On the Bit Complexity of Solving Bilinear Polynomial Systems

Ioannis Emiris, Angelos Mantzaflaris, Elias Tsigaridas

On the Bit Complexity of Solving Bilinear Polynomial Systems

Ioannis Z. Emiris
Dept Informatics & Telecoms
University of Athens, Greece;
ATHENA Research Center
Maroussi, Greece
emiris@di.uoa.gr

Angelos Mantzaflaris
RICAM, Austrian Academy of Sciences
Altenberger Str. 69, 4040
Linz, Austria
angelos.mantzaflaris@oeaw.ac.at

Elias Tsigaridas
Sorbonne Universités, UPMC Univ Paris 06, CNRS, INRIA, Laboratoire d'Informatique de Paris 6 (LIP6), Équipe POLSYS, 4 place Jussieu, 75252 Paris Cedex 05, France
elias.tsigaridas@inria.fr

ABSTRACT

We bound the Boolean complexity of computing isolating hyperboxes for all complex roots of systems of bilinear polynomials. The resultant of such systems admits a family of determinantal Sylvester-type formulas, which we make explicit by means of homological complexes. The computation of the determinant of the resultant matrix is a bottleneck for the overall complexity. We exploit the quasi-Toeplitz structure to reduce the problem to efficient matrix-vector products, corresponding to multivariate polynomial multiplication. For zero-dimensional systems, we arrive at a primitive element and a rational univariate representation of the roots. The overall bit complexity of our probabilistic algorithm is $\widetilde{\mathcal{O}}_B(n^4D^4 + n^2D^4\tau)$, where n is the number of variables, D equals the bilinear Bézout bound, and τ is the maximum coefficient bitsize. Finally, a careful infinitesimal symbolic perturbation of the system allows us to treat degenerate and positive dimensional systems, thus making our algorithms and complexity analysis applicable to the general case.

CCS Concepts

•**Computing methodologies** → *Symbolic calculus algorithms;*

Keywords

bilinear polynomial, polynomial system solving, separation bounds, DMM, resultant matrix

1. INTRODUCTION

Efficient algorithms for solving of polynomial systems is one of the most active areas of research in computational algebra. Its importance stems from the fact that, using polynomial systems, we can model many problems in various research disciplines. We focus on efficient algorithms for solving bilinear polynomial systems. Such systems are common in various applications, for example in cryptography [21, 30], coding theory [37], real algebraic geometry [42], and game theory [15]. We derive explicit Boolean complexity estimates for isolating all the roots of bilinear polynomial systems without assumptions on the input.

Bilinear systems. Let $S_1(d) = \mathbb{R}[x_1, \ldots, x_{n_1}]_d$, $S_2(d) = \mathbb{R}[y_1, \ldots, y_{n_1}]_d$ and $n = n_1 + n_2$. The space of polynomials of bidegree (d_1, d_2) is denoted by $S(d_1, d_2) = S_1(d_1) \otimes S_2(d_2)$. We consider a square system $f_1, \ldots, f_n$ of n polynomials $f_k \in S(1,1)$. We call these polynomials $(n_1, n_2)-$bilinear, that is, they are homogeneous of degree 1 with respect to each of the sets of (affine) variables $\{x_1, \ldots, x_{n_1}\}$ and $\{y_1, \ldots, y_{n_2}\}$. We aim to isolate all complex roots of the system

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

where $\mathbf{x} = (x_1, \ldots, x_{n_1})$, $\mathbf{y} = (y_1, \ldots, y_{n_2})$. Each polynomial is written in matrix form as $f_k(\mathbf{x}, \mathbf{y}) = (1, \mathbf{x}) A_k (1, \mathbf{y})^\top$, where $A_k \in \mathbb{Z}^{(n_1+1)\times(n_2+1)}$, $1 \le k \le n$ are coefficient matrices.

Assuming for the moment that this system is zero-dimensional, the number of roots in $\mathbb{P}^{n_1} \times \mathbb{P}^{n_2}$ equals the bilinear Bézout bound

$$D = D(n_1, n_2) := \binom{n_1 + n_2}{n_1} = \binom{n_1 + n_2}{n_2}. \tag{2}$$

This number is the mixed volume of the system, and, even in case of infinitely many roots, it is equal to the degree of the variety [15]. This quantity appears frequently in the complexity of our algorithms. By Stirling's formula and by assuming the worst case $n_1 \approx n_2$, we can estimate that $D \sim \frac{4^n}{\sqrt{\pi n}}$. However, if we consider $\min(n_1, n_2) = q$, with q constant, then $D \sim n^q$, i.e. polynomial in the number of variables.

EXAMPLE 1.1. *Consider the $(1,2)-$bilinear, 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 $(x_1; y_1, y_2) = (\pm\frac{\sqrt{3}}{3}; -1, 2 \mp \sqrt{3})$ plus the root $(x_0, x_1; y_0, y_1, y_2) = (0, 1; 0, -1, 1)$ at infinity.

ACM ISBN .
DOI:

Previous Work. Systems homogeneous in distinct groups of variables were firstly discussed by Sylvester [46]. Moreover, Muir [36] and McCoy [34] gave the first expressions of the resultant of such forms as the determinant of a matrix. Sturmfels and Zelevinski [45] first discovered Sylvester-type formulas for unmixed (having the same support), multigraded systems. These formulas can be seen as certain choices of a Weyman complex of modules [48, 49]. Indeed, several classical resultant matrices can be discovered via such complexes [50], 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 from Bézout-type, Sylvester-type blocks (and toric Jacobians). Similar maps have been identified between the terms of Tate resolutions as well [10]. It turns out that for bilinear 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.

In [22], see also [44], the problem of computing the roots of bilinear systems is tackled by means of Gröbner bases. They present a modification of the F5 algorithm that avoids all reductions to zero. For generic bilinear systems the complexity of computing the Gröbner basis is $\mathcal{O}(\binom{n+n_1+1}{n_1+1}^{\omega})$ arithmetic operations, where $n_1 \leq n_2$ and ω is the exponent of matrix multiplication. To isolate the roots we need to convert the basis to a shape form. The bit complexity of such an approach w.r.t. the degree bound is not straightforward. Moreover, this arithmetic bound does not hold, when the systems are not generic or if they are not 0-dimensional. In this regard, our results are not directly comparable with [22]. Our approach is "orthogonal". It uses (a variant of) u-resultant based on the determinantal resultant matrices that we describe in Sec. 2.2. This step is fundamental for estimating bit complexity of the whole process. The arithmetic complexity of our approach is $\widetilde{\mathcal{O}}(D^3)$. To the best of knowledge, our result is the first one regarding the bit complexity of solving (i.e. computing the isolated roots) of any bilinear polynomial system, without regularity or any other assumptions on the dimension of the zero locus. We employ the primitive element representation (PER) of the roots Canny [7] and the rational univariate representation (RUR) [2, 5, 41]; 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 [25]. Our references, with respect to polynomial system solving algorithm is by no means exhaustive.

