is i. This means that

$$\deg_w U_i(x_1,\ldots,x_n) = \deg_w (U_{i,i}(x_1)\, b_i)$$

and

$$\deg_w (U_{i,j}(x_1)\, b_j) < \deg_w (U_{i,i}(x_1)\, b_i)$$

for all $j>i$. If $j<i$ is such that $\deg_w(U_{i,j}(x_1)\, b_j) = \deg_w(U_{i,i}(x_1)\, b_i)$, then the definition of the ordering $\prec_{\mathcal{B}}$ ensures that $b_j \prec_{\text{lex}} x_1^{f_1} b_i$, where $f_1 := w_1^{-1}(\deg_w b_j - \deg_w b_i)$. It follows that $f_1 = \deg U_{i,i}(x_1) - \deg U_{i,j}(x_1)$, whence $x_1^{\deg U_{i,j}(x_1)} b_j \prec_w x_1^{\deg U_{i,i}(x_1)} b_i$. In other words, the leading monomial of U_i for $\prec_w$ is the leading monomial of $U_{i,i}(x_1)\, b_i$. Consequently, the leading monomial of B_w is the leading monomial of $U_{j,j} b_j$, where $j := \max\{i \in \mathcal{S} : a_i \neq 0\}$. Finally, the minimality of B_w implies that B_w is $\mathbb{K}$-proportional to $U_{\min \mathcal{S}}$.

In order to obtain the Popup form U, we first compute the univariate representations of $\boldsymbol{\alpha}_j$ for $j=1,\ldots,L$: this takes $\tilde{O}(N)$ operations in $\mathbb{K}$ by Lemma 2.2. The univariate representation of $\boldsymbol{\alpha}_j$ is given by a monic polynomial $\chi_j \in \mathbb{K}[x_1]$ of degree $\leqslant |\boldsymbol{\alpha}_j| \leqslant M$ and polynomials $v_{j,2},\ldots,v_{j,n} \in \mathbb{K}[x_1]$ of degrees $<M$. Now consider the matrix $F \in \mathbb{K}[x_1]^{r\times L}$ defined by

$$F_{i,j} := b_i(v_{j,2},\ldots,v_{j,n}) \operatorname{mod} \chi_j$$

for $i=1,\ldots,r$ and $j=1,\ldots,L$: the entries $F_{i,j}$ can be computed in softly linear time $\tilde{O}(rLM) = \tilde{O}(rN)$. Since (3.3) and (3.4) imply $r=O(L)$, we may apply Proposition 3.2. We conclude that U can be computed using

$$\tilde{O}(r^{\omega-1}N) = \tilde{O}((N^{(n-1)/n})^{\omega-1}N) = \tilde{O}(N^{1+(\omega-1)(n-1)/n}).$$

operations in $\mathbb{K}$. □

3.4. Heterogeneous bases

Given $D \geqslant N$, we define

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

Given a finite subset $S \subseteq \{1, \ldots, n\}$, we write $\delta_i^S := 1$ if $i \in S$ and $\delta_i^S := 0$ otherwise. We introduce

$$\begin{array}{rcl} \pi_{|S}\colon \quad \mathbb{K}^n & \to & \mathbb{K}^n \\ (x_1, \ldots, x_n) & \mapsto & (\delta_1^S x_1, \ldots, \delta_n^S x_n). \end{array}$$

We denote by $\boldsymbol{\alpha}_{|S}$ the image of $\boldsymbol{\alpha}$ under $\pi_{|S}$ and we define

$$\begin{array}{rcl} \mathfrak{M}_{|S} & := & \{x_1^{e_1} \cdots x_n^{e_n} \in \mathfrak{M} : i \notin S \Longrightarrow e_i = 0\} \\ \mathbb{K}[x_1, \ldots, x_n]_{|S} & := & \{P \in \mathbb{K}[x_1, \ldots, x_n] : \operatorname{supp} P \subseteq \mathfrak{M}_{|S}\} \\ I_{|S} & := & I \cap \mathbb{K}[x_1, \ldots, x_n]_{|S}. \end{array}$$

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 $\boldsymbol{w}_{\backslash E} \in (\mathbb{R}^{>} \cup \{\perp\})^n$ to be the weight $\boldsymbol{w}'$ with $w_i' = w_i$ if $i \notin E$ and $w_i' = \perp$ if $i \in E$. If $\boldsymbol{w}$ is admissible, then we note that $\boldsymbol{w}_{\backslash E}$ is again admissible for $\mathbb{K}[x_1, \ldots, x_n]_{|S}$ where $S := \{1, \ldots, n\} \setminus E$. We further let

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

The family of simplest polynomials $(B_{\boldsymbol{w}})_{\boldsymbol{w} \in \boldsymbol{\Omega}_D^{\#}}$ is called an **heterogenous basis** for $\boldsymbol{\alpha}$ and D.

LEMMA 3.4. *Let M and L be as in (3.4) and assume that $x_1, \ldots, x_n$ are joined separating forms for $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_L$. Then a heterogenous basis $(B_{\boldsymbol{w}})_{\boldsymbol{w} \in \boldsymbol{\Omega}_D^{\#}}$ can be computed using*

$$\tilde{O}(N^{1 + (\omega - 1)(n - 1)/n} \log^{n-1} D)$$

operations in $\mathbb{K}$.

