Computing Multihomogeneous Resultants Using Straight-Line Programs

Gabriela Jeronimo [*,1], Juan Sabia [1]

Departamento de Matemática, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Ciudad Universitaria, 1428 Buenos Aires, Argentina

---

Abstract

We present a new algorithm for the computation of resultants associated with multihomogeneous (and, in particular, homogeneous) polynomial equation systems using straight-line programs. Its complexity is polynomial in the number of coefficients of the input system and the degree of the resultant computed.

Key words: Sparse resultant, multihomogeneous system, Poisson-type product formula, symbolic Newton's algorithm.
MSC 2000: Primary: 14Q20, Secondary: 68W30.

---

1 Introduction

The resultant associated with a polynomial equation system with indeterminate coefficients is an irreducible multivariate polynomial in these indeterminates which vanishes when specialized in the coefficients of a particular system whenever it has a solution.

Resultants have been used extensively for the resolution of polynomial equation systems, particularly because of their role as eliminating polynomials. In the last years, the interest in the computation of resultants has been renewed not only because of their computational usefulness, but also because

[*] Corresponding author.
Email addresses: `jeronimo@dm.uba.ar` (Gabriela Jeronimo), `jsabia@dm.uba.ar` (Juan Sabia).
[1] Partially supported by the following Argentinian research grants: UBACyT X198 (2001-2003), UBACyT X112 (2004-2007) and CONICET PIP 02461/01.

they turned to be an effective tool for the study of complexity aspects of polynomial equation solving.

The study of classical homogeneous resultants goes back to Bézout, Cayley and Sylvester (see [1], [5] and [27]). In [20], Macaulay obtained explicit formulas for the classical resultant as a quotient of two determinants. More recently, Gelfand, Kapranov and Zelevinski generalized the classical notion to the sparse case (see [11]) and several effective procedures were proposed to compute classical and sparse resultants (see for instance [3], [4], [7], [8], [10], [25], [26]).

A particular case of sparse polynomial systems are the *multihomogeneous* systems; this means systems in which the set of variables can be partitioned into subsets so that every polynomial of the system is homogeneous in the variables of each subset. Multihomogeneous polynomial equation systems appear in several areas such as geometric modeling, game theory and computational economics. The problem of computing resultants for this subclass of polynomial systems was already considered by McCoy, who presented in [21] a formula involving determinants for the resultant of a multihomogeneous system. More recently, several results in this line of work have been obtained (see for instance [29], [9]).

Due to the well-known estimates for the degree of the resultant, any algorithm for the computation of resultants which encodes the output as an array of coefficients (dense form) cannot have a polynomial complexity in the size of the input (that is, the number of coefficients of the generic polynomial system whose resultant is computed). Then, in order to obtain these order of complexity, a different way of representing polynomials should be used. An alternative data structure which was introduced in the polynomial equation solving framework yielding a significant reduction in the previously known complexities is the *straight-line program* representation of polynomials (see for instance [13], [14]). Roughly speaking, a straight-line program which encodes a polynomial is a program which enables us to evaluate it at any given point.

The first algorithm for the computation of (homogeneous and) sparse resultants using straight-line programs was presented in [18]. Its complexity is polynomial in the dimension of the ambient space and the volume associated to the input set of exponents, but it deals only with a subclass of *unmixed* resultants.

In this paper we construct an algorithm for the computation of *arbitrary* multihomogeneous (and, in particular, homogeneous) resultants by means of straight-line programs. Its complexity is *polynomial* in the degree and the number of variables of the computed resultant. (See Theorem 5 for the precise statement of this result).

Our algorithm can be applied, in particular, to compute *any* classical homogeneous resultant. In this case, it can be seen as an extension of the one in [18, Corollary 4.1], which works only for polynomials of the same degree.

In the multihomogeneous case, the algorithm in [18, Corollary 4.2] can be applied to compute multihomogeneous resultants only when the multi-degrees of the polynomials coincide, and it is *probabilistic*. On the contrary, our algorithm can compute *any* multihomogeneous resultant and it is always *deterministic*. Furthermore, when computing unmixed multihomogeneous resultants, the complexity of our algorithm matches the expected complexity of the one in [18].

The paper is organized as follows:

In Section 2 we recall some basic definitions, fix the notation and describe the algorithmic model and data structures we will consider. We also introduce the main algorithmic tools that will be used. In Section 3 we first recall some elementary properties of multihomogeneous polynomial equation systems and we prove a Poisson-type formula for the multihomogeneous resultant. Applying this formula recursively, we obtain a product formula for the multihomogeneous resultant that enables us to derive an algorithm for its computation, which is the main result in Section 4.

2 Preliminaries

2.1 Definitions and Notation

Throughout this paper $\mathbb{Q}$ denotes the field of rational numbers, $\mathbb{N}$ denotes the set of positive integers and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.