If we use the data-structure of straight-line programs there are also efficient algorithms to compute the multihomogeneous resultants, cf. [29]. For efficient algorithms for computing general resultant matrices we refer to [9, 17, 18]. For general algorithms for computing the determinant we refer to [31] and references therein.

Outline. We characterize all possible Sylvester-type formulas for the resultant of a (n_1, n_2)-bilinear system of equations. We give explicit algorithms to construct the three possible resultant matrices, which are *optimal*, in the sense that there are no extraneous factors involved (Sec. 2). We adapt the approach of Canny [7] to represent the roots using the primitive element and the rational univariate representation [2, 41] (Sec. 3) for bilinear systems, and we bound the height of the corresponding u-resultant (Sec. 3.1). We provide explicit bounds for the separation of the roots (Sec. 3.2) that depend on the bilinear Bézout bound. We present explicit bit complexity bounds for isolating all the roots of a system of bilinear polynomials (Thm. 4.1). Our algorithm runs in $\widetilde{\mathcal{O}}_B(n^2 n_1 n_2 D^4 \lg(D) + n^2 D^4 \tau)$. We also tackle the cases where the system is overdetermined (Sec. 4.2), has roots at infinity, or is positive-dimensional (Sec. 4.3). If one of the groups of unknowns has a constant number of elements, our bounds are polynomial in the number of variables.

2. SYLVESTER-TYPE FORMULAS FOR THE BILINEAR RESULTANT

In this section we describe three determinantal Sylvester formulas for the resultant of bilinear systems. Sylvester type formulas refer to matrices where the entries are single coefficients of the input polynomials (possibly with a sign change). This kind of expressions are very convenient for both the analysis and the implementation of resultant methods, since the matrix entries have known bitsize and are computable in constant time.

First, let us define the bilinear resultant in $\mathbb{P}^{n_1} \times \mathbb{P}^{n_2}$. Given a sequence $f_0, \ldots, f_n$ of $(n_1, n_2)-$bilinear forms with variables $\mathbf{x}$ and $\mathbf{y}$ (as in Sect. 1) with symbolic coefficients, their resultant $R(f_0, \ldots, f_n)$ is a multihomogeneous polynomial in $\mathbb{Z}[S(1,1)^{n+1}]$, i.e., in the coefficients of the polynomials, with degree w.r.t. (the coefficients of) f_k equal to the Bézout bound

$$\deg_{f_k} R(f_0, \ldots, f_n) = D(n_1, n_2)\ ,\ k = 0, \ldots, n\ .$$

The total degree of the resultant is $(n+1)D(n_1, n_2)$ and vanishes if and only if the polynomials have a common root in $\mathbb{P}^{n_1} \times \mathbb{P}^{n_2}$. That is, this resultant is an instance of the sparse resultant [23], where the toric variety is $\mathbb{P}^{n_1} \times \mathbb{P}^{n_2}$; the resultant is unique up to integer constants. Our aim is to obtain square matrices, with entries coefficients of the polynomials f_k, whose determinant is equal to their resultant, i.e. determinantal Sylvester-type formulas.

2.1 The Weyman resultant complex

In this section we recall some tools from representation theory, which will help us arrive at the bilinear resultant matrices. The bigraded resultant polynomial can be computed as the determinant of the Weyman complex [48], which arises by applying the, so called, geometric technique of [49] to the incidence variety of $f_0, \ldots, f_n$ and generalizes the Cayley-Koszul complex [23]. Note that the classical Koszul complex cannot be used to derive resultant matrices in the bigraded case (cf. [43]).

For any $(m_1, m_2) \in \mathbb{Z}^2$ and $f_0, \ldots, f_n \in S(1,1)$, Weyman's construction is a complex $K_\bullet = K_\bullet(m_1, m_2)$ of finite dimensional vector spaces:

$$0 \to K_{n+1} \to \cdots \to K_1 \xrightarrow{\varphi} K_0 \to \cdots \to K_{-n} \to 0\ , \qquad (3)$$

where the terms are defined as $K_\nu = \oplus_{p=0}^{n} K_{\nu,p}$ with

$$K_{\nu,p} = \left(H^{a_1}_{n_1}(m_1 - p) \otimes H^{a_2}_{n_2}(m_2 - p)\right)^{\binom{n+1}{p}} \qquad (4)$$

with $a_1 + a_2 = p - \nu$, $\nu = n+1, \ldots, -n$, and $H^a_{n_i}(b)$, $b \in \mathbb{Z}$ is the $a-$th cohomology group of $\mathbb{P}^{n_i}$ with coefficients in the sheaf $\mathscr{O}(b)$ [26]. These terms depend solely on n_1, n_2 and m_1, m_2. The maps between the terms (e.g. φ) depend polynomially on the coefficients of $f_0, \ldots, f_n$. This construction appeared in [48] and a detailed presentation is given in [49].

The crucial property of the complex (3) is that its determinant is equal to (a power of) $R(f_0, \ldots, f_n)$. The determinant of the complex is, in principle, a rational expression involving the determinants of the maps in the complex, and is usually 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), then we obtain the resultant as the determinant of the square matrix expressing the map φ 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 φ is an epimorphism if and only if the complex is exact or, equivalently, the polynomials do not have a common root in $\mathbb{P}^{n_1} \times \mathbb{P}^{n_2}$. Moreover, if n_1, n_2 are fixed, the possible values of (m_1, m_2) 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:

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

Here $S_i^*(d)$ denotes the dual space of $S_i(d)$, i.e. the space of linear functionals $\lambda : S_i(d) \to \mathbb{R}$. This space is isomorphic to (evaluations of) polynomials in formal partial derivatives, e.g. $S_1^*(d) \cong \mathbb{R}[\partial_{x_1}, \ldots, \partial_{x_{n_1}}]_d$, see [32, 33] and references therein.

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

EXAMPLE 2.1. *The resultant of three $(1,1)-$bilinear forms, i.e. $n_1 = n_2 = 1$ corresponds to the determinantal complex $K_\bullet(2,1)$, i.e. $(m_1, m_2) = (2,1)$. Using (6), the complex (5) becomes $K_{1,1} \xrightarrow{\varphi} K_{0,0}$, i.e. it implies the existence of 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].*

All classically known resultant formulas can be obtained as the determinant of a map φ in (5), for particular integers $(m_1, m_2) \in \mathbb{Z}^2$. Moreover, the existence of a determinantal complex implies a determinantal formula for the resultant.

2.2 Determinantal Sylvester formulas

In this section we identify all determinantal Sylvester complexes for $(n_1, n_2)-$bilinear systems. These formulas are obtained 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 $p \in \{0, \ldots, n\}$ (cf. [50]). General such formulas for multilinear systems are identified in [14, 16, 45, 50]. We specialize these results to bilinear systems in the following lemma.

