SIAM J. COMPUT.
Vol. 32, No. 2, pp. 435–454

ACCELERATED SOLUTION OF MULTIVARIATE POLYNOMIAL SYSTEMS OF EQUATIONS*

B. MOURRAIN†, V. Y. PAN‡, AND O. RUATTA†

Abstract. We propose new Las Vegas randomized algorithms for the solution of a square nondegenerate system of equations, with well-separated roots. The algorithms use $\mathcal{O}(\delta\, 3^n D^2 \log(D)\log(b))$ arithmetic operations (in addition to the operations required to compute the normal form of the boundary monomials modulo the ideal) to approximate all real roots of the system as well as all roots lying in a fixed n-dimensional box or disc. Here D is an upper bound on the number of all complex roots of the system (e.g., Bezout or Bernshtein bound), δ is the number of real roots or the roots lying in the box or disc, and $\epsilon = 2^{-b}$ is the required upper bound on the output errors. For computing the normal form modulo the ideal, the efficient practical algorithms of [B. Mourrain and P. Trébuchet, in *Proceedings of the International Symposium on Symbolic and Algebraic Computation,* ACM, New York, 2000, pp. 231–238] or [J. C. Faugère, *J. Pure Appl. Algebra,* 139 (1999), pp. 61–88] can be applied. We also yield the bound $\mathcal{O}(3^n D^2 \log(D))$ on the complexity of counting the numbers of all roots in a fixed box (disc) and all real roots. For a large class of inputs and typically in practical computations, the factor δ is much smaller than D, $\delta = o(D)$. This improves by the order of magnitude the known complexity estimates of the order of at least $3^n D^4 + D^3 \log(b)$ or D^4, which so far are the record estimates even for the approximation of a single root of a system and for each of the cited counting problems, respectively. Our progress relies on proposing several novel techniques. In particular, we exploit the structure of matrices associated to a given polynomial system and relate it to the associated linear operators, dual space of linear forms, and normal forms of polynomials in the quotient algebra; furthermore, our techniques support the new nontrivial extension of the matrix sign and quadratic inverse power iterations to the case of multivariate polynomial systems, where we emulate the recursive splitting of a univariate polynomial into factors of smaller degree.

Key words. multivariate polynomials, systems of equations, quotient algebras, dual spaces, quasi-Toeplitz matrices, quasi-Hankel matrices, matrix sign iteration, quadratic power iteration

AMS subject classifications. 65H10, 68W30, 68W25, 68W20

PII. S0097539701385168

1. Introduction.

The classical problem of solving a multivariate polynomial system of equations is presently the subject of intensive research and one of the central practical and theoretical problems in the area of algebraic computation (see [21], [5], [32], [15].) It has major applications, for instance, to robotics, computer modelling and graphics, molecular biology, and computational algebraic geometry.

The oldest approach to the solution is the elimination method, reducing the problem to the computation of the associated resultant or its multiples. This classical method evolved in the old works by Bezout, Dixon, and Macaulay (see, e.g., [21], [45]) but later remained largely ignored by the researchers and algorithm designers until it was resurrected first by Chistov and Grigoriev [8], who designed a deterministic solution algorithm, then in a randomized approach by Canny [4], and later by

*Received by the editors February 20, 2001; accepted for publication (in revised form) September 19, 2002; published electronically February 4, 2003. Some results of this paper were presented at the 30th Annual ACM Symposium on Theory of Computing in 1998 and at the Smalefest (Hong Kong, 2000).

http://www.siam.org/journals/sicomp/32-2/38516.html

†Inria, Galaad, BP 93, 06902 Sophia-Antipolis, France (mourrain@sophia.inria.fr, oruatta@sophia.inria.fr).

‡Department of Mathematics and Computer Science, Lehman College, City University of New York, Bronx, NY 10468 (vpan@lehman.cuny.edu). The research of this author was supported by NSF grant CCR 9732206 and PSC CUNY award 62435-0031.

Giusti and Heintz [18] and has since become a very popular approach. One of the major further steps was the reduction of the solution of a multivariate polynomial system to matrix operations, in particular, by rational transformation of the original problem into a matrix eigenproblem (cf. [1], [16], [15], [27], [25], [10]).

The approach has been explored and extended by many researchers, has been exploited in the practice of algebraic computing, and has also supported the record asymptotic upper bound $\mathcal{O}^*(D^4)$ on the arithmetic computational complexity of the solution of a nondegenerate polynomial system having a finite number of roots [40]. Here and hereafter, $\mathcal{O}^*(s)$ stands for $\mathcal{O}(s \log^c s)$, c denoting a constant independent of s, and D is an upper bound on the number of roots of the given polynomial system. (For D, one may choose either the Bezout bound, $\prod_i d_i$, d_i denoting the maximum degree in the ith variable in all monomials of the system, or the Bernshtein bound, which is much smaller for sparse systems and equals the mixed volume of the associated Newton polytope, defined by the exponents of the monomials.) Even for many subproblems and related problems, no known algorithms support any better bound than $\mathcal{O}(D^4)$. This includes approximation of all real roots of a polynomial system (which is highly important due to applications to robotic and computer graphics), all its roots lying in a fixed n-dimensional box or disc, counting all roots in such a box or disc or all real roots, and even approximation of a single root. Some progress was achieved in [30], where a single root was approximated in $\mathcal{O}^*(3^n D^2)$ time, but under a strong restriction on the input polynomials.

Against this background, our new algorithms support the computational cost estimate of $\mathcal{O}^*(3^n D^2)$ for all of the subproblems listed above, that is, for both of the counting problems, the computation of a single root, all real roots, and all roots in a fixed box or disc. More precisely, our bound is $\mathcal{O}^*(\delta\, 3^n D^2)$ in the latter two cases, where δ is the number of real roots or roots in the selected box or disc, respectively. In practical applications, such a number is typically much less than D. The number of real roots grows as $\sqrt{D}$ for a large class of input systems [41]. See also the sparse case [24]. Thus, for all listed problems, we improve the known complexity estimates by an order of magnitude.

We have a reservation from a theoretical point of view; that is, our main algorithm relies on the known effective algorithms for the computation of the normal form of monomials on the boundary of the monomial basis (see section 4). These algorithms exploit structured matrices and in practice appear to run faster than our subsequent computations (see [17], [33]), but their known theoretical cost bounds are greater than the order of $e^{3n} D^3$ (see [22]).

Our paper addresses the problem of the asymptotic acceleration of the resolution stage, where the structure of the quotient algebra $\mathcal{A}$ (associated with the polynomial system) is already described by using the minimal number of parameters, that is, via the normal form of the monomials on the boundary of the basis. From a purely theoretical point of view, we have an alternative approach that avoids the normal-form algorithms at the price of using the order of $\mathcal{O}(12^n\, D^2)$ additional arithmetic operations [31]. This should be technically interesting because no other known approach yields this bound, but in this paper, we prefer to stay with our present, practically promising version, referring the reader to [31] on the cited theoretical approach. Our practically promising solution relies on fast computation of normal forms of polynomials modulo the ideal, based on the algorithm of [33]. Some limited amount of experimental evidence to the efficiency of this algorithm has been reported in [33], and further experimentation is ongoing.

Our algorithms approximate the roots numerically, and in terms of the required upper bound 2^{-b} (b is the bit precision) on the output errors of the computed solution, we obtain the running time estimate $\mathcal{O}(\log b)$ due to quadratic convergence of our algorithms. Within a constant factor, such an estimate matches the lower bound of [39] and enables us to yield a high output precision at relatively low cost; this gives us a substantial practical advantage versus the algorithms that reach only $\mathcal{O}(b)$ because the solution of a polynomial system is usually needed with a high precision. We achieve this by using the matrix sign iteration and the inverse quadratic iteration, both of which converge at a quadratic rate right from the start. All techniques and results can be extended to the case of sparse input polynomials (see Remark 3.16). In this case, the computation cost bounds become $\mathcal{O}(D\,C_{PolMult})$, where $C_{PolMult}$ is the cost of polynomial multiplication, which is small when the polynomials are sparse. (This cost depends on the degree of the polynomials and not only on an upper bound D on the number of roots.)

The factor 3^n is a substantial deficiency, of course, but it is still much less than D for the large and important class of input polynomials of degree higher than 3.

Our results require some other restrictions. First, we consider systems with simple roots or well-separated roots. In the presence of a cluster, a specific analysis is needed [43] and deserves additional work, which is not in the scope of this paper. Second, we need the existence of a nondegenerate linear form, which implies that the quotient algebra $\mathcal{A}$ is a Gorenstein algebra [12], [14]. This is the case in which the solution set is 0-dimensional and is defined by n equations. If we have more than n equations defining a 0-dimensional variety, we may take their n-random linear combination (see, e.g., [13]), which yields the required Gorenstein property, but this may introduce extra solutions that we will have to remove at the end. Finally, for approximation, our algorithms converge quadratically (using $\mathcal{O}(\log(b))$ steps) but require certain nondegeneracy assumptions (such as uniqueness of the minimum of the value of $|h(\zeta)|$, where ζ is a root and $h(x)$ is a polynomial). The latter assumptions can be ensured with a high probability by a random linear transformation of the variables. Even if these assumptions are barely satisfied, the slowdown of the convergence is not dramatic because the convergence is quadratic right from the start.

Similarly, we apply randomization to regularize the computations at the counting stages and for the auxiliary computation of the nondegenerate linear form in the dual space $\widehat{\mathcal{A}}$. Then again, nondegeneracy is ensured probabilistically and is verified in the subsequent computation. (That is, we stay under the Las Vegas probabilistic model, where failure may occur, with a small probability, but otherwise the correctness of the output is ensured.)

Some of our techniques should be of independent interest. In particular, we extend the theory of structured matrices to the ones associated to multivariate polynomials and show correlation among computations with such matrices and dual spaces of linear forms. We show some new nontrivial applications of the normal forms of polynomials of the quotient algebra. Furthermore, we establish new reduction from multivariate polynomial computations to some fundamental operations of linear algebra (such as the matrix sign iteration, the quadratic inverse power iteration, and the computation of Schur's complements).

Our progress has some technical similarity to the acceleration of the solution of linear systems of equations via fast matrix multiplication (in particular, we also rely on faster multiplication in the quotient algebra defined by the input polynomials) but even more so to the recent progress in the univariate polynomial rootfinding

via recursive splitting of the input polynomial into factors (cf. [6], [34], [36], [37]). Although recursive splitting into factors may be hard to even comprehend in the case of multivariate polynomial systems, this is exactly the basic step of our novel recursive process, which finally reduces our original problem to ones of small sizes. Of course, we could not achieve splitting in the original space of the variables, but we yield it in terms of idempotent elements of the associated quotient algebra (such elements represent the roots), and for this purpose we have to apply all of our advanced techniques. This approach generalizes the methods of [6] and [36] to the multivariate case. The only missing technical point of our extension of the univariate splitting construction of [36] is the balancing of the splitting, which was the most recent and elusive step in the univariate case (cf. [36], [37]). It is a major challenge to advance our approach to achieve balancing in our recursive splitting process even in the worst case (possibly by using the geometry of discriminant varieties) and, consequently, to approximate all of the roots of any specific polynomial system in $\mathcal{O}^*(3^n D^2 \log b)$ arithmetic time. Another goal is the computations in the dual space, as well as with structured matrices. The latter subject is of independent interest as well [44], [32].

