# Multilinear Polynomial Systems: Root Isolation and Bit Complexity

Ioannis Emiris, Angelos Mantzaflaris, Elias Tsigaridas

# Multilinear Polynomial Systems: Root Isolation and Bit Complexity

Ioannis Z. Emiris

*National and Kapodistrian University of Athens, Greece*
*ATHENA Research and Innovation Center, Maroussi, Greece*

Angelos Mantzaflaris

*Université Côte d'Azur, Inria Sophia Antipolis - Méditerranée, France*

Elias P. Tsigaridas

*Sorbonne Université,* CNRS, INRIA, *Laboratoire d'Informatique de Paris 6,* LIP6, *Équipe* PolSys, *4 place Jussieu, F-75005, Paris, France*

**Abstract**

We exploit structure in polynomial system solving by considering polynomials that are linear in subsets of the variables. We focus on algorithms and their Boolean complexity for computing isolating hyperboxes for all the isolated complex roots of well-constrained, unmixed systems of multilinear polynomials based on resultant methods. We enumerate all expressions of the multihomogeneous (or multigraded) resultant of such systems as a determinant of Sylvester-like matrices, aka *generalized Sylvester matrices*. We construct these matrices by means of Weyman homological complexes, which generalize the Cayley-Koszul complex.

The computation of the determinant of the resultant matrix is the bottleneck for the overall complexity. We exploit the quasi-Toeplitz structure to reduce the problem to efficient matrix-vector multiplication, which corresponds to multivariate polynomial multiplication, by extending the seminal work on Macaulay matrices of Canny, Kaltofen, and Yagati [9] to the multihomogeneous case.

*Email addresses:* emiris@di.uoa.gr (Ioannis Z. Emiris), angelos.mantzaflaris@inria.fr (Angelos Mantzaflaris), elias.tsigaridas@inria.fr (Elias P. Tsigaridas)

We compute a rational univariate representation of the roots, based on the primitive element method. In the case of 0-dimensional systems we present a Monte Carlo algorithm with probability of success $1 - 1/2^{\varrho}$, for a given $\varrho \geq 1$, and bit complexity $\widetilde{\mathcal{O}}_B(n^2 D^{4+\epsilon}(n^{N+1} + \tau) + nD^{2+\epsilon}\varrho(D + \varrho))$ for any $\epsilon > 0$, where $n$ is the number of variables, $D$ equals the multilinear Bézout bound, $N$ is the number of variable subsets, and $\tau$ is the maximum coefficient bitsize. We present an algorithmic variant to compute the isolated roots of overdetermined and positive-dimensional systems. Thus our algorithms and complexity analysis apply in general with no assumptions on the input.

*Keywords:* multilinear polynomial, DMM separation bound, resultant matrix, Cayley-Koszul complex, primitive element, rational univariate representation, bit complexity

---

## 1. Introduction

Efficient algorithms for solving polynomial systems represents a core activity in computational algebra. Its importance stems from the fact that polynomial systems model many problems in various scientific and engineering disciplines. We focus on efficient algorithms for *multilinear* systems, which are common in numerous applications, for example in cryptography [24, 33], coding theory [42], real algebraic geometry [47], and game theory [22]. We derive explicit Boolean complexity estimates for isolating all roots of multilinear polynomial systems without any assumptions on the input.

We consider $N$ variable subsets and denote by $S_k(d) = \mathbb{R}[x_{k,1}, \ldots, x_{k,n_k}]_d$, where $1 \leq k \leq N$, the set of polynomials of degree at most $d$ in variables $x_{k,1}, \ldots, x_{k,n_k}$. We denote the space of polynomials of multidegree $(d_1, d_2, \ldots, d_N)$ by

$$S(d_1, d_2, \ldots, d_N) := S_1(d_1) \otimes \cdots \otimes S_N(d_N)\,.$$

The total number of (affine) variables is then $n = n_1 + \cdots + n_N$. We consider a system of equations of $n$ polynomials, $f_1, \ldots, f_n$, such that $f_k \in S(1, \ldots, 1)$, for $1 \leq k \leq n$. We call these polynomials $(n_1, \ldots, n_N)-$multilinear, that is, after group-wise homogenization they have degree one with respect to each of the variable groups $\{x_{1,0}, x_{1,1}, \ldots, x_{1,n_1}\}$, $\ldots$, $\{x_{N,0}, x_{N,1}, \ldots, x_{N,n_N}\}$. Using this notation, our goal is to isolate all the complex roots of the system

$$(\Sigma):\ f_1(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \cdots = f_n(\mathbf{x}_1, \ldots, \mathbf{x}_N) = 0, \tag{1}$$

where $\mathbf{x}_k = (x_{k,1}, \ldots, x_{k,n_k})$ and $1 \leq k \leq N$. Assuming for the moment that this system is zero-dimensional in the multiprojective space, the number of roots in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_N}$ equals the multilinear Bézout bound

$$D = D(n_1, \ldots, n_N) := \frac{(n_1 + \cdots + n_N)!}{n_1! n_2! \cdots n_N!} = \frac{n!}{\prod_{r=1}^{N} n_r!} \,. \tag{2}$$

This number is the mixed volume of the system, and, even in case of infinitely many roots, bounds the degree of the variety [22]. This quantity appears frequently in the complexity of our algorithms. We can use Stirling's approximation to estimate the multinomial as

$$D \sim \frac{n^n \sqrt{2\pi n}}{\prod_{k=1}^{N} n_k^{n_k} \sqrt{2\pi n_k}} \,.$$

In particular, for $N = 2$ the maximum number of roots the systems, using the previous approximation, occurs when $n_1 \approx n_2$, which results $D \sim \frac{4^n}{\sqrt{\pi n}}$. However, if we assume $\min(n_1, n_2) = q$, for some constant $q$, then $D \sim n^q$, that is the bound on the roots is polynomial in the number of variables.

**Example 1.** *Consider the* $(1,2)-$*multilinear, square system*

$$\begin{aligned}
f_1 &= b_0 + b_1 y_2 + b_2 y_1 + b_3 x_1 + b_4 x_1 y_2 + b_5 x_1 y_1 \\
&= 1 + y_2 + y_1 + x_1 + 2\, x_1 y_2 + 2\, x_1 y_1 \\
f_2 &= c_0 + c_1 y_2 + c_2 y_1 + c_3 x_1 + c_4 x_1 y_2 + c_5 x_1 y_1 \\
&= 1 + y_2 + 2\, y_1 + 2\, x_1 + x_1 y_2 + x_1 y_1 \\
f_3 &= d_0 + d_1 y_2 + d_2 y_1 + d_3 x_1 + d_4 x_1 y_2 + d_5 x_1 y_1 \\
&= 2 + y_2 + 2\, y_1 + x_1 + 2\, x_1 y_2 + 2\, x_1 y_1
\end{aligned}$$

*with variable groups* $\{x_1\}$ *and* $\{y_1, y_2\}$*. The number of roots of the system in* $\mathbb{P}^1 \times \mathbb{P}^2$ *is* $D(1,2) = 3$*. These are the two affine points* $(x_1; y_1, y_2) = (\pm\frac{\sqrt{3}}{3}; -1, 2 \mp \sqrt{3})$ *and the root* $(x_0, x_1; y_0, y_1, y_2) = (0, 1; 0, -1, 1)$ *at infinity.*

We use $\mathcal{O}_B$, resp. $\mathcal{O}$, to denote bit, resp. arithmetic, complexity and the $\widetilde{\mathcal{O}}_B$, resp. $\widetilde{\mathcal{O}}$, notation means that we are ignoring logarithmic factors.

*Previous work and our contribution.* Systems that are homogeneous in distinct groups of variables were first discussed by Sylvester [51]. Muir [41] and McCoy [39] gave the first expressions of the resultant of such forms as the determinant of a matrix. We can consider multilinear resultants as a special case of hyperdeterminants, in the so-called "boundary case" in [53]. Sturmfels and Zelevinski [50] first discovered Sylvester-type formulas for unmixed, that is systems of polynomial with the same support, multigraded systems. We can interpret these formulas as certain choices of a Weyman complex of modules [53, 54]. Indeed, we can discover several classical resultant matrices via such complexes [55], including the projective resultant [12]. The discovery of new resultant formulas coming from these homological constructions was fully explored in [14, 16], where combinatorial bounds were established for possible determinantal complexes, which allowed for an implementation discovering all such complexes. These works extended the study of resultant formulas from Sylvester-type matrices to Bézout-type and hybrid matrices. Interestingly, hybrid resultant matrices are made up of Bézout-type, Sylvester-type blocks (and toric Jacobians). Similar maps have been identified between the terms of Tate resolutions as well [15, 10]. It turns out that for multilinear systems, the most appealing of determinantal formulas are available; these are optimal, pure Sylvester formulas, that are quite analogous to the classical Sylvester matrix of two homogeneous polynomials in one variable. For mixed bilinear systems with two supports we refer the reader to [3].

In the present work we revisit the explicit matrix construction of [50], and we elaborate on a new matrix construction. In particular we extend [50, Theorem 1] by introducing a Koszul-type formula for miltilinear resultants.

Regarding the computational complexity, in [25], see also [49], the problem of computing the roots of multilinear systems is tackled by means of Gröbner bases. There, a modification of the F5 criterion which avoids all reductions to zero during the computations is presented. For generic $(n_1,n_2)$–bilinear systems the arithmetic complexity of computing a Gröbner basis is $\mathcal{O}(\binom{2n_1+n_2+1}{n_1+1}^{\omega})$, where $n_1 \leq n_2$ are the sizes of the two blocks of variables and $\omega$ is the exponent of matrix multiplication [23]. To isolate the roots we need to convert the basis to a shape form. The bit complexity of such an approach with respect to the degree bound is not straightforward. Moreover, this arithmetic bound does not hold for non-generic and for positive-dimensional polynomial systems. In this regard, our results are not directly comparable to [25].

Our approach is "orthogonal" to [25]. We introduce a new variant of the $u$-resultant, which relies on the determinantal resultant matrices that we describe in Sec. 2.2. The construction and the properties of these matrices are fundamental for estimating bit complexity of the whole process. The arithmetic complexity of our approach is $\widetilde{\mathcal{O}}(D^3)$, which we derive by adopting to our case the seminal work of Canny, Kaltofen, and Yagati [9] on computing the determinant of a Macaulay matrix by a reduction to multivariate polynomial multiplication. For solving symbolically generic multihomogeneous systems we refer the reader to [48].

To the best of our knowledge, our result is the first one regarding the bit complexity of solving, that is computing the isolated roots, of any multilinear polynomial system, without regularity or any other assumption on the dimension of the zero locus.

We employ the primitive element representation (PER) of the roots [7] and the rational univariate representation (RUR) [1, 5, 38, 46]; for an improved version of RUR in the bivariate case we refer to [6]. A Gröbner free alternative for solving polynomial systems is explored in [28]. We should emphasize that our references with respect to polynomial system solving algorithm are by no means exhaustive and we encourage the interested reader to study the references of the stated bibliography.

If we use the data-structure of straight-line programs there are also efficient algorithms to compute the multihomogeneous resultants, for example [32], with arithmetic complexity which is polynomial in the number of variables and the degree of the resultant and it seems to be cubic to the Bézout bound; we refer the reader to [32, Theorem 5] for a precise statement. For efficient algorithms for computing general resultant matrices we refer to [9, Theorem 3] for a Las Vegas randomized algorithm with arithmetic complexity $\widetilde{\mathcal{O}}(nK^3)$, where $K = \binom{d+n-1}{n-1}$, $d$ is the degree of the polynomials, and $n$ is the number of variables. We also refer to [20, 21] to various improvements that exploit (randomized) computations with structured matrices. For general algorithms for computing the determinant we refer to [34] and references therein.