LEMMA 2.2. *There exist three determinantal Sylvester maps ϕ_1, ϕ_2, ϕ_3 for the (n_1, n_2)-bilinear forms $f_0, \ldots, f_n$, up to dual pairs. These are*

(i) $\phi_1 : S(0, n_1)^{n+1} \to S(1, n_1 + 1)$, *which is given as*

$$(g_0, g_1, \ldots, g_n) \mapsto \sum_{k=0}^{n} g_k f_k \;,\; g_k \in S_2(n_1)\,. \tag{8}$$

(ii) $\phi_2 : S(n_2, 0)^{n+1} \to S(n_2 + 1, 1)$ *defined the same way as (8), with $g_k \in S_1(n_2)$.*

(iii) $\phi_3 : S_1^*(1)^{\binom{n+1}{n_1+1}} \to S_2(1)^{\binom{n+1}{n_1}}$ *with*

$$(\lambda_1, \lambda_2, \ldots, \lambda_r) \mapsto (q_1, q_2, \ldots, q_s) \tag{9}$$

where $r = \binom{n+1}{n_2}$, $s = \binom{n+1}{n_1}$ and

$$q_j = \sum\nolimits_{k=0, k \notin J_j}^{n} (-1)^\sigma \lambda_\mu f_k \;,\; j = 1, \ldots, s$$

where $J_j \subset \{0, \ldots, n\}, |J_j| = n_1$ denotes the combination with lexicographic index j, $\mu = \mu(j,k)$ is the lexicographic index of the (n_1+1)-combination $J_j \cup \{k\}$ of $\{0, \ldots, n\}$, $\sigma = \sigma(j,k) = |\{t \in J_j : t < k\}|$ and $\lambda_\mu \in S_1^*(1)$, $\mu = 1, \ldots, r$.

Moreover, these are the only Sylvester maps arising from complex (3).

PROOF. The formulas (i) and (ii) are depicted in [45] for the multilinear case. They correspond to the choices $(m_1, m_2) = (1, n_1 + 1)$ and $(m_1, m_2) = (n_2 + 1, 1)$ of degree vectors in (3), respectively. They have the form of a classical Sylvester map, expressing multiplication by f_k's. The third formula is found in [48] and corresponds to the degree vector $(m_1, m_2) = (-1, n_1 + 1)$. The map expresses a linear combination of applications of dual functionals $w_0 \mathbf{1} + \sum_i w_i \partial_{x_i}$ to the f_k's [16].

For any of the three determinantal degree vectors (m_1, m_2) listed above, the vector $(n_2 - m_1, n_1 - m_2)$ is also determinantal. Due to Serre duality, this additional vector yields the transpose of the matrix corresponding to (m_1, m_2).

For the uniqueness, we look at all possible complexes of the form (7). A case by case analysis shows that the possible Sylvester maps arise from the values $p = 0$, $p = n_1$, $p = n_2$, or $p = n$, due to (6). Using (4) we obtain the complexes $K_\bullet(1, n_1 + 1)$, $K_\bullet(n_2 + 1, 1)$, $K_\bullet(-1, n_1 + 1)$ as well as their duals $K_\bullet(n_2 - 1, -1)$, $K_\bullet(-1, n_1 - 1)$, $K_\bullet(n_2 + 1, -1)$, all corresponding to the listed maps (i–iii). □

Let us remark that the dual (or adjoint) Sylvester map of (i) is a map $\Phi_1 : S^*(1, n_1 + 1) \to S_2^*(n_1)^{(n+1)}$, with

$$\lambda \mapsto (f_0 \cdot \lambda,\, f_1 \cdot \lambda,\, \ldots,\, f_n \cdot \lambda)\,,\;\; \lambda \in S^*(1, n_1 + 1)\,, \tag{10}$$

where the functionals $f_k \cdot \lambda$ are defined as $g \mapsto \lambda(g f_k)$.

EXAMPLE 2.3. *We demonstrate these Sylvester-type resultant matrices. Consider the $(1,2)-$bilinear system of Example 1.1, augmented by an extra polynomial f_0*

$$f_0 = a_0 + a_1 y_2 + a_2 y_1 + a_3 x_1 + a_4 x_1 y_2 + a_5 x_1 y_1\,. \tag{11}$$

The set $f_0, \ldots, f_3$ is an overdetermined system of equations. The resultant has degree $\deg R(f_0, \ldots, f_3) = 4\,D(1,2) = 12$, so this will be the dimension of the Sylvester-type matrices.

The maps (i) and (ii) of Lemma 2.2 are quite similar to the classical Macaulay map but also take into account the special bihomogeneous structure of the system. We have

$$\phi_1 : S(0,1)^4 \to S(1,2)\,,$$

which yields the following (transposed) matrix

$$\begin{array}{c} \\ f_0 \\ y_1 f_0 \\ y_2 f_0 \\ f_1 \\ y_1 f_1 \\ y_2 f_1 \\ f_2 \\ y_1 f_2 \\ y_2 f_2 \\ f_3 \\ y_1 f_3 \\ y_2 f_3 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 1 & y_2 & y_2^2 & y_1 & y_1 y_2 & y_1^2 & x_1 & x_1 y_2 & x_1 y_2^2 & x_1 y_1 & x_1 y_1 y_2 & x_1 y_1^2 \end{array} \\ \begin{bmatrix} a_0 & a_1 & 0 & a_2 & 0 & 0 & a_3 & a_4 & 0 & a_5 & 0 & 0 \\ 0 & 0 & 0 & a_0 & a_1 & a_2 & 0 & 0 & 0 & a_3 & a_4 & a_5 \\ 0 & a_0 & a_1 & 0 & a_2 & 0 & 0 & a_3 & a_4 & 0 & a_5 & 0 \\ b_0 & b_1 & 0 & b_2 & 0 & 0 & b_3 & b_4 & 0 & b_5 & 0 & 0 \\ 0 & 0 & 0 & b_0 & b_1 & b_2 & 0 & 0 & 0 & b_3 & b_4 & b_5 \\ 0 & b_0 & b_1 & 0 & b_2 & 0 & 0 & b_3 & b_4 & 0 & b_5 & 0 \\ c_0 & c_1 & 0 & c_2 & 0 & 0 & c_3 & c_4 & 0 & c_5 & 0 & 0 \\ 0 & 0 & 0 & c_0 & c_1 & c_2 & 0 & 0 & 0 & c_3 & c_4 & c_5 \\ 0 & c_0 & c_1 & 0 & c_2 & 0 & 0 & c_3 & c_4 & 0 & c_5 & 0 \\ d_0 & d_1 & 0 & d_2 & 0 & 0 & d_3 & d_4 & 0 & d_5 & 0 & 0 \\ 0 & 0 & 0 & d_0 & d_1 & d_2 & 0 & 0 & 0 & d_3 & d_4 & d_5 \\ 0 & d_0 & d_1 & 0 & d_2 & 0 & 0 & d_3 & d_4 & 0 & d_5 & 0 \end{bmatrix} \end{array}\,.$$