Let us conclude this section with a high-level description of our approach. Our solution of polynomial systems consists of the following stages:

1. Compute a basic nondegenerate linear form on the quotient algebra $\mathcal{A}$ associated to a given system of polynomial equations.

2. Compute nontrivial idempotent elements of $\mathcal{A}$.

3. Recover the roots of the given polynomial system from the associated idempotents.

The quotient algebra $\mathcal{A}$ and the dual space of linear forms on it are defined and initially studied in section 2. Stage 1 is elaborated in section 4. Idempotents are computed by iterative algorithms of section 6. Section 7 shows how to recover or to count the roots efficiently when the idempotents are available. The computations are performed in the quotient algebra, and they are reduced to operations in the dual space by using the associated structured (quasi-Toeplitz and quasi-Hankel) matrices. In section 3, we define the classes of such matrices, show their correlation to polynomial computations, and exploit it to operate with such matrices faster. In section 5, we show how the combined power of the latter techniques and the ones developed for working in the dual space enable us to rapidly perform the basic operations in the quotient algebra and, consequently, the computations of sections 6 and 7.

Stage 1 contributes $O(3^n D^2 \log D)$ ops to the overall complexity bound, assuming that the normal form of the monomials on the boundary of a basis is known. The computation of a nontrivial idempotent at stage 2 has cost $O(3^n D^2 \log D \log b)$, which dominates the cost of the subsequent root counting or their recovery from the idempotents. The overall complexity depends on the number of idempotents that one has to compute, which in turn depends on the number δ of roots of interest. So far, we cannot utilize here the effective tools of balanced splitting, available in the similar situation for the univariate polynomial rootfinding. Thus, in the worst case, in each step we split out only a single root from the set of all roots, and then we need δ idempotents.

2. Definitions and preliminaries. Hereafter, $R = \mathbb{C}[x_1, \dots, x_n]$ is the ring of multivariate polynomials in the variables $x_1, \dots, x_n$, with coefficients in the complex field $\mathbb{C}$. $\mathbb{Z}$ is the set of integers, $\mathbb{N}$ is its subset of nonnegative integers, and $L = \mathbb{C}[x_1^{\pm}, \dots, x_n^{\pm}]$ is the set of Laurent polynomials with monomial exponents in $\mathbb{Z}^n$. For any $\mathbf{a} = (a_1, \dots, a_n) \in \mathbb{Z}^n$, $\mathbf{x}^{\mathbf{a}}$ is the monomial $\mathbf{x}^{\mathbf{a}} = x_1^{a_1} \cdots x_n^{a_n}$. $\lfloor E \rceil$ is the cardinality

(that is, the number of elements) of a finite subset E of $\mathbb{Z}^n$. "ops" will stand for "arithmetic operations" in the underlying coefficient ring or field.

2.1. Quotient algebra. To motivate and to demonstrate our study, we will next consider the univariate case, where we have a fixed polynomial $f \in \mathbb{C}[x]$ of degree d with d simple roots: $f(x) = f_d \prod_{i=1}^{d}(x-\zeta_i)$. The quotient algebra of residue polynomials modulo f, denoted by $\mathcal{A} = \mathbb{C}[x]/(f)$, is a vector space of dimension d. Its basis is $(1, x, \ldots, x^{d-1})$. Consider the Lagrange polynomials

$$\mathbf{e}_i = \prod_{j \neq i} \frac{x-\zeta_j}{\zeta_i - \zeta_j}.$$

One immediately sees that $\sum_i \mathbf{e}_i = 1$ and $\mathbf{e}_i\mathbf{e}_j \equiv \mathbf{e}_i(\mathbf{e}_i - 1) \equiv 0$ (for these two polynomials vanish at the roots of f). In other words, the Lagrange polynomials $\mathbf{e}_i$ are orthogonal idempotents in $\mathcal{A}$, and we have $\mathcal{A} = \sum_i \mathbb{C}\,\mathbf{e}_i$. Moreover, for any polynomial $a \in \mathcal{A}$, we also have $(a - a(\zeta_i))\mathbf{e}_i \equiv 0$, so that $\mathbf{e}_i$ is an eigenvector for the operator of multiplication by a in $\mathcal{A}$, for the eigenvalue $a(\zeta_i)$. These multiplication operators have a diagonal form in the basis $(\mathbf{e}_i)$ of $\mathcal{A}$. According to a basic property of Lagrange polynomials, we have $a \equiv \sum_i a(\zeta_i)\,\mathbf{e}_i(x)$ for any $a \in \mathcal{A}$. Therefore, the dual basis of $(\mathbf{e}_i)$ (formed by the coefficients of the $\mathbf{e}_i$ in this decomposition) consists of the linear forms associating to a its values at the points ζ_i. We will extend this approach to the case of multivariate polynomial systems, which, of course, will require substantial further elaboration and algebraic formalism. We refer the reader to [26], [27], [32], [42] for further details.

Let $f_1, \ldots, f_m$ be m polynomials of R, defining the polynomial system $f_1(x) = 0, \ldots, f_m(x) = 0$. Let I be the ideal generated by these polynomials, that is, the set of polynomial combinations $\sum_i f_i q_i$ of these elements. $\mathcal{A} = R/I$ denotes the quotient ring (algebra) defined in R by I, and $\equiv$ denotes the equality in $\mathcal{A}$. We consider the case in which the quotient algebra $\mathcal{A} = R/I$ *is of finite dimension* D *over* $\mathbb{C}$. This implies that the set of roots or solutions $\mathcal{Z}(I) = \{\zeta \in \mathbb{C}^n; f_1(\zeta) = \cdots = f_m(\zeta) = 0\}$ is finite: $\mathcal{Z}(I) = \{\zeta_1, \ldots, \zeta_d\}$ with $d \leq D$. Then we have a decomposition of the form

$$\mathcal{A} = \mathcal{A}_1 \oplus \cdots \oplus \mathcal{A}_d, \tag{1}$$

where $\mathcal{A}_i$ is a local algebra, for the maximal ideal $\mathbf{m}_{\zeta_i}$ defining the root ζ_i. From decomposition (1), we deduce that there exist orthogonal idempotents $\mathbf{e}_1, \ldots, \mathbf{e}_d$ satisfying

$$\mathbf{e}_1 + \cdots + \mathbf{e}_d \equiv 1 \text{ and } \mathbf{e}_i\,\mathbf{e}_j \equiv \begin{cases} 0 & \text{if } i \neq j, \\ \mathbf{e}_i & \text{if } i = j. \end{cases}$$

If $I = Q_1 \cap \cdots \cap Q_d$ is the minimal primary decomposition of I, we have $\mathbf{e}_i\mathcal{A} \sim R/Q_i$, where $\mathcal{A}_i = \mathbf{e}_i\,\mathcal{A}$ is a local algebra, for the maximal ideal $\mathbf{m}_{\zeta_i}$ defining the root ζ_i. Thus, to any root $\zeta \in \mathcal{Z}$, we associate an idempotent $\mathbf{e}_\zeta$.

2.2. Dual space. Let $\widehat{R}$ denote the dual of the $\mathbb{C}$-vector space R, that is, the space of linear forms

$$\begin{aligned} \Lambda : R &\to \mathbb{C}, \\ p &\mapsto \Lambda(p). \end{aligned}$$

(R will be the primal space for $\widehat{R}$.) Let us recall two celebrated examples, that is, the *evaluation at a fixed point* ζ,

$$\begin{aligned} \mathbf{1}_\zeta \,:\, & R \to \mathbb{C}, \\ & p \mapsto p(\zeta), \end{aligned}$$

and the map

$$\begin{aligned} (\mathbf{d}^{\mathbf{a}} = (\mathbf{d}_1)^{a_1} \cdots (\mathbf{d}_n)^{a_n}) : R &\to \mathbb{C} \\ p &\mapsto \frac{1}{\prod_{i=1}^n a_i!}\, (d_{x_1})^{a_1} \cdots (d_{x_n})^{a_n}\, (p)(0), \end{aligned} \tag{2}$$

where $\mathbf{a} = (a_1, \ldots, a_n)$ is any vector from $\mathbb{N}^n$ and d_{x_i} is the partial derivative with respect to the variable x_i. For any $\mathbf{b} = (b_1, \ldots, b_n) \in \mathbb{N}^n$, we have

$$\mathbf{d}^{\mathbf{a}}(\mathbf{x}^{\mathbf{b}}) = \begin{cases} 1 & \text{if } \forall i, a_i = b_i, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, $(\mathbf{d}^{\mathbf{a}})_{\mathbf{a}\in\mathbb{N}^n}$ is the dual basis of the primal monomial basis. Thus we decompose any linear form $\Lambda \in \widehat{R}$ as

$$\Lambda = \sum_{\mathbf{a}\in\mathbb{N}^n} \Lambda(\mathbf{x}^{\mathbf{a}})\, \mathbf{d}^{\mathbf{a}}. \tag{3}$$

Hereafter, *we will identify* $\widehat{R}$ *with* $\mathbb{C}[[\mathbf{d}_1, \ldots, \mathbf{d}_n]]$. The map $\Lambda \to \sum_{\mathbf{a}\in\mathbb{N}^n} \Lambda(\mathbf{x}^{\mathbf{a}})\, \mathbf{d}^{\mathbf{a}}$ defines a one-to-one correspondence between the set of linear forms Λ and the set $\mathbb{C}[[\mathbf{d}_1, \ldots \mathbf{d}_n]] = \mathbb{C}[[\mathbf{d}]] = \{\sum_{\mathbf{a}\in\mathbb{N}^n} \lambda_{\mathbf{a}} \mathbf{d}_1^{a_1} \cdots \mathbf{d}_n^{a_n}\}$ of polynomials in the variables $\mathbf{d}_1, \ldots, \mathbf{d}_n$.

The evaluation at 0 corresponds to the constant 1 under this definition. It will also be denoted by $\delta_0 = \mathbf{d}^0$.

We will denote by $\widehat{\mathcal{A}}$ and also by $I^\perp$ the subspace of $\widehat{R}$ made of those linear forms that vanish on the ideal I.

