Asymptotic Acceleration of Solving Multivariate Polynomial Systems of Equations

Bernard Mourrain

INRIA, SAGA
BP 93, 06902 Sophia-Antipolis
France
mourrain@sophia.inria.fr

Victor Y. Pan

Department of Mathematics and Computer Science
Lehman College, City University of New York
Bronx, NY 10468
VPAN@LCVAX.LEHMAN.CUNY.EDU
(Supported by NSF Grant CCR 9625344 and PSC CUNY Award 668365)

Abstract

We propose new Las Vegas randomized algorithms for the solution of a multivariate generic or sparse polynomial system of equations. The algorithms use $\mathcal{O}^*((\delta+4^n)3^n D^2 \log b)$ arithmetic operations 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 the roots of the system, δ is the number of real roots or the roots lying in the box or disc, $\epsilon = 2^{-b}$ is the required upper bound on the output errors, and $\mathcal{O}^*(s)$ stands for $\mathcal{O}(s \log^c s)$, c being a constant independent of s. We also yield the bounds $\mathcal{O}^*(12^n D^2)$ for the complexity of counting the numbers of all roots in a fixed box (disc) and all real roots and $O^*(12^n D^2 \log b)$ for the complete solution of generic system. 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 order of magnitude the known complexity estimates of order at least $D^3 \log b$ or D^3, which so far are the record ones even for approximating 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 algebraic residues; 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.

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 some bibliography in [11], [3], [16], [8].) It has major applications, for instance, to robotics, computer modeling 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 has evolved in the old works by Bezout, Dixon, and Macaulay (see e.g. [11], [24]), then remained largely ignored by the researchers and algorithm designers but was resurrected by Canny in the 80s to become a very popular approach since then. 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], [7], [14]).

The approach has been explored and extended by many researchers, has been exploited in practice of algebraic computing, and also supported the record asymptotic upper bound $\mathcal{O}^*(D^3)$ on the arithmetic computational complexity of the solution of a polynomial system having a finite number of roots. 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 we may choose either the Bezout bound, $\prod_i d_i$, d_i denoting the maximum degree in the i-th variable in all monomials of the system, or the Bernstein 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.) The cited record bound $\mathcal{O}^*(D^3)$ is due to [22] but also has several other derivations and has been staying as a stable landmark for the multivariate polynomial system solving, somewhat similarly to the complexity bound $\mathcal{O}^*(N^3)$ for solving a nonsingular linear system of N equations, which was supported by Gaussian elimination and stayed as a landmark and a record until Strassen's result of 1969. In fact, even in the case of solving $generic$ polynomial system (including no degeneracy) as well as for many subproblems and related problems, no known algorithms support any better bound than $\mathcal{O}^*(D^3)$. 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 approximating a single root. Some progress was achieved in [16], where a single root was approximated in $\mathcal{O}^*(3^n D^2)$ time, but under a certain strong restriction on the input polynomials.

In the light of this background, the main result of our paper should be quite surprising: our new algorithms support the computational cost estimate of $\mathcal{O}^*(12^n D^2)$, for all the listed above subproblems, including the complete solution of generic system, 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 $\mathcal{O}^*((\delta + 4^n)3^n D^2)$ in the latter two cases, where δ is at most the number d of real roots or roots in the selected box or disc, respectively. In practical applications, such a number d is typically much less than D, and furthermore, δ grows as $\log D$ for a large class of input systems. Thus, for all listed problems, we improve the known complexity estimates by the order of magnitude. Furthermore, the factor $3^n(\delta + 4^n)$ can be replaced by $3^n\delta + 4^n M$, where M is the overall number of monomials of the input polynomials, which for sparse systems is dominated by 3^n.

Our algorithms approximate the roots numerically, and in terms of the required upper bound 2^{-b} on the output errors of the computed solution, we obtain the estimate $\mathcal{O}(\log b)$. Within a constant factor, such an estimate matches the lower bound of [23] and enables us to yield a high output precision at relatively low cost; this gives us a substantial practical advantage versus the algorithms that only reach $\mathcal{O}(b)$, because the solution of a polynomial system is usually needed with a high precision. We achieve this by using the matrix sign and inverse quadratic iterations, which converge with quadratic rate right from the start.