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 (ii) we obtain

$$\phi_2 : S(2,0)^4 \to S(3,1)$$

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

$$\begin{array}{c} \\ f_0 \\ x_1 f_0 \\ x_1^2 f_0 \\ f_1 \\ x_1 f_1 \\ x_1^2 f_1 \\ f_2 \\ x_1 f_2 \\ x_1^2 f_2 \\ f_3 \\ x_1 f_3 \\ x_1^2 f_3 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 1 & y_2 & y_1 & x_1 & x_1y_2 & x_1y_1 & x_1^2 & x_1^2y_2 & x_1^2y_1 & x_1^3 & x_1^3y_2 & x_1^3y_1 \end{array} \\ \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & b_0 & b_1 & b_2 & b_3 & b_4 & b_5 \\ c_0 & c_1 & c_2 & c_3 & c_4 & c_5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_0 & c_1 & c_2 & c_3 & c_4 & c_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_0 & c_1 & c_2 & c_3 & c_4 & c_5 \\ d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & d_0 & d_1 & d_2 & d_3 & d_4 & d_5 \end{bmatrix} \end{array}.$$

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 map in (iii)

$$\phi_3 \,:\, S_1^*(1)^6 \to S_2(1)^4$$

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:

$$\begin{array}{c} \\ \mathbf{1} \\ \partial_{x_1} \\ \mathbf{1} \\ \partial_{x_1} \\ \mathbf{1} \\ \partial_{x_1} \\ \mathbf{1} \\ \partial_{x_1} \\ \mathbf{1} \\ \partial_{x_1} \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 1 & y_2 & y_1 & 1 & y_2 & y_1 & 1 & y_2 & y_1 & 1 & y_2 & y_1 \end{array} \\ \begin{bmatrix} -b_0 & -b_1 & -b_2 & a_0 & a_1 & a_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -b_3 & -b_4 & -b_5 & a_3 & a_4 & a_5 & 0 & 0 & 0 & 0 & 0 & 0 \\ -c_0 & -c_1 & -c_2 & 0 & 0 & 0 & a_0 & a_1 & a_2 & 0 & 0 & 0 \\ -c_3 & -c_4 & -c_5 & 0 & 0 & 0 & a_3 & a_4 & a_5 & 0 & 0 & 0 \\ -d_0 & -d_1 & -d_2 & 0 & 0 & 0 & 0 & 0 & 0 & a_0 & a_1 & a_2 \\ -d_3 & -d_4 & -d_5 & 0 & 0 & 0 & 0 & 0 & 0 & a_3 & a_4 & a_5 \\ 0 & 0 & 0 & -c_0 & -c_1 & -c_2 & b_0 & b_1 & b_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -c_3 & -c_4 & -c_5 & b_3 & b_4 & b_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & -d_0 & -d_1 & -d_2 & 0 & 0 & 0 & b_0 & b_1 & b_2 \\ 0 & 0 & 0 & -d_3 & -d_4 & -d_5 & 0 & 0 & 0 & b_3 & b_4 & b_5 \\ 0 & 0 & 0 & 0 & 0 & 0 & -d_0 & -d_1 & -d_2 & c_0 & c_1 & c_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & -d_3 & -d_4 & -d_5 & c_3 & c_4 & c_5 \end{bmatrix} \end{array}.$$

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

The three above matrices all 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 $\{a_0, \ldots, a_5\}$, $\{b_0, \ldots, b_5\}$, $\{c_0, \ldots, c_5\}$, $\{d_0, \ldots, d_5\}$. This polynomial is $R(f_0, \ldots, f_3)$.

3. REPRESENTATION OF THE ROOTS

We represent the roots of the system as rational function of univariate polynomials, evaluated at the roots of a univariate polynomial. We employ the primitive element representation (PER) [7] and the rational univariate representation (RUR) [2, 5, 41]. In RUR the denominator is the same for all the 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 bilinear polynomials as in Eq. (1). We add the polynomial

$$f_0(\mathbf{x}, \mathbf{y}) = u_{0,0} + \sum_{i=1}^{n_1} u_{i,0} x_i + \sum_{j=1}^{n_2} u_{0,j} y_j + \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} u_{i,j} x_i y_j \,,$$

where the $u_{i,j}$ are parameters, to obtain the overconstrained system

$$(\Sigma_0) : \; f_0(\mathbf{x}, \mathbf{y}) = f_1(\mathbf{x}, \mathbf{y}) = \cdots = f_n(\mathbf{x}, \mathbf{y}) = 0. \qquad (12)$$

We can compute the resultant of (Σ_0) as the determinant of any of the matrices presented in the previous section.

The PER. We follow Canny's approach [7] but instead of using a linear polynomial f_0, we use a bilinear one. This corresponds to the "trick" of hiding a variable. Another reason for choosing such a polynomial is that if all the $n+1$ polynomials have the same support, then the determinant of the resultant matrices 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 have need to treat this extraneous factor.

The resultant $R(f_0)$ of the system (Σ_0) is a polynomial; it is a product of factors of the form

$$\sum_{i=0}^{n_1} \sum_{j=0}^{n_2} u_{i,j} \alpha_{k,i} \beta_{k,j} \,.$$

We choose $(n_1+1)(n_2+1) - 1$ constants and we set $r(z) = R(-z, c_{1,0}, \ldots, c_{0,1}, \ldots, c_{1,1}, \ldots, c_{n_1,n_2})$, where z is a new variable. With this substitution the factors of R corresponding to roots at infinity become constants. The other factors of R are of the form

$$z\,\alpha_{k,0}\beta_{k,0} - \sum_{i=1}^{n_1} c_{i,0}\alpha_{k,i}\beta_{k,0} - \sum_{j=1}^{n_2} c_{0,j}\alpha_{k,0}\beta_{k,j} - \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} c_{i,j}\alpha_{k,i}\beta_{k,j}.$$