We now define multiplication of a linear form by a polynomial ($\widehat{R}$ is an R-module) as follows. For any $p \in R$ and $\Lambda \in \widehat{R}$, we write

$$\begin{aligned} p \star \Lambda : R &\to \mathbb{C}, \\ q &\mapsto \Lambda(p\, q). \end{aligned}$$

For any pair of elements $p \in R$ and $a \in \mathbb{N},\ a > 1$, we have

$$(d_{x_i})^a\, (x_i\, p)(0) = a\, (d_{x_i})^{a-1}\, p(0).$$

Consequently, for any pair $(p, \mathbf{a})$, $p \in R$, $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n$ (where $a_i \neq 0$ for a fixed i), we obtain

$$\begin{aligned} x_i \star \mathbf{d}^{\mathbf{a}}(p) &= \mathbf{d}^{\mathbf{a}}(x_i\, p) \\ &= \mathbf{d}_1^{a_1} \cdots \mathbf{d}_{i-1}^{a_{i-1}} \mathbf{d}_i^{a_i - 1} \mathbf{d}_{i+1}^{a_{i+1}} \cdots \mathbf{d}_n^{a_n}(p); \end{aligned}$$

that is, x_i acts as the *inverse* of $\mathbf{d}_i$ in $\mathbb{C}[[\mathbf{d}]]$. For this reason, such a representation is referred to as the *inverse systems* (see, for instance, [23]). If $a_i = 0$, then $x_i \star \mathbf{d}^{\mathbf{a}}(p) = 0$, which allows us to redefine the product $p \star \Lambda$ as follows.

PROPOSITION 2.1. *For any pair $p, q \in R$ and any $\Lambda(\mathbf{d}) \in \mathbb{C}[[\mathbf{d}]]$, we have*

$$p \star \Lambda(q) = \Lambda(p\,q) = \pi_+(p(\mathbf{d}^{-1})\,\Lambda(\mathbf{d}))(q),$$

where π_+ is the projection mapping Laurent series onto the space generated by the monomials in $\mathbf{d}$ with positive exponents.

This yields the following algorithm.

ALGORITHM 2.2. *For any polynomial $p \in \langle \mathbf{x}^\alpha \rangle_{\alpha \in E}$ and a vector $[\Lambda(\mathbf{x}^\beta)]_{\beta \in E+F}$, compute the vector $[p \star \Lambda(\mathbf{x}^\beta)]_{\beta \in F}$ as follows:*

- *Write $\tilde{\Lambda}(\mathbf{d}) = \sum_{\beta \in E+F} \Lambda(\mathbf{x}^\beta)\,\mathbf{d}^\beta$.*
- *Compute the product $\rho(\mathbf{d}) = p(\mathbf{d}^{-1})\tilde{\Lambda}(\mathbf{d})$ in $\mathbb{C}[\mathbf{d}, \mathbf{d}^{-1}]$.*
- *Keep the coefficients ρ_α of $\mathbf{d}^\alpha$ for $\alpha \in F$.*

3. Quasi-Toeplitz and quasi-Hankel matrices. In this section, we describe the structure of the matrices and some tools that we will use for our algorithm design.

Let us recall first the known arithmetic complexity bounds for polynomial multiplication (see [2, pp. 56–64]), which is the basic step of our subsequent algorithms. Let $C_{PolMult}(E, F)$ denote the number of ops (that is, of arithmetic operations) required for the multiplication of a polynomial with support in E by a polynomial with support in F.

THEOREM 3.1. *Let $E + F = \{\alpha^i = (\alpha_1^{(i)}, \ldots, \alpha_n^{(i)}),\ i = 1, \ldots, N\}$ with $|\alpha^{(i)}| = \sum_j \alpha_j^{(i)} = d_i$ for $i = 1, \ldots, N$ and $d = \max_i(d_i)$. Let $C_{K;Eval}(G)$ ops suffice to evaluate a polynomial with a support G on a set of K points. Then we have*

$$C_{PolMut}(E, F) = \mathcal{O}\left(C_{N;Eval}(E) + C_{N;Eval}(F) + N\left(\log^2(N) + \log(d)\right)\right).$$

Proof. Apply the evaluation-interpolation techniques to multiply the two polynomials (cf. [2]). That is, first evaluate the input polynomials on a fixed set of N points, then multiply pairwise the computed values to obtain the values of the product on the same set, and finally interpolate from these values and compute the coefficients of the product by applying the (sparse) polynomial interpolation algorithm (cf. [2]). By summarizing the computational cost estimates, we obtain the theorem. □

For special sets E and F, we have better bounds.

THEOREM 3.2. *Let $E_d = [0, \ldots, d-1] \subset \mathbb{N}$. Then*

$$C_{PolMult}(E_d, E_d) = \mathcal{O}(d \log(d)).$$

THEOREM 3.3. *Let $E_{\mathbf{c}} = \{(\alpha_1, \ldots, \alpha_n)\ ;\ 0 \le \alpha_i \le c_i - 1\}$, $E_{\mathtt{d}} = \{(\beta_1, \ldots, \beta_n)\ ;\ 0 \le \beta_i \le d_i - 1\}$, $c = \max\{c_1, \ldots, c_n\}$, and $d = \max\{d_1, \ldots, d_n\}$. Then we have*

$$C_{PolMult}(E_{\mathbf{c}}, E_{\mathtt{d}}) = \mathcal{O}(M \log(M)),$$

where $M = f^n$ and $f = c + d + 1$.

THEOREM 3.4. *Let $E_{f,n}$ be the set of exponents having total degree at most f in n variables. Then*

$$C_{PolMult}(E_{c,n}, E_{d,n}) = \mathcal{O}(T \log^2(T)),$$

where $T = \binom{n+c+d}{n}$ is the number of monomials of degree at most $c+d$ in n variables.

REMARK 3.5. *Theorems 3.1 and 3.3 correspond, respectively, to lattice points in a product of intervals and in the scaled standard simplex and can be extended to the computations over any ring of constants (rather than over the complex field) at*

the expense of increasing their complexity bounds by at most the factors of $\log\log(N)$ *or* $\log\log(M)$*, respectively* [2]. *Theorem* 3.4 *can be extended similarly to any field of constants having characteristic* 0.

Next, by following [32], [31], we will extend the definitions of Toeplitz and Hankel matrices to the multivariate case. As we will see, these structures are omnipresent when we solve polynomial systems.

DEFINITION 3.6. *Let* E *and* F *be two finite subsets of* $\mathbb{N}^n$*, and let* $M = (m_{\alpha,\beta})_{\alpha\in E,\beta\in F}$ *be a matrix whose rows are indexed by the elements of* E *and columns by the elements of* F*. Let* $\mathtt{i}$ *denote the* i*th basis coordinate vector of* $\mathbb{N}^n$*.*

- $M = [m_{\alpha,\beta}]_{\alpha\in E,\beta\in F}$ *is an* (E,F) quasi-Toeplitz *matrix if and only if, for all* $\alpha \in E, \beta \in F$*, the entries* $m_{\alpha,\beta} = t_{\alpha-\beta}$ *depend only on* $\alpha - \beta$*, that is, if and only if, for* $i = 1,\dots,n$*, we have* $m_{\alpha+\mathtt{i},\beta+\mathtt{i}} = m_{\alpha,\beta}$*, provided that* $\alpha, \alpha+\mathtt{i} \in E; \beta, \beta+\mathtt{i} \in F$*; such a matrix* M *is associated with the polynomial* $T_M(\mathbf{x}) = \sum_{\mathbf{u}\in E+F} t_{\mathbf{u}}\,\mathbf{x}^{\mathbf{u}}$.
- M *is an* (E,F) quasi-Hankel *matrix if and only if, for all* $\alpha \in E, \beta \in F$*, the entries* $m_{\alpha,\beta} = h_{\alpha+\beta}$ *depend only on* $\alpha+\beta$*, that is, if and only if, for* $i = 1,\dots,n$*, we have* $m_{\alpha-\mathtt{i},\beta+\mathtt{i}} = m_{\alpha,\beta}$ *provided that* $\alpha, \alpha-i \in E; \beta, \beta+\mathtt{i} \in F$*; such a matrix* M *is associated with the Laurent polynomial* $H_M(\mathbf{d}) = \sum_{\mathbf{u}\in E-F} h_{\mathbf{u}}\mathbf{d}^{\mathbf{u}}$.

For $E = [0,\dots,m-1]$ and $F = [0,\dots,n-1]$ (resp., $F = [-n+1,\dots,0]$), Definition 3.6 turns into the usual definition of Toeplitz (resp., Hankel) matrices (see [2]). Quasi-Toeplitz matrices have also been studied under the name of multilevel Toeplitz matrices (see, e.g., [44]) in the restricted special case, where the sets E and F are rectangular (i.e., a product of intervals). For our study of polynomial systems of equations, using the latter restricted case is not sufficient, and our more general definitions are required.

The definitions can be extended immediately to all subsets E, F of $\mathbb{Z}^n$ if we work with the Laurent polynomials.

The classes of quasi-Toeplitz and quasi-Hankel matrices can be transformed into each other by means of multiplication by the reflection matrix, having ones on its antidiagonal and zeros elsewhere.

DEFINITION 3.7. *Let* $\pi_E : L \to L$ *be the projection map such that* $\pi_E(\mathbf{x}^\alpha) = \mathbf{x}^\alpha$ *if* $\alpha \in E$ *and* $\pi_E(\mathbf{x}^\alpha) = 0$ *otherwise. Also let* $\pi_E : \mathbb{C}[[\mathbf{d}]] \to \mathbb{C}[[\mathbf{d}]]$ *denote the projection map such that* $\pi_E(\mathbf{d}^\alpha) = \mathbf{d}^\alpha$ *if* $\alpha \in E$ *and* $\pi_E(\mathbf{d}^\alpha) = 0$ *otherwise.*

We can describe the quasi-Toeplitz and quasi-Hankel operators in terms of polynomial multiplication (see [30], [29]), and the next proposition reduces multiplication of an (E,F) quasi-Toeplitz (resp., quasi-Hankel) matrix by a vector $\mathbf{v} = [v_\beta] \in \mathbb{C}^F$ to (Laurent's) polynomial multiplication.