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

Our progress has some technical similarity to accelerating 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) and, even more so, with the recent progress in the univariate polynomial rootfinding via recursive splitting of the input polynomial into factors (cf. [4], [18], [19], [20]). Although recursive splitting into factors may be hard even to 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 yielded it in terms of idempotent elements of the associated quotient algebra (such elements represent the roots), and for this purpose we had to apply all our sophisticated and advanced techniques. This approach generalizes the methods of [4] and [19] to the multivariate case. The only missing technical point of our extension of the univariate splitting construction of [19] is the balancing of the splitting, which was the most recent and elusive step in the univariate case (cf. [19], [20]). 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 the roots of any specific polynomial system in $\mathcal{O}^*(12^n D^2 \log b)$ arithmetic time. Another major goal is to decrease or remove the factor 12^n from our complexity bounds, perhaps by means of improving our entire construction or its blocks of computations with structured matrices.

Let us conclude this section with a high level description of our approach. (For further details, we refer the reader to the next sections andS to our full paper). Our solution of polynomial systems consists of the following stages:

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

2. Compute non-trivial 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 correlations to polynomial computations and exploit some of these correlations to operate with such matrices faster. In section 5 we show how the combined power of the latter techniques and ones developed for working in the dual space enables us to perform rapidly the basic operations in the quotient algebra and, consequently, the computations of sections 6 and 7.

In terms of the complexity bounds, stage 1 contributes the terms $O(12^n D^2 \log D)$. 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, $R = \mathbb{C}[x_1, \ldots, x_n]$ is the ring of multivariate polynomials in the variables $x_1, \ldots, x_n$, with coefficients in the complex field $\mathbb{C}$. $\mathbb{Z}$ is the set of integers, $\mathbb{N}$ is its subset of nonnegative integers, $L = \mathbb{C}[x_1^{\pm}, \ldots, x_n^{\pm}]$ is the set of Laurent polynomials with monomial exponents in $\mathbb{Z}^n$. For any $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{Z}^n$, $\mathbf{x}^{\mathbf{a}}$ is the monomial $\mathbf{x}^{\mathbf{a}} = x_1^{a_1} \cdots x_n^{a_n}$. $\lfloor E \rceil$ is the number of the elements of a finite subset E of $\mathbb{Z}^n$.

2.1 Quotient algebra

Hereafter, $I = (f_1, \ldots, f_m)$ is the ideal of $R = \mathbb{C}[\mathbf{x}]$ generated by the elements $f_1, \ldots, f_m$, 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 will consider polynomial systems $f_1 = 0, \ldots, f_n = 0$ of n equations in n variables with finite sets of common roots $\mathcal{Z} = \mathcal{Z}(I) = \{\zeta \in \mathbb{C}^n ; f_1(\zeta) = \cdots = f_n(\zeta) = 0\}$. In this *case of complete intersection*, the vector space $\mathcal{A}$ has a finite dimension D, $D \geq d$ (D is the number of roots counted with their multiplicities). 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 exists orthogonal idempotents $(\mathbf{e}_i)_{i=1,\ldots,d}$ such that $1 = \mathbf{e}_1 + \cdots + \mathbf{e}_d$, $\mathbf{e}_i\,\mathbf{e}_j = 0$ if $i \neq j$, $\mathbf{e}_i^2 = \mathbf{e}_i$ and $\mathcal{A}_i = \mathbf{e}_i\,\mathcal{A}$. 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 $\lambda : \ p \mapsto \lambda(p)$, $p \in R$, $\lambda(p) \in \mathbb{C}$. ($R$ will be the primal space for $\widehat{R}$.) Let us recall two celebrated examples, that is, $\delta_\zeta : p \mapsto p(\zeta)$, the *evaluation at a fixed point* ζ, and the map

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

where $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n$ is any vector, and d_{x_i} is the 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** $\mathbb{C}[[\mathbf{d}_1, \ldots, \mathbf{d}_n]]$ and will also write *"f.p.s."* to abbreviate "formal power series". 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 $\delta_0 = \mathbf{d}^0$. We can multiply 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 define $p \star \Lambda : q \mapsto \Lambda(p\,q)$, $q \in R$, $\Lambda(p, q) \in \mathbb{C}$. For any pair of elements $p \in R$ and $a \in \mathbb{N},\ a > 1$, we have

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

Consequently, for any pair $p \in R, \mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n$ (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, [13]). 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 $p, q \in R$ 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 on 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 $[p \star \Lambda(\mathbf{x}^\beta)]_{\beta\in F}$:

1. *Let $\tilde{\Lambda}(\mathbf{d}) = \sum_{\beta\in E+F} \Lambda(\mathbf{x}^\beta)\,\mathbf{d}^\beta$.*
2. *Compute the product $\rho(\mathbf{d}) = p(\mathbf{d}^{-1})\tilde{\Lambda}(\mathbf{d})$ in $\mathbb{C}[\mathbf{d}, \mathbf{d}^{-1}]$.*
3. *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.