A preliminary version of the present article appeared in [17], treating only the case of two variable subsets, that is, the case of bilinear systems.

*Paper organization.* We give explicit algorithms to construct all possible resultant matrices, which are *optimal*, in the sense that there are no extraneous factors involved (Sec. 2). We adapt the approach of Canny's [7] to represent

the roots using the primitive element and the rational univariate representation [1, 46] (Sec. 3.1) for multilinear systems, and we bound the height of the corresponding $u$-resultant (Sec. 3.3). We provide explicit bounds for the separation of the roots (Sec. 3.4) that depend on the multilinear Bézout bound. We present explicit bit complexity bounds for isolating all the roots of a system of multilinear polynomials (Thm. 8). The Monte Carlo algorithm has $1 - 1/2^{\varrho}$ probability of success, for a given $\varrho \geq 1$, and runs in $\widetilde{\mathcal{O}}_B(n^2 D^{4+\epsilon}(n^{N+1} + \tau) + nD^{2+\epsilon}\varrho(D + \varrho))$ for any $\epsilon > 0$, where $\tau$ is the maximum coefficient bitsize assuming an oracle that provides random primes of bounded value. We also tackle the cases where the system is overdetermined (Sec. 4.2), has roots at infinity, or is positive-dimensional (Sec. 4.3). In the bilinear case, if one variable subset has constant size, then our bounds are polynomial in $n$.

## 2. Determinantal formulas for the multilinear resultant

In this section we describe determinantal generalized Sylvester formulas for the resultant of multilinear systems. Generalized Sylvester type formulas refer to matrices where the entries are the coefficients of the input polynomials (possibly with a sign change). These correspond to certain Koszul morphisms coming from determinantal complexes. Such expressions are very convenient for both the analysis and the implementation of resultant methods, since the matrix entries have known bitsize and we can compute them in constant time.

First, let us define the multilinear resultant in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_N}$. Given a sequence $f_0, \ldots, f_n$ of $(n_1, \ldots, n_N)-$multilinear forms with variables $\mathbf{x}_1, \ldots, \mathbf{x}_N$ (as in Sect. 1) with symbolic coefficients, their resultant $R(f_0, \ldots, f_n)$ is a multihomogeneous polynomial with integer coefficients having as variables the coefficients of the polynomials of the input system. The degree of the resultant with respect to the coefficients of the polynomial $f_k$ equals the Bézout bound, that is

$$\deg_{f_k} R(f_0, \ldots, f_n) = D(n_1, \ldots, n_N) \ , \quad \text{for } k = 0, \ldots, n \ .$$

The total degree of the resultant is $(n+1)D(n_1, \ldots, n_N)$. It vanishes if and only if the polynomials have a common root in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_N}$. This resultant is an instance of the sparse resultant [26] where the toric variety is $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_N}$; the resultant is unique up to an integer constant. Our

aim is to obtain square matrices, having as entries the coefficients of the polynomials $f_k$, whose determinant equals the resultant; in other words our goal is to obtain determinantal formulas.

*2.1. The Kempf–Lascoux–Weyman resultant complex*

In this section we recall some tools from representation theory, which will help us in the development of multilinear resultant matrices. Extensive introduction to these techniques can be found in [26, Chapter 13]. We may compute the multigraded resultant polynomial as the determinant of the Weyman complex [53] (also known as Kempf–Lascoux–Weyman construction), which arises by applying the, so called, *geometric technique for computing syzygies* [54, Chapter 5] to our case. More precicely, we define the line bundle

$$\mathcal{L} := S(1,\dots,1) \cong \mathbb{P}^{(1+n_1)\cdots(1+n_N)-1},$$

and the projective variety $\mathcal{L}^{n+1}$, which yields a very ample vector bundle of rank $n+1$ on the irreducible projective variety $\mathcal{L}$. On the total space $\mathcal{L}^{n+1}\times X$, we have the natural projection $\pi : \mathcal{L}^{n+1}\times X \to \mathcal{L}^{n+1}$ where $X = \mathbb{P}^{n_1}\times\cdots\times\mathbb{P}^{n_N}$. We consider the incidence variety (cl stands for the Zariski closure of the set)

$$\mathcal{Z} = \mathrm{cl}\big\{(f_0,\dots,f_n,\boldsymbol{\alpha})\in\mathcal{L}^{n+1}\times X :\ f_0(\boldsymbol{\alpha})=\cdots=f_n(\boldsymbol{\alpha})=0\big\}$$

and the restriction $\pi^* : \mathcal{Z}\to\mathcal{L}^{n+1}$ of $\pi$, which is a birational isomorphism (cf. [54, Proposition 9.1.1]). Then, $\mathcal{Z}$ is smooth and provides a desingularization of the resultant variety. The closure of the image

$$\pi^*(\mathcal{Z}) = \big\{\,(f_0,\dots,f_n)\in\mathcal{L}^{n+1} :\ \exists\boldsymbol{\alpha}\in X, f_0(\boldsymbol{\alpha})=\cdots=f_n(\boldsymbol{\alpha})=0\big\}$$

is equal to the resultant variety. The latter is described by the equation $R(f_0,\dots,f_n)=0$ that we mentioned previously.

The variety $\mathcal{Z}$ is irreducible of codimension 1 in $\mathcal{L}^{n+1}$ and as such it admits a locally free resolution in terms of Koszul complexes [26, Chapter 2, Section 1B]. More generally, for each vector bundle $\mathcal{M} = S(\mathbf{m})$ on $X$, for some $\mathbf{m}\in\mathbb{Z}^N$, we can construct a generically exact complex $K_\bullet = K_\bullet(\mathcal{L}^{n+1},\mathcal{M})$ of graded free $\mathcal{L}-$modules which implies a free resolution of the coordinate ring of the resultant variety [54, Theorem 9.1.2], which we write down as

$$0\to K_{n+1}\to\cdots\to K_1\to K_0\to\cdots\to K_{-n}\to 0\ . \tag{3}$$

In the sequel we elaborate on the combinatorics of the complex (3) and we develop the main properties that we will need later on. In particular, the sygygy modules in (3) depend solely on $n_1, \ldots, n_N$ and $\mathbf{m} = (m_1, \ldots, m_N)$, which is a "degree vector" that corresponds to a choice of the vector bundle $\mathcal{M}$ [54, Proposition 4.1.3]. They are defined as $K_\nu = \oplus_{p=0}^{n+1} K_{\nu,p}$ and each summand $K_{\nu,p}$ is either vanishing or has the form

$$K_{\nu,p} = \left(H^{a_1}_{n_1}(m_1 - p) \otimes \cdots \otimes H^{a_N}_{n_N}(m_N - p)\right)^{\binom{n+1}{p}}, \tag{4}$$

for some integers $a_k \in \{0, n_k\}$ with $a_1 + \cdots + a_N = p - \nu$, $-n \leq \nu \leq n+1$. Here $H^a_{n_k}(b)$, $b \in \mathbb{Z}$ is the $a-$th cohomology group of the projective space $\mathbb{P}^{n_k}$ with coefficients in the sheaf $\mathscr{O}(b)$, see [29, Sections 2.5 & 3.5] for more details. The maps between the terms depend polynomially on the coefficients of $f_0, \ldots, f_n$. This construction appears in [53] and [54] gives a detailed presentation.

The crucial property of the complex (3) is that its determinant equals (a power of) $R(f_0, \ldots, f_n)$ [54, Proposition 9.1.3]. The determinant of the complex is, in principle, a rational expression involving the determinants of the maps in the complex [26, Appendix A], and, usually, it is not given as the determinant of a single matrix. However, when the complex has only two non-zero terms (for specific integers $m_1, m_2, \ldots, m_N$), then we obtain the resultant as the determinant of the square matrix expressing the map $\varphi$ at the non-zero part of the complex. We call such complexes and the induced square matrix expressions *determinantal*. The determinantal complexes are of the form

$$0 \to \bigoplus_{p=0}^{n+1} K_{1,p} \xrightarrow{\varphi} \bigoplus_{p=0}^{n+1} K_{0,p} \to 0 \ . \tag{5}$$

The linear map $\varphi$ is an epimorphism if and only if the complex is exact or, equivalently, the polynomials do not have a common root in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_N}$. Moreover, if $n_1, \ldots, n_N$ are fixed, the possible values of $(m_1, \ldots, m_N)$ which lead to determinantal complexes is a finite set [14, 16]. Each non-zero cohomology group in (4) is identified by a (dual) vector space of polynomials (cf. [29, Theorem 5.1]):

$$H^a_{n_i}(b) \cong \begin{cases} S_i(b) & , a = 0 \text{ and } b \geq 0 \\ S_i^*(-b - n_i - 1) & , a = n_i \text{ and } b < -n_i \\ 0 & , \text{otherwise.} \end{cases} \tag{6}$$

Here $S_i^*(d)$ denotes the dual space of $S_i(d)$, that is the space of linear functionals $\lambda : S_i(d) \to \mathbb{R}$. This space is isomorphic to (evaluations of) polynomials in formal partial derivatives, that is $S_1^*(d) \cong \mathbb{R}[\partial_{x_1}, \dots, \partial_{x_{n_1}}]_d$, see [36, 37] and references therein.

This identification allows us to choose bases and to express the maps $\varphi$ between the modules of (5) as a square matrix depending on the coefficients of $f_0, \dots, f_n$.

**Example 2.** *The resultant of three $(1,1)-$bilinear forms, that is when $n_1 = n_2 = 1$, corresponds to the determinantal complex $K_\bullet(2,1)$, thus $\mathbf{m} = (2,1)$. Using (6), the complex (5) becomes $K_{1,1} \xrightarrow{\varphi} K_{0,0}$. Therefore, there exists a map $\varphi : S(1,0)^3 \to S(2,1)$ whose determinant equals the resultant. This resultant matrix is depicted in [14, Sect. 7.1].*

We can obtain all classically known resultant formulas as the determinant of a map $\varphi$ in (5), by choosing a particular integer vector $\mathbf{m} \in \mathbb{Z}^2$. Moreover, the existence of a determinantal complex implies a determinantal formula for the resultant.

### *2.2. Determinantal Sylvester and Koszul formulas*

We identify determinantal formulas from generalized Sylvester complexes for multilinear systems. These formulas are valid if and only if the non-zero terms in (5) are

$$0 \to K_{1,p+1} \xrightarrow{\varphi} K_{0,p} \to 0 \tag{7}$$

for some $p \in \{0, \dots, n\}$ (cf. [55]). General such formulas for multilinear systems are identified in [50, 55, 14, 16] and for bilinear systems with two different supports in [3]. We specialize these results to multilinear systems in the following lemma.

The part *(i)* are classical Sylvester matrices and were discovered in [50]. The formulas in *(ii)* are new and their explicit degree vectors are a generalization of the Sylvester ones. All of these formulas depend on an ordering of the variable groups. Therefore in what follows we fix a permutation $\sigma \in \mathfrak{S}_N$, and we require that variables and degree vectors are sorted according to $\sigma$.