PROPOSITION 3.8. *The matrix* M *is an* (E,F) *quasi-Toeplitz (resp., an* (E,F) *quasi-Hankel) matrix if and only if it is the matrix of the operator* $\pi_E \circ \mu_{T_M} \circ \pi_F$ *(resp.,* $\pi_E \circ \mu_{H_M} \circ \pi_F$*), where, for any* $p \in L$*,* $\mu_p : q \mapsto p\,q$ *is the operator of multiplication by* p *in* L*.*

Proof (see [29]). We will give a proof only for an (E,F) quasi-Toeplitz matrix $M = (M_{\alpha,\beta})_{\alpha\in E,\beta\in F}$. (The proof is similar for a quasi-Hankel matrix.) The associated polynomial is $T_M(\mathbf{x}) = \sum_{\mathbf{u}\in E+F} t_{\mathbf{u}}\mathbf{x}^{\mathbf{u}}$. For any vector $\mathbf{v} = [v_\beta] \in \mathbb{C}^F$, let $v(\mathbf{x})$ denote

the polynomial $\sum_{\beta\in F} v_\beta \mathbf{x}^\beta$. Then

$$T_M(\mathbf{x})\, v(\mathbf{x}) = \sum_{\mathbf{u}\in E+F,\beta\in F} \mathbf{x}^{\mathbf{u}+\beta}\, t_{\mathbf{u}}\, v_\beta$$
$$= \sum_{\alpha=\mathbf{u}+\beta\in E+2\,F} \mathbf{x}^\alpha \left(\sum_{\beta\in F} t_{\alpha-\beta}\, v_\beta\right),$$

where we assume that $t_{\mathbf{u}} = 0$ if $\mathbf{u} \notin E + F$. Therefore, for $\alpha \in E$, the coefficient of $\mathbf{x}^\alpha$ equals

$$\sum_{\beta\in F} t_{\alpha-\beta}\, v_\beta = \sum_{\beta\in F} M_{\alpha,\beta}\, v_\beta,$$

which is precisely the coefficient α of $M\mathbf{v}$. □

ALGORITHM 3.9. *Multiplication of the (E, F) quasi-Toeplitz (resp., quasi-Hankel) matrix $M = (M_{\alpha,\beta})_{\alpha\in E,\beta\in F}$ by a vector $\mathbf{v} = [v_\beta] \in \mathbb{C}^F$:*

- *multiply the polynomials $T_M = \sum_{\mathbf{u}\in E+F} t_{\mathbf{u}}\, \mathbf{x}^{\mathbf{u}}$ (resp., $H_M(\mathbf{d}) = \sum_{\mathbf{u}\in E-F} h_{\mathbf{u}}\mathbf{d}^{\mathbf{u}}$) by $v(\mathbf{x}) = \sum_{\beta\in F} v_\beta \mathbf{x}^\beta$ (resp., $v(\mathbf{d}^{-1}) = \sum_{\beta\in F} v_\beta \mathbf{d}^{-\beta}$),*
- *and output the projection of the product on $\mathbf{x}^E$ (resp., $\mathbf{d}^E$).*

DEFINITION 3.10. *$C_{PolMult}(E, F)$ denotes the number of ops required to multiply a polynomial with a support in E by a polynomial with a support in F.*

Clearly, Algorithm 3.9 uses $C_{PolMult}(E+F, F)$ (resp., $C_{PolMult}(E-F, -F)$) ops.

PROPOSITION 3.11.

(a) *An (E, F) quasi-Hankel (resp., an (E, F) quasi-Toeplitz) matrix M can be multiplied by a vector by using $\mathcal{O}(N \log^2(N) + N \log(d) + C_{M,N})$ ops, where $d = \deg H_M$ (resp., $\deg T_M$) , $N = \lfloor E - 2\,F \rceil$ (resp., $\lfloor E + 2F \rceil$), and $C_{M,N}$ denotes the cost of the evaluation of all monomials of the polynomial H_M (resp., T_M) on a fixed set of N points.*

(b) *In particular, the ops bound becomes $\mathcal{O}(M \log(M))$, where $E + F = E_{\mathbf{c}}, F = E_{\mathbf{d}}$ and $E_{\mathbf{c}}, E_{\mathbf{d}}$ and $M = (c + d + 1)^n$ are defined as in Theorem 3.3, whereas*

(c) *the bound turns into $\mathcal{O}(T \log^2(T))$, where $E + F = E_{c,n}, F = E_{d,n}$ and $E_{c,n}, E_{d,n}$, and $T = \binom{n+c+d}{n}$ are defined as in Theorem 3.4.*

Proof. Reduce the problem to computing the product of the two polynomials $H_M(\mathbf{x})$ (resp., $T_M(\mathbf{x})$) and $V(\mathbf{x})$, and then apply Theorems 3.1–3.4. □

Applying these results, we can bound the number of ops in Algorithm 2.2 as follows.

PROPOSITION 3.12. *For any polynomial $p \in R$ with support in E and any vector $[\Lambda(\mathbf{x}^\alpha)]_{\alpha\in E+F}$ (with $\Lambda \in \widehat{R}$), the vector $[p \star \Lambda(\mathbf{x}^\beta)]_{\beta\in F}$ can be computed in $\mathcal{O}(\lfloor E + F \rceil \log^2(\lfloor E + F \rceil))$ ops.*

Once we have a fast matrix-by-vector multiplication, a nonsingular linear system of equations can also be solved quickly by means of the conjugate gradient algorithm, which is based on the following theorem [19, section 10.2].

THEOREM 3.13. *Let $W\,\mathbf{v} = \mathbf{w}$ be a nonsingular linear system of N equations. Then N multiplications of each of the matrices W and W^T by vectors and $\mathcal{O}(N^2)$ additional ops suffice to compute the solution $\mathbf{v}$ to this linear system.*

Note that W^T is a quasi-Toeplitz (resp., quasi-Hankel) matrix if W is, and then both matrices can be multiplied by a vector quickly (see Proposition 3.11). Therefore, in the cases of quasi-Toeplitz and quasi-Hankel matrices W, Theorem 3.13 yields a

fast algorithm for solving the linear system $W\,\mathbf{v} = \mathbf{w}$. We will also need the following related result.

THEOREM 3.14 (see [32]). *Let W be an $N \times N$ real symmetric or Hermitian matrix. Let S be a fixed finite set of complex numbers. Then there is a randomized algorithm that selects N random parameters from the set S independently of each other (under uniform probability distribution on S) and either fails with a probability of at most $\frac{(N+1)N}{2\lfloor S\rceil}$ or performs $\mathcal{O}(N)$ multiplications of the matrix W by vectors and $\mathcal{O}(N^2\log(N))$ other ops to compute the rank and the signature of W.*

Hereafter, random selection of elements of a set S as in Theorem 3.14 will be called *sampling*.

Proof. To support the claimed estimate, we first tridiagonalize the matrix W by the Lanczos randomized algorithm [2, pp. 118–119], which involves an initial vector of dimension N and fails with a probability of $\frac{(N+1)N}{2\lfloor S\rceil}$ if the N coordinates of the vector have been sampled at random from the set S. The above bound on the failure probability and the cost bound of $\mathcal{O}(N)$ multiplications of the matrix W by vectors and $\mathcal{O}(N^2\log(N))$ other ops of this stage have been proved in [38]. Then, in $\mathcal{O}(N)$ ops, we compute the Sturm sequence of the N values of the determinants of all of the $k \times k$ northwestern (leading principal) submatrices of W for $k = 1, \dots, N$ and obtain the numbers N_+ and N_- of positive and negative eigenvalues of W from the Sturm sequence (cf., e.g., [3]). These two numbers immediately define the rank and the signature of W. ∎

Combining Proposition 3.11 with Theorems 3.13 and 3.14 gives us the next corollary.

COROLLARY 3.15. *For an $N \times N$ quasi-Toeplitz or quasi-Hankel matrix W, the estimates of Theorems* 3.13 *and* 3.14 *turn into $O(N^2\log(N))$ ops if the matrix has a maximal (c,d) support where $c+d=N$. They turn into $\mathcal{O}(N^2\log^2(N))$ ops if the matrix has a total degree (c,d) support where $c+d=O(N)$ and into $\mathcal{O}((\log^2(N)+\log(d))N^2+C_{W,N})$ otherwise, where d and $C_{W,N}$ are defined as in Proposition* 3.11 (a) *for $M=W$.*

REMARK 3.16. *Hereafter, we will refer to the matrices of case* (b) *in Proposition* 3.11 *as the matrices with support of the maximal degree (c,d) and to the matrices of case* (c) *as the ones with support of the total degree (c,d). Furthermore, stating our estimates for the arithmetic complexity of computations, we will assume that the input polynomials have the maximal degree (c,d) support. That is, we will rely on Theorem* 3.3 *and Proposition* 3.11 (b)*, and we will express the estimates in terms of the cardinality of the supports E and/or F or in terms of an upper bound D on the number of common roots of the input polynomials. The estimates can be easily extended to the other cases based on Theorem* 3.1 *or* 3.4 *and Proposition* 3.11 (a) *or* (c) *instead of Theorem* 3.3 *and Proposition* 3.11 (b)*. In the latter case (Theorem* 3.4 *and Proposition* 3.11 (c)*), the cost estimates increase by the factors* $\log(D)$, $\log(\lfloor E\rceil)$, *or* $\log(\lfloor F\rceil)$, *respectively. In case Theorems* 3.1 *and Proposition* 3.11 (a) *are used, the estimates are expressed in terms of the bounds $C_{PolMult}(G,H)$ or $C_{M,N}$ for appropriate sets G and H, matrix M, and integer N. The latter case covers sparse input polynomials for which the respective bounds $C_{PolMult}(G,H)$ and $C_{M,N}$ are smaller than for the general (or dense) input, although they are not expressed solely in terms of the cardinality D. (They also depend on the degree of the monomials or the cardinality of the supports of the input polynomial system.)*

4. Computation of a nondegenerate linear form. In this section, we will compute a nondegenerate linear form on $\mathcal{A}$ provided that we are given a basis $(\mathbf{x}^\alpha)_{\alpha\in E}$ of $\mathcal{A}$ and the normal form of the elements on the boundary of this basis. This is the case, for instance, when we have computed a Gröbner basis of our ideal I for any monomial ordering [9] or when we apply any other normal-form algorithm [28], [33].

DEFINITION 4.1.

- *Let $v_i = (\delta_{i,1}, \dots, \delta_{i,n}) \in \mathbb{N}^n$, where $\delta_{i,j}$ is the Kronecker symbol.*
- *For all $A \subset \mathbb{N}^n$, $\Omega(A) = \{\alpha \in \mathbb{N}^n : \alpha \in A \text{ or } \exists i \in \{1, \dots, n\}, \alpha - v_i \in A\}$.*
- *N_α for $\alpha \in \Omega(E)$ is the normal form of the monomial $\mathbf{x}^\alpha \bmod I$, i.e., the canonical representative of its class modulo the ideal I. $N_\alpha = \mathbf{x}^\alpha$ if $\alpha \in E$, and*

$$N_\alpha = \sum_{\beta\in E} n_{\alpha,\beta}\mathbf{x}^\beta$$

if $\alpha \in \Omega(E) - E$.

Our goal is to obtain the coefficients $\tau(\mathbf{x}^\alpha)$ for $\alpha \in E+E+E$, where $\tau \in \widehat{\mathcal{A}} = I^\perp$ is a generic linear form. We will compute them, by induction, under the following hypothesis.

HYPOTHESIS 4.2.