Definition 3.1 *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 $\mathtt{i}$ be the i^{th} canonical vector of $\mathbb{N}^n$.*

- *M 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 $i = 1, \ldots, n$, 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 $i = 1, \ldots, n$, 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 $H_M(\mathbf{d}) = \sum_{\mathbf{u}\in E-F} h_{\mathbf{u}} \mathbf{d}^{\mathbf{u}}$.*

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

For $E = [0, \ldots, m-1]$ and $F = [0, \ldots, n-1]$ (resp. $F = [-n+1, \ldots, 0]$), definition 3.1 turns into the usual definition of Hankel (resp. Toeplitz) matrices (see [2]).

Definition 3.2 *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. We 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 [16], [15]), 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 3.3 *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. $\pi_E \circ \chi_{H_M} \circ \pi_F$), where for any $p \in L$, $\mu_p : q \mapsto p\,q$ is the operator of multiplication by p in L.*

Proof. 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 $v(\mathbf{x}) = \sum_{\beta\in F} v_\beta \mathbf{x}^\beta$. Then

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

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 $M\,\mathbf{v}$ (see [16]). □

Algorithm 3.4 MULTIPLY 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 polynomial $T_M = \sum_{\mathbf{u}\in E+F} t_{\mathbf{u}}\, \mathbf{x}^{\mathbf{u}}$ (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 project the product on $\mathbf{x}^E$ (resp. $\mathbf{d}^E$).*

Definition 3.5 *Hereafter, "ops" stand for "arithmetic operations", and $C_{PolMult}(E, F)$ denotes the number of ops required to multiply a polynomial with a support in E by a polynomial with a support in F.*

Now, we estimate that algorithm 3.4 can be performed by using $C_{PolMult}(E + F, F)$, resp. $C_{PolMult}(E - F, -F)$ ops. This cost is bounded in the following proposition:

Proposition 3.6 *An (E, F) quasi-Hankel (resp. an (E, F) quasi-Toeplitz) matrix M can be multiplied by a vector in $\mathcal{O}(N \log^2 N + C_{M,N})$ ops, where $N = \lfloor E - 2\,F \rceil$ (resp. $\lfloor E + 2F \rceil$) and where $C_{M,N}$ bounds the cost of the evaluation of the polynomial H_M (resp. T_M) on a fixed set of N points.*

Proof. See [16]. □

In the same spirit, we can bound the number of ops in algorithm 2.2:

Proposition 3.7 *For any fixed pair of a linear form Λ and a vector $[\Lambda(\mathbf{x}^\alpha)]_{\alpha\in E+F}$, 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, solving a linear system can also be performed efficiently using the the following result (cf. e.g. [2]).

Theorem 3.8 *Let S be a finite set with $|S|$ elements and let $W\,\mathbf{v} = \mathbf{w}$ be a non-singular linear system of N equations. Then choosing $2\,N$ random parameters from S (independently of each other and under the uniform probability distribution on S) and performing $2N$ multiplications of W by vectors and $\mathcal{O}(N^2)$ other arithmetic operations suffice either to compute the solution $\mathbf{v}$ to the linear system $W\,\mathbf{v} = \mathbf{w}$ with a probability at most $1 - \frac{2\,N}{|S|}$ or to output* FAILURE *with a probability at most $\frac{2\,N}{|S|}$.*

This method involves $2N$ multiplications of W by a vector. In the case of structured matrices, it yields a fast algorithm for solving the linear system $W\,\mathbf{v} = \mathbf{w}$.