We may assume, without loss of generality, that $\alpha_{k,0} = \beta_{k,0} = 1$ for the roots that are not at infinity. In this way the roots of r, which we denote as ζ_m, are

$$\zeta_m = \sum_{i=1}^{n_1} c_{i,0}\alpha_{m,i} + \sum_{j=1}^{n_2} c_{0,j}\beta_{m,j} + \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} c_{i,j}\alpha_{m,i}\beta_{m,j}, \quad (13)$$

where m runs over the roots of the system. By abuse of notation, we also denote by r the square-free part of r. With this notation $r(z) = \mathtt{lc}(r)\prod_m (z - \zeta_m)$, where m runs over all the distinct roots of r, and

$$r'(z) = \mathtt{lc}(r) \sum_m \prod_{\nu \neq m} (z - \zeta_m). \qquad (14)$$

For the PER of the x coordinates of the solutions of the bilinear system, we consider the polynomials $\hat{a}_k^+(z)$ and $\hat{a}_k^-(z)$ where

$$\begin{aligned} \hat{a}_k^\pm(z) = R(-z, c_{1,0}, \ldots, (c_{k,0} \pm 1), \ldots, c_{n_1,0}, \\ c_{0,1}, \ldots, c_{0,n_2}, c_{1,1}, \ldots, c_{n_1,n_2}), \end{aligned} \qquad (15)$$

for $1 \le k \le n_1$. That is, we consider a pair of polynomials for each of the x variables. Let $a_k^\pm$ be the corresponding square-free parts of $\hat{a}_k^\pm$, respectively. The roots of $a_k^\pm(z)$ are $\zeta_m \pm \alpha_{m,k}$ and so we have the factorization

$$a_k^\pm(z) = \mathtt{lc}(a_k^\pm) \prod_m (z - \zeta_m \mp \alpha_{m,k}).$$

Consider the polynomial $a_k^+(2\theta - z)$. It holds

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

The resultant w.r.t. z of $a_k^-(z)$ and $a_k^+(2\theta - z)$, where θ is a new parameter, is

$$\begin{aligned} \mathtt{res}(a_k^-(z), a_k^+(2\theta - z)) &= \prod_m a_k^+(2\theta - \zeta_m + a_{m,k}) \\ &= \prod_m \prod_\nu (\zeta_m - \alpha_{m,k} - 2\theta + \zeta_\nu + \alpha_{\nu,k}). \end{aligned} \qquad (16)$$

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

$$\zeta_m - \alpha_{m,k} - 2\zeta_l + \zeta_\nu + \alpha_{\nu,k} = 0, \tag{17}$$

and then the resultant is 0. Eq. (17) holds for sure if $m = l = \nu$. However, there might be "bad" choices of the constants $c_{i,j}$ that result in spurious roots for different tuples of roots of the system. We will characterize these "bad" values in Sec. 3.1. When these values are avoided, we refer to the resulting f_0 as a *separating polynomial*.

Assuming that there are no spurious roots, we consider $a_k^-(z)$ and $a_k^+(2\theta - z)$ as a bivariate polynomial system. Its solutions are described by univariate polynomials $s_{k,0}$ and $s_{k,1}$ of degree $\mathcal{O}(D^2)$ [35, Thm. 1] as $s_{k,0}(\theta) = 0$ and $z = s_{k,1}(\theta)/s'_{k,0}(\theta)$. We notice from Eq. (16) that 2θ, and so the roots of $s_{k,0}(\theta)$, encode all the possible sums of the roots of $a_k^-(z)$ and $a_k^+(z)$. Hence, r divides $s_{k,0}$, since when $m = \nu$ in (16), then $\theta = \zeta_m$, which are exactly the roots of r, see (13). Therefore, for θ such that $r(\theta) = 0$, the polynomials $a_k^-(z)$ and $a_k^+(2\theta - z)$ have a common root, say $\zeta_\ell - \alpha_{\ell,k}$, which is described as $s_{k,1}(\theta)/s'_{k,0}(\theta)$. Thus, the PER for the k-th x-coordinate is

$$r_k(z) = z - \frac{s_{k,1}(z)}{s'_{k,0}(z)} = \frac{z\, s'_{k,0}(z) - s_{k,1}(z)}{s'_{k,0}(z)} = \frac{\widetilde{s}_{k,1}(z)}{s'_{k,0}(z)}, \tag{18}$$

where z runs over all the roots of r. In addition the polynomials r and $s'_{k,0}$ are relative prime, and so we can compute the inverse of $s'_{k,0}$ modulo r. Thus, we can express $r_k(z)$ as a $r_k(z) = p_k(z) \mod r(z)$, for a polynomial $p_k = \widetilde{s}_{k,1}\,(s'_{k,0})^{-1}$.

The RUR. To compute the RUR we slightly modify the approach in [2], see also [11, 41], to fit our needs. Consider $\hat{f}_{0,k}(\mathbf{x}, \mathbf{y}) = f_0(\mathbf{x}, \mathbf{y}) + \mu\, x_k$, where μ is a new variable, for $1 \le k \le n_1$, and f_0, as in PER, is a bilinear separating polynomial, that forces the roots at infinity to be constants that multiply the resultant. An explicit construction of f_0 is in Sec. 3.1. We replace f_0 with $\hat{f}_{0,k}$ in (Σ_0) to obtain a new system $(\Sigma_{0,k})$. We substitute $u_{0,0} = -z$ in f_0. Let $g_k \in (\mathbb{Z}[\mu])[z]$ be (the square-free part of) the resultant of $(\Sigma_{0,k})$ after we eliminate $\mathbf{x}$ and $\mathbf{y}$. It holds $\deg(g_k) = \mathcal{O}(D)$, and $g_k(z) = \mathtt{lc}(g_k) \prod_m (z - \zeta_m - \mu\, \alpha_{m,k})$, and

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

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

$$-s_k(z) = \mathtt{lc}(g_k) \sum\nolimits_m \alpha_{m,k} \prod\nolimits_{\nu \neq m} (z - \zeta_m).$$

If we combine it with Eq. (14) then we obtain the representation

$$x_k = -\frac{\mathtt{lc}(r)\, s_k(z)}{\mathtt{lc}(g_k)\, r'(z)} \tag{19}$$

for the k-th x-coordinate of the roots of the system. We can obtain a similar representation for y-coordinates.

To recover the y-coordinates of the solutions, we consider the polynomials $\hat{b}_\ell^+(z)$ and $\hat{b}_\ell^-(z)$ where

$$\hat{b}_\ell^\pm(z) = R(-z, c_{1,0}, \ldots, c_{n_1,0}, c_{0,1}, \ldots, (c_{0,\ell} \pm 1), \ldots, c_{0,n_2}, c_{1,1}, \ldots, c_{n_1,n_2})$$

for $1 \le \ell \le n_2$. We work similarly and we obtain a representation $-\mathtt{lc}(r) q_k(z)/(\mathtt{lc}(g_k)\, r'(z))$, for $1 \le k \le n_2$, for the y-coordinates. In total, we have to perform this procedure $n = n_1 + n_2$ times to obtain a representation for all the coordinates.