Proof. The cost for computing a single $B_{\boldsymbol{w}}$ is given in Lemma 3.3. On the other hand, we have $\operatorname{card} \boldsymbol{\Omega}_D^{\#} \leqslant 2^n \operatorname{card} \boldsymbol{\Omega}_D$, and $\operatorname{card} \boldsymbol{\Omega}_D = O(\log^{n-1} D)$ by [17, Lemma 2]. □

Let M and L are still as in (3.4). Assume that N is a power 2^{ℓ}. A **recursive heterogeneous basis** for $\boldsymbol{\alpha}$ and D consists of the following data:

- a heterogenous basis for $\boldsymbol{\alpha}$ and D,
- for all $m = 1, 2, \ldots, 2^{\ell - 1}$ and $i = 0, \ldots, N / m - 1$, a heterogeneous basis for $\boldsymbol{\alpha}_{im+1,m} := (\alpha_{im+1}, \ldots, \alpha_{im+m}) \in (\mathbb{K}^n)^m$ and $D_m := 4^n n! \, (2m + 1)$.

We say that linearly independent linear forms $u_1, \ldots, u_n$ **weakly separate** $\boldsymbol{\alpha}$ if each of them is a joined separating form for $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_L$. We say that they **recursively weakly separate** $\boldsymbol{\alpha}$ if they weakly separate each of the above $\boldsymbol{\alpha}_{im+1,m}$ for $m = 1, 2, \ldots, 2^{\ell - 1}$ and $i = 0, \ldots, N / m - 1$.

In order to construct recursive heterogeneous bases, we need coordinates $x_1, \ldots, x_n$ that recursively weakly separate $\boldsymbol{\alpha}$. This is the purpose of the following lemma.

LEMMA 3.5. *Assume that we are given $(\ell + 1) N^{1 + 1/n} + 1$ points in $\mathbb{K}$. A basis $u_1, \ldots, u_n$ of linear forms that recursively weakly separate $\boldsymbol{\alpha}$ can be computed using $\tilde{O}(N^{1 + 1/n})$ operations in $\mathbb{K}$.*

Proof. With M and L as in (3.4) we have

$$L \binom{M}{2} \leqslant \left(\frac{N}{M} + 1\right) \binom{M}{2} = \frac{1}{2} (N + M) \, (M - 1) \leqslant \frac{1}{2} (N + N^{1/n}) \, N^{1/n} \leqslant N^{1 + 1/n}. \tag{3.7}$$

Let us call $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_L$ the standard split of $\boldsymbol{\alpha}$. We construct the sequence of sequences of points $\boldsymbol{\beta}$ that consists of the standard split of $\boldsymbol{\alpha}$ and the standard splits of $\boldsymbol{\alpha}_{im+1,m}$ for $m = 1, 2, \ldots, 2^{\ell - 1}$ and $i = 0, \ldots, N / m - 1$.

Let $M_m := \lfloor m^{1/n} \rfloor$. The standard split of $\boldsymbol{\alpha}_{im+1,m}$ consists of $L_m := \lceil m / M_m \rceil$ sequences of cardinality $\leqslant M_m$. The total number of points in $\boldsymbol{\beta}$ is at most $(\ell+1)\,N = O(N \log N)$. Using (3.7), we verify that

$$\begin{aligned}\sum_{\gamma \in \boldsymbol{\beta}} \binom{|\gamma|}{2} &\leqslant L\binom{M}{2} + \sum_{m=1,2,\ldots,2^{\ell-1}} \sum_{0 \leqslant i < N/m} L_m \binom{M_m}{2}\\ &\leqslant N^{1+1/n} + \ell \frac{N}{m} m^{1+1/n}\\ &\leqslant (\ell+1)\,N^{1+1/n}.\end{aligned}$$

By Lemma 2.1, we may compute a joined separating form $u_1 = x_1 + \lambda_2 x_2 + \cdots + \lambda_n x_n$ for $\boldsymbol{\beta}$, in time $O((\ell+1)\,N^{1+1/n}) = \tilde{O}(N^{1+1/n})$, using the given subset of $\mathbb{K}$. For $j = 2, \ldots, n$, we take $u_j := x_j + \lambda_j' u_1$ for some λ_j' in $\mathbb{K}$ such that

$$\lambda_j' \neq -\frac{x_j(b) - x_j(a)}{u_1(b) - u_1(a)}$$

for all distinct points $a, b \in \mathbb{K}^n$ in γ, for all γ in $\boldsymbol{\beta}$. Since λ_j' must be different from $\leqslant (\ell+1)\,N^{1+1/n}$ elements in $\mathbb{K}$, a suitable λ_j' can be found in the given subset of $\mathbb{K}$. In this way, u_i is a joined separating form for $\boldsymbol{\beta}$.

By construction, $u_1, \ldots, u_n$ are $\mathbb{K}$-linearly independent. The evaluation of u_1 at all points $a \in \mathbb{K}^n$ in $\boldsymbol{\beta}$ takes $O(N \log N)$ operations in $\mathbb{K}$. For each $j = 2, \ldots, n$, the set of unsuitable values for λ_j' can be computed using $O(N^{1+1/n} \log N)$ operations in $\mathbb{K}$. □

LEMMA 3.6. *Assume that $x_1, \ldots, x_n$ recursively weakly separate $\boldsymbol{\alpha}$. Then a recursive heterogeneous basis for $\boldsymbol{\alpha}$ and $D \geqslant N$ can be computed using*

$$\tilde{O}(N^{1+(\omega-1)(n-1)/n} \log^{n-1} D)$$

operations in $\mathbb{K}$.