Theorem 3.9 *[17]. Under the notation of theorem 3.8, let W be an N-by-N real symmetric matrix. Then $\mathcal{O}(N)$ multiplications of W by vectors and $\mathcal{O}(N^2)$ other ops suffice to compute the rank and the signature of W.*

Sketch of proof. Tridiagonalize the matrix W by the Lanczos randomized algorithm. Then obtain the numbers n_+ and n_- of positive and negative eigenvalues of W from the Sturm sequences of the values of the characteristic polynomials of all leading principal submatrices of W. These two numbers immediately define the rank and signature of W. Apply the estimates of [2] for the complexity of all steps of this computation. □

4 Computing a non-degenerate linear form

Our algorithms of the next sections perform computations in A efficiently based on the knowledge of a certain linear form on $\mathcal{A}$, which induces a non-degenerate inner product. More precisely, we assume the following items 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 non-degenerate,*
- *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 construct this linear form in the case of a system of n equations in n unknowns having finite number of isolated roots (that is, in the case of complete intersection); furthermore, we assume some genericity conditions. Our construction is based on the resultant matrix approach to the solution (either using the classical or sparse resultant), and we assume that the input system is *generic for this resultant construction*. Our construction can be generalized to other resultant matrices in the complete intersection case (see [5]). We can also obtain our basic linear form from a normal form algorithm (e.g. if a Groebner basis is available) in the case of complete intersection.

We denote by E_i the support of the polynomial f_i and by E_0 a given set of monomials. Next, we construct a non-degenerate linear form τ on $\mathcal{A}$, and the set $[\tau(\mathbf{x}^{\alpha})]_{\alpha \in F}$ for $F = E_0 + E_1 + \cdots + E_n$. This construction is based on the resultant matrix computations.

Let f_0 be a *random* polynomial with support in $\mathbf{x}^{E_0}$. To construct resultants, as in the work of Macaulay [12] (see also [9]), we may use matrices associated to maps of the form:

$$\begin{array}{rcl} \Phi : \mathcal{V}_0 \times \cdots \times \mathcal{V}_n & \rightarrow & \mathcal{V} \\ (q_0, \ldots, q_n) & \mapsto & \displaystyle\sum_{i=0}^{n} f_i\, q_i, \end{array} \tag{4}$$

where $\mathcal{V}_i = \langle \mathbf{x}^{F_i} \rangle$ is a vector space generated by a finite number of monomials. We denote by F_i the set of the exponents of these monomials: $F_i = \{\beta_{i,1}, \beta_{i,2}, \ldots\}$. The vector space $\mathcal{V} = \langle \mathbf{x}^{F} \rangle$ is also a vector space generated by the monomials, whose exponents are in the set F. The matrix of Φ can be divided into blocks $[N_0, N_1, \ldots, N_n]$. The columns of the blocks correspond to the multiples of f_i expressed in the monomial basis $\mathbf{x}^{F}$. The matrix generalizes the Sylvester matrix of two univariate polynomials to the case of multivariate polynomials. It belongs to the class of quasi-Toeplitz matrices (see section 3). >From the matrix of this map, it is possible to extract a maximal square submatrix S, which is generically of maximal rank (see [12],[3], for more details). The latter submatrix can be partitioned into four blocks as follows:

$$S = \begin{bmatrix} U & V \\ Z & W \end{bmatrix}$$

where W is invertible, and all blocks are quasi-Toeplitz matrices.

Proposition 4.1 *For any vector $\Lambda_0 = [\Lambda_{\alpha}]_{\alpha \in E_0}$, the vector*

$$\Lambda_0^{\mathtt{t}} \left[\begin{array}{c|c} \mathbb{I}_D & -VW^{-1} \end{array} \right] = [\Lambda_{\beta}]_{\beta \in F}$$

is the vector of the coefficients $(\mathbf{d}^{\beta})_{\beta \in F}$ of an element Λ of $\widehat{\mathcal{A}}$. For a random choice of Λ_0, the linear form Λ defines a non-degenerate inner product on $\mathcal{A}$.