**Lemma 3.** *Let $\sigma \in \mathfrak{S}_N$ and let $n_1, \dots, n_N$ be the cardinalities of the variable groups $\mathbf{x}_{\sigma(1)}, \dots, \mathbf{x}_{\sigma(N)}$, respectively.*

*The following are* $\lceil (N+1)/2 \rceil$ *determinantal maps for the* $(n_1, \ldots, n_N)$*-multilinear forms* $f_0, \ldots, f_n$*, which are linear with respect to their coefficients. They are given by the degree vectors* $\mathbf{m}^{(q)} \in \mathbb{Z}^N$ *with*

$$m_k^{(q)} = 1 - 2\,\chi(k,q) + \sum_{j=1}^{k-1} n_j \quad , \quad 1 \le k \le N, \quad 0 \le q \le \left\lceil \frac{N-1}{2} \right\rceil$$

*where* $\chi(k,q) = \begin{cases} 1 & \text{if } k \le q \\ 0 & \text{otherwise} \end{cases}$ *. In particular, we obtain*

*(i) the Sylvester map*

$$\phi : S(\mathbf{m}^{(0)} - \mathbf{1})^{n+1} \to S(\mathbf{m}^{(0)})\,, \tag{8}$$

$$(g_0,\, g_1,\, \ldots,\, g_n) \;\mapsto\; g_0 f_0 + \cdots + g_n f_n \tag{9}$$

*with* $m_k^{(0)} = 1 + n_1 + \cdots + n_{k-1}$ *,* $1 \le k \le N$

*(ii) for any* $q$ *such that* $1 \le q \le \lceil \frac{N-1}{2} \rceil$ *the Koszul maps*

$$\psi_q : [S^*(z_1+1, \ldots, z_q+1) \otimes S(z_{q+1}, \ldots, z_N)]^r \;\to\; [S^*(z_1, \ldots, z_q) \otimes S(z_{q+1}+1, \ldots, z_N+1)]^s$$

*with* $z_k = \begin{cases} \sum_{j=k+1}^{q} n_j & \text{if } k \le q \\ \sum_{j=q+1}^{k-1} n_j & \text{if } k > q \end{cases}$ *and* $r = \binom{n+1}{n_{q+1}+\cdots+n_N}$, $s = \binom{n+1}{n_1+\cdots+n_q}$,

$$(\lambda_1, \ldots, \lambda_r) \;\mapsto\; \left( \sum_{k=0, k\notin J_1}^{n} (-1)^{\delta}\, \lambda_\mu f_k, \ldots, \sum_{k=0, k\notin J_s}^{n} (-1)^{\delta}\, \lambda_\mu f_k \right) \tag{10}$$

*and* $J_j \subset \{0, \ldots, n\}, |J_j| = n_1 + \cdots + n_q$ *denotes the combination with lexicographic index* $j$*,* $\mu = \mu(j,k)$ *is the lexicographic index of the* $|J_j|+1$ *combination* $J_j \cup \{k\}$ *of* $\{0, \ldots, n\}$*,* $\delta = \delta(j,k) = |\{t \in J_j : t < k\}|$ *and* $\lambda_1, \ldots, \lambda_r \in S^*(z_1+1, \ldots, z_q+1) \otimes S(z_{q+1}, \ldots, z_N)$*.*

*Proof:* The Sylvester formula (i) is depicted in [50]. The formula corresponds to a classical Sylvester map, expressing multiplication by $f_k$'s, and we can see that it has dimension $\dim \mathrm{Dom}\phi = \dim \mathrm{Im}\phi = (n+1)D$. The Koszul formulas (ii) correspond to determinantal complexes (5) with non-zero part

$$0 \to \left[ \bigotimes_{k=1}^{N} H_{n_k}^{a_k}\left( m_k^{(q)} - n_1 - \cdots - n_q - 1 \right) \right]^r \to \left[ \bigotimes_{k=1}^{N} H_{n_k}^{a_k}\left( m_k^{(q)} - n_1 - \cdots - n_q \right) \right]^s \to 0\,.$$

where $a_k = \chi(k,q)n_k$. We will show that each of them provides a square matrix expressing the resultant of the system. Indeed, using the dimension implied by (6), we can check that

$$\dim \mathrm{Dom}\psi_q = \prod_{k=1}^{q} \binom{\sum_{j=k}^{q} n_j}{n_k} \cdot \prod_{k=q+1}^{N} \binom{\sum_{j=q+1}^{k} n_j}{n_k} \cdot r = (n+1)D$$

and similarly for $\dim \mathrm{Im}\psi_q$. By the surjectivity of the complex (cf. [53],) and as a consequence of [54, Theorem 9.1.2] we deduce that the determinant of $\psi_q$ is the resultant $R(f_0, \ldots, f_n)$. □

Observe that $m_1 = 1$ for the Sylvester map and $m_1 = -1$ for all the Koszul maps. Also note that for $q = 0$ it is $\chi(k,q) = 0$, and we obtain the Sylvester map. However, for $q \neq 0$ the maps (and matrices) are structurally different. They expresses a linear combination of applications of dual functionals to the $f_k$'s, see also [3, 16].

**Example 4.** *We illustrate these Sylvester- and Koszul-type resultant matrices. Consider the* $(1,2)-$*bilinear system of Example 1, augmented by an extra polynomial* $f_0$*, that is*

$$f_0 = a_0 + a_1y_2 + a_2y_1 + a_3x_1 + a_4x_1y_2 + a_5x_1y_1 \ . \tag{11}$$

*The number of common solutions of* $f_1, \ldots, f_3$ *is* $D(1,2) = 3$ *and the set of polynomials* $f_0, \ldots, f_3$ *form an overdetermined system of equations. The resultant of the system has degree* $\deg R(f_0, \ldots, f_3) = 12$*, which will also be the dimension of the matrices.*

*Map (i) of Lem. 3 is quite similar to the classical Macaulay map but it also takes into account the special bihomogeneous structure of the system. We have, for the identity permutation,*

$$\phi \ : \ S(0,1)^4 \to S(1,2) \ ,$$

*which yields the following (transposed) matrix*

$$\begin{array}{c} \\ f_0 \\ y_1f_0 \\ y_2f_0 \\ f_1 \\ y_1f_1 \\ y_2f_1 \\ f_2 \\ y_1f_2 \\ y_2f_2 \\ f_3 \\ y_1f_3 \\ y_2f_3 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 1 & y_2 & y_2^2 & y_1 & y_1y_2 & y_1^2 & x_1 & x_1y_2 & x_1y_2^2 & x_1y_1 & x_1y_1y_2 & x_1y_1^2 \end{array} \\ \left[\begin{array}{cccccccccccc} a_0 & a_1 & & a_2 & & & a_3 & a_4 & & a_5 & & \\ & & & a_0 & a_1 & a_2 & & & & a_3 & a_4 & a_5 \\ & a_0 & a_1 & & a_2 & & & a_3 & a_4 & & a_5 & \\ b_0 & b_1 & & b_2 & & & b_3 & b_4 & & b_5 & & \\ & & & b_0 & b_1 & b_2 & & & & b_3 & b_4 & b_5 \\ & b_0 & b_1 & & b_2 & & & b_3 & b_4 & 0 & b_5 & \\ c_0 & c_1 & & c_2 & & & c_3 & c_4 & & c_5 & & \\ & & & c_0 & c_1 & c_2 & & & & c_3 & c_4 & c_5 \\ & c_0 & c_1 & & c_2 & & & c_3 & c_4 & & c_5 & \\ d_0 & d_1 & & d_2 & & & d_3 & d_4 & & d_5 & & \\ & & & d_0 & d_1 & d_2 & & & & d_3 & d_4 & d_5 \\ & d_0 & d_1 & & d_2 & & & d_3 & d_4 & & d_5 & \end{array}\right] \end{array}.$$

*This matrix expresses the polynomial multiplication* (8) *with* $g_k \in S_2(1)$, $k = 0, \ldots, n$. *It has block structure* $(n+1) \times 1$ *and each quasi-Toeplitz block is of size* $D \times (n+1)D$. *From the permutation* $\sigma = (2\,1)$ *we obtain*

$$\phi \,:\, S(2,0)^4 \to S(3,1)$$

*which implies a matrix of the same block structure as above:*

$$\begin{array}{c} \\ f_0 \\ x_1f_0 \\ x_1^2f_0 \\ f_1 \\ x_1f_1 \\ x_1^2f_1 \\ f_2 \\ x_1f_2 \\ x_1^2f_2 \\ f_3 \\ x_1f_3 \\ x_1^2f_3 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 1 & y_2 & y_1 & x_1 & x_1y_2 & x_1y_1 & x_1^2 & x_1^2y_2 & x_1^2y_1 & x_1^3 & x_1^3y_2 & x_1^3y_1 \end{array} \\ \left[\begin{array}{cccccccccccc} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & & & & & & \\ & & & a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & & & \\ & & & & & & a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & & & & & & \\ & & & b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & & & \\ & & & & & & b_0 & b_1 & b_2 & b_3 & b_4 & b_5 \\ c_0 & c_1 & c_2 & c_3 & c_4 & c_5 & & & & & & \\ & & & c_0 & c_1 & c_2 & c_3 & c_4 & c_5 & & & \\ & & & & & & c_0 & c_1 & c_2 & c_3 & c_4 & c_5 \\ d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & & & & & & \\ & & & d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & & & \\ & & & & & & d_0 & d_1 & d_2 & d_3 & d_4 & d_5 \end{array}\right] \end{array}.$$

*The multiplication map which is expressed here is again as* (8) *but with* $g_k \in S_1(2)$, $k = 0, \ldots, n$.

*Finally, the Koszul map (ii)*

$$\psi_1 \,:\, S_1^*(1)^6 \otimes S_2(0) \to S_1^*(0) \otimes S_2(1)^4$$

*is given by the Koszul-type formula*

$$(\lambda_1, \ldots, \lambda_6) \mapsto \begin{pmatrix} -\lambda_1 f_1 - \lambda_2 f_2 - \lambda_3 f_3 \\ \lambda_1 f_0 - \lambda_4 f_2 - \lambda_5 f_3 \\ \lambda_2 f_0 + \lambda_4 f_3 - \lambda_6 f_3 \\ \lambda_3 f_0 + \lambda_5 f_1 + \lambda_6 f_2 \end{pmatrix}^\top ,$$

*with dual functionals $\lambda_i \in S_1^*(1)$ of the form $w_0\mathbf{1} + w_1\partial_{x_1}$. Note that $\mathbf{1}f_k$ picks the constant term of $f_k$, regarded as a form in $S_1(1)$. We arrive at the (transposed) matrix:*

$$\begin{array}{c|cccccccccccc}
 & 1 & y_2 & y_1 & 1 & y_2 & y_1 & 1 & y_2 & y_1 & 1 & y_2 & y_1 \\
\mathbf{1} & -b_0 & -b_1 & -b_2 & a_0 & a_1 & a_2 & & & & & & \\
\partial_{x_1} & -b_3 & -b_4 & -b_5 & a_3 & a_4 & a_5 & & & & & & \\
\mathbf{1} & -c_0 & -c_1 & -c_2 & & & & a_0 & a_1 & a_2 & & & \\
\partial_{x_1} & -c_3 & -c_4 & -c_5 & & & & a_3 & a_4 & a_5 & & & \\
\mathbf{1} & -d_0 & -d_1 & -d_2 & & & & & & & a_0 & a_1 & a_2 \\
\partial_{x_1} & -d_3 & -d_4 & -d_5 & & & & & & & a_3 & a_4 & a_5 \\
\mathbf{1} & & & & -c_0 & -c_1 & -c_2 & b_0 & b_1 & b_2 & & & \\
\partial_{x_1} & & & & -c_3 & -c_4 & -c_5 & b_3 & b_4 & b_5 & & & \\
\mathbf{1} & & & & -d_0 & -d_1 & -d_2 & & & & b_0 & b_1 & b_2 \\
\partial_{x_1} & & & & -d_3 & -d_4 & -d_5 & & & & b_3 & b_4 & b_5 \\
\mathbf{1} & & & & & & & -d_0 & -d_1 & -d_2 & c_0 & c_1 & c_2 \\
\partial_{x_1} & & & & & & & -d_3 & -d_4 & -d_5 & c_3 & c_4 & c_5
\end{array} .$$

*This resultant matrix has block structure $\binom{n+1}{n_2} \times \binom{n+1}{n_1}$ and each block has size $(n_1+1) \times (n_2+1)$.*

*All the matrices that we have presented have the same determinant, which is a homogeneous polynomial of degree 12 in $\mathbb{Z}[a_0, \ldots, d_5]$ and, additionally, it is homogeneous of degree 3 in each of the variable sets $\{a_0, \ldots, a_5\}$, $\{b_0, \ldots, b_5\}$, $\{c_0, \ldots, c_5\}$, and $\{d_0, \ldots, d_5\}$. This polynomial is $R(f_0, \ldots, f_3)$.*

## 3. Representation of the roots

This section elaborates on representing the roots of the system as a rational function of univariate polynomials, evaluated at the roots of another

univariate polynomial. We employ the primitive element representation (PER) [7] and the rational univariate representation (RUR) [1, 46, 5, 38]. In RUR the denominator is the same for all rational functions; it is the derivative of the square-free part of a factor of the resultant.