Proof. This is a direct consequence of Lemma 3.4; recall that n is fixed. □

3.5. Amortized evaluation

We gather the preceding results of this section into the following statement.

LEMMA 3.7. *Let $d \in \mathbb{N}$ and $D := (d+1)^n$. Assume that $N \leqslant D$ is a power 2^ℓ, and that we are given and element of order*

$$\geqslant \max((\ell+1)\,N^{1+1/n} + 1, d+1, 4^n n!\,(D+1))$$

in $\mathbb{K}$. Then we can compute an invertible $n \times n$ matrix U over $\mathbb{K}$ such that $x_1, \ldots, x_n$ recursively weakly separate $U(\boldsymbol{\alpha})$, together with a recursive heterogeneous basis for $U(\boldsymbol{\alpha})$ and D, using

$$\tilde{O}(N^{1+(\omega-1)(n-1)/n} \log^{n-1} d)$$

operations in $\mathbb{K}$. Having computed such a basis, any polynomial P of total degree $\leqslant d$ can be evaluated at $\boldsymbol{\alpha}$ using $\tilde{O}(D)$ operations in $\mathbb{K}$.

Proof. The case $n = 1$ is well-known; see [9, chapter 10]. So let us assume $n \geqslant 2$. The computation of U and of the recursive heterogeneous basis has been addressed in Lemmas 3.5 (using the element of order $> (\ell+1)\,N^{1+1/n}$) and 3.6.

Given P and U, the computation of the polynomial $P \circ U^{-1}$ takes $\tilde{O}(d^n)$ operations in $\mathbb{K}$ thanks to [16, Appendix A, Proposition A.5] and the element of order $>d$. By [17, Theorem 5], $P \circ U^{-1}$ can be evaluated at $U(\boldsymbol{\alpha})$ using $\tilde{O}(\mathsf{SM}(D))$ operations in $\mathbb{K}$. Here $\mathsf{SM}(s)$ is a cost function for multiplying two sparse polynomials P and Q in $\mathbb{K}[x_1, \ldots, x_n]$, where s is the maximum of the sizes of the supports of P, Q, and $P Q$. As detailed in the proof of [17, Theorem 1], we may take $\mathsf{SM}(D) = \tilde{O}(D)$, thanks to the element of order $\geqslant 4^n n!\,(D+1)$. □

THEOREM 3.8. *Let $n \geqslant 1$ be a fixed dimension, let $\boldsymbol{\alpha} \in (\mathbb{K}^n)^N$ and $d \in \mathbb{N}$ be such that $N = O(d^n)$. After the precomputation of suitable data, as a function of $\boldsymbol{\alpha}$ and d only, any polynomial P of total degree $\leqslant d$ can be evaluated at $\boldsymbol{\alpha}$ using*

$$\tilde{O}(d^n)$$

operations in $\mathbb{K}$. Moreover, the precomputation can be done using

$$\tilde{O}(N^{1+(\omega-1)(n-1)/n})\,(\log d)^{O(1)}$$

operations in $\mathbb{K}$.

Proof. Let $D := (d+1)^n$. We first handle the case where $2N \leqslant D$. In this way, up to repeating some points in $\boldsymbol{\alpha}$, we may assume that N is a power of two such that $N \leqslant D$.

If we are given an element of order $\geqslant \max((\ell+1)\,N^{1+1/n}+1, d+1, 4^n n!\,(D+1))$, then the complexity bounds follow from Lemma 3.7. Otherwise, we may appeal to Proposition A.2 to ensure the existence of this element of high order. In this way, note that the precomputed data are in general defined over an extension of the form $\mathbb{K}[z]/(\mu(z))$, and that P must then be evaluated over this extension, following the rules described in the proof of Proposition A.2.

Finally, if $2N > D$ then we subdivide $\boldsymbol{\alpha}$ into $O(N/D) = O(1)$ subsequences of size $\leqslant \lfloor D/2 \rfloor$, and then repeat the precomputations and the evaluations for each subsequence. □

3.6. Non-amortized evaluation

Using the preceding amortized evaluation strategy, we analyze the cost of a single multipoint evaluation in n variables as a function of N.

THEOREM 3.9. *Let $n \geqslant 1$ be a fixed dimension. A polynomial $P \in \mathbb{K}[x_1, \ldots, x_n]$ of total degree $\leqslant d$ can be evaluated at $N = (d^n)^\theta$ points in $\mathbb{K}^n$ using $\tilde{O}((N + d^n)^{\mu_n(\theta)})$ operations in $\mathbb{K}$, where*

$$\mu_n(\theta) := \begin{cases} 1 \quad \text{if } \theta \leqslant \frac{n}{(n-1)\,\omega+1} \\ \theta + 1 - \frac{n}{(n-1)\,\omega+1} \quad \text{if } \frac{n}{(n-1)\,\omega+1} \leqslant \theta \leqslant 1 \\ 1 + \left(1 - \frac{n}{(n-1)\,\omega+1}\right)\frac{1}{\theta} \quad \text{if } 1 \leqslant \theta. \end{cases}$$

Proof. We start with the case $\theta \leqslant 1$. We subdivide $\boldsymbol{\alpha}$ into subsequences $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_L$ of cardinality $\leqslant M$, where $M \leqslant N$ and $L M = O(N)$. The evaluations of P at $\boldsymbol{\alpha}_i$ for $i = 1, \ldots, L$ take

$$\tilde{O}((d^n + M^{1+(\omega-1)(n-1)/n})\,L) = \tilde{O}((d^n + M^{((n-1)\omega+1)/n})\,L) \tag{3.8}$$

operations in $\mathbb{K}$ by Theorem 3.8. We distinguish the following cases:

- If $N^{((n-1)\omega+1)/n} \leqslant d^n$, then we set $M := N$ and $L := 1$, so the cost (3.8) is at most $\tilde{O}(d^n)$.