Proof. The rows of the matrix $\left[\begin{array}{c|c} \mathbb{I}_D & -VW^{-1} \end{array} \right]$ are orthogonal to the columns of the matrix $\begin{bmatrix} V \\ W \end{bmatrix}$, representing multiples of the polynomials $f_1, \ldots, f_n$. Therefore, these rows are the coefficients vectors of the elements of $\widehat{\mathcal{A}} = I^{\perp}$ in the dual basis $(\mathbf{d}^{\alpha})$. Taking a random combination Λ_0 of these rows yields, with a high probability, the coordinate vector of the linear form on $\mathcal{A}$ that induces a non-degenerate bilinear form. □

This computation involves the solution of a linear system of equations defined by the quasi-Toeplitz matrix W. Using proposition 3.6, we have the following property:

Proposition 4.2 *The coefficients $[\tau(\mathbf{x}^{\beta})]_{\alpha \in E+E+E}$ where $\mathbf{x}^{E}$ is a basis of $\mathcal{A}$ can be computed in $\mathcal{O}(12^n D^2 \log(D))$ ops.*

Proof. The set $E_1 + \cdots + E_n$ contains a set E such that $\mathbf{x}^{E}$ is a basis of $\mathcal{A}$ (see [7], [21], [17]). If we let $E_0 = E + E$, the previous computation yields the coefficients $[\tau(\mathbf{x}^{\beta})]_{\alpha \in E+E+E}$, We observe that the size of the matrix W that we need to invert is a (E', E')quasi-Toeplitz where $E' = E + E$ (with $|E'| \leq 2^n |D|$) and we apply proposition 3.6 and theorem 3.8 to W. □

We also refer the reader to the definitions of section 3.

Remark. Our alternative algorithm computes all the coefficients $[\tau(\mathbf{x})^{\beta})]_{\alpha}$ in $O(M 4^n D^2 \log(D))$ ops, where M is the overall number of all monomials in the polynomials of the input system, M is relatively small for sparse systems.

5 Arithmetic in $\mathcal{A}$

In this section, we assume that we have a basic set of items (including linear form τ) defined in the previous section. 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 H_{Λ}^{F} denote the quasi-Hankel matrix, $\mathtt{H}_{\Lambda}^{F} = (\Lambda(\mathbf{x}^{\alpha+\beta}))_{\alpha,\beta \in F}$.*

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

Proof. We apply proposition 3.4 to the (F, F) quasi-Hankel matrix $\mathtt{H}_\Lambda^F$. □

Combining theorem 3.8 and proposition 5.2 implies the following result:

Proposition 5.3 *Checking if the linear system $\mathtt{H}_\Lambda^F \mathbf{u} = \mathbf{v}$ has a unique solution and computing its solution (resp. computing its rank) requires $\mathcal{O}(3^n \lfloor F \rceil^2 \log(\lfloor F \rceil))$ ops.*

5.1 Dual basis

As τ defines a non-degenerate bilinear form, there exists a family $(\mathbf{w}_\alpha)_{\alpha\in E}$ such that $\tau(\mathbf{x}^\alpha\, \mathbf{w}_\beta) = \delta_{\alpha,\beta}$, $\delta_{\alpha,\beta}$ being Kronecker's symbol, $\delta_{\alpha,\alpha} = 1$, $\delta_{\alpha,\beta} = 0$ if $\alpha \neq \beta$. The family $(\mathbf{w}_\alpha)_{\alpha\in E}$ 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*

$$p \equiv \sum_{\alpha\in E} \tau(p\, \mathbf{w}_\alpha)\mathbf{x}^\alpha \equiv \sum_{\alpha\in E} \tau(p\, \mathbf{x}^\alpha)\mathbf{w}_\alpha\,. \tag{5}$$

Proof. See [5], [6]. □

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

$$\tau(\mathbf{w}_\alpha\, \mathbf{x}^\gamma) = \sum_{\beta\in E} w_{\alpha,\beta}\, \tau(\mathbf{x}^{\beta+\gamma}) \tag{6}$$

is 1 if $\alpha = \gamma$ and 0 elsewhere. In terms of matrices, equation (6) implies that

$$\mathtt{H}_\tau\, \mathtt{W}_\tau = \mathbb{I}_D \tag{7}$$

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 relation (7), we deduce that

$$[p]_\mathbf{x} = \mathtt{W}_\tau\, [p]_\mathbf{w},\ [p]_\mathbf{w} = \mathtt{H}_\tau\, [p]_\mathbf{x}. \tag{8}$$

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 $\mathcal{O}(3^n D^2\, \log(D))$ ops.*

5.2 Product in $\mathcal{A}$

We apply the projection formula (5) 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 an f.p.s., 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 $f, g \in \langle \mathbf{x}^\alpha \rangle_{\alpha\in E}$, COMPUTE THE PRODUCT $f\, g$ IN THE BASIS $\langle \mathbf{x}^\alpha \rangle_{\alpha\in E}$ OF $\mathcal{A}$.