- *$(x^\alpha)_{\alpha\in E}$ is stable under derivation, that is, $\alpha = \alpha' + v_i \in E$ implies that $\alpha' \in E$.*
- *N_α, the normal form of $\mathbf{x}^\alpha$, is available for every $\alpha \in \Omega(E)$.*
- *The values $\tau_\alpha = \tau(x^\alpha)$ are available for all $\alpha \in E$, where τ is not degenerate $\in \widehat{\mathcal{A}} = I^\perp$.*

For the third part, we can remark that a random choice of $\tau(\mathbf{x}^\alpha)$ will imply with a high probability that τ does not degenerate. Our procedure is based on the following property.

PROPOSITION 4.3. *For each $\alpha \in \Omega(E)$, we have $\tau_\alpha = \tau(N_\alpha) = \sum_{\beta\in E} n_{\alpha,\beta}\tau_\beta$. This value can be computed by applying $\mathcal{O}(D)$ ops, where $D = \lfloor E \rceil$. More generally, for all $\gamma \in E$ we have the following inductive relation:*

$$\tau_{\alpha+\gamma} = \sum_{\beta\in E} n_{\alpha,\beta}\tau_{\beta+\gamma}.$$

Now assume that we have computed all of the values τ_β for $\beta \in \Omega(E)$, and let $\alpha = \alpha_0 + v_i \in \Omega(\Omega(E))$ with $\alpha_0 \in \Omega(E)$. Then

$$\tau(\mathbf{x}^\alpha) = \tau(x_i N_{\alpha_0}) = \sum_{\beta\in E} n_{\alpha_0,\beta}\tau\left(x_i\mathbf{x}^\beta\right).$$

We know all of the $n_{\alpha_0,\beta}$ and all of the $\tau\left(x_i\mathbf{x}^\beta\right)$ because $\beta + v_i \in \Omega(E)$. Therefore, we obtain $\tau_\alpha = \sum_{\beta\in E} n_{\alpha_0,\beta}\tau_{\beta+v_i}$ by computing a scalar product. Recursively, this leads us to the following inductive definition of the "levels" Ω_i.

DEFINITION 4.4. *Write $\Omega_0 = E$, $\Omega_1 = \Omega(E)$ and $\Omega_i = \Omega(\Omega_{i-1}) \cap (E + E + E)$, $i = 2, 3, \dots$, and write $h = \max\{|\alpha| : \alpha \in E\}$ so that $E + E + E = \Omega_{2h}$.*

PROPOSITION 4.5. *For every $\alpha \in \Omega_i$, there is $\alpha' \in \mathbb{N}^n$ and $\alpha_1 \in \Omega_1 - \Omega_0$ such that $\alpha = \alpha_1 + \alpha'$ with $|\alpha'| \le i - 1$ and for all $\beta \in E$ we have $\beta + \alpha' \in \Omega_{i-1}$.*

Proof. Assume that $i > 0$. Let $\alpha \in \Omega_i \subset E + E + E$. Then α can be decomposed as follows: $\alpha = \gamma_0 + \gamma_1 + \gamma_2$ with $\gamma_0, \gamma_1, \gamma_2 \in E$ and $|\gamma_1 + \gamma_2| = i$. As $i > 1$, there exists $\alpha' = \gamma_1 + \gamma_2 - v_j \in \mathbb{N}^n$, and because $(\mathbf{x}^\alpha)_{\alpha\in E}$ is stable by Hypothesis 4.2, we have

$\alpha' \in E + E$. It follows that $\alpha = \alpha_1 + \alpha'$, where $\alpha_1 = \gamma_0 + v_j \in \Omega_1$ and $|\alpha'| \leq i - 1$. Therefore, for all $\beta \in E$, $\beta + \alpha' \in \Omega_{i-1}$, which completes the proof. ∎

Assume now that we have already computed all of the values τ_β for $\beta \in \Omega_{i-1}$. Then, according to Proposition 4.5, for any $\alpha \in \Omega_i$, we have $\alpha = \alpha_1 + \alpha'$, with $\alpha_1 \in \Omega_1$ and $|\alpha'| \leq i - 1$. Thus, if $\alpha_1 \in \Omega_1 - \Omega_0$, we have

$$\tau(\mathbf{x}^\alpha) = \tau(\mathbf{x}^{\alpha_1}\, \mathbf{x}^{\alpha'}) = \sum_{\beta \in E} n_{\alpha_1,\beta} \tau(\mathbf{x}^{\beta+\alpha'})$$

with $\beta + \alpha' \in \Omega_{i-1}$; otherwise, if $\alpha_1 \in \Omega_0$, we have $\alpha = \alpha_1 + \alpha' \in \Omega_{i-1}$. In other words, we can compute by induction the values of τ on Ω_i from its values on Ω_{i-1}. This yields the following recursive algorithm for the computation of $\tau(\mathbf{x}^\alpha)$ with $\alpha \in E + E + E$.

ALGORITHM 4.6. *Compute the first coefficients of the series associated with a linear form τ of $I^\perp$ as follows:*

1. *For i from 1 to $2h$ do for each $\alpha = \alpha_0 + \alpha_1 \in \Omega_i$ with α_0 and α_1 as in Proposition 4.5 compute $\tau_\alpha = \sum_{\beta \in E} n_{\alpha_1,\beta} \tau_{\alpha_0+\beta}$*
   End for
2. *Compute and output the polynomial $S = \sum_{\alpha \in E+E+E} \tau_\alpha \mathtt{d}^\alpha$.*

PROPOSITION 4.7. *The arithmetic complexity of Algorithm 4.6 is $\mathcal{O}\left(3^n D^2\right)$.*

Proof. For each element $\alpha \in E + E + E$, we compute τ_α in $\mathcal{O}(D)$ arithmetic operations, and there are at most $\mathcal{O}(3^n D)$ elements in $E + E + E$, which gives us the claimed arithmetic complexity estimate. ∎

5. Arithmetic in the algebra $\mathcal{A}$. Our algorithms in the next sections perform computations in $\mathcal{A}$ efficiently based on the knowledge of a certain linear form on $\mathcal{A}$ (such as the one computed in the previous section), which induces a nondegenerate inner product. More precisely, we assume that the following items are available.

Basic set of items:

- *a linear form $\tau \in \widehat{\mathcal{A}} = I^\perp$, such that the bilinear form $\tau(a\, b)$ from $\mathcal{A} \times \mathcal{A}$ to $\mathbb{C}$ is nondegenerate,*
- *a monomial basis $(\mathbf{x}^\alpha)_{\alpha \in E}$ of $\mathcal{A}$,*
- *the coefficients $(\tau(\mathbf{x}^\alpha))_{\alpha \in F}$, where $F = E + E + E$.*

The number of elements in E is the dimension D of $\mathcal{A}$ over $\mathbb{C}$. We describe basic operations in the quotient ring $\mathcal{A}$ in terms of the following quasi-Hankel matrix.

DEFINITION 5.1. *For any Λ in $\widehat{\mathcal{A}}$ and for any subset F of $\mathbb{N}^n$, let $\mathtt{H}_\Lambda^F$ denote the quasi-Hankel matrix, $\mathtt{H}_\Lambda^F = (\Lambda(\mathbf{x}^{\alpha+\beta}))_{\alpha,\beta \in F}$.*

By default we will assume we are dealing with the maximal degree support whenever we state our arithmetic complexity estimates (see Remark 3.16).

PROPOSITION 5.2. *The matrix $\mathtt{H}_\Lambda^F$ can be multiplied by a vector by using $\mathcal{O}(3^n \lfloor F \rceil \log(3^n \lfloor F \rceil))$ ops.*

Proof. Apply Proposition 3.11 (b) to the (F, F) quasi-Hankel matrix $\mathtt{H}_\Lambda^F$, and observe that $\lfloor F + F + F \rceil = 3^n \lfloor F \rceil$. ∎

Combining Corollary 3.15 and Proposition 5.2 implies the following result.

PROPOSITION 5.3. *Check if the linear system $\mathtt{H}_\Lambda^F \mathbf{u} = \mathbf{v}$ has a unique solution, and, if so, computing the solution requires $\mathcal{O}(3^n \lfloor F \rceil^2 \log(3^n \lfloor F \rceil))$ ops. The same cost estimate applies to the computation of the rank of the matrix $\mathtt{H}_\Lambda^F$, which involves randomization with $\lceil F \rceil$ random parameters and has a failure probability of at most $(\lfloor F \rceil + 1) \lfloor F \rfloor / (2 \lfloor S \rceil)$ provided that the parameters have been sampled from a fixed finite set S.*

5.1. Dual basis. As τ defines a nondegenerate bilinear form, there exists a set of polynomials $(\mathbf{w}_\alpha)_{\alpha\in E}$ such that $\tau(\mathbf{x}^\alpha\,\mathbf{w}_\beta)=\delta_{\alpha,\beta}$, $\delta_{\alpha,\beta}$ being Kronecker's symbol, $\delta_{\alpha,\alpha}=1$, and $\delta_{\alpha,\beta}=0$ if $\alpha\neq\beta$. The set $(\mathbf{w}_\alpha)_{\alpha\in E}$ is called the *dual basis* of $(\mathbf{x}^\alpha)_{\alpha\in E}$ for τ.

PROPOSITION 5.4 (projection formula). *For any $p\in R$, we have*

$$p\equiv\sum_{\alpha\in E}\tau(p\,\mathbf{w}_\alpha)\mathbf{x}^\alpha\equiv\sum_{\alpha\in E}\tau(p\,\mathbf{x}^\alpha)\mathbf{w}_\alpha. \tag{4}$$

Proof. See [7], [11]. □

DEFINITION 5.5. *For any $p\in\mathcal{A}$, denote by $[p]_{\mathbf{x}}$ and $[p]_{\mathbf{w}}$ the coordinate vectors of p in the bases $(\mathbf{x}^\alpha)_{\alpha\in E}$ and $(\mathbf{w}_\alpha)_{\alpha\in E}$, respectively.*

Let $\mathbf{w}_\alpha=\sum_{\beta\in E}w_{\beta,\alpha}\,\mathbf{x}^\beta$, and let $\mathtt{W}_\tau=(w_{\alpha,\beta})_{\alpha,\beta\in E}$ be the coefficient matrix. By the definition of the dual basis,

$$\tau(\mathbf{w}_\alpha\,\mathbf{x}^\gamma)=\sum_{\beta\in E}w_{\alpha,\beta}\,\tau(\mathbf{x}^{\beta+\gamma}) \tag{5}$$

is 1 if $\alpha=\gamma$ and 0 elsewhere. In terms of matrices, (5) implies that

$$\mathtt{H}_\tau\,\mathtt{W}_\tau=\mathbb{I}_D, \tag{6}$$

where $\mathtt{H}_\tau=\mathtt{H}_\tau^E=(\tau(\mathbf{x}^{\beta+\gamma}))_{\beta,\gamma\in E}$. From the definition of $\mathtt{W}_\tau$ and (6), we deduce that

$$[p]_{\mathbf{x}}=\mathtt{W}_\tau\,[p]_{\mathbf{w}},\ [p]_{\mathbf{w}}=\mathtt{H}_\tau\,[p]_{\mathbf{x}}. \tag{7}$$

The next result follows from Proposition 5.3.