If K is a field, we denote an algebraic closure of K by $\overline{K}$. The ring of polynomials in the variables $x_1, \ldots, x_n$ with coefficients in K is denoted by $K[x_1, \ldots, x_n]$. For a polynomial $f \in K[x_1, \ldots, x_n]$ we write $\deg f$ to refer to the total degree of f.

Let $r \in \mathbb{N}$ be a positive integer. Fix positive integers $n_1, \ldots, n_r$ and consider r groups of variables $X_j := (x_{j0}, \ldots, x_{j\,n_j})$, $j = 1, \ldots, r$. We say that the polynomial $F \in K[X_1, \ldots, X_r]$ is *multihomogeneous* of *multi-degree* $(v_1, \ldots, v_r)$, where $(v_1, \ldots, v_r)$ is a sequence of non-negative integers, if F is homogeneous of degree v_j in the group of variables X_j for every $1 \leq j \leq r$.

For $n \in \mathbb{N}$ and an algebraically closed field k, we denote by $\mathbb{A}^n(k)$ and $\mathbb{P}^n(k)$ (or simply by $\mathbb{A}^n$ or $\mathbb{P}^n$ if the base field is clear from the context) the n-

dimensional affine space and projective space over k respectively, equipped with their Zariski topologies. If $S \subset \mathbb{A}^n$, $\overline{S}$ denotes the closure of S with respect to the Zariski topology of $\mathbb{A}^n$.

We adopt the usual notions of dimension and degree of an algebraic variety V, which will be denoted by $\dim V$ and $\deg V$ respectively. See, for instance, [23] and [15] for the definitions of these notions.

2.2 Data Structures and Algorithmic Model

The algorithms we consider in this paper are described by arithmetic networks over the base field $\mathbb{Q}$ (see [28]). An arithmetic network is represented by means of a directed acyclic graph. The external nodes of the graph correspond to the input and output of the algorithm. Each of the internal nodes of the graph is associated with either an arithmetic operation in $\mathbb{Q}$ or a comparison ($=$ or $\neq$) between two elements in $\mathbb{Q}$ followed by a selection of another node. These are the only operations allowed in our algorithms.

We assume that the cost of each operation in the algorithm is 1 and so, we define the *complexity* of the algorithm as the number of internal nodes of its associated graph.

The objects our algorithm deals with are polynomials with coefficients in $\mathbb{Q}$. We represent each of them by means of one of the following data structures:

- *Dense form*, that is, as the array of all its coefficients (including zeroes) in a prefixed order of monomials. The size of this representation equals the number of coefficients of the polynomial.
- *Sparse encoding,* that is, as an array of the coefficients corresponding to monomials in a fixed set, provided that we know in advance that the coefficient of any other monomial of the polynomial must be zero. The size in this case is the cardinal of the fixed set of monomials.
- *Straight-line programs,* which are arithmetic circuits (i.e. networks without branches). Roughly speaking, a straight-line program over $\mathbb{Q}$ encoding a polynomial $f \in \mathbb{Q}[x_1, \dots, x_n]$ is a program which enables us to evaluate the polynomial f at any given point in $\mathbb{Q}^n$. Each of the instructions in this program is an addition, subtraction or multiplication between two precalculated elements in $\mathbb{Q}[x_1, \dots, x_n]$, or an addition or multiplication by a scalar. The number of instructions in the program is called the *length* of the straight-line program. For a precise definition of straight-line program we refer to [2, Definition 4.2] (see also [17]).

Let us remark that from the dense form of a polynomial it is straightforward to obtain a straight-line program encoding it. The length of this straight-line program is essentially the number of coefficients (including zeroes) of the polynomial.

We will deal with a particular class of sparse polynomials, which appear when dehomogenizing multihomogeneous polynomials. As in the previous case, we can provide estimates for the length of a straight-line program encoding the polynomial in terms of the number of its coefficients and of the number of groups of variables.

More precisely, using the notation of Section 2.1, let $F \in K[X_1, \dots, X_r]$ be a multihomogeneous polynomial of multi-degree $(v_1, \dots, v_r) \in \mathbb{N}_0^r$ given by the vector of all the coefficients of monomials of multi-degree $(v_1, \dots, v_r)$, and let $f \in K[X'_1, \dots, X'_r]$ be the polynomial obtained by specializing $x_{j\, n_j} = 1$ for $j = 1, \dots, r$, where $X'_j := (x_{j0}, \dots, x_{j\, n_j - 1})$. We can obtain a straight-line program encoding f as follows:
First, for $j = 1, \dots, r$, we compute a straight-line program of length $\binom{n_j+v_j}{v_j}$ whose result sequence is the set of all monomials of degree v_j in n_j variables. Then, for every $\alpha = (\alpha_1, \dots, \alpha_r)$ such that $\alpha_j \in \mathbb{N}_0^{n_j}$ and $|\alpha_j| \le v_j$ for $j = 1, \dots, r$, compute the monomial $a_\alpha \, {X'_1}^{\alpha_1} \dots {X'_r}^{\alpha_r}$, where a_α is the coefficient of this monomial in f. Each of these monomials is obtained with r products from the coefficients of f and the monomials computed in the previous step and so, the length of the straight-line program increases in $r \prod_{1 \le j \le r} \binom{n_j+v_j}{v_j}$. Finally, add all the monomials obtained in the second step in order to obtain the straight-line program encoding f. The length of this straight-line program is $\sum_{1 \le j \le r} \binom{n_j+v_j}{v_j} + (r+1) \prod_{1 \le j \le r} \binom{n_j+v_j}{v_j}$, that is, of order $O(rN)$, where $N := \prod_{1 \le j \le r} \binom{n_j+v_j}{v_j}$ denotes the number of coefficients of f.