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 coefficient of $f\, g \star \tau$.*
3. *Solve the system $[f\, g]_\mathbf{w} = \mathtt{H}_\tau\, \mathbf{u}$.*

The vector $\mathbf{u}$ 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 $\mathcal{O}(3^n D^2 \log(D))$ 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 $\mathcal{O}^*(4^n D)$ ops (see proposition 3.6). The complexity of the third step is bounded according to proposition 5.3 (with $F = E$). □

5.3 Inverse in $\mathcal{A}$

The projection formula of proposition 5.4 also 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 (8). 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 inverse in $\mathcal{A}$:

Algorithm 5.9 FOR ANY $f \in \langle \mathbf{x}^\alpha \rangle_{\alpha\in E}$, COMPUTE THE INVERSE OF $f \in \mathcal{A}$ IF IT EXISTS.

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

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

By combining propositions 5.2, 5.3, and 3.7, we obtain

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

6 Iterative methods

Our algorithms will amount to computing non-trivial idempotents by iterative processes. The algorithms work in $\mathbb{C}$, and we write $\mathbf{i} = \sqrt{-1}$. More rudimentary univariate versions of these algorithms were studied in [4]. We will use the basic operations in the quotient algebra $\mathcal{A}$ in order to devise two

iterative methods, which eventually yield non-trivial idempotents. We will first consider iteration associated to a slight modification of the so-called *Joukovski* map (see [10],[4]): $z \mapsto \frac{1}{2}(z+\frac{1}{z})$ 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. $u_0 = h \in \langle \mathbf{x}^\alpha \rangle_{\alpha \in E}$. $u_{n+1} \equiv \frac{1}{2}(u_n - \frac{1}{u_n}) \in \mathcal{A},\ n = 0, 1, \ldots$.

By proposition 5.10, we have

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

Proof. By Proposition 5.3, an element of $\mathcal{A}$ can be inverted in $\mathcal{O}(3^n D^2 \log(D))$ ops. A linear combination of u_n and u_n^{-1} can be computed in D ops, which yields the required bound. □

Proposition 6.3 *Assume that for any root $\zeta \in \mathcal{Z}$, $\Re(h(\zeta)) \neq 0$. Then the sequence (u_n) converges quadratically to $\sigma = \sum_{\Im(h(\zeta))>0} \mathbf{e}_\zeta - \sum_{\Im(h(\zeta))<0} \mathbf{e}_\zeta$ ($\Im$ is the imaginary part), and we have*

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

(for some constant K), where

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

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

and $\rho = \max\{\rho^+, \rho^-\}$.

Proof. We apply the classical convergence analysis of the Joukovski map (see [10]) to the matrices of multiplication by u_n in $\mathcal{A}$, whose eigenvalues are $\{u_n(\zeta), \zeta \in \mathcal{Z}(I)\}$. □

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

denote the idempotents associated to the roots $\zeta \in \mathcal{Z}$ such that $\Im(h(\zeta)) > 0$ and $\Im(h(\zeta)) < 0$, respectively. The choice of $h = x_i - \epsilon$ and $h = x_i + \epsilon$ allows us to recover the two idempotents,

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

Their product can be computed in $\mathcal{O}^*(3^n D^2)$ 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 $\mathcal{O}^*(3^n D^2)$ ops, to yield the sum of the fundamental idempotents, whose roots are nearly real.

Algorithm 6.4 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 $\mathbf{e}^\pm_{i,\epsilon}$ *and* $\mathbf{r}_{i,\epsilon}$.
- *Compute the product* $\mathbf{r}_\epsilon \equiv \mathbf{r}_{1,\epsilon} \cdots \mathbf{r}_{n,\epsilon}$ *in* $\mathcal{A}$.