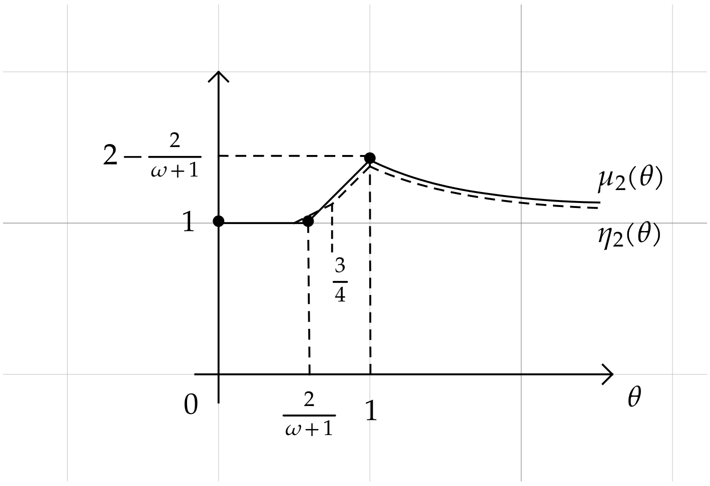

Figure 3.1. Complexity exponent $\mu_2(\theta)$ for bivariate evaluation in degree d at $d^{2\theta}$ points via Theorem 3.9. For comparison, we also show the complexity exponent $\eta_2(\theta)$ from Theorem 2.8.

- Otherwise we take

$$M := \left\lfloor d^{\frac{n^2}{(n-1)\omega+1}} \right\rfloor$$

and $L := \lceil N / M \rceil$, so the cost (3.8) simplifies into

$$\tilde{O}\left(d^n\left(\frac{N}{M}+1\right)\right) = \tilde{O}\left(N d^{n-\frac{n^2}{(n-1)\omega+1}} + d^n\right) = \tilde{O}\left(N d^{n-\frac{n^2}{(n-1)\omega+1}}\right).$$

Finally, if $N > d^n$ then we subdivide $\boldsymbol{\alpha}$ into $O(N/d^n)$ subsequences of size $\leqslant d^n$, so the evaluation cost becomes

$$\tilde{O}((d^n)^{\mu_n(1)} N / d^n) = \tilde{O}(N^{1+(\mu_n(1)-1)/\theta}). \qquad \square$$

Remark 3.10. If $n=2$ and $\omega<3$, then Theorem 3.9 always improves on Theorem 2.7.

Remark 3.11. We have plotted the complexity exponent $\mu_2(\theta)$ for bivariate multi-point evaluation in Figure 3.1. Comparing with Theorem 2.8, while assuming that $\omega<3$, we observe that

- $\mu_2(\theta) = \eta_2(\theta)$ if $\theta \leqslant 1/2$,
- $\mu_2(\theta) < \eta_2(\theta)$ if $1/2 < \theta < \theta_c$,
- $\mu_2(\theta) \geqslant \eta_2(\theta)$ if $\theta_c \leqslant \theta$, where $\theta_c := \frac{\omega+2}{2(\omega+1)} < \frac{3}{4}$.

Remark 3.12. Comparing with Theorem 2.6, the complexity exponents $\eta_n(1)$ and $\mu_n(1)$ tend to $(\omega+1)/2$ and $2-1/\omega$ respectively, when n tends to infinity. If $\omega>2$, then our new bound improves on the Nüsken–Ziegler algorithm for large n. More precisely, we have

$$\eta_n(1) - \gamma_n(1) = \frac{(\omega-1)\,((\omega-2)\,n+1-\omega)\,(n-1)}{2n\,((n-1)\,\omega+1)},$$

so our method is faster when

$$n > \frac{\omega-1}{\omega-2}.$$

3.7. Three or more variables

As in section 2.3, let us now analyze the variant when we apply the amortized evaluation method for $n-1$ variables to the non-amortized problem for n variables.