We start by introducing some notation. For a (multivariate) polynomial $f$ with integer coefficients, we denote by $\mathtt{H}(f)$ the height (largest absolute value) of its coefficients and by $\mathfrak{h}(f) = \lg(\mathtt{H}(f))$ the maximum bitsize of its coefficients. For a univariate polynomial $g$, $\mathtt{lc}(g)$ denotes its leading coefficient.

Consider a square system $f_1, \ldots, f_n$ of $n$ multilinear polynomials as in Eq. (1). We add the polynomial

$$f_0(\boldsymbol{x}) = f_0(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) = F_0(1, \boldsymbol{x}_1, \ldots, 1, \boldsymbol{x}_N),$$

where

$$F_0(x_{1,0}, \boldsymbol{x}_1, \ldots, x_{N,0}, \boldsymbol{x}_N) = \sum_{i_1=0}^{n_1} \cdots \sum_{i_N=0}^{n_N} u_{0,i_1,\ldots,i_N} x_{1,i_1} \cdots x_{N,i_N}$$

and the $u_{0,i_1,\ldots,i_N}$ are parameters. In this way we obtain the overconstrained system

$$(\Sigma_0): \ f_0(\boldsymbol{x}) = f_1(\boldsymbol{x}) = \cdots = f_n(\boldsymbol{x}) = 0. \tag{12}$$

We can compute the resultant of $(\Sigma_0)$ as the determinant of any of the matrices we have presented in the previous section.

We assume system Eq. (1) has only isolated solutions in the multi-projective space and the resultant is not identically zero. The resultant might be identically zero when there are positive dimensional components, even at infinity; we handle this case in Sec. 4.3 using symbolic perturbation.

### *3.1. The primitive element representation (PER)*

We follow Canny's approach [7] with the important difference that the linear polynomial $f_0$ is replaced by a *multilinear* one. The main reason for choosing a multilinear polynomial as $f_0$ is that if all the $n+1$ polynomials have the same support, then the determinant of the resultant matrices we have presented in the previous section gives exactly the resultant. A different choice for $f_0$ might yield a determinant equal to a multiple of the resultant. In this case we would need to treat this extraneous factor. Nevertheless, this extra factor is generically non-zero.

The resultant $R(f_0)$ of the system $(\Sigma_0)$ is a polynomial; due to Poisson formula it is a product of factors of the form

$$\sum_{i_1=0}^{n_1} \cdots \sum_{i_N=0}^{n_N} u_{0,i_1,\ldots,i_N}\, \alpha_{k,1,i_1} \cdots \alpha_{k,N,i_N} \quad ,$$

where $(\alpha_{k,1,0}, \ldots, \alpha_{k,1,n_1}, \ldots, \alpha_{k,N,0}, \ldots, \alpha_{k,N,n_N})$ is a root of the system. We specialize $u_{0,0,\ldots,0}$ to $-z$, where $z$ is a new variable and we choose $\prod_{j=1}^{N}(n_j+1)-1$ other constants, say $c_{0,i_1,\ldots,i_N}$, for the $u_{0,i_1,\ldots,i_N}$; we denote all of them as $\boldsymbol{c}$. Let this specialized resultant be $r(z) = R(-z, \boldsymbol{c})$. With this substitution the factors of $R$ corresponding to roots at infinity become constants. The other factors of $R$ are of the form

$$z\,\alpha_{k,1,0} \cdots \alpha_{k,N,0} - \sum_{\substack{i_1=0,\ldots,i_N=1 \\ (i_1,\ldots,i_N)\neq(0,\ldots,0)}}^{n_1,\ldots,n_N} c_{0,i_1,\ldots,i_N} \alpha_{k,1,i_1} \cdots \alpha_{k,N,i_N} \quad .$$

We may assume, without loss of generality, that $\alpha_{k,1,0} = \cdots \alpha_{k,N,0} = 1$, as the roots are not at infinity. Under these assumptions, we denote the roots of $r$ as $\zeta_m$; then

$$\zeta_m = \sum_{\substack{i_1=0,\ldots,i_N=1 \\ (i_1,\ldots,i_N)\neq(0,\ldots,0)}}^{n_1,\ldots,n_N} c_{0,i_1,\ldots,i_N} \alpha_{k,1,i_1} \cdots \alpha_{k,N,i_N} \quad , \tag{13}$$

where $m$ runs over (the indices of) all the roots of the system. By abuse of notation, we also denote by $r$ the square-free part of $r$. With this notation

$$r(z) = \mathtt{lc}(r) \prod\nolimits_m (z - \zeta_m), \tag{14}$$

where $m$ runs over (the indices of) all the distinct roots of $r$. The derivative of $r$ with respect to $z$ is

$$r'(z) = \mathtt{lc}(r) \sum\nolimits_m \prod\nolimits_{\nu \neq m} (z - \zeta_\nu). \tag{15}$$

To obtain the PER of the coordinates of the solutions of the multilinear system we consider the polynomials $\hat{a}^+_{k,i}(z)$ and $\hat{a}^-_{k,i}(z)$ where

$$\hat{a}^{\pm}_{k,i}(z) = R(-z, \boldsymbol{c}, c_{0,0,\ldots,i,\ldots,0} \pm 1), \tag{16}$$

where the $i$ in the index of $c$ appears in the $k$-th position, for $1 \leq k \leq N$ and $1 \leq i \leq n_k$. The semantics of the notation are as follows: we consider $f_0$ as before, but to the coefficient of the monomial $x_{k,i}$, which is $c_{0,0,\ldots,i,\ldots,0}$, we add 1, resp. subtract 1, to obtain $\hat{a}^+_{k,i}$, resp. $\hat{a}^-_{k,i}$, as a resultant of the corresponding system.

Consequently, we have a pair of polynomials for each coordinate $x_{k,i}$, $1 \leq k \leq N$ and $1 \leq i \leq n_k$.

Let $a^\pm_{k,i}$ be the square-free parts of the polynomials $\hat{a}^\pm_{k,i}$. The roots of the polynomial $a^\pm_{k,i}(z)$ are $\zeta_m \pm \alpha_{m,k,i}$. Hence, we have the factorization

$$a^\pm_{k,i}(z) = \mathtt{lc}(a^\pm_k) \prod_m (z - \zeta_m \mp \alpha_{m,k,i}).$$

Consider the polynomial $a^+_{k,i}(2\theta - z)$. It holds

$$a^+_{k,i}(2\theta - z) = \mathtt{lc}(a^+_k) \prod_\nu (2\theta - z + \zeta_\nu + \alpha_{\nu,k,i}).$$

The resultant w.r.t. $z$ of $a^-_{k,i}(z)$ and $a^+_{k,i}(2\theta - z)$, where $\theta$ is a new parameter, is

$$\begin{aligned} \mathtt{res}(a^-_{k,i}(z), a^+_{k,i}(2\theta - z)) &= \prod_m a^+_{k,i}(2\theta - \zeta_m + \alpha_{m,k,i}) \\ &= \mathtt{c} \prod_m \prod_\nu (\zeta_m - \alpha_{m,k,i} - 2\theta + \zeta_\nu + \alpha_{\nu,k,i}), \end{aligned} \tag{17}$$

for some constant $\mathtt{c}$ that is the leading coefficient of the polynomial.

If $\theta = \zeta_l$, then the polynomials $a^-_{k,i}(z)$ and $a^+_{k,i}(2\theta - z)$ have a common root if and only if

$$\zeta_m - \alpha_{m,k,i} - 2\zeta_l + \zeta_\nu + \alpha_{\nu,k,i} = 0, \tag{18}$$

for some indices $m$ and $\nu$, and in this case the resultant is 0. Eq. (18) holds for sure if $m = l = \nu$. However, there might be "bad" choices of the constants $c_{0,i_1,\ldots,i_N}$ that result in spurious roots for different tuples of roots of the system. We will characterize these "bad" values in Sec. 3.3. We call a polynomial $f_0$ that avoids these bad values *separating polynomial*.

Assuming there are no spurious roots, we consider $a^-_{k,i}(z)$ and $a^+_{k,i}(2\theta - z)$ as a bivariate polynomial system, in unknowns $\theta$ and $z$. We can represent

the solutions of this system using univariate polynomials in $\theta$, say $s_{k,i,0}$ and $s_{k,i,1}$, of degree $\mathcal{O}(D^2)$ [40, Thm. 1], as $s_{k,i,0}(\theta) = 0$ and $z = s_{k,i,1}(\theta)/s'_{k,i,0}(\theta)$. We notice from Eq. (17) that $2\theta$, and so the roots of $s_{k,i,0}(\theta)$, encode all the possible sums of the roots of $a^-_{k,i}(z)$ and $a^+_{k,i}(z)$. Hence, $r$ divides $s_{k,i,0}$, since when $m = \nu$ in (17) it holds $\theta = \zeta_m$. The latter are exactly the roots of $r$, see (13). Therefore, for $\theta$ such that $r(\theta) = 0$, the polynomials $a^-_{k,i}(z)$ and $a^+_{k,i}(2\theta - z)$ have a common root, say $\zeta_\ell - \alpha_{\ell,k,i}$, which also has the representation $s_{k,i,1}(\theta)/s'_{k,i,0}(\theta)$. Thus, the PER for the $k$-th $x$-coordinate is

$$r_{k,i}(z) = z - \frac{s_{k,i,1}(z)}{s'_{k,i,0}(z)} = \frac{z\, s'_{k,i,0}(z) - s_{k,i,1}(z)}{s'_{k,i,0}(z)} = \frac{\widetilde{s}_{k,i,1}(z)}{s'_{k,i,0}(z)}, \tag{19}$$

where $z$ runs over all the roots of $r$.

We also notice that the polynomials $r$ and $s'_{k,i,0}$ are relative prime (because $r$ is a divisor of $s_{k,i,0}$, which is square-free) and so we can compute the inverse of $s'_{k,i,0}$ modulo $r$. Therefore, we can express $r_{k,i}(z)$ as a

$$r_{k,i}(z) = p_{k,i}(z) \mod r(z), \tag{20}$$

for a polynomial $p_{k,i} = \widetilde{s}_{k,i,1}\,(s'_{k,i,0})^{-1}$.