According to propositions 6.2 and 6.3, we have

Proposition 6.5 *An approximation of $\mathbf{r}_\epsilon$ (with the error $\epsilon = 2^{-b}$) can be computed in $\mathcal{O}(3^n D^2 \log(D) \log(|\frac{b}{\log(\rho)}|))$ ops, where*

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

The second iterative method is the quadratic inverse power method:

Algorithm 6.6 QUADRATIC INVERSE POWER ITERATION. $u_0 = h \in \langle \mathbf{x}^\alpha \rangle_{\alpha \in E}$. $u_{n+1} \equiv \frac{1}{u_n^2} \in \mathcal{A},\ n = 0, 1, \ldots$.

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

Proposition 6.7 *An approximation (up to the error ϵ) of the idempotent $\mathbf{e}_\zeta$ such that a simple root ζ minimizes $|h|$ on $\mathcal{Z}(I)$ can be computed in $\mathcal{O}(3^n D^2 \log(D) \log(|\frac{\rho}{b}|))$ ops, where $\rho = |\frac{h(\zeta)}{h(\zeta')}|$ and $|h(\zeta')|$ is the second smallest value of $|h|$ over $\mathcal{Z}(I)$.*

Proof. We rely on the convergence analysis of the quadratic inverse 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 Computing the (real) roots

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

We may restrict our computation on $\mathcal{A}^{\mathbb{R}}_\epsilon$ by computing the linear form $\tau' = \mathbf{r}_\epsilon \star \tau$ (in $\mathcal{O}^*(3^n D^2)$ ops, according to proposition 3.7), and we have the following properties:

Proposition 7.1 *The linear form $\tau' = \mathbf{r}_\epsilon \star \tau$ defines a non-degenerate inner product on $\mathcal{A}^{\mathbb{R}}_\epsilon$.*

The number of nearly real roots (counted with multiplicities) is the rank of $H^E_{r_\epsilon \star \tau}$.

Let E' be a subset of E such that the submatrix $H^{E'}_{\tau'}$ is of maximal rank. Then E' is a basis of $\mathcal{A}_\epsilon$.

Proof. See [17]. □

This leads to an algorithm for computing the rank of $H^E_{\tau'}$ and, by theorem 3.9, we have:

Proposition 7.2 *The number of real roots can be computed in $\mathcal{O}(3^n D^2 \log(D))$ ops.*

To compute (real) root minimizing a given function $|h|$, we may apply algorithm 6.6 in $\mathcal{A}$ (or $\mathcal{A}_\epsilon^{\mathbb{R}}$) and obtain the following theorem:

Theorem 7.3 *The idempotent corresponding to the (real) root ζ, which minimizes a function $|h|$, can be computed (up to an error $\epsilon = 2^{-b}$) in $\mathcal{O}(3^n D^2 \log(D) \log(b))$ ops.*

This process can be used to compute the other roots via deflation. 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 algorithm 6.1 to compute the sum of idempotents corresponding to the roots inside the box. The complexity of each step beeing bounded in theorem 7.3, this leads to the following result for δ real roots in a given box:

Theorem 7.4 *The idempotent corresponding to the δ (real) roots ζ in a given box can be computed (up to an error $\epsilon = 2^{-b}$) by using $\mathcal{O}(3^n \delta\, D^2\, \log(D) \log(b))$ ops.*

The final step of our algorithms determines a root ζ from the idempotent $\mathbf{e}_\zeta$.

Proposition 7.5 *The n coordinates of the root ζ can be determined from the idempotent $\mathbf{e}_\zeta$ in $\mathcal{O}^*(3^n D^2)$ ops.*

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 [15], [17], in the case of a simple root, we have

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

This vector is computed within the complexity bound of proposition 5.2 and immediately gives us the coordinates of the root ζ if $\mathbf{x}^E$ contains $1, x_1, \ldots, x_n$, 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 [15], [17], [6]), we recover the coordinates of ζ, by computing $n+1$ products in $\mathcal{A}$ (by algorithm 5.7). □

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 *Proc. Intern. Conf. on Numerical Math.*, volume 86 of *Int. Series of Numerical Math*, pages 11–30. Birkhäuser, 1988.

[2] D. Bini and V. Pan. *Polynomial and Matrix Computations, Volume 1 : Fundamental Algorithms*. Birkhäuser, Boston, 1994.

[3] J. Canny and I. Emiris. An efficient algorithm for the sparse mixed resultant. In G. Cohen, T. Mora, and O. Moreno, editors, *Proc. Intern. Symp. Applied Algebra, Algebraic Algor. and Error-Corr. Codes (Puerto Rico)*, volume 263 of *Lect. Notes in Comp. Science*, pages 89–104. Springer, 1993.