PROPOSITION 5.6. *For any $p\in\mathcal{A}$, the coordinates $[p]_{\mathbf{x}}$ of p in the monomial basis can be computed from its coordinates $[p]_{\mathbf{w}}$ in the dual basis by using $\mathcal{O}(3^nD^2\,\log(3^nD))$ ops.*

5.2. Product in $\mathcal{A}$. We apply projection formula (4) and, for any $f\in R$, deduce that $f\equiv\sum_{\alpha\in E}\tau(f\,\mathbf{x}^\alpha)\,\mathbf{w}_\alpha=\sum_{\alpha\in E}f\star\tau(\mathbf{x}^\alpha)\,\mathbf{w}_\alpha$ in $\mathcal{A}$. Furthermore, by expressing the linear form $f\star\tau$ as a formal power series, we obtain $f\star\tau=\sum_{\alpha\in\mathbb{N}^n}\,f\star\tau(\mathbf{x}^\alpha)\,\mathbf{d}^\alpha$ so that the coefficients of $(\mathbf{d}^\alpha)_{\alpha\in E}$ in the expansion of $f\star\tau$ are the coefficients $[f]_{\mathbf{w}}$ of f in the dual basis $(\mathbf{w}_\alpha)_{\alpha\in E}$.

Similarly, for any $f,g\in\mathcal{A}$, the coefficients of $(\mathbf{d}^\alpha)_{\alpha\in E}$ in $fg\star\tau$ are the coefficients $[fg]_{\mathbf{w}}$ of $f\,g$ in the dual basis $(\mathbf{w}_\alpha)_{\alpha\in E}$. This leads to the following algorithm for computing the product in $\mathcal{A}$.

ALGORITHM 5.7. *For any pair $f,g\in\langle\mathbf{x}^\alpha\rangle_{\alpha\in E}$, compute the product fg in the basis $\langle\mathbf{x}^\alpha\rangle_{\alpha\in E}$ of $\mathcal{A}$ as follows:*

1. *Compute the coefficients of $(\mathbf{d}^\alpha)_{\alpha\in E}$ in the product $f\,g\star\tau$.*
2. *Obtain the coefficients $[f\,g]_{\mathbf{w}}$ from the first coefficients of $fg\star\tau$.*
3. *Solve in $\mathbf{u}$ the linear system $[f\,g]_{\mathbf{w}}=\mathtt{H}_\tau\,\mathbf{u}$.*

Output the vector $\mathbf{u}$, which is the coordinate vector $[f\,g]_{\mathbf{x}}$ of $f\,g$ in the monomial basis of $\mathcal{A}$.

PROPOSITION 5.8. *The product $f\,g$ can be computed in $\mathcal{O}(3^nD^2\log(3^nD))$ ops.*

Proof. $f\,g\star\tau$ is the product of polynomials with supports in $-E$ or $E+E+E$. Such a product can be computed in $\mathcal{O}(3^nD\log^2(3^nD))$ ops (see Proposition 3.11 and Remark 3.16 and observe that $\lfloor E+E+E\rceil=\mathcal{O}\,(3^n\lfloor E\rceil)$). The complexity of the third step is bounded according to Proposition 5.3 (with $F=E$). □

5.3. Inversion in $\mathcal{A}$. The projection formula of Proposition 5.4 implies that $f\,\mathbf{x}^\alpha = \sum_{\beta\in E} f \star \tau(\mathbf{x}^{\alpha+\beta})\,\mathbf{w}_\beta$, which means that $[f\,\mathbf{x}^\alpha]_{\mathbf{w}}$ is the coordinate vector $[f \star \tau(\mathbf{x}^{\alpha+\beta})]_{\beta\in E}$, that is, the column of the matrix $\mathtt{H}_{f\star\tau}$ indexed by α. In other words, $[f\,\mathbf{x}^\alpha]_{\mathbf{w}} = \mathtt{H}_{f\star\tau}\,[\mathbf{x}^\alpha]_{\mathbf{x}}$. By linearity, for any $g \in \mathcal{A}$, we have

$$[f\,g]_{\mathbf{w}} = \mathtt{H}_{f\star\tau}[g]_{\mathbf{x}} = \mathtt{H}_\tau\,[f\,g]_{\mathbf{x}},$$

according to (7). Thus, if $fg = 1$, that is, if $g = f^{-1}$, we have $\mathtt{H}_{f\star\tau}[g]_{\mathbf{x}} = \mathtt{H}_\tau\,[1]_{\mathbf{x}}$. This leads to the following algorithm for computing the inverses (reciprocals) in $\mathcal{A}$.

ALGORITHM 5.9. *For any $f \in \langle \mathbf{x}^\alpha\rangle_{\alpha\in E}$, verify whether there exists the inverse (reciprocal) of $f \in \mathcal{A}$, and, if so, compute it.*

1. *Compute* $\mathbf{v} = \mathtt{H}_\tau\,[1]_{\mathbf{x}}$.
2. *Solve in* $\mathbf{u}$ *the linear system* $\mathtt{H}_{f\star\tau}\mathbf{u} = \mathbf{v}$ *or output* FAILURE *if the matrix* $\mathtt{H}_\tau$ *is not invertible.*

Output the vector $\mathbf{u}$*, which is the coordinate vector* $[f^{-1}]_{\mathbf{x}}$ *of* f^{-1} *in the monomial basis of* $\mathcal{A}$.

By combining Propositions 5.2 and 5.3 and Remark 3.16, we obtain the following proposition.

PROPOSITION 5.10. *The inverse (reciprocal)* f^{-1} *of an element* f *of* $\mathcal{A}$ *can be computed by using* $\mathcal{O}(3^n D^2 \log(3^n D))$ *ops.*

6. Iterative methods. Our algorithms for the root approximation will essentially amount to computing nontrivial idempotents in the quotient algebra $\mathcal{A}$ by iterative processes with the subsequent simple recovery of the roots from the idempotents. The algorithms work in $\mathbb{C}^D$, and we will write $\mathbf{i} = \sqrt{-1}$. More rudimentary univariate versions of such algorithms were studied in [6]. We will use the basic operations in the quotient algebra $\mathcal{A}$ in order to devise two iterative methods, which will converge to nontrivial idempotents. We will first consider an iteration associated to a slight modification of the so-called *Joukovski* map (see [20], [6]): $z \mapsto \frac{1}{2}(z + \frac{1}{z})$ and its variant $z \mapsto \frac{1}{2}(z - \frac{1}{z})$. The two attractive fixed points of this map are 1 and -1; for its variant, they turn into $\mathbf{i}$ and $-\mathbf{i}$.

ALGORITHM 6.1. *Sign iteration. Choose* $u_0 = h \in \langle \mathbf{x}^\alpha\rangle_{\alpha\in E}$*, and recursively compute* $u_{k+1} \equiv \frac{1}{2}(u_k - \frac{1}{u_k}) \in \mathcal{A},\ k = 0, 1, \dots$.

By applying Proposition 5.10 and Remark 3.16, we obtain the following result.

PROPOSITION 6.2. *Each iteration of Algorithm* 6.1 *requires* $\mathcal{O}(3^n D^2 \log(3^n D))$ *ops.*

Proof. Apply Proposition 5.3 and Remark 3.16 to estimate the arithmetic cost of the computation of the inverse (reciprocal) of an element of $\mathcal{A}$. To yield the claimed cost bound of Proposition 6.2, it remains to compute a linear combination of u_n and u_n^{-1} in $\mathcal{O}(D)$ ops by direct operations on vectors of size D. ∎

Hereafter, $\Re(h)$ and $\Im(h)$ denote the real and the imaginary parts of a complex number h, respectively. Recall that we write ζ to denote the common roots $\zeta \in \mathcal{Z}(I)$ of given polynomials $f_1, \dots, f_m$.

REMARK 6.3. *In Proposition* 6.4*, we will assume that* $J(h(\zeta)) \neq 0$ *for all* $\zeta \in \mathcal{Z}(I)$ *and, in Proposition* 6.6*, that* $|h(\zeta)|$ *is minimized for a unique root* $\zeta \in \mathcal{Z}(I)$*. These assumptions are satisfied for a generic system of polynomials or a generic polynomial* h.

PROPOSITION 6.4. *The sequence* $(u_0, u_1, \dots)$ *of Algorithm* 6.1 *converges quadratically to* $\sigma = \sum_{\Im(h(\zeta))>0} \mathbf{e}_\zeta - \sum_{\Im(h(\zeta))<0} \mathbf{e}_\zeta$*, and we have*

$$\|u_n - \sigma\| \le K \times \rho^{2^n}$$

(for some constant K), where

$$\rho^+ = \max_{\Im(h(\zeta))>0, \zeta\in\mathcal{Z}(I)} \left|\frac{h(\zeta)-\mathbf{i}}{h(\zeta)+\mathbf{i}}\right|,$$

$$\rho^- = \max_{\Im(h(\zeta))<0, \zeta\in\mathcal{Z}(I)} \left|\frac{h(\zeta)+\mathbf{i}}{h(\zeta)-\mathbf{i}}\right|,$$

$\mathbf{i}=\sqrt{-1}$, *and* $\rho=\max\{\rho^+,\rho^-\}$.

Proof. Apply the classical convergence analysis of the Joukovski map (see [20]) to the matrices of multiplication by u_n in $\mathcal{A}$, whose eigenvalues are $\{u_n(\zeta), \zeta\in\mathcal{Z}(I)\}$. □

Let

$$\mathbf{e}^+ = \sum_{\Im(h(\zeta))>0} \mathbf{e}_\zeta = \frac{1}{2}(1+\sigma), \quad \mathbf{e}^- = \sum_{\Im(h(\zeta))\le 0} \mathbf{e}_\zeta = \frac{1}{2}(1-\sigma)$$

denote the two sums of the idempotents associated to the roots $\zeta\in\mathcal{Z}$ such that $\Im(h(\zeta))>0$ and $\Im(h(\zeta))<0$, respectively.