THEOREM 3.13. *Let $n\geqslant 3$ be a fixed dimension. A polynomial $P\in\mathbb{K}[x_1,\ldots,x_n]$ of total degree $\leqslant d$ can be evaluated at $N=(d^n)^\theta$ points in $\mathbb{K}^n$ using $\tilde{O}((N+d^n)^{\kappa_n(\theta)})$ operations in $\mathbb{K}$, where*

$$\kappa_n(\theta):=\begin{cases} 1 & \text{if } \theta\leqslant\frac{n-1}{(n-2)\,\omega+1}\\ \theta+\frac{(\omega-1)\,(n-2)}{(n-2)\,\omega+1} & \text{if } \frac{n-1}{(n-2)\,\omega+1}\leqslant\theta\leqslant 1\\ 1+\frac{(\omega-1)\,(n-2)}{(n-2)\,\omega+1}\frac{1}{\theta} & \text{if } 1\leqslant\theta.\end{cases}$$

Proof. Again, we subdivide $\boldsymbol{\alpha}$ into subsequences $\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_L$ of cardinality $\leqslant M$ such that $M=O(d^{n-1})$. We expand P as a polynomial in x_n,

$$P(x_1,\ldots,x_n)=\sum_{0\leqslant i\leqslant d}P_i(x_1,\ldots,x_{n-1})\,x_n^i,$$

and let $\pi\colon\mathbb{K}^n\to\mathbb{K}^{n-1}$ denote the projection $(x_1,\ldots,x_n)\mapsto(x_1,\ldots,x_{n-1})$.

We apply Theorem 3.8 successively with $\pi(\boldsymbol{\alpha}_1),\ldots,\pi(\boldsymbol{\alpha}_L)$ in the role of $\boldsymbol{\alpha}$. The total cost of the precomputations is

$$\tilde{O}(LM^{1+(\omega-1)(n-2)/(n-1)})\,(\log d)^{O(1)},$$

after which the cost of the evaluations of $P_0,\ldots,P_d$ at $\pi(\boldsymbol{\alpha})$ becomes

$$\tilde{O}(Ld^n).$$

We deduce $P(\boldsymbol{\alpha})$ in time $O(dN)$. Now we set

$$M:=\lceil(d^n)^{1/(1+(\omega-1)(n-2)/(n-1))}\rceil \text{ and } L:=\lceil N/M\rceil.$$

Using $n\geqslant 3$, we verify that $M=O(d^{n-1})$. In total, the computation of $P(\boldsymbol{\alpha})$ costs

$$\begin{aligned}&\tilde{O}(L\,(M^{1+(\omega-1)(n-2)/(n-1)}+d^n)+dN)\\ =\;&\tilde{O}(Ld^n+dN)\\ =\;&\tilde{O}((N/M+1)\,d^n+dN)\\ =\;&\tilde{O}(Nd^{n-n/(1+(\omega-1)(n-2)/(n-1))}+d^n+dN)\\ =\;&\tilde{O}(Nd^{n(\omega-1)(n-2)/((n-2)\omega+1)}+d^n+dN).\end{aligned}$$

Still using $n\geqslant 3$, we note that

$$\frac{n\,(\omega-1)\,(n-2)}{(n-2)\,\omega+1}-1=\frac{(n-1)\,((n-2)\,\omega-n+1)}{(n-2)\,\omega+1}\geqslant 0.$$

Consequently, the total cost simplifies to $\tilde{O}(Nd^{n(\omega-1)(n-2)/((n-2)\omega+1)}+d^n)$. □

Remark 3.14. Since the map $t\mapsto\frac{t}{(t-1)\,\omega+1}$ is decreasing for $t\geqslant 1$, we have

$$\frac{n}{(n-1)\,\omega+1}<\frac{n-1}{(n-2)\,\omega+1}.$$

If $\frac{n-1}{(n-2)\,\omega+1}\leqslant\theta\leqslant 1$, then

$$\mu_n(\theta)-\kappa_n(\theta)\;=\;\frac{\omega-1}{((n-2)\,\omega+1)\,((n-1)\,\omega+1)}>0.$$

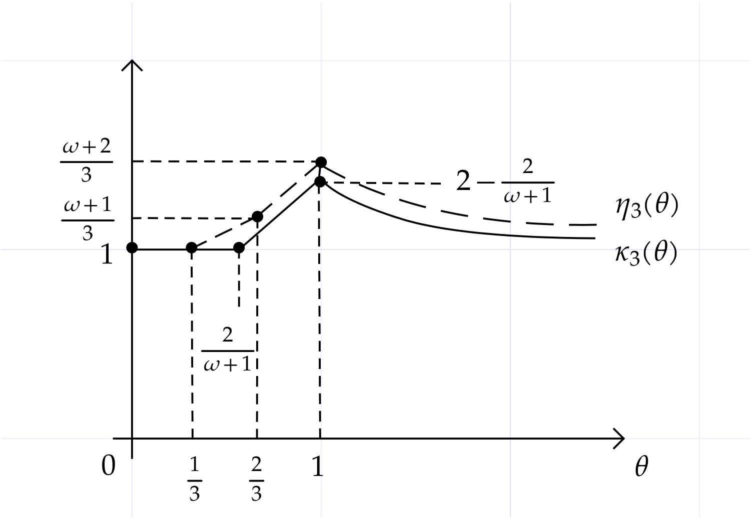

Figure 3.2. Complexity exponent $\kappa_3(\theta)$ for the evaluation of a polynomial in three variables of degree d at $d^{3\theta}$ points via Theorem3.13.

Consequently, if $n\geqslant 3$, then Theorem 3.13 always improves upon Theorem 3.9.