**Example 5.** *Consider the $(1,2)-$bilinear system of Example 1, augmented by the extra polynomial $f_0$ in Example 4 with symbolic coefficients $a_i$. The resultant factors as*

$$\begin{aligned} &R(f_0,\ldots,f_3) = -9(a_4 - a_5)\\ &(a_0 + (2-\sqrt{3})a_1 - a_2 + \frac{\sqrt{3}}{3}a_3 - \frac{2\sqrt{3}+3}{3}a_4 + \frac{\sqrt{3}}{3}a_5)\\ &(a_0 + (\sqrt{3}+2)a_1 - a_2 - \frac{\sqrt{3}}{3}a_3 + \frac{2\sqrt{3}-3}{3}a_4 - \frac{\sqrt{3}}{3}a_5). \end{aligned}$$

*The three linear factors correspond to the three roots of the system. Note that for the root at infinity the constant term of the linear factor vanishes.*

*If we consider the substitution $a_0 = -z, a_1 = 1, a_2 = 1, a_3 = -1, a_4 = -1, a_5 = 1$, then $r(z) = -222 - 72\,z + 18\,z^2$. The solutions of $r$ are $2 \pm \frac{7}{3}\sqrt{3}$.*

*We also obtain the polynomials $\hat{b}^+_2(z) = -312 - 144\,z + 18\,z^2$ and $\hat{b}^-_2(z) = -96 + 18\,z^2$. Both are square-free and so $b^\pm_2 = \hat{b}^\pm_2$.*

*If we compute subresultant sequence of $b^-_2(z)$ and $b^+_2(2\theta - z)$, then the linear polynomial is $(46656 - 23328\,\theta)\,z + (-69984 - 93312\,\theta + 23328\,\theta^2)$.*

*Therefore the representation of the $y_2$ coordinates of the solutions is $r_2(z) = z + \frac{-69984-93312\,z+23328\,z^2}{46656-23328\,z}$. To obtain all the values of the $y_2$-coordinate we should evaluate $r_2$ at the roots of $r$. It holds that $r_2(2 \pm \frac{7}{3}\sqrt{3}) = 2 \pm \sqrt{3}$.*

*3.2. The rational univariate representation (RUR)*

To compute the RUR we slightly modify the approach in [1], see also [46, 11, 38], to fit our needs. Consider

$$\hat{f}_{0,k,i}(\mathbf{x}) = f_0(\mathbf{x}) + \mu\, x_{k,i} \ ,$$

where $\mu$ is a new variable, for $1 \leq k \leq N$, $1 \leq i \leq n_k$, and $f_0$ is, as in PER, a multilinear separating polynomial, that forces the roots at infinity to be constants that multiply the resultant. An explicit construction of $f_0$ follows the same steps as in Sec. 3.3. We replace $f_0$ by $\hat{f}_{0,k,i}$ in $(\Sigma_0)$ to obtain a new system $(\Sigma_{0,k,i})$. We substitute $u_{0,0,\ldots,0} = -z$ in $f_0$. Let $g_{k,i} \in (\mathbb{Z}[\mu])[z]$ be (the square-free part of) the resultant of $(\Sigma_{0,k,i})$ after we eliminate $\boldsymbol{x}$. It holds $\deg(g_{k,i}) = \mathcal{O}(D)$, and

$$g_{k,i}(z) = \mathtt{lc}(g_{k,i}) \prod\nolimits_m (z - \zeta_m - \mu\, \alpha_{m,k,i}).$$

The derivative with respect to $\mu$ is

$$\tfrac{\partial}{\partial \mu} g_{k,i} = -\mathtt{lc}(g_{k,i}) \sum\nolimits_m \alpha_{m,k,i} \prod\nolimits_{\nu \neq m} (z - \zeta_\nu - \mu\, \alpha_{\nu,k,i}),$$

where $m$ runs over (the indices of) all the distinct roots the system. If we set $\mu = 0$ to the previous expression, we get

$$-h_{k,i}(z) = \mathtt{lc}(g_{k,i}) \sum\nolimits_m \alpha_{m,k,i} \prod\nolimits_{\nu \neq m} (z - \zeta_\nu).$$

If we combine it with Eq. (15), then we obtain the representation

$$x_{k,i} = -\frac{\mathtt{lc}(r)\, h_{k,i}(z)}{\mathtt{lc}(g_{k,i})\, r'(z)}, \tag{21}$$

for the $i$-th coordinate of the $k$-th block of the roots of the system. In total, we have to perform this procedure $n = n_1 + \cdots + n_N$ times to obtain a similar representation for all the coordinates.

**Lemma 6.** *Assume system* $(\Sigma_0)$ *in Eq.* (12) *has only isolated roots in the corresponding multiprojective space and that the input polynomials have integer coefficients. Let the resultant of the* $(n_1, \ldots, n_N)$*-multilinear system be a univariate polynomial in* $z$ *of degree* $D$ *and bitsize* $\mathcal{O}(L)$*. The representation of the roots using Eq. (19) consists of polynomials of degree* $\mathcal{O}(D^2)$ *and bitsize* $\widetilde{\mathcal{O}}(D^2 + DL)$*. The representation of the roots using Eq. (21) consists of polynomials of degree* $\mathcal{O}(D)$ *and bitsize* $\widetilde{\mathcal{O}}(D + L)$*.*

*We can compute the representation of the roots, using the results of [40], with a Monte Carlo algorithm with probability of success* $1-1/2^\varrho$*, for a given* $\varrho \geq 1$*, and complexity* $\widetilde{\mathcal{O}}_B(D^{4+\epsilon} + D^{3+\epsilon}(L + \varrho) + D^{2+\epsilon}\varrho^2)$ *for any* $\epsilon > 0$*, assuming an oracle that returns a random prime less than* $(2^\varrho DL)^{\mathcal{O}(1)}$*.*

*Proof:* The resultant is a univariate polynomial in $z$ of degree $\mathcal{O}(D)$ and bitsize $\mathcal{O}(L)$. Let $r$ be its square-free part, then $\deg(r) = \mathcal{O}(D)$ and $\mathfrak{h}(r) = \mathcal{O}(D+L)$; the additional term $D$ in the bitsize is due to Mignotte's bound for the divisors of polynomials [56, Lecture IV, Theorem 17]. The same bounds holds for $a^{\pm}_{k,i}$, since they are also computed as determinants of resultant matrices.

With these bounds the complexity of PER is a direct consequence of [40, Thm. 1] that computes a rational univariate representation for the roots of a bivariate polynomial system by a Monte Carlo algorithm in the stated complexity bound. We compute a representation of the common root of $a^-_{k,i}(z)$ and $a^+_{k,i}(2\theta - z)$ as the rational function using [40] and then by exploiting Eq. (19) we obtain a the PER representation, $r_{k,i}(z)$ of the roots of the system. The maximum bitsize of the coefficients of all the involved polynomials is $\widetilde{\mathcal{O}}(D^2 + DL)$. Then, we compute $p_{k,i} = \widetilde{s}_{k,i,1}\,(s'_{k,i,0})^{-1}$, Eq. (20), by performing $\mathcal{O}(D^2)$ operations.

For the RUR, Eq. (21), we need to compute $g_{k,i}$, its square-free part, its derivative w.r.t. $\mu$, and finally $h_{k,i}$. As $g_{k,i}$ is also a determinant of a resultant matrix, it holds $\mathfrak{h}(g_{k,i}) = \widetilde{\mathcal{O}}(L)$, and so $\mathfrak{h}(h_{k,i}) = \widetilde{\mathcal{O}}(D + L)$. Moreover, $\deg(h_{k,i}) = \mathcal{O}(D)$. To compute $h_{k,i}$ we notice that $\mathtt{lc}(r)h_{k,i} = \mathtt{lc}(g_k)\, r'\, p_{k,i} \mod r$; an operation that costs $\widetilde{\mathcal{O}}(D)$.

Therefore, the cost of the procedure is dominated by the cost of computing the rational univariate representation of bivariate polynomial system of $a^-_{k,i}(z)$ and $a^+_{k,i}(2\theta - z)$. As we have to perform this task for all the coordinates the overall complexity of the Monte Carlo algorithm is $\widetilde{\mathcal{O}}_B(n(D^{4+\epsilon} + D^{3+\epsilon}(L + \varrho) + D^{2+\epsilon}\varrho^2))$ for any $\epsilon > 0$ and probability of success $1 - 1/2^\varrho$, for

a given $\varrho \geq 1$. In addition, this approach [40] assumes oracle that returns a random prime less than $(2^{\varrho}DL)^{\mathcal{O}(1)}$. $\square$

### *3.3. "Bad" values for $c_{0,i_1,\ldots,i_N}$ and the height of $f_0$*

Following Sec. 3, we choose a multilinear $f_0$ to augment $(\Sigma)$ and to obtain a system $(\Sigma_0)$, Eq. (12). The polynomial is of the form

$$f_0(\mathbf{x}) = -z + \sum_{\substack{i_1=0,\ldots,i_N=1 \\ (i_1,\ldots,i_N)\neq(0,\ldots,0)}}^{n_1,\ldots,n_N} c_{0,i_1,\ldots,i_N} x_{1,i_1} \cdots x_{N,i_N},$$

where $c_{0,i_1,\ldots,i_N} \in \mathbb{Z}$ are constants to be specified in the sequel and $z$ is a new parameter. As we have already mentioned there are values of $c_{0,i_1,\ldots,i_N}$ that force our algorithm to fail. The goal of this section is to identify these "bad" values. In addition we estimate the height of the coefficients of $f_0$.

Assume that $D$ bounds the number of the affine as well as the projective isolated roots of the system, see Eq. (2). First, we replace each $c_{0,i_1,\ldots,i_N}$ with a power of $t$, where $t$ is a new indeterminate. Then, the resultant of the system is a polynomial in $t$.

We argue that this polynomial is not identically zero. If this was the case, then it would mean that we can augment any system of multilinear polynomials $f_1, \ldots, f_N$ with only isolated roots in the multiprojective space, with the polynomial $f_0$ that involves the parameter $t$, and the resultant of the overdetermined system would be identically zero. Notice that our matrix construction of Sec. 2.2 gives exactly the resultant. In this case, $f_0$ vanishes at the zeros of all these polynomial systems, which is a contradiction.

Consider a solution $(\alpha_{k,1,0}, \ldots, \alpha_{k,1,n_1}, \ldots, \alpha_{k,N,0}, \ldots, \alpha_{k,1,n_N})$ of the system. Evaluating $f_0$ at this solution results to a polynomial in $t$, that is $f_{0,k} =$

$$\sum_{i_1=0}^{n_1} \cdots \sum_{i_N=0}^{n_N} t^{i_N+(n_N+1)i_{N-1}+(n_N+1)(n_{N-1}+1)i_{N-2}+\cdots+(n_N+1)\cdots(n_2+1)i_1} \alpha_{k,i_1} \cdots \alpha_{k,i_N} \tag{22}$$

which has degree $\sum_{i=1}^{N} n_i \prod_{k=i+1}^{N}(n_k+1)$. With this substitution, to avoid the "bad" values it suffices to choose suitable constant(s) for $t$.

Hence, our goal now is to compute the values of $t$ that cause spurious roots to appear. The first class of "bad" choices for $t$, and hence for the constants $c_{0,i_1,\ldots,i_N}$, are due to (isolated) roots of the system at projective

infinity. These roots might force $f_{0,k}$, and hence $r(z)$, to vanish identically. Recall that by our choice of $f_0$, the roots at infinity evaluate to constants and thus they multiply the resultant. Therefore, if these constants are zero, then they make the determinant of the resultant to vanish identically, even in the case where the resultant of the system is not zero.

For a root(s) at infinity we may have, for some $k$, $\{\alpha_{k,j,0} = 0\}_{j\in J}$, where $J$ runs over all the subsets of $\{1, \ldots, N\}$.