[4] J.P. Cardinal. On two iterative methods for approximating the roots of a polynomial. In J. Renegar, M. Shub, and S. Smale, editors, *Proc. AMS-SIAM Summer Seminar on Math. of Numerical Analysis, (Park City, Utah, 1995)*, volume 32 of *Lectures in Applied Math.*, pages 165–188. Am. Math. Soc. Press, 1996.

[5] J.P. Cardinal and B. Mourrain. Algebraic approach of residues and applications. In J. Renegar, M. Shub, and S. Smale, editors, *Proc. AMS-SIAM Summer Seminar on Math. of Numerical Analysis, (Park City, Utah, 1995)*, volume 32 of *Lectures in Applied Math.*, pages 189–210. Am. Math. Soc. Press, 1996.

[6] M. Elkadi and B. Mourrain. Approche Effective des Résidus Algébriques. Rapport de Recherche 2884, INRIA, 1996.

[7] I. Emiris and A. Rege. Monomial bases and polynomial system solving. In *Proc. ACM Intern. Symp. on Symbolic and Algebraic Computation, Oxford*, pages 114–122, 1994.

[8] I.Z. Emiris and V.Y. Pan. The structure of sparse resultant matrices. In *Proc. ACM Intern. Symp. Symbolic Algebraic Comput. (ISSAC)*, pages 189–196, Maui, Hawaii, July 1997.

[9] I.M. Gelfand, M.M. Kapranov, and A.V. Zelevinsky. *Discriminants, Resultants and Multidimensional Determinants*. Birkhäuser, Boston-Basel-Berlin, 1994.

[10] P. Henrici. *Applied and Computational Complex Analysis*, volume I. Wiley, 1988.

[11] D.Kapur and Y.N. Lakshman. Elimination methods: an introducton. In B. Donald, D. Kapur, and J. Mundy, editors, *Symbolic and Numerical Computation for Artifitial Intellingence*, pages 45–89. Academic Press, New York, 1992.

[12] F.S. Macaulay. Some formulae in elimination. *Proc. London Math. Soc.*, 1(33):3–27, 1902.

[13] F.S. Macaulay. *The Algebraic Theory of Modular Systems*. Cambridge Univ. Press, 1916.

[14] B. Mourrain. Solving polynomial systems by matrix computations. Preprint, submitted, 1997.

[15] B. Mourrain and V. Y. Pan. Multidimensional structured matrices and polynomial systems. *Calcolo, (Special Issue, Workshop on Toeplitz Matrices: Structure, Algorithms and Applications)*, 33:389–401, 1997.

[16] B. Mourrain and V. Y. Pan. Solving special polynomial systems by using structured matrices and algebraic residues. In F. Cucker and M. Shub, editors, *Proc. of the Workshop on Foundations of Computational Mathematics (Rio de Janeiro. 1997)*, pages 287–304. Springer, 1997.

[17] B. Mourrain and V.Y. Pan. Multivariate polynomials, duality and structured matrices. Preprint, submitted for publication, 1997.

[18] V. Y. Pan. Optimal (up to polylog factors) sequential and parallel algorithms for approximating complex polynomial zeros. In *Proceedings, 27th Annual ACM Symp. on Theory of Computing*, 741-750, ACM Press, New York, 1995.

[19] V. Y. Pan. Optimal and nearly optimal algorithms for approximating complex polynomial zeros. *Computers and Math. Appls.*, 31(12):97–138, 1996.

[20] V. Y. Pan. Solving a polynomial equation: some history and recent progress. *SIAM Review*, 39(2):187–220, 1997.

[21] P. S. Pedersen and B. Sturmfels. Product formulas for resultants and Chow forms. *Math. Zeitschrift*, 214:377–396, 1993.

[22] J. Renegar, On the worst-case arithmetic complexity of approximating zeros of systems of polynomials, *SIAM J. Com-*

put. , **18**, 350-370, 1989.

[23] J. Renegar. On the worst-case arithmetic complexity of approximating zeros of polynomials, *J. Complexity*, **3**, 90-113, 1987.

[24] B.L. Van der Waerden. *Modern Algebra, Vol. II*. Frederick Ungar Publishing Co, 1948.