Remark 3.15. The function $\kappa_3(\theta)$ from Theorem 3.13 is plotted in Figure 3.2. In comparison with Theorem 2.6, we note that $\kappa_3(1)=\eta_3(1)$ at the limit when $\omega=2$. However, for the best currently known upper bound for ω, we have $\kappa_3(\theta)<\eta_3(\theta)$ for all $\theta>1/3$, even when taking into account Remark 2.4. More precisely, for $\omega\approx 2.371552$, we have $\kappa_3(1)\approx 1.406801$. At the same time, for $\omega_2\approx 3.250385$, the exponent of the Nüsken–Ziegler algorithm is $(\omega_2+1)/3\approx 1.416795$.

Remark 3.16. For $\theta=1$, it is instructive to compare Theorem 3.13 with the best Nüsken–Ziegler style complexity bound $\tilde{O}(d^{(n-1)(\omega_2/2)+1})$ of Remark 2.4:

- If $\omega=3$ and $\omega_2=4$, then the bound from Theorem 3.13 is always better.
- For the best currently known values $\omega\approx 2.371552$ and $\omega_2\approx 3.250385$, we verify numerically that
$$1+\frac{(\omega-1)(n-2)}{(n-2)\omega+1}=\kappa_n(1)<\frac{(n-1)(\omega_2/2)+1}{n}$$
if and only if $n=3$ or $n\geqslant 7$.
- At the limit when $\omega=2$ and $\omega_2=3$, we have $\eta_3(1)=\kappa_3(1)={}^4\!/_3$ and $\eta_n(1)<\kappa_n(1)$ for $n>3$.

APPENDIX A. ELEMENTS OF LARGE ORDER

Let $\mathbb{K}$ be an effective field, and let $\mathbb{K}^\times:=\mathbb{K}\setminus\{0\}$ denote its multiplicative group. We are interested in finding elements of $\mathbb{K}^\times$ of a sufficiently high order, possibly after replacing $\mathbb{K}$ by a suitable extension. The following algorithm finds such elements whenever the cardinality of $\mathbb{K}$ is sufficiently large.

Algorithm A.1

Input. A subset $\mathcal{S}$ of $\mathbb{K}^\times$ of cardinality $N\geqslant 1$.

Output. An element of $\mathbb{K}^{\times}$ of order $\geqslant N$.

```
1. Set $g := 1$, $m := 1$, and $\ell := \lfloor \log_2 N \rfloor$.
2. For each element $a$ in $\mathcal{S}$ do:
    a. If $a^m = 1$ then continue the loop of step 2 with the next element in $\mathcal{S}$.
    b. Compute $a^i$ for $i = 2, \ldots, N-1$.
    c. If the order $e$ of $a$ is $\geqslant N$, then return $a$.
    d. Compute $d := \gcd(m, e)$, $\tilde{e} := \gcd(e, (e/d)^{\ell})$, $\tilde{m} := m / \gcd(m, (e/d)^{\ell})$, $\tilde{a} := a^{e/\tilde{e}}$,
       and $\tilde{g} := g^{m/\tilde{m}}$.
    e. Compute the integers $u$ and $v$ of the Bézout relation $1 = u\tilde{m} + v\tilde{e}$. Replace $g$
       by $\tilde{g}^u \tilde{a}^v$ and $m$ by $\tilde{m}\tilde{e}$.
    f. If $m \geqslant N$ then return $g$.
```

PROPOSITION A.1. *Algorithm A.1 is correct and performs $O(N \log N)$ operations in $\mathbb{K}$.*

Proof. Let $\mathcal{G}$ denote the current subset of the elements of $\mathcal{S}$ that have been handled when entering step 2.a. We will show by induction that the elements of $\mathcal{G}$ have order dividing m, and that g has order m. Of course, these properties hold at the beginning, when $\mathcal{G}$ is empty.

If $a^m = 1$ then the induction hypothesis is preserved. If the algorithm exits at step 2.c, then the output is clearly correct. Otherwise, the order $e \leqslant N-1$ of a can be read off from the computations of step 2.b.

Let $p_1, \ldots, p_s$ be the prime numbers occurring in the factorization of $m e$, of respective multiplicities $e_1, \ldots, e_s$ in e and $m_1, \ldots, m_s$ in m, so we have

$$e = p_1^{e_1} \cdots p_s^{e_s} \text{ and } m = p_1^{m_1} \cdots p_s^{m_s}.$$

Some of the e_i or m_i may be zero here. Since e and m are at most N, the e_i and m_i are at most ℓ. From

$$d = \gcd(e, m) = \prod_{1 \leqslant i \leqslant s} p_i^{\min(e_i, m_i)},$$

and since e does not divide m, we note that the integer

$$e/d = \prod_{1 \leqslant i \leqslant s} p_i^{e_i - \min(e_i, m_i)}$$

is at least $\geqslant 2$ and only divisible by the primes p_i such that $e_i > m_i$. It follows that

$$\tilde{e} = \prod_{\substack{1 \leqslant i \leqslant s \\ e_i > m_i}} p_i^{e_i}, \quad \gcd(m, (e/d)^{\ell}) = \prod_{\substack{1 \leqslant i \leqslant s \\ e_i > m_i}} p_i^{m_i}, \quad \tilde{m} = \prod_{\substack{1 \leqslant i \leqslant s \\ e_i \leqslant m_i}} p_i^{m_i},$$

hence $\tilde{m}$ and $\tilde{e}$ are coprime and

$$\tilde{m}\tilde{e} = m \prod_{\substack{1 \leqslant i \leqslant s \\ e_i > m_i}} p_i^{e_i - m_i} \geqslant 2m.$$