If $h(\mathbf{x})$ is a linear function in $\mathbf{x}$, then each of the idempotents $\mathbf{e}^+$ and $\mathbf{e}^-$ is associated with all of the roots lying in a fixed half-space of $\mathbb{C}^n$ defined by the inequalities $\Im(h(\zeta))>0$ or $\Im(h(\zeta))<0$. Conversely, an appropriate linear function $h(\mathbf{x})$ defines the idempotents $\mathbf{e}^+$ and $\mathbf{e}^-$ associated with any fixed half-space of $\mathbb{C}^n$. Furthermore, for any fixed polytope in $\mathbb{C}^n$ defined as the intersection of half-spaces, we may compute the family of the associated idempotents whose product will be associated with the polytope. In particular, any bounded box is the intersection of $4n$ half-spaces, and the associated idempotent can be computed in $4n$ applications of Algorithm 6.1. Let us specify the case in which the polytope is the almost flat unbounded box approximating the real manifold $R^n=\{\mathbf{x}:\Im(x_i)=0, i=1,\dots,n\}$. In this case, the choices of $h=x_i-\epsilon$ and $h=x_i+\epsilon$ allow us to approximate the two idempotents

$$\mathbf{e}^-_{i,\epsilon} = \sum_{\Im(\zeta_i)<\epsilon} \mathbf{e}_\zeta, \quad \mathbf{e}^+_{i,\epsilon} = \sum_{\Im(\zeta_i)>-\epsilon} \mathbf{e}_\zeta.$$

Their product can be computed in $\mathcal{O}(3^nD^2\log(3^nD))$ ops to yield $\mathbf{r}_{i,\epsilon}=\sum_{|\Im(\zeta_i)|<\epsilon}\mathbf{e}_\zeta$, and the product $\mathbf{r}_\epsilon\equiv\mathbf{r}_{1,\epsilon}\cdots\mathbf{r}_{n,\epsilon}$ can be computed in $\mathcal{O}(3^nD^2\log(3^nD))$ ops to yield the sum of the fundamental idempotents whose associated roots of the polynomial system are nearly real.

ALGORITHM 6.5. *Computing the sum of the fundamental (nearly real) idempotents.*

- *For i from 1 to n do*
  $u_0=x_i\pm\epsilon$; $u_1:\equiv\frac{1}{2}(u_0-\frac{1}{u_0})$ in $\mathcal{A}$; $k:=1$;
  while $\|u_k-u_{k-1}\|<2^{-b}$ do $\{\ u_{k+1}:=\frac{1}{2}(u_k-\frac{1}{u_k})$; $k:=k+1\ \}$
  Compute $\mathbf{e}^\pm_{i,e}$ and $\mathbf{r}_{i,\epsilon}$.
- *Compute and output the product $\mathbf{r}_\epsilon\equiv\mathbf{r}_{1,\epsilon}\cdots\mathbf{r}_{n,\epsilon}$ in $\mathcal{A}$.*

According to Propositions 6.2 and 6.4 and Remark 3.16, we have the following proposition.

PROPOSITION 6.6. *An approximation of $\mathbf{r}_\epsilon$ (within the error bound $\epsilon=2^{-b}$) can be computed in $\mathcal{O}(\mu\,3^nD^2\ \log(3^nD))$ ops, where*

$$\mu=\mu(b,\rho)=\log|b/\log(\rho)| \tag{8}$$

and

$$\rho = \max_i\{ \quad \max_{\Im(\zeta_i)>0,\zeta\in\mathcal{Z}(I)} |\tfrac{\zeta_i-\mathbf{i}}{\zeta_i+\mathbf{i}}|, \quad \max_{\Im(\zeta_i)<0,\zeta\in\mathcal{Z}(I)} |\tfrac{\zeta_i+\mathbf{i}}{\zeta_i-\mathbf{i}}| \quad \}. \tag{9}$$

The second iterative method is the quadratic power method.

ALGORITHM 6.7. *Quadratic power iteration. Choose* $u_0 = h \in \langle \mathbf{x}^\alpha \rangle_{\alpha\in E}$, *and recursively compute* $u_{n+1} \equiv u_n^2 \in \mathcal{A}$, $n = 0, 1, \ldots$.

Each step of this iteration requires at most $\mathcal{O}\left(3^n D^2 \log\left(3^n D\right)\right)$ ops, and we have the following property.

PROPOSITION 6.8. *An approximation (within the error bound* $\epsilon = 2^{-b}$*) of the idempotent* $\mathbf{e}_\zeta$ *such that a unique simple root* ζ *minimizes* $|h|$ *on* $\mathcal{Z}(I)$ *can be computed in* $\mathcal{O}\left(\nu 3^n D^2 \log\left(3^n D\right)\right)$ *ops, where*

$$\nu = \nu\left(b, \gamma\right) = \log\left(b/|\log\left(\gamma\right)|\right), \tag{10}$$

$$\gamma = \left|\frac{h(\zeta)}{h(\zeta')}\right|, \tag{11}$$

and $|h(\zeta')|$ *is the second smallest value of* $|h|$ *over* $\mathcal{Z}(I)$.

Proof. We rely on the convergence analysis of the quadratic power method applied to the matrices of multiplication by u_n in $\mathcal{A}$, whose eigenvalues are $\{u_n(\zeta), \zeta \in \mathcal{Z}(I)\}$. ∎

7. Counting and approximating the roots and the real roots. In this section, we will apply the techniques and algorithms of the previous sections to the problems of counting and approximation of the roots of the system $\mathbf{p} = \mathbf{0}$.

In the algorithms for counting roots, we will use the randomization required to apply Theorem 3.13. The resulting randomized algorithms and the computational complexity estimates for counting (excluding the preprocessing stage of subsection 7.5) will apply to any 0-dimensional polynomial system.

In the approximation algorithms, we do not need randomization except for the ensurance of the assumption of Propositions 6.4 (cf. Remark 6.3), but the estimates for the computational cost depend on the parameters ρ and γ of the two latter propositions (cf. (8), (11)) and remain meaningful unless these parameters are extremely close to 1.

7.1. Counting the roots and the real roots.

THEOREM 7.1 (see [29]). *The number of the roots (resp., real roots) of the system* $\mathbf{p} = \mathbf{0}$ *is given by the rank (resp., the signature) of the quasi-Hankel matrix* H_τ^E.

Theorem 7.1, Corollary 3.15, and Remark 3.16 together imply the following result.

COROLLARY 7.2. *The numbers of the roots and of the real roots of the polynomial system* $\mathbf{p} = \mathbf{0}$ *can be computed by a randomized algorithm that generates* D *random parameters and, in addition, performs* $O(3^n D^2 \log(3^n D))$ *ops. If the random parameters are sampled from a fixed finite set* S, *then the algorithm may fail with a probability at most* $\left(3^n D + 1\right) 3^n D / \left(2\lfloor S \rceil\right)$.

7.2. Approximation of a root. Application of Algorithm 6.7 in $\mathcal{A}$ yields the following theorem.

THEOREM 7.3. *The idempotent corresponding to a root* ζ *that maximizes the absolute values* $|h(\zeta)|$ *of a fixed polynomial* $h(\mathbf{x})$ *can be approximated (within an error bound* $\epsilon = 2^{-b}$*) by using* $O(3^n D^2 \nu \log(3^n D))$ *ops, where* ν *is defined in* (10) *and* (11).

The latter cost bound dominates the cost of the subsequent transition from the idempotent to a root.

THEOREM 7.4. *The n coordinates of a simple root ζ can be determined from the idempotent $\mathbf{e}_\zeta$ in $\mathcal{O}(3^n D^2 \log(3^n D))$ ops. This bound increases by the factor of n if the root is multiple.*

Proof. We compute $J\mathbf{e}_\zeta$ in $\mathcal{A}$ (where J is the Jacobian of the n equations) by Algorithm 5.7. According to [29], [32], in the case of a simple root, we have

$$\mathtt{H}_\tau^E\,[J\,\mathbf{e}_\zeta]_{\mathbf{x}} = \lambda\,[\zeta^\alpha]_{\alpha\in E}, \quad \lambda \in \mathbb{C}.$$

This vector is computed at the arithmetic cost within the complexity bound of Proposition 5.2 (cf. [32]), and this immediately gives us the coordinates of the root ζ if $\mathbf{x}^E$ contains $1, x_1, \dots, x_n$, which is generically the case. If the root is not simple, then, according to the relation

$$x_i\,J\,\mathbf{e}_\zeta \equiv \zeta_i\,J\,\mathbf{e}_\zeta$$

(see [29], [32], [11]), we recover the coordinates of ζ by computing $n+1$ products in $\mathcal{A}$ (by Algorithm 5.7). □

7.3. Approximation of a selected root. In view of Theorem 7.4, it is sufficient to approximate the idempotents associated to the roots.

Suppose that we seek a root of the system $\mathbf{p} = \mathbf{0}$ whose coordinate x_1 is the closest to a given value $u \in \mathbb{C}$. Let us assume that u is not a projection of any root of the system $\mathbf{p} = \mathbf{0}$ so that $x_1 - u$ has the inverse (reciprocal) in $\mathcal{A}$. Let $h(\mathbf{x})$ denote such an inverse (reciprocal). We have $h\,(\mathbf{x})(x_1 - u) \equiv 1$ and $h(\zeta) = \frac{1}{\zeta_1 - u}$. Therefore, a root whose coordinate x_1 is the closest to u_1 is a root for which $|h(\zeta)|$ is the largest. Consequently, iterative squaring of $h = h(\mathbf{x})$ shall converge to this root.

The polynomial h can be computed by using $O(3^n D^2 \nu \log(3^n D))$ ops for ν of (10) and (11) (see [32, section 3.3.4]).

One may compute several roots of the polynomial system by applying the latter computation (successively or concurrently) to several initial values u.

7.4. Counting nearly real roots and the roots in a polytope. As long as we have (a close approximation to) the idempotent $\mathbf{r}$ associated with a fixed polytope, we may restrict our counting and approximation algorithms to such a polytope simply by moving from the basic nondegenerate linear form τ to the form $\mathbf{r} \star \tau$ (by using $O(3^n D^2 \log(3^n D))$ ops). Let us specify this in the case in which the polytope is the nearly flat box approximating the real space $\mathbb{R}^n$ (cf. Algorithm 6.5 and Proposition 6.6).

Let $\mathcal{A}_\epsilon^{\mathbb{R}} = \mathbf{r}_\epsilon \mathcal{A}$ denote the subalgebra of $\mathcal{A}$ corresponding to the (nearly) real idempotents for a fixed $\epsilon = 2^{-b}$.

We may restrict our computation on $\mathcal{A}_\epsilon^{\mathbb{R}}$ by computing the linear form $\tau' = \mathbf{r}_\epsilon \star \tau$ (in $\mathcal{O}(3^n D^2 \log(D))$ ops, according to Proposition 3.12), and we have the following properties.

PROPOSITION 7.5.

- *The linear form $\tau' = \mathbf{r}_\epsilon \star \tau$ defines a nondegenerate inner product on $\mathcal{A}_\epsilon^{\mathbb{R}}$.*
- *The number of nearly real roots (counted with their multiplicities) is the rank of the matrix $H^E_{\mathbf{r}_\epsilon \star \tau} = (\mathbf{r}_\epsilon \star \tau(\mathbf{x}^{\beta+\gamma}))_{\beta,\gamma \in F}$.*
- *Let E' be a subset of E such that the submatrix $H^{E'}_{\tau'}$ is of the maximal rank. Then E' is a basis of $\mathcal{A}_\epsilon$.*

Proof. See [32]. □

We thus require an algorithm for computing the rank of $H_{\tau'}^{E}$ (see [35] on fast computation of the rank). Assuming (8) and (9), we deduce the following result from Theorem 3.14.