LEMMA 3.1. *Let the polynomials of the system (Σ_0) in Eq. (12) have integer coefficients. Let the resultant of the $(n_1, n_2)-$bilinear system be of degree D and bitsize $\mathcal{O}(L)$. The representation of the roots using Eq. (18) consists of polynomials of degree $\mathcal{O}(D^2)$ and bitsize $\widetilde{\mathcal{O}}(D^2+DL)$. The representation of the roots using Eq. (19) consists of polynomials of degree $\mathcal{O}(D)$ and bitsize $\widetilde{\mathcal{O}}(D+L)$. We compute them in $\widetilde{\mathcal{O}}_B(n(D^4 + D^3L))$, with a Monte Carlo algorithm for Eq. (18) and deterministically for Eq. (19), whenever a separating polynomial f_0 is known.*

PROOF. The resultant is univariate polynomial in z. Let r be its square-free part; then $\deg(r) = \mathcal{O}(D)$ and $\mathfrak{h}(r) = \mathcal{O}(D + L)$. The same bounds holds for $a_k^\pm$, since they are also computed as determinants of the resultant matrices. With these bounds the complexity of PER is a direct consequence of [35, Thm. 1].

For Eq. (18) we proceed as follows. We pick a prime, $\mathtt{p}$, of bitsize $\widetilde{\mathcal{O}}(D^2 + DL)$, and we perform all the computations in the ring $(\mathbb{Z}/\mathbb{Z}_\mathtt{p})[z]/r(z)$. Then, in this ring, we compute a representation of the common root of $a_k^-(z)$ and $a_k^+(2\theta - z)$, $\frac{\widetilde{s}_{k,1}(z)}{s_{k,0}(z)}$, by exploiting their first subresultant polynomial. For this we use fast subresultant algorithms [1, 47]. Next, we compute the inverse of $s_{k,0}$, and finally the representation $r_k(z) = p_k(z) \mod r(z)$, for $p_k = \widetilde{s}_{k,1}\,(s_{k,0})^{-1}$. In total we perform $\widetilde{\mathcal{O}}(D)$ operations in the ring. Each operation costs $\widetilde{\mathcal{O}}_B(D(D^2+DL)) = \widetilde{\mathcal{O}}_B(D^3+D^2L)$, and so the overall cost is $\widetilde{\mathcal{O}}_B(D^4 + D^3L)$ [1].

For the RUR, Eq. (19), we need to compute g_k, its square-free part, its derivative w.r.t. μ, and finally s_k. As g_k is also a determinant of a resultant matrix, it holds $\mathfrak{h}(g_k) = \widetilde{\mathcal{O}}(L)$, and so $\mathfrak{h}(s_k) = \widetilde{\mathcal{O}}(D + L)$. Moreover, $\deg(s_k) = \mathcal{O}(D)$. To compute s_k we notice that $\mathtt{lc}(r) s_k = \mathtt{lc}(g_k)\, r'\, p_k \mod r$; an operation that costs $\widetilde{\mathcal{O}}(D)$. So the overall cost is $\widetilde{\mathcal{O}}_B(n(D^4 + D^3L))$ as we have to compute the representation for all the coordinates. □

3.1 "Bad" values for $c_{i,j}$ and the height of f_0

Following Sec. 3, we choose a bilinear f_0 to add to (Σ) and we obtain the system (Σ_0), Eq. (12), of the form

$$f_0(\mathbf{x}, \mathbf{y}) = -z + \sum_{i=1}^{n_1} c_{i,0} x_i + \sum_{j=1}^{n_2} c_{0,j} y_j + \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} c_{i,j} x_i y_j\ ,$$

where $c_{i,j} \in \mathbb{Z}$ are constants to be specified in the sequel, and z is a new parameter. There are values of $c_{i,j}$ that make our algorithm fail. The goal of this section is to identify these "bad" values. Moreover, We also estimate the height of the coefficients of f_0.

Assume that D bounds the affine as well as the projective isolated roots of the system, see Eq. (2). First, we replace each $c_{i,j}$ with a power of t, where t is a new indeterminate. In this way, the evaluation of f_0 at each solution of the system, say $(\alpha_{k,0}, \ldots, \alpha_{k,n_1}, \beta_{k,0}, \ldots, \beta_{k,n_2})$, results a polynomial in t

$$f_{0,k} = -\sum\nolimits_{i=0}^{n_1} \sum\nolimits_{j=0}^{n_2} t^{i(n_2+1)+j} \alpha_{k,i} \beta_{k,j}, \tag{20}$$

that has degree $n_1 n_2 + n_1 + n_2$. With this substitution, it suffices to choose a constant for t.

Our goal 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 $c_{i,j}$, 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 resultant to vanish identically.

For a root at infinity we may have $\alpha_{k,0} = 0$, or $\beta_{k,0} = 0$, or $\alpha_{k,0} = \beta_{k,0} = 0$, for some k. We can forget the last case as it is contained in the first two. In each of the first two cases, $f_{0,k}$ is a

polynomial in t of degree $\leq n_1n_2+n_1+n_2 \leq 2\,n_1n_2$. The product of these two polynomials is a polynomial of degree $\leq 4\,n_1n_2$, 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. (20), and the product of all these polynomials is a polynomial of degree at most $4n_1n_2D$. In a similar way, we obtain such a polynomial when we consider the computation of the polynomials $a_k^{\pm}$ and $b_k^{\pm}$. There are $2n_1$ and $2n_2$ such polynomials, respectively, each of degree at most $4n_1n_2D$ in t. Now we consider the product of all these polynomials. We obtain a polynomial in t of degree $\leq 8n_1n_2(n_1+n_2)D = 8n_1n_2nD$. The roots of this polynomial describe all the first class of "bad" values of t, hence of the constants $c_{i,j}$. Any value for t which does not nullify this polynomial ensures that $r(z)$, $a_k^{\pm}$, and $b_k^{\pm}$ do not vanish identically, even in the presence of isolated roots of (Σ) at infinity.

The second class of "bad" values for t are those that force Eq. (17) to vanish for distinct indices m, l, and ν. After substituting each $c_{i,j}$ by a suitable power of t, Eq. (17) becomes a polynomial in t of degree $\leq 2n_1n_2$. We have one such polynomial for each triple of roots ζ_m, ζ_l, and ζ_ν, see Eq. (13), and for all possible k. Therefore, there are $\binom{D}{3}n_1$ plus $\binom{D}{3}n_2$ polynomials that correspond to "bad" values for the x and y coordinates, respectively. The product of all of them results a polynomial of degree at most $2nn_1n_2D^3$.

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 $2n_1n_2nD^3 + 8n_1n_2nD \leq 10n_1n_2nD^3$. Hence, if we consider the integers in the interval $I = [0,\, 10n_1n_2nD^3]$ 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 $c_{i,j}$. Obviously, $|t_0| = \mathtt{H}(t_0) \leq 10n_1n_2nD^3$.

Substituting $t = t_0$ in $f_{0,k}$, we obtain $c_{i,j}$ and f_0 such that

$$\mathtt{H}(f_0) \leq (10n_1n_2nD^3)^{n_1n_2+n}$$

and so $\mathfrak{h}(f_0) = \mathcal{O}(n_1n_2\lg(n_1n_2) + n_1n_2\lg(D))$.