Now $\tilde{g}$ has order $\tilde{m}$ and $\tilde{a}$ has order $\tilde{e}$, whence $(\tilde{g}^v \tilde{a}^u)^{\tilde{m}\tilde{e}} = 1$. If p is a prime divisor of $\tilde{m}$, then

$$(\tilde{g}^v \tilde{a}^u)^{(\tilde{m}\tilde{e})/p} = \tilde{g}^{v\tilde{e}(\tilde{m}/p)} = \tilde{g}^{(1-u\tilde{m})(\tilde{m}/p)} = \tilde{g}^{(\tilde{m}/p)} \neq 1.$$

Similarly, if p is a prime divisor of $\tilde{e}$, then $(\tilde{g}^v \tilde{a}^u)^{(\tilde{m}\tilde{e})/p} \neq 1$. Consequently, $\tilde{g}^v \tilde{a}^u$ has order $\tilde{m}\tilde{e}$. In particular, if the algorithm exits in step 2.f, then the output is correct. From

$$\tilde{m}\tilde{e} = m \prod_{\substack{1 \leqslant i \leqslant s \\ e_i > m_i}} p_i^{e_i - m_i} = \prod_{\substack{1 \leqslant i \leqslant s \\ e_i \leqslant m_i}} p_i^{m_i} \prod_{\substack{1 \leqslant i \leqslant s \\ e_i > m_i}} p_i^{e_i}$$

we note that e and m divide $\tilde{m}\tilde{e}$. Therefore the induction hypothesis again holds at the end of step 2.

Since the orders of the elements of $\mathcal{G}$ divide m, they are roots of the polynomial $x^m - 1$, whence $|\mathcal{G}| \leqslant m$. This shows that the algorithm works as expected.

Steps 2.a, 2.d, and 2.e take $O(\log N)$ operations in $\mathbb{K}$. Step 2.b takes $O(N)$ operations. In step 2.e, the integer m is at least doubled, so step 2.b can occur only $O(\log N)$ times. Overall, the total cost is $O(N \log N)$. We finally observe that the gcds and the Bézout relations can be computed using a negligible number of $\tilde{O}((\log N)^2)$ bit operations. In order to be painstakingly precise, we note that such bit operations can be emulated by operations over $\mathbb{K}$ in our algebraic complexity model. □

In many usual cases, it is known that Algorithm A.1 is suboptimal. In particular, if the characteristic of $\mathbb{K}$ is zero, then 2 has always order $\geqslant N$. If $\mathbb{K} = \mathbb{F}_q$ is a finite field, then $\mathbb{K}^\times$ is cyclic and primitive elements can be obtained more efficiently; see [8, 28, 29] for instance, but also the survey [7]. As an advantage, Algorithm A.1 is field agnostic. This property seems mostly of theoretical interest, but it turns out to be useful when programming generic algorithms, that do not require specific properties of $\mathbb{K}$: for instance the characteristic or the cardinality of $\mathbb{K}$ are not computable. The following proposition explains how to use Algorithm A.1 even without any piece of information about $\mathbb{K}$.

PROPOSITION A.2. *Let T be a computation tree of total cost L over $\mathbb{K}$, such that*

- *One of the input values of T must contain an element ζ of order $\geqslant N$;*
- *The output values of T are independent of the value taken for ζ, even when ζ is taken in an algebraic extension $\mathbb{E}$ of $\mathbb{K}$ and when T is evaluated over $\mathbb{E}$.*

Then there exists a computation tree T' of total cost

$$O(L\,\mathsf{M}(\log N) \log\log N + N \log N)$$

that computes the same as T but without requiring an input element of order $\geqslant N$.

Proof. We are interested in computing an element of order $\geqslant N$. First, we can compute the sequence of integers $1, 2, \ldots, N$ in $\mathbb{K}$ using $O(N)$ additions, and then determine whether char $\mathbb{K}$ is $>N$ or not. If char $\mathbb{K} > N$, then we can use Algorithm A.1 in order to obtain an element ζ of $\mathbb{K}$ of order $\geqslant N$. In this case, T' simply runs T with ζ.

Otherwise, $p := \operatorname{char} \mathbb{K} \leqslant N$. We shall compute in a sufficiently large algebraic extension of $\mathbb{K}$ as follows. Let $e := \lceil \log(N+1) / \log p \rceil$ be the first integer such that

$$p^e - 1 \geqslant N.$$

Thanks to [27, Theorem 3.2], we may deterministically construct a monic irreducible polynomial $\mu(z) \in \mathbb{F}_p[z]$ of degree e in $\mathbb{F}_p[z]$ using $\tilde{O}(p^{1/2})\, e^{O(1)} = O(N)$ operations in $\mathbb{K}$. In this way, $\mathbb{F}_p[z] / (\mu(z))$ is the finite field with p^e elements. We can enumerate N non-zero elements of $\mathbb{F}_p[z] / (\mu(z))$ and then use Algorithm A.1 in order to obtain an element ζ of $\mathbb{F}_p[z] / (\mu(z))$ of order $\geqslant N$. We regard $\zeta(z)$ as a polynomial in $\mathbb{K}[z]$ of degree $<e$.