It suffices to consider the cases $\alpha_{k,1,0} = 0$, or $\alpha_{k,2,0} = 0$, $\ldots$, or $\alpha_{k,N,0} = 0$ as the rest of the cases are contained in one of these $N$ cases.

For each of these $N$ cases, $f_{0,k}$ is a polynomial in $t$ of degree

$$\sum_{i=1}^{N} n_i \prod_{k=1}^{N} (n_k + 1) \leq 2^N N \prod_{i=1}^{N} n_i.$$

The product of these $N$ polynomials is a polynomial of degree $\leq 2^N N^2 \prod_{i=1}^{N} n_i$, in $t$. As there are at most $D$ isolated roots at infinity, then each of them gives rise to a polynomial in $t$, as in Eq. (22), and the product of all these polynomials is a polynomial of degree at most $D\, 2^N N^2 \prod_{i=1}^{N} n_i$. In a similar way, we obtain such a polynomial when we consider the computation of the polynomials $a_{k,i}^{\pm}$. There are $2n$ such polynomials, respectively, each of degree at most $D\, 2^N N^2 \prod_{i=1}^{N} n_i$ in $t$. Now we consider the product of all these polynomials. We obtain a polynomial in $t$ of degree $\leq nD\ 2^{N+1} N \prod_{i=1}^{N} n_i$. The first class of "bad" values of $t$, hence of the constants $c_{0,i_1,\ldots,i_N}$, are the roots of this polynomial. Any value for $t$ which does not nullify this polynomial ensures that $r(z)$ and $a_{k,i}^{\pm}$ do not vanish identically, even in the presence of isolated roots of $(\Sigma)$ at infinity.

The second class of "bad" values for $t$ are those that force Eq. (18) to vanish for distinct indices $m$, $l$, and $\nu$. After substituting each $c_{0,i_1,\ldots,i_N}$ by a suitable power of $t$, Eq. (18) becomes a polynomial in $t$ of degree less or equal to $2N \prod_{i=1}^{N} n_i$. We have one such polynomial for each triple of roots $\zeta_m$, $\zeta_l$, and $\zeta_\nu$, see Eq. (13), and for all possible $k$. Therefore, there are $\binom{D}{3} n$ polynomials that correspond to "bad" values for the coordinates. The product of all of them results a polynomial of degree at most $2nD^3 N \prod_{i=1}^{N} n_i$.

Finally, we consider the product of the two polynomials that correspond to the two classes of "bad" values for the constants. This is a polynomial in

$t$ of degree at most

$$2nD^3N\prod_{i=1}^{N} n_i + 4nDN^2\prod_{i=1}^{N} n_i \leq 6nD^3N^2\prod_{i=1}^{N} n_i.$$

Hence, if we consider the integers in the interval $I = [0, ..6nD^3N^2\prod_{i=1}^{N} n_i]$ there is at least one, say $t_0 \in I$, that it is not a root of this polynomial. This integer implies a safe choice of the values $c_{0,i_1,\ldots,i_N}$. Obviously, $|t_0| = \mathtt{H}(t_0) \leq 6nD^3N^2\prod_{i=1}^{N} n_i$.

Substituting $t = t_0$ in $f_{0,k}$, we obtain $c_{0,i_1,\ldots,i_N}$ and $f_0$ such that

$$\mathtt{H}(f_0) \leq \left(6nD^3N^2\prod_{i=1}^{N} n_i\right)^{2N\prod_{i=1}^{N} n_i}$$

and so $\mathfrak{h}(f_0) = \mathcal{O}(N\prod_{i=1}^{N} n_i \lg(nDN\prod_{i=1}^{N} n_i))$.

**Lemma 7.** *In the worst case* $\mathfrak{h}(f_0) = \mathcal{O}(N\prod_{i=1}^{N} n_i \lg(nDN\prod_{i=1}^{N} n_i))$. *We also have the simpler bound* $\mathfrak{h}(f_0) = \mathcal{O}(n^{N+2}\lg(nD))$, *which is less accurate.*

The previous analysis and the derived bound allows us to introduce a probabilistic version for computing $f_0$. Using the Schwartz-Zippel lemma, if we choose $t_0$ from an interval that contains $6nD^3N^2\prod_{i=1}^{N} n_i$ integers, for a constant $c \in \mathbb{N}$, then the probability to obtain a "bad" $f_0$ is $1/c$. By repeated applications we can amplify this probability.

*3.4. Separation and representation bounds*

In this section we present bounds on the number of roots that we need to represent the solutions of multilinear polynomial systems.

Separation bounds are bounds on the minimum distance between two isolated roots of a polynomial system. We use `DMM` bound [19, 18], which is an output sensitive aggregate version to estimate the separation bound of a system of multilinear polynomials.

Assume that $f_i \in \mathbb{Z}[\boldsymbol{x}]$ and $\mathfrak{h}(f_i) \leq \tau$. Then, following Lem. 7, $\mathfrak{h}(f_0) = \sigma = \mathcal{O}(N\prod_{i=1}^{N} n_i \lg(nDN\prod_{i=1}^{N} n_i))$, or $\mathfrak{h}(f_0) = \sigma = \mathcal{O}(n^{N+2}\lg(nD))$. We will use the second, less accurate bound, to simplify the presentation of the various bounds.

For example, for the bilinear case we can derive more accurate bounds for the separation of the solutions and the bit complexity of the overall solving algorithm [17]. We leave the derivation of the accurate bounds for the multilinear case to the reader.

By Lem. 3, we know exactly the form of the resultant of multilinear systems. In particular, it is homogeneous of degree $D$ in the coefficients of each $f_i$. It has the form

$$r(z) = \cdots + \varrho_i \, z^i \, \mathbf{c}_i^{D-i} \mathbf{a}_{1,i}^D \mathbf{a}_{2,i}^D \cdots \mathbf{a}_{n,i}^D + \ldots, \tag{23}$$

where $\varrho_i \in \mathbb{Z}$, $\mathbf{a}_{k,i}^D$ denotes a monomial in coefficients of $f_k$ with total degree $D$, and $\mathbf{c}_i^{D-i}$ denotes a monomial in the coefficients of $f_0$ of total degree $D-i$. The degree of $r$, with respect to $z$, is $D$ and corresponds to the number of solutions of the system. It is nonzero because we have assumed that the system has only isolated solutions, even at infinity. We bound $|\varrho_i| \leq (n+1)^{2(n+1)D}$ using the fact the Newton polytopes of $f_k$ are products of simplices. Following [18, 19] we get $\mathfrak{h}(r) = \mathcal{O}(nD \lg(nD) + D\sigma + nD\tau)$, that expands to

$$\mathfrak{h}(r) = \mathcal{O}(n^{N+2} D \lg(nD) + nD\tau). \tag{24}$$

The same bounds hold for $\hat{a}_k^{\pm}$, $\hat{b}_\ell^{\pm}$ and $r'$, and this $L$ that appears in Lem. 6. Therefore, for the representation of the roots, of Eq. (21) we have that $\deg(s_k) = \mathcal{O}(D)$ and

$$\mathfrak{h}(s_k) = \mathcal{O}(n^{N+2} D \lg(nD) + nD\tau). \tag{25}$$

Let $\Delta_i$ be the (local) separation bound of the $i$-th isolated root of $(\Sigma)$, that is the minimum distance of the $i$-th root to any other isolated root of the system. Then, using `DMM` [18, Cor. 10] we have

$$-\lg \prod\nolimits_i \Delta_i = \mathcal{O}(n^{N+2} D^2 \lg(nD) + nD^2\tau). \tag{26}$$

Moreover, for any non-zero coordinate of any root of the system it holds

$$2^{-\mathcal{O}(n^{N+2} D \lg(nD) + nD\tau)} \leq |\alpha_{k,i_1,\ldots,i_N}| \leq 2^{\mathcal{O}(n^{N+2} D \lg(nD) + nD\tau)} \ , \tag{27}$$

where $1 \leq i_k \leq n_k$ for all $k$ from 1 to $N$.

## 4. The complexity of solving

In this section we establish the bit complexity of the algorithm to compute isolating hyperboxes for all the roots of the system.

*4.1. Square systems of dimension 0*

First, we consider the complexity of computing the determinant of the first resultant matrix, say $M$, of Sec. 2.2, when it is a scalar matrix. The dimension of the matrix is $(n+1)D \times (n+1)D$. We use Wiedemann's algorithm, following the approach in [9]. The arithmetic complexity is dominated by the cost of performing $(n+1)D$ applications of matrix-vector products $Mb$, for a vector $b$.

Lem. 3 implies that $b^\top M$ corresponds to $(n+1)$ polynomial multiplications. Each involves two polynomials: one multilinear of degree 1 in each block of variables, this is $f_i$ and one of multi-degree $(0, n_1, n_1+n_2, \ldots, n_1 + \ldots n_{N-1})$, for each block of variables respectively. The latter polynomial has $\widetilde{\mathcal{O}}(D)$ terms. The output has $O(D)$ terms, and the cost to compute it is $\widetilde{\mathcal{O}}_B(D)$. Using Tellegen's principle [4] we obtain, at almost the same cost, an algorithm for $Mb$. Thus, the determinant computation costs $\widetilde{\mathcal{O}}(nD^2)$ arithmetic operations. Similar results hold for the other resultant matrices.

Computing $r$ requires the determinant of a (resultant) matrix depending on one parameter $z$; this is done using interpolation. Recall that $r$ has degree $\mathcal{O}(D)$ and $\mathfrak{h}(r) = \mathcal{O}(n^{N+2}D\lg(nD) + nD\tau)$.

We need to perform $\mathcal{O}(D)$ scalar determinant evaluations; each reduces to $D$ times $(n+1)$ polynomial multiplications. We assume the polynomials have the worst possible bitsize, that is $\mathfrak{h}(r) = \mathcal{O}(n^{N+2}D\lg(nD) + nD\tau)$. The cost of each multiplication is $\widetilde{\mathcal{O}}_B(n^{N+2}D^2\lg(nD) + nD^2\tau)$ using a probabilistic [2, Thm. 7.1] or a worst case [31, Cor. 21] algorithm. Thus, the total cost for computing the polynomial $r$ is $\widetilde{\mathcal{O}}_B(n^{N+3}D^4\lg(nD) + n^2D^4\tau)$.

To compute the representation of the roots of Eq. (21) we need to compute $a_k^{\pm}(z)$, $b_\ell^{\pm}$, and $s_k(z)$. The cost of computing $a_k^{\pm}(z)$ is the same as that for computing $r$. Since there are $n$ coordinates, the cost for computing all $a_k^{\pm}(z)$ and $b_k^{\pm}(z)$ is $\widetilde{\mathcal{O}}_B(n^{N+3}D^4\lg(nD) + n^2D^4\tau)$. Following Lem. 6, the degree of $s_k$'s is $\mathcal{O}(D)$ and their bitsize is $\mathcal{O}(n^{N+2}D\lg(nD) + nD\tau)$. The cost to construct them is $\widetilde{\mathcal{O}}_B(n^{N+3}D^{4+\epsilon}\lg(nD) + n^2D^{4+\epsilon}\tau + nD^{3+\epsilon}\varrho + nD^{2+\epsilon}\varrho^2)$, or $\widetilde{\mathcal{O}}_B(n^2D^{4+\epsilon}(n^{N+1}\lg(nD)+\tau) + nD^{2+\epsilon}\varrho(D+\varrho))$, using the Monte Carlo supported by Lem. 6.