LEMMA 3.2. $\mathfrak{h}(f_0) = \mathcal{O}(n_1n_2\lg(n_1n_2) + n_1n_2\lg(D))$.

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 $c\,10\,n_1n_2nD^3$ 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.

3.2 Separation and representation bounds

Separation bounds are bounds on the minimum distance between two isolated roots of a polynomial system. We use DMM bound [19, 20], which is an output sensitive aggregate version to estimate the separation bound of a system of bilinear polynomials.

We assume that $f_i \in \mathbb{Z}[\mathbf{x},\mathbf{y}]$ and $\mathfrak{h}(f_i) \leq \tau$. Then $\mathfrak{h}(f_0) = \sigma = \mathcal{O}(n_1n_2\lg(n_1n_2) + n_1n_2\lg(D))$ (Lem. 3.2). By Lem. 2.2, we know exactly the form of the resultant for bilinear systems. In particular, it is homogeneous of degree D in the coefficients of each f_i. It has the form

$$r(z) = \cdots + \varrho_i\, z^i\, \mathbf{c}_i^{D-i}\mathbf{a}_{1,i}^D\mathbf{a}_{2,i}^D\cdots\mathbf{a}_{n,i}^D + \ldots, \qquad (21)$$

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 product of simplices. Following [19, 20] we get $\mathfrak{h}(r) = \mathcal{O}(nD\lg(nD) + D\sigma + nD\tau)$, that expands to

$$\begin{aligned}\mathfrak{h}(r) &= \mathcal{O}(nD\lg(nD) + n_1n_2D\lg(n_1n_2D) + nD\tau)\\ &= \widetilde{\mathcal{O}}(n_1n_2D\lg(D) + nD\tau).\end{aligned} \qquad (22)$$

The same bounds hold for $\hat{a}_k^{\pm}$, $\hat{b}_\ell^{\pm}$ and r', and this L that appears in Lem. 3.1. Therefore, for the representation of the roots, of Eq. (19) we have that $\deg(s_k) = \mathcal{O}(D)$ and

$$\mathfrak{h}(s_k) = \widetilde{\mathcal{O}}(n_1n_2D\lg(D) + nD\tau). \qquad (23)$$

If Δ_i is the separation bound of the i-th isolated root of (Σ), then using DMM [20, Cor. 10],

$$-\lg\textstyle\prod_i \Delta_i = \mathcal{O}(nD^2\lg(nD) + nD^2\tau). \qquad (24)$$

Moreover, for any root of the system it holds

$$2^{-\mathcal{O}(nD\lg(n)+nD\tau)} \leq |\alpha_{k,i}|, |\beta_{\ell,j}| \leq 2^{\mathcal{O}(nD\lg(n)+nD\tau)}, \qquad (25)$$

where $1 \leq i \leq n_1$ and $1 \leq j \leq n_2$.

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. 2.2 implies that $b^\top M$ corresponds to $(n+1)$ polynomial multiplications. Each involves two polynomials: one of bidegree (1,1) and one of degree n_1 with n_2 variables (the case of the first resultant matrix), or of degree n_2 in n_1 variables (the case of the first resultant matrix). 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 two resultant matrices.

It is worth to mention that for the third matrix we can exploit directly its block structure and we can avoid to use polynomial multiplication. This is so due to its construction using differentials. In this way we can compute the matrix-vector product using $\mathcal{O}(n\min(n_1,n_2)^2D)$ operations.

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 $\mathfrak{h}(r) = \widetilde{\mathcal{O}}(n_1n_2D\lg(D) + nD\tau)$. 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 $\widetilde{\mathcal{O}}(n_1n_2D\lg(D) + nD\tau)$. The cost of each multiplication is $\widetilde{\mathcal{O}}_B(n_1n_2D^2\lg(D) + nD^2\tau)$, using a probabilistic [3, Thm. 7.1] or a worst case [28, Cor. 21] algorithm. Thus, the total cost for computing the polynomial r is $\widetilde{\mathcal{O}}_B(n\,n_1n_2D^4\lg(D) + n^2D^4\tau)$.

To compute the representation of the roots of Eq. (19) 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 $\widetilde{\mathcal{O}}_B(n^2n_1n_2D^4\lg(D) + n^3D^4\tau)$. Following Lem. 3.1, the degree of s_k's is $\mathcal{O}(D)$ and their bitsize is $\widetilde{\mathcal{O}}(n^2D + n_1n_2D\lg(D) + nD\tau)$. The cost to construct them is $\widetilde{\mathcal{O}}_B(n^3D^4 + nn_1n_2D^4\lg(D) + n^2D^4\tau)$.

Next, we isolate all the complex roots of r, with cost bounded by $\widetilde{\mathcal{O}}_B(D^2\mathfrak{h}(r)) = \widetilde{\mathcal{O}}_B(D^3(n_1n_2\lg(D)+n\tau))$ [38]. We obtain isolating boxes for the complex roots of r. Then, we refine the roots up to accuracy $2^{-\lambda}$ in $\widetilde{\mathcal{O}}_B(D^2\mathfrak{h}(r)+D\lambda)$ [39]. We perform all the computations with λ bits of accuracy. We need to determine the value for λ such that the evaluation of the RUR, see Eq. (19), at the approximations of the roots of r results to disjoint hyperboxes for the roots of the system.

After refinement, for every root γ_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. (19), 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 [27, Sec. 5.1] we have $\mathtt{wid}(s_k([\gamma_i])) \leq \boldsymbol{c}_{2D}\, \overline{s}_k(|\gamma_i|)$. The constant $\boldsymbol{c}_{2D}$, under the mild assumption that the precision used to perform the computations is bigger than $\lg(n)$, is $\boldsymbol{c}_{2D} \leq 5D\,2^{-\lambda} = 2^{-\lambda+\mathcal{O}(\lg(D))}$. 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. (25). A similar bound holds for $r'([\gamma_i])$. Putting everything together we have

$$\mathtt{wid}(I_{k,i}) \leq 2^{-\lambda+\widetilde{\mathcal{O}}(n_1n_2D\lg(D)+nD^2\tau)}.$$

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 λ such that

$$\mathtt{wid}(I_{k,i}) \leq 2^{-\lambda+\widetilde{\mathcal{O}}(n_1n_2D\lg(D)+nD^2\tau)} \leq 2^{-\lg\prod_i \Delta_i},$$

which results the following bound $\lambda > \mathcal{O}(nD^2\lg(D)+nD^2\tau)$ on 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 $\widetilde{\mathcal{O}}_B(nD^3\tau+D\lambda)$ [40, Lemma 21]. The previous discussion leads to the theorem.