Now we regard T as a computation tree over $\mathbb{K}[z]/(\mu(z))$, using ζ in place of the input element of order $\geqslant N$. While executing T for given input in $\mathbb{K}$, sums and products can be lifted straightforwardly to $\mathbb{K}[z]/(\mu(z))$: elements of $\mathbb{K}[z]/(\mu(z))$ are represented by polynomials of degree $<e$. When testing whether an element

$$a(z)\in\mathbb{K}[z]/(\mu(z))$$

is invertible or not, we proceed as follows:

- If $a(z)$ is identically zero, then the test returns false.
- If $a(z)$ is invertible modulo $\mu(z)$, then the test returns true.
- Otherwise, μ can be factored $\mu=\mu_1\mu_2$, with $\mu_1:=\gcd(\mu,a)$, $\deg\mu_1\geqslant 1$, and $\deg\mu_2\geqslant 1$. We continue our computations with μ_1 in the role of μ, while projecting all previously computed results from $\mathbb{K}[z]/(\mu(z))$ in $\mathbb{K}[z]/(\mu_1(z))$. In particular, $a(z)$ becomes identically zero after this projection, and the test returns false.

Note that $\mathbb{F}_q$ remains embedded in $\mathbb{K}[z]/(\mu(z))$ whenever μ is replaced by any non-trivial factor over $\mathbb{K}$. In particular, the order of ζ remains $\geqslant N$ after any such replacement. At the end, we thus obtain the evaluation of T over $\mathbb{K}[z]/(\mu^*(z))$, where μ^* is a non-constant factor of the original μ. This proves the correctness of our method.

Computing ζ takes $O(N\log N)$ operations thanks to Proposition A.1. The evaluation of T over $\mathbb{K}[z]/(\mu(z))$ requires L ring operations or extended gcd computations for polynomials over $\mathbb{K}$ of degree at most e. This contributes $O(L\,\mathsf{M}(e)\log e)=O(L\,\mathsf{M}(\log N)\log\log N)$ to the cost. The overall cost is therefore as claimed. □

BIBLIOGRAPHY

[1] S. Abelard, A. Couvreur, and G. Lecerf. Efficient computation of Riemann–Roch spaces for plane curves with ordinary singularities. *Appl. Algebra Eng. Commun. Comput.*, 35(6):739–804, 2024.

[2] V. Bhargava, S. Ghosh, Z. Guo, M. Kumar, and C. Umans. Fast multivariate multipoint evaluation over all finite fields. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 221–232. New York, NY, USA., 2022. IEEE.

[3] V. Bhargava, S. Ghosh, M. Kumar, and C. K. Mohapatra. Fast, algebraic multivariate multipoint evaluation in small characteristic and applications. *J. ACM*, 2023. Article 42.

[4] 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.

[5] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der Mathematischen Wissenschaften*. Springer-Verlag, 1997.

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

[7] Q. Cheng. On the construction of finite field elements of large order. *Finite Fields their Appl.*, 11(3):358–366, 2005.

[8] S. Gao. Elements of provable high orders in finite fields. *Proc. Am. Math. Soc.*, 127(6):1615–1623, 1999.

[9] J. von zur Gathen and J. Gerhard. *Modern computer algebra.* Cambridge University Press, New York, 3rd edition, 2013.

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

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

[12] J. van der Hoeven and R. Larrieu. Fast reduction of bivariate polynomials with respect to sufficiently regular Gröbner bases. In C. Arreche, editor, *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*, ISSAC '18, pages 199–206. New York, NY, USA, 2018. ACM.

[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. van der Hoeven and G. Lecerf. Amortized multi-point evaluation of multivariate polynomials. *J. Complexity*, 74:101693, 2022.

[18] J. van der Hoeven, G. Lecerf, B. Mourrain et al. Mathemagix. 2002. `http://www.mathemagix.org`.

[19] J. van der Hoeven and É. Schost. Multi-point evaluation in higher dimensions. *Appl. Alg. Eng. Comm. Comp.*, 24(1):37–52, 2013.

[20] K. S. Kedlaya and C. Umans. Fast polynomial factorization and modular composition. *SIAM J. Comput.*, 40(6):1767–1802, 2011.

[21] 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.

[22] V. Neiger. *Bases of relations in one or several variables: fast algorithms and applications*. PhD thesis, École Normale Supérieure de Lyon (France) – University of Waterloo (Canada), 2016. `https://tel.archives-ouvertes.fr/tel-01431413`.

[23] V. Neiger. Fast computation of shifted Popov forms of polynomial matrices via systems of modular polynomial equations. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '16, pages 365–372. New York, NY, USA, 2016. ACM.

[24] 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.

[25] V. Neiger, B. Salvy, É. Schost, and G. Villard. Faster modular composition. *J. ACM*, 71(2):1–79, 2023. Article No. 11.

[26] 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.

[27] V. Shoup. New algorithms for finding irreducible polynomials over finite fields. *Math. Comp.*, 54(189):435–447, 1990.

[28] V. Shoup. Searching for primitive roots in finite fields. *Math. Comp.*, 58:369–380, 1992.

[29] I. Shparlinski. On finding primitive roots in finite fields. *Theor. Comput. Sci.*, 157(2):273–275, 1996.

[30] V. V. Williams, Y. Xu, Z. Xu, and R. Zhou. New bounds for matrix multiplication: from alpha to omega. In D. P. Woodruff, editor, *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3792–3835. Philadelphia, PA 19104 USA, 2024. SIAM.