Next, we isolate all the complex roots of $r$, with cost bounded by $\widetilde{\mathcal{O}}_B(D^2\mathfrak{h}(r)) = \widetilde{\mathcal{O}}_B(n^{N+3}D^4\lg(nD) + n^2D^4\tau)$ [43]. We obtain isolating boxes for the complex roots of $r$. Then, we refine the roots up to accuracy $2^{-\lambda}$ in $\widetilde{\mathcal{O}}_B(D^2\mathfrak{h}(r) + D\lambda)$ [44]. We perform all the computations with $\lambda$ bits of accuracy. We need to determine the value for $\lambda$ such that the evaluation of

the RUR, see Eq. (21), at the approximations of the roots of $r$ results to disjoint hyperboxes for the roots of the system.

After refinement, for every root $\gamma_i$ of $r$, we have an interval $[\gamma_i]$, such that its width is less than $2^{-\lambda}$; that is $\mathtt{wid}([\gamma_i]) \leq 2^{-\lambda}$. For each coordinate $k$, using interval or multiprecision floating point arithmetic, we evaluate Eq. (21), at $[\gamma_i]$. In this way we obtain intervals, $I_{k,i}$ for all the possible values of the $k$-th coordinates, where $I_{k,i} = -\frac{\mathtt{lc}(r)\, s_k([\gamma_i])}{\mathtt{lc}(g)\, r'([\gamma_i])}$. Using Horner's rule for the evaluation [30, Sec. 5.1] we have $\mathtt{wid}(s_k([\gamma_i])) \leq \boldsymbol{c}_{2D}\, \overline{s}_k(|\gamma_i|)$ . Under the mild assumption that the precision used for the computations exceeds $\lg(n)$, the constant $\boldsymbol{c}_{2D}$ is $\leq 5D\, 2^{-\lambda} = 2^{-\lambda+\mathcal{O}(\lg(D))}$. The polynomial $\overline{s}_k$ is $s_k$ but with its coefficients replaced by their absolute value. Thus,

$$\overline{s}_k(|\gamma_i|) \leq (D+1)\, \mathtt{H}(s_k) \max\{1, |\gamma_i|^D\} \ .$$

We bound $|\gamma_i|$ using Eq. (27). A similar bound holds for $r'([\gamma_i])$. Putting everything together we have

$$\mathtt{wid}(I_{k,i}) \leq 2^{-\lambda+\widetilde{\mathcal{O}}(n^{N+2}D^2 \lg(nD)+nD^2\tau)}.$$

For these intervals to be disjoint it suffices that $\mathtt{wid}(I_{k,i}) \leq 2^{-\lg \prod_i \Delta_i}$ holds. Hence, we choose a proper constant $\lambda$ such that

$$\mathtt{wid}(I_{k,i}) \leq 2^{-\lambda+\widetilde{\mathcal{O}}(n^{N+2}D^2 \lg(nD)+nD^2\tau)} \leq 2^{-\lg \prod_i \Delta_i},$$

which yields $\lambda > \widetilde{\mathcal{O}}(n^{N+2}D^2 \lg(nD) + nD^2\tau)$ for the precision. To actually obtain the isolating boxes for the roots we evaluate $s_k$ and $r'$ at the isolating boxes of the roots of $r$ using an approximate multipoint evaluation algorithm in $\widetilde{\mathcal{O}}_B(nD^3\tau + D\lambda)$ [45, Lemma 21]. The previous discussion leads to the following.

**Theorem 8.** *There is a Monte Carlo algorithm for isolating the roots of a 0-dimensional system of $(n_1, \ldots, n_N)$-multilinear polynomials with integer coefficients of maximum bitsize $\tau$ with probability of success $1 - 1/2^{\varrho}$, for a given $\varrho \geq 1$, and complexity $\widetilde{\mathcal{O}}_B(n^2D^{4+\epsilon}(n^{N+1}\lg(nD) + \tau) + nD^{2+\epsilon}\varrho(D + \varrho))$, for any $\epsilon > 0$, where $n = \sum_{i=1}^{N} n_i$, $D$ is the multilinear Bézout bound of Eq. (2), and assuming an oracle that returns a random prime less than $(2^{\varrho}n^{N+2}D^2\tau)^{\mathcal{O}(1)}$.*

*4.2. Overdetermined systems of dimension 0*

Let us assume the input consists of more than $n$ multilinear polynomials, say $p$, that is $f_1, \ldots, f_p$, and nevertheless has a finite number of (multi-projective) roots. Our first task is to make the system square, using the technique of [27]. We consider $n$ random linear combinations of the input polynomials, that is $h_k = \sum_{i=1}^{p} r_i f_i$, for $1 \leq k \leq n$, where $r_i$ are random integers. The bitsize of the polynomials $h_k$ of this new square system is asymptotically the same, up to logarithmic factors, with the bitsize of the polynomials $f_i$. We refer to [18] for details. Then, we solve the system using Thm. 8 and we obtain a representation for the roots of the new system.

The bounds of the representation and the complexity of computing it are the same as in the case of square system, that we have presented in the previous section. However, the procedure that we use to construct a square system might introduce additional isolated points. Thus, not all isolated roots of the new system correspond to roots of the original one. Equivalently, not all the roots of the resultant, $r(z)$, correspond to roots of the original system. To identify the roots of interest we proceed as follows. We substitute the RUR of the $x$ and $y$ coordinates in $f_k$, for $1 \leq k \leq p$. That is

$$f_k \left( \frac{-\mathtt{lc}(r)\, s_{1,1}}{\mathtt{lc}(s_{1,1})\, r'}, \ldots, \frac{-\mathtt{lc}(r)\, s_{1,n_1}}{\mathtt{lc}(s_{1,n_1})\, r'}, \ldots, \frac{-\mathtt{lc}(r)\, s_{N,1}}{\mathtt{lc}(s_{N,1})\, r'}, \ldots, \frac{-\mathtt{lc}(r)\, s_{N,n_N}}{\mathtt{lc}(s_{N,n_N})\, r'} \right) .$$

In this way we obtain a rational function in $z$, say $\frac{f_{k,0}(z)}{f_{k,1}(z)}$. Let $\sigma$ be the bitsize of the polynomials in RUR. We can compute this rational function in $\widetilde{\mathcal{O}}_B(n^{N+2}D\sigma)$ [45]. It holds that $\deg(f_{k,0}) = \deg(f_{k,1}) = \mathcal{O}(D)$ and $\mathfrak{h}(f_{k,0}) = \widetilde{\mathcal{O}}(n\sigma)$. To determine which are the roots of the new system that correspond to roots of the original system, it suffices to compute the $\gcd(r, f_{1,0}, \ldots, f_{p,0})$. This corresponds to, at most, $p$ computations of GCD's of two polynomials of degree $\mathcal{O}(D)$ and bitsize $\widetilde{\mathcal{O}}(n\sigma)$. The cost for each GCD computation is $\widetilde{\mathcal{O}}_B(nD^2\sigma)$ [52], and so the total cost is $\widetilde{\mathcal{O}}_B(pnD^2\sigma)$. If we substitute $\sigma = \widetilde{\mathcal{O}}(n^{N+2}D\lg(nD) + nD\tau)$, then we obtain the bound $\widetilde{\mathcal{O}}_B(p(n^{N+3}D^3\lg(D) + n^2D^3\tau))$. If we use a probabilistic algorithm for the GCD of $p+1$ polynomials using one GCD operation [52], then we can eliminate the factor $p$.

*4.3. Positive dimensional systems*

The resultant computation and thus the algorithm of Thm. 8 fails when the system is not zero-dimensional, including the case of excess components

at infinity. This section handles this situation by an infinitesimal symbolic perturbation of the given polynomials.

The resultant may vanish identically for random choices of the coefficients of $f_0$ (where $f_0$ as in Sect. 3.3), if there are infinitely many roots of the system $f_0 = f_1 = \cdots = f_n = 0$ (projective or affine). This case is not covered by the "bad" values' computation of Sect. 3.3. This is so because we perform this computation under the assumption that there are finitely many common solutions in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_N}$ for the input equations $f_1, \ldots, f_n$. Note that there are no extraneous factors coming from the determinantal matrix expressions of our resultant, therefore there is no other possibility of an identically zero determinant, because of the vanishing of this extra factor.

To compute the roots in these cases we consider the general approach of generalized characteristic polynomial (GCP) [8]. In particular, we apply a perturbation scheme for sparse systems [13]. We select a square $(n_1, \ldots, n_N)$-multilinear system $p_1, \ldots, p_n$ that leads to a non-zero resultant; this happens for any choice of polynomials $p_1, \ldots, p_n$ with finitely many roots in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_N}$, and a generic polynomial $f_0$. To define such a system, consider the resultant matrix of $f_0, p_1, \ldots, p_n$; it is possible to permute the rows and columns to minimize the number of different monomials from the polynomials $p_1, \ldots, p_n$ that appear in the diagonal. Of course, the diagonal shall contain at least one monomial for each of the equations $f_0, p_1, \ldots, p_n$. Then, we assign random coefficients for those monomials per equation $p_1, \ldots, p_n$ that appear on the (permuted) diagonal, while all other monomials are assigned the zero coefficient. The theory of toric GCP in [13] implies that $R(f_0, p_1, \ldots, p_n) \neq 0$ and, therefore, the matrices in Lem. 3 are non-singular.

Let us return to our original problem of a given polynomial system $f_1, \ldots, f_n$ having an infinite number of solutions. We introduce the perturbed system $\tilde{f}_k = f_k + \varepsilon p_k$, $k = 1, \ldots, n$, for a positive infinitesimal $\varepsilon$, augmented by $\tilde{f}_0 = f_0$. The polynomial $C(\varepsilon) := R(\tilde{f}_0, \ldots, \tilde{f}_n)$ has degree equal to $nD$ with respect to $\varepsilon$. We are interested in its trailing (nonzero) coefficient, because it is the most significant, since $\varepsilon \to 0^+$. By [13, Prop. 3.4], this trailing coefficient is equal to $R(f_0, p_1, \ldots, p_n)$. Therefore, if we regard $C(\varepsilon)$ as a univariate polynomial in $\varepsilon$, there exists a non-zero trailing coefficient, which is a polynomial in the coefficients of $f_0$. Moreover, it has the same property as the unperturbed resultant in the zero-dimensional case, namely it factors as a product of linear forms, and the coefficients of the linear forms provide us with one point per connected component of the solution set of $f_1, \ldots, f_n$. The complexity of computing the trailing non-vanishing coefficient is the

complexity of the zero-dimensional case multiplied by the degree, say $d$, with respect to $\varepsilon$, of this term, as one may perform the computations mod $\varepsilon^d$, following the method of [21]. A powerful method for handling single-parameter matrices is developed in [35].