THEOREM 4.1. *There is a probabilistic algorithm for isolating the roots of a 0-dimensional system of (n_1,n_2)-bilinear polynomials with integer coefficients of maximum bitsize τ, in $\widetilde{\mathcal{O}}_B(n^2n_1n_2D^4\lg(D)+n^2D^4\tau)$, where $n=n_1+n_2$ and D is the bilinear Bézout bound of Eq. (2).*

We would have obtained the same complexity results if we had used the PER, Eq. (18), representation of the roots. In this case λ would have been $\lambda > \mathcal{O}(nD^3\lg(D)+nD^3\tau)$.

4.2 Overdetermined systems of dimension 0

Assume that the input consists of more than n bilinear polynomials, say p, that is $f_1,\ldots,f_p$, which possess a finite number of (biprojective) roots. First, we have to make the system square, using the technique from [24], by considering 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 of this new, square system is asymptotically the same, up to logarithmic factors. We refer to [20] for details. Then, we solve the system using Thm. 4.1. 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 previous section. However, the procedure 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

$$f_k\left(\frac{-\mathtt{lc}(r)\,s_1}{\mathtt{lc}(g_1)\,r'},\ldots,\frac{-\mathtt{lc}(r)\,s_{n_1}}{\mathtt{lc}(g_{n_1})\,r'},\frac{-\mathtt{lc}(r)\,q_1}{\mathtt{lc}(h_1)\,r'},\ldots,\frac{-\mathtt{lc}(r)\,q_{n_2}}{\mathtt{lc}(h_{n_2})\,r'}\right).$$

In this way we obtain a rational function in z, say $\frac{f_{k,0}(z)}{f_{k,1}(z)}$. Let σ be the bitsize of the polynomials in RUR. We can compute this rational function in $\widetilde{\mathcal{O}}_B(n(n+n_1n_2)D\sigma)$ [40]. 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 $\widetilde{\mathcal{O}}_B(nD^2\sigma)$ [47], and so the total cost is $\widetilde{\mathcal{O}}_B(pnD^2\sigma)$. If we substitute $\sigma = \widetilde{\mathcal{O}}(n_1n_2D\lg(D)+nD\tau)$, then we obtain the bound $\widetilde{\mathcal{O}}_B(p(nn_1n_2D^3\lg(D)+n^2D^3\tau))$. If we use a probabilistic algorithm for the GCD of $p+1$ polynomials using one GCD operation [47], then we can eliminate the factor p.

4.3 Positive dimensional systems

The resultant computation and thus the algorithm of Thm. 4.1 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. 3.1), 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 of Sect. 3.1, which are derived under the assumption of finitely many solutions in $\mathbb{P}^{n_1}\times\mathbb{P}^{n_2}$ 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.

To compute the roots in these cases we consider the general approach of generalized characteristic polynomial of [8]. In particular, we apply the perturbation scheme for sparse systems of [13]. Assume a square $(n_1,n_2)-$bilinear system $p_1,\ldots,p_n$ that has a non-zero resultant. This happens for any choice of n polynomials $p_1,\ldots,p_n$ with finitely many roots in $\mathbb{P}^{n_1}\times\mathbb{P}^{n_2}$, and a generic polynomial f_0. In particular we consider random coefficients for a set of monomials per equation that appear on a (permuted) diagonal of the matrix, and zero elsewhere. Then $R(f_0,p_1,\ldots,p_n)\neq 0$ implies that the matrices in Lem. 2.2 are non-singular.

We assume that a given polynomial system $f_1,\ldots,f_n$ has an infinite number of solutions. Now consider the perturbed system $\tilde{f}_k = f_k + \varepsilon p_k$, $k=1,\ldots,n$ augmented by f_0. The polynomial $C(\varepsilon) := R(\tilde{f}_0,\ldots,\tilde{f}_n)$ has degree equal to nD w.r.t. ε. By [13, Prop. 3.4] its leading coefficient is equal to $R(f_0,p_1,\ldots,p_n)$, therefore non-zero. Therefore, if we regard $C(\varepsilon)$ as a univariate polynomial in ε, there exists a non-zero trailing coefficient, which is a polynomial in the coefficients of f_0. This trailing term of $C(\varepsilon)$ has the same property as the unperturbed resultant in the zero-dimensional case. That is, the coefficient of the ε-power 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 ε, of this term, as we can perform the computations $\mod \varepsilon^d$.

EXAMPLE 4.2. *Consider the $(1,2)-$bilinear, square system $f_1 = 1+2\,y_2+2\,y_1+x_1+2\,x_1y_2+2\,x_1y_1$, $f_2 = 1+2\,y_2+y_1+2\,x_1+2\,x_1y_2+2\,x_1y_1$, $f_3 = 1+2\,y_2+2\,y_1+x_1+2\,x_1y_2+2\,x_1y_1$. 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 $\rho,\sigma\in\mathbb{R}$ plus the root $(x_0,x_1;y_0,y_1,y_2) = (0,1;0,-1,1)$ at infinity. We perturb the system by ε; 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 2.3, 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 of f_0. The (factored) coefficient of the trailing term $q_0\varepsilon^6$ is:*

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

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

Acknowledgments: The authors are grateful to the reviewers for their careful reading and corrections they brought the original draft. ET is partially supported by ANR-11-BS02-0013 HPAC project and an FP7 Marie Curie Career Integration Grant.

References

[1] C. J. Accettella, G. M. D. Corso, and G. Manzini. Inversion of two level circulant matrices over $\mathbb{Z}_p$. *Lin. Alg. Appl.*, 366:5 – 23, 2003.

[2] 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.

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

[4] A. Bostan, G. Lecerf, and E. Schost. Tellegen's principle into practice. In *Proc. ISSAC*, pages 37–44, 2003. ISBN 1-58113-641-2.

[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. 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. ISBN 1-58113-827-X.

[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. Emiris. Computing sparse projection operators. *Contemporary Mathematics*, 286:121–140, 2001.

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

[15] I. 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.

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

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

[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, B. Mourrain, and E. Tsigaridas. Separation bounds for polynomial systems. Technical report, INRIA, Dec. 2015.

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

[22] 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.

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

[24] 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.

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

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

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

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

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

[30] 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.

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

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

[33] 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.

[34] 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.

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

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

[37] 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.

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

[39] V. Y. Pan and E. Tsigaridas. Accelerated Approximation of the Complex Roots and Factors of a Univariate Polynomial. *Theoretical Computer Science*, 2015. (To appear).

[40] V. Y. Pan and E. Tsigaridas. Nearly optimal computations with structured matrices. *Theoretical Computer Science*, 2015. (To appear).

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

[42] 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.

[43] H. Schenck, D. Cox, and A. Dickenstein. A case study in bigraded commutative algebra. *Syzygies & Hilbert Functions*, 254:67–112, 2007.

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

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

[46] 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.

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

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

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

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