PROPOSITION 7.6. *The number of all nearly real roots can be computed by using* $\mathcal{O}(\mu\, 3^n\, D^2 \log(3^n D))$ *ops (for* μ *of* (8) *and* (9)*).*

7.5. Approximation of nearly real roots and the roots in a box.. To compute a nearly real root as well as a root lying in a fixed box in $\mathbb{C}^n$ maximizing a given function $|h|$, we may apply Algorithm 6.7 in $\mathcal{A}$ (or $\mathcal{A}_{\epsilon}^{\mathbb{R}}$) and Proposition 6.8 and obtain the following theorem.

THEOREM 7.7. *A nearly real root (as well as a root lying in a fixed box) that maximizes a function* $|h|$ *can be computed (up to an error* $\epsilon = 2^{-b}$*) by using* $\mathcal{O}((\mu+\nu)\ 3^n D^2 \log(3^n D))$ *ops for* μ *and* ν *of* (8)–(11)*.*

This process can be extended to compute the other roots via deflation. That is, we replace $\mathbf{r}_\epsilon$ by $\mathbf{r}'_\epsilon = \mathbf{r}_\epsilon - \mathbf{e}_\zeta$, compute $\tau'' = \mathbf{r}'_\epsilon \star \tau$, and apply the same iteration to compute the next (real) root, where $|h|$ takes on its second smallest value over $\mathcal{Z}(I)$. We can also restrict our computation to a fixed box by using the algorithm of subsection 7.4 to compute the sum of the idempotents corresponding to the roots lying inside the box. The complexity of each step is bounded in Theorem 7.7, leading to the following result for δ (real) roots in a given box.

THEOREM 7.8. *The* δ *(real) roots* ζ *lying in a given box can be computed (up to an error* $\epsilon = 2^{-b}$*) by using* $\mathcal{O}((\mu+\nu)\, n 3^n \delta\, D^2\, \log(D)\log(b))$ *ops for* μ *and* ν *of* (8)–(11)*.*

REFERENCES

[1] W. AUZINGER AND H. J. STETTER, *An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations*, in Numerical Mathematics (Singapore, 1988), Internat. Schriftenreihe Numer. Math. 86, Birkhäuser, Basel, 1988, pp. 11–30.

[2] D. BINI AND V. Y. PAN, *Polynomial and matrix computations, Vol.* 1: *Fundamental Algorithms*, Birkhäuser Boston, Boston, 1994.

[3] D. BINI AND V. Y. PAN, *Computing matrix eigenvalues and polynomial zeros where the output is real*, SIAM J. Comput., 27 (1998), pp. 1099–1115.

[4] J. CANNY, *Generalised characteristic polynomials*, J. Symbolic Comput., 9 (1990), pp. 241–250.

[5] J. CANNY AND I. EMIRIS, *An efficient algorithm for the sparse mixed resultant*, in Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (San Juan, PR, 1993), Lecture Notes in Comput. Sci. 673, G. Cohen, T. Mora, and O. Moreno, eds., Springer-Verlag, Berlin, 1993, pp. 89–104.

[6] J. P. CARDINAL, *On two iterative methods for approximating the roots of a polynomial*, in The Mathematics of Numerical Analysis (Park City, UT, 1995), Lectures in Appl. Math. 32, J. Renegar, M. Shub, and S. Smale, eds., AMS, Providence, RI, 1996, pp. 165–188.

[7] J. P. CARDINAL AND B. MOURRAIN, *Algebraic approach of residues and applications*, in The Mathematics of Numerical Analysis (Park City, UT, 1995), Lectures in Appl. Math. 32, J. Renegar, M. Shub, and S. Smale, eds., AMS, Providence, RI, 1996, pp. 189–210.

[8] A. L. CHISTOV AND D. Y. GRIGORIEV, *Complexity of Quantifier Elimination in the Theory of Algebraically Closed Fields*, Lecture Notes in Comput. Sci. 176, Springer-Verlag, New York, 1984.

[9] D. COX, J. LITTLE, AND D. O'SHEA, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, Undergrad. Texts Math., Springer-Verlag, New York, 1992.

[10] D. COX, J. LITTLE, AND D. O'SHEA, *Using Algebraic Geometry*, Undergrad. Texts Math., Springer-Verlag, New York, 1998.

[11] M. ELKADI AND B. MOURRAIN, *Approche effective des résidus algébriques*, Rapport de recherche 2884, INRIA, Sophia-Antipolis, France, 1996.

[12] M. Elkadi and B. Mourrain, *Some Applications of Bezoutians in Effective Algebraic Geometry*, Rapport de recherche 3572, INRIA, Sophia-Antipolis, France, 1998.

[13] M. Elkadi and B. Mourrain, *A new algorithm for the geometric decomposition of a variety*, in Proceedings of the International Symposium on Symbolic and Algebraic Computation, S. Dooley, ed., ACM, New York, 1999, pp. 9–16.

[14] M. Elkadi and B. Mourrain, *Algorithms for residues and Lojasiewicz exponents*, J. Pure Appl. Algebra, 153 (2000), pp. 27–44.

[15] I. Z. Emiris and V. Y. Pan, *The structure of sparse resultant matrices*, in Proceedings of the International Symposium on Symbolic and Algebraic Computation, ACM, New York, 1997, pp. 189–196.

[16] I. Z. Emiris and A. Rege, *Monomial bases and polynomial system solving*, in Proceedings of the International Symposium on Symbolic and Algebraic Computation, ACM, New York, 1994, pp. 114–122.

[17] J. C. Faugère, *A new efficient algorithm for computing Gröbner basis (F4)*, J. Pure Appl. Algebra, 139 (1999), pp. 61–88.

[18] M. Giusti and J. Heintz, *La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial*, in Proceedings of the International Meeting on Commutative Algebra, Sympos. Math. 34, Cambridge University Press, Cambridge, UK, 1991, pp. 216–255.

[19] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., John Hopkins University Press, Baltimore, MD, 1996.

[20] P. Henrici, *Applied and Computational Complex Analysis. Volume* I, Wiley, New York, 1988.

[21] D. Kapur and Y. N. Lakshman, *Elimination methods: An introduction*, in Symbolic and Numerical Computation for Artificial Intelligence, B. Donald, D. Kapur, and J. Mundy, eds., Academic Press, New York, 1992, pp. 45–89.

[22] Y. N. Lakshman and D. Lazard, *On the complexity of zero-dimensional algebraic systems*, in Effective Methods in Algebraic Geometry, Progr. Math. 94, Birkhäuser Boston, Boston, 1991, pp. 217–225.

[23] F. S. Macaulay, *The Algebraic Theory of Modular Systems*, Cambridge University Press, Cambridge, UK, 1916.

[24] J. M. Rojas, *On the average number of real roots of certain random sparse polynomial systems*, in The Mathematics of Numerical Analysis (Park City, UT, 1995), Lectures in Appl. Math. 32, J. Renegar, M. Shub, and S. Smale, eds., AMS, Providence, RI, 1996, pp. 689–699.

[25] D. Manocha, *Solving polynomial systems using matrix computations*, in Advances in Computational Mathematics, Ser. Approx. Decompos. 4, World Scientific Publishing, River Edge, NJ, 1994, pp. 99–129.

[26] B. Mourrain, *Isolated points, duality and residues*, J. Pure Appl. Algebra, 117/118 (1996), pp. 469–493.

[27] B. Mourrain, *Computing isolated polynomial roots by matrix methods*, J. Symbolic Comput., 26 (1998), pp. 715–738.

[28] B. Mourrain, *A new criterion for normal form algorithms*, in Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, Lecture Notes in Comput. Sci. 1719, M. Fossorier, H. Imai, S. Lin, and A. Poli, eds., Springer-Verlag, Berlin, 1999, pp. 430–443.

[29] B. Mourrain and V. Y. Pan, *Multidimensional structured matrices and polynomial systems*, Calcolo, 33 (1997), pp. 389–401.

[30] B. Mourrain and V. Y. Pan, *Solving special polynomial systems by using structured matrices and algebraic residues*, in Foundations of Computational Mathematics (Rio de Janeiro), F. Cucker and M. Shub, eds., Springer-Verlag, New York, 1997, pp. 287–304.

[31] B. Mourrain and V. Y. Pan, *Asymptotic acceleration of solving multivariate polynomial systems of equations*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, ACM, New York, 1998, pp. 488–496.

[32] B. Mourrain and V. Y. Pan, *Multivariate polynomials, duality and structured matrices*, J. Complexity, 16 (2000), pp. 110–180.

[33] B. Mourrain and P. Trébuchet, *Solving projective complete intersection faster*, in Proceedings of the International Symposium on Symbolic and Algebraic Computation, ACM, New York, 2000, pp. 231–238.

[34] V. Y. Pan, *Optimal (up to polylog factors) sequential and parallel algorithms for approximating complex polynomial zeros*, in Proceedings of the 27th Annual Symposium on Theory of Computing, ACM, New York, 1995, pp. 741–750.

[35] V. Y. Pan, *Structured Matrices and Polynomials: Unified Superfast Algorithms*, Birkhäuser Boston, Springer-Verlag, New York, 2001.

[36] V. Y. Pan, *Optimal and nearly optimal algorithms for approximating complex polynomial zeros*, Comput. Math. Appl., 31 (1996), pp. 97–138.

[37] V. Y. Pan, *Solving a polynomial equation: Some history and recent progress*, SIAM Rev., 39 (1997), pp. 187–220.

[38] V. Y. Pan and Z. Chen, *The complexity of matrix eigenproblem*, in Proceedings of the 31st Annual ACM Symposium on Theory of Computing, ACM, New York, 1999, pp. 507–516.

[39] J. Renegar, *On the worst-case complexity of approximating zeros of polynomials*, J. Complexity, 3 (1987), pp. 90–113.

[40] J. Renegar, *On the worst-case arithmetic complexity of approximating zeros of systems of polynomials*, SIAM J. Comput., 18 (1989), pp. 350–370.

[41] M. Shub and S. Smale, *On the complexity of Bezout's theorem* I—*geometric aspects*, J. Amer. Math. Soc., 6 (1993), pp. 459–501.

[42] H. J. Stetter, *Eigenproblems are at the heart of polynomial system solving*, SIGSAM Bulletin, 30 (1996), pp. 22–25.

[43] H. J. Stetter, *Analysis of zero clusters in multivariate polynomial systems*, in Proceedings of the International Symposium on Symbolic and Algebraic Computation, ACM, New York, 1996, pp. 127–135.

[44] E. E. Tyrtyshnikov, *A unifying approach to some old and new theorems on distribution and clustering*, Linear Algebra Appl., 232 (1996), pp. 1–43.

[45] B. L. van der Waerden, *Modern Algebra, Volume* II, Frederick Ungar Publishing, New York, 1948.