**Example 9.** *Consider the* $(1,2)-$*bilinear, square system of the following equations*

$$\begin{aligned} f_1 &= 1 + 2\,y_2 + 2\,y_1 + x_1 + 2\,x_1y_2 + 2\,x_1y_1, \\ f_2 &= 1 + 2\,y_2 + y_1 + 2\,x_1 + 2\,x_1y_2 + 2\,x_1y_1, \\ f_3 &= 1 + 2\,y_2 + 2\,y_1 + x_1 + 2\,x_1y_2 + 2\,x_1y_1. \end{aligned}$$

*The system has infinitely many roots, which are* $(x_1; y_1, y_2) = (-1; -1, \rho)$, $(x_1; y_1, y_2) = (-\frac{1}{2} - \sigma; -\frac{1}{2} - \sigma, \sigma)$, *for any* $\rho, \sigma \in \mathbb{C}$, *in addition to the root* $(x_0, x_1; y_0, y_1, y_2) = (0, 1; 0, -1, 1)$ *at projective infinity. We perturb the system by* $\varepsilon$*; for this example, perturbing one linear term per equation is sufficient. Let* $\tilde{f}_1 = f_1 + \varepsilon x_1$ *and* $\tilde{f}_{1+j} = f_{1+j} + \varepsilon y_j$, $j = 1, 2$. *Adding the polynomial* $f_0$ *as in Example 4, we obtain the resultant* $R(f_0, \ldots, f_3) = q_0\varepsilon^6 + q_1\varepsilon^7 + q_2\varepsilon^8$, *where* $q_0, q_1, q_2$ *depend on the coefficients* $a_i$ *of* $f_0$. *The (factored) coefficient of the trailing term* $q_0\varepsilon^6$ *is:*

$$\begin{aligned} q_0(a_0, \ldots, a_5) = &-32(a_4 - a_5)(a_0 - a_1 - a_2 - a_3 + a_4 + a_5) \\ &(a_0 - \frac{1}{4}\,a_1 - \frac{1}{4}\,a_2 - \frac{1}{4}\,a_3 + \frac{1}{16}\,a_4 + \frac{1}{16}\,a_5), \end{aligned}$$

*which corresponds to the root at infinity and to one point on each component, namely* $\rho = -1$ *and* $\sigma = -\frac{1}{4}$.

## 5. Conclusions

Future work includes the study of mixed systems of equations, e.g., [3], or in particular multihomogeneous equations with scaled support [16]. We shall also examine alternatives for determinant computation achieving good bit complexity, such as [34], juxtaposed to our current approach. The latter achieves record complexity for integer determinants, but it does not exploit any quasi-Toeplitz structure present in the matrix.

**Acknowledgments**

The authors are grateful to the reviewers for their careful reading and their constructive comments. IZE participates in team Aromath, joint between Inria Sophia-Antipolis and NKUA. ET is partially supported by ANR JCJC GALOP (ANR-17-CE40-0009), the PGMO grant ALMA and the PHC GRAPE.

[1] M. E. Alonso, E. Becker, M.-F. Roy, and T. Wörmann. Algorithms in algebraic geometry and applications. chapter Zeros, Multiplicities, and Idempotents for Zero-dimensional Systems, pages 1–15. 1996.

[2] A. Arnold and D. S. Roche. Output-sensitive algorithms for sumset and sparse polynomial multiplication. In *Proc. ACM ISSAC*, pages 29–36, 2015.

[3] M. Bender, J.-C. Faugère, A. Mantzaflaris, and E. Tsigaridas. Bilinear systems with two supports: Koszul resultant matrices, eigenvalues, and eigenvectors. In *Proc. of Int'l Symposium on Symbolic and Algebraic Computation (ISSAC'18)*, pages 63–70. ACM, 2018.

[4] A. Bostan, G. Lecerf, and E. Schost. Tellegen's principle into practice. In *Proc. ISSAC*, pages 37–44, 2003.

[5] A. Bostan, B. Salvy, and É. Schost. Fast algorithms for zero-dimensional polynomial systems using duality. *Applicable Algebra in Engineering, Communication and Computing*, 14(4):239–272, 2003.

[6] Y. Bouzidi, S. Lazard, G. Moroz, M. Pouget, F. Rouillier, and M. Sagraloff. Improved algorithms for solving bivariate systems via Rational Univariate Representations. Tech. report, Inria, June 2015.

[7] J. Canny. Some algebraic and geometric computations in PSPACE. In *Proc. 20th STOC*, pages 460–467. ACM, 1988.

[8] J. Canny. Generalised characteristic polynomials. *JSC*, 9(3):241–250, 1990.

[9] J. F. Canny, E. Kaltofen, and L. Yagati. Solving systems of nonlinear polynomial equations faster. In *Proc. ACM ISSAC*, pages 121–128, 1989.

[10] D. Cox and E. Materov. Tate resolutions for Segre embeddings. *Algebra & Number Theory*, 2(5):523–550, 2008.

[11] X. Dahan and E. Schost. Sharp estimates for triangular sets. In *Proc. ACM ISSAC*, pages 103–110, 2004.

[12] C. D'Andrea and A. Dickenstein. Explicit formulas for the multivariate resultant. *J. Pure and Applied Algebra*, 164(1-2):59–86, 2001.

[13] C. D'Andrea and I. Z. Emiris. Computing sparse projection operators. In *Symbolic Computation: Solving Equations in Algebra, Geometry, and Engineering*, volume 286 of *Contemporary Mathematics*, pages 121–140. AMS, Providence, 2001.

[14] A. Dickenstein and I. Z. Emiris. Multihomogeneous resultant formulae by means of complexes. *JSC*, 36(3):317–342, 2003.

[15] D. Eisenbud, F.-O. Schreyer, and J. Weyman. Resultants and chow forms via exterior syzygies. *Journal of the American Mathematical Society*, 16(3):537–579, 2003.

[16] I. Z. Emiris and A. Mantzaflaris. Multihomogeneous resultant formulae for systems with scaled support. *JSC*, 47(7):820–842, 2012.

[17] I. Z. Emiris, A. Mantzaflaris, and E. Tsigaridas. On the bit complexity of solving bilinear polynomial systems. In *Proc. of Int'l Symposium on Symbolic and Algebraic Computation (ISSAC'16)*, pages 215–222, New York, NY, USA, 2016. ACM.

[18] I. Z. Emiris, B. Mourrain, and E. Tsigaridas. Separation bounds for polynomial systems. Technical report, INRIA, Dec. 2015.

[19] I. Z. Emiris, B. Mourrain, and E. P. Tsigaridas. The DMM bound: Multivariate (aggregate) separation bounds. In *Proc. ACM ISSAC*, pages 243–250, 2010.

[20] I. Z. Emiris and V. Y. Pan. Symbolic and Numeric Methods for Exploiting Structure in Constructing Resultant Matrices. *JSC*, 33(4):393–413, Apr. 2002.

[21] I. Z. Emiris and V. Y. Pan. Improved algorithms for computing determinants and resultants. *J. of Complexity*, 21(1):43–71, Feb. 2005.

[22] I. Z. Emiris and R. Vidunas. Root counts of semi-mixed systems, and an application to counting Nash equilibria. In *Proc. ACM ISSAC*, pages 154–161, Kobe, Japan, 2014. ACM Press.

[23] J.-C. Faugere, M. S. El Din, and P.-J. Spaenlehauer. Gröbner bases of bihomogeneous ideals generated by polynomials of bidegree (1, 1): Algorithms and complexity. *JSC*, 46(4):406–437, 2011.

[24] J.-C. Faugère, F. Levy-Dit-Vehel, and L. Perret. Cryptanalysis of minrank. In *Advances in Cryptology*, pages 280–296. Springer, 2008.

[25] J.-C. Faugère, M. Safey El Din, and P.-J. Spaenlehauer. Gröbner bases of bihomogeneous ideals generated by polynomials of bidegree (1,1): Algorithms and complexity. *JSC*, 46:406–437, 2011.

[26] I. Gelfand, M. Kapranov, and A. Zelevinsky. *Discriminants, Resultants and Multidimensional Determinants*. Birkhäuser, Boston, 1994.

[27] 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 *Proc. Int. Meeting on Commutative Algebra, Cortona*, 1993.

[28] M. Giusti, G. Lecerf, and B. Salvy. A Gröbner free alternative for polynomial system solving. *J. of Complexity*, 17(1):154–211, 2001.

[29] R. Hartshorne. *Algebraic Geometry*. Springer, New York, 1977.

[30] N. J. Higham. *Accuracy and stability of numerical algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, 2002.

[31] J. v. d. Hoeven and G. Lecerf. On the bit-complexity of sparse polynomial multiplication. *JSC*, 50:227–254, 2013.

[32] G. Jeronimo and J. Sabia. Computing multihomogeneous resultants using straight-line programs. *JSC*, 42(1–2):218–235, 2007.

[33] A. Joux. A new index calculus algorithm with complexity $L(1/4+o(1))$ in small characteristic. In *Selected Areas in Cryptography–SAC 2013*, pages 355–379. Springer, 2014.

[34] E. Kaltofen and G. Villard. On the complexity of computing determinants. *Computational Complexity*, 13(3-4):91–130, 2005.

[35] E. L. Kaltofen and M. Nehring. Supersparse black box rational function interpolation. In *Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation*, ISSAC '11, pages 177–186, New York, NY, USA, 2011. ACM.

[36] A. Mantzaflaris and B. Mourrain. Deflation and certified isolation of singular zeros of polynomial systems. In *Proc. ACM ISSAC*, pages 249–256, 2011.

[37] A. Mantzaflaris and B. Mourrain. Singular zeros of polynomial systems. In T. Dokken and G. Muntingh, editors, *Advances in Shapes, Geometry, and Algebra*, volume 10 of *Geometry and Computing*, pages 77–103. Springer, 2014.

[38] A. Mantzaflaris, E. Schost, and E. Tsigaridas. Sparse rational univariate representation. In *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '17, pages 301–308, New York, NY, USA, 2017. ACM.

[39] N. McCoy. On the resultant of a system of forms homogeneous in each of several sets of variables. *Trans. AMS*, 35(1):215–233, 1933.

[40] E. Mehrabi and É. Schost. A softly optimal monte carlo algorithm for solving bivariate polynomial systems over the integers. *J. of Complexity*, 34:78 – 128, 2016.

[41] T. Muir. The resultant of a set of lineo-linear equations. *Proc. Royal Soc. of South Africa*, 2(1):373–380, 1910.

[42] A. V. Ourivski and T. Johansson. New technique for decoding codes in the rank metric and its cryptography applications. *Problems of Information Transmission*, 38(3):237–246, 2002.

[43] V. Y. Pan. Univariate polynomials: Nearly optimal algorithms for numerical factorization and root-finding. *JSC*, 33(5):701 – 733, 2002.

[44] V. Y. Pan and E. Tsigaridas. Accelerated approximation of the complex roots and factors of a univariate polynomial. *Theoretical Computer Science*, 681:138–145, 2017.

[45] V. Y. Pan and E. Tsigaridas. Nearly optimal computations with structured matrices. *Theoretical Computer Science*, 681:117–137, 2017.

[46] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, 1999.

[47] M. Safey El Din and É. Schost. Polar varieties and computation of one point in each connected component of a smooth real algebraic set. In *Proc. ACM ISSAC*, pages 224–231, 2003.

[48] M. Safey El Din and É. Schost. Bit complexity for multi-homogeneous polynomial system solving – application to polynomial minimization. *Journal of Symbolic Computation*, 87:176 – 206, 2018.

[49] P.-J. Spaenlehauer. *Solving multi-homogeneous and determinantal systems: algorithms, complexity, applications.* Thesis, Université Pierre et Marie Curie (Univ. Paris 6), Oct. 2012.

[50] B. Sturmfels and A. Zelevinsky. Multigraded resultants of Sylvester type. *J. Algebra*, 163(1):115–127, 1994.

[51] J. Sylvester. On the degree and weight of the resultant of a multipartite system of equations. *Proc. Royal Soc. of London*, 12:674–676, 1862.

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

[53] J. Weyman. Calculating discriminants by higher direct images. *Trans. of AMS*, 343(1):367–389, 1994.

[54] J. Weyman. *Cohomology of Vector Bundles and Syzygies*. Cambridge Univ. Press, 2003. [Cambridge Tracts in Mathematics 149].

[55] J. Weyman and A. Zelevinsky. Determinantal formulas for multigraded resultants. *J. Alg. Geom.*, 3(4):569–597, 1994.

[56] C.-K. Yap. *Fundamental problems of algorithmic algebra*. Oxford University Press, New York, 2000.