2.3 Algorithmic Tools

The algorithms we construct in this paper rely on different subroutines dealing with polynomials encoded by straight-line programs. We describe in this section several procedures that will be used in the intermediate steps of our computations.

Our main algorithmic tool is a symbolic version of the Newton-Hensel algorithm for the approximation of zeroes of polynomial equation systems. We will describe the algorithm briefly in order to state the hypotheses needed for its application and to estimate its complexity. For a complete description of this procedure and a proof of its correctness we refer to [12] and [16]. See also [18] for a detailed statement in a context similar to ours.

Let $f_1, \dots, f_n \in \mathbb{Q}[T_1, \dots, T_N][x_1, \dots, x_n]$ be polynomials such that

$$W := \{f_1(\tau, x) = 0, \dots, f_n(\tau, x) = 0\} \subset \mathbb{A}^N \times \mathbb{A}^n$$

is an equidimensional variety of dimension N and the projection map $\pi : W \to \mathbb{A}^N$ is dominant. Let

$$\mathcal{D}_{\mathcal{F}} := \left(\frac{\partial f_i}{\partial x_j}\right)_{1 \le i,j \le n} \in \mathbb{Q}[T_1, \dots, T_N][x_1, \dots, x_n]^{n \times n}$$

be the Jacobian matrix of $\mathcal{F} := (f_1, \dots, f_n)$ with respect to the variables $x_1, \dots, x_n$, and let $\mathcal{J}_{\mathcal{F}} := \det(\mathcal{D}_{\mathcal{F}}) \in \mathbb{Q}[T_1, \dots, T_N][x_1, \dots, x_n]$ be the Jacobian determinant of the system.

Assume that for a point $t := (t_1, \dots, t_N) \in \mathbb{A}^N$, we have $\pi^{-1}(t) = \{t\} \times Z$, where Z is a 0-dimensional variety of cardinality

$$\delta := \max\{\#\pi^{-1}(\tau) : \tau \in \mathbb{A}^N \text{ and } \pi^{-1}(\tau) \text{ is finite}\}$$

such that $\mathcal{J}_{\mathcal{F}}(t, \xi) \neq 0$ for every $\xi \in Z$.

Set $K := \mathbb{Q}(T_1, \dots, T_N)$ and consider the variety

$$W^e := \{f_1(x) = 0, \dots, f_n(x) = 0\} \subset \mathbb{A}^n(\overline{K}),$$

which is a 0-dimensional variety of degree δ, since δ is the cardinality of the generic fiber of π.

Under the above conditions, the points in W^e can also be considered as power series vectors: the implicit function theorem implies that for every $\xi \in Z$, there exists a unique $\gamma_\xi \in \mathbb{C}[[T_1 - t_1, \dots, T_N - t_N]]^n$ such that

$$\gamma_\xi(t) = \xi \qquad \text{and} \qquad f_i(T_1, \dots, T_N, \gamma_\xi) = 0 \quad \forall\, 1 \le i \le n.$$

These power series vectors can be approximated by means of the Newton operator

$$\mathcal{N}_{\mathcal{F}}^T := x^T - \mathcal{D}_{\mathcal{F}}(x)^{-1}\, \mathcal{F}(x)^T \ \in K(x)^{n \times 1}$$

from the points in Z (see [16, Section 2]): if we set $\mathcal{N}_{\mathcal{F}}^{(m)}$ for the m-times iteration of $\mathcal{N}_{\mathcal{F}}$, for every $\xi \in Z$,

$$\mathcal{N}_{\mathcal{F}}^{(m)}(\xi) \equiv \gamma_\xi \qquad \mathrm{mod}\ (T_1 - t_1, \dots, T_N - t_N)^{2^m}.$$

Observe that $\mathcal{N}_{\mathcal{F}}$ is a vector of n rational functions in $K(x)$, and the same holds for $\mathcal{N}_{\mathcal{F}}^{(m)}$ for every $m \in \mathbb{N}$.

From the algorithmic point of view, we are interested in the computation of numerators and denominators for these rational functions. We denote by

`NumDenNewton` a procedure which computes polynomials $g_1^{(m)}, \ldots, g_n^{(m)}, h^{(m)}$ in $\mathbb{Q}[T_1, \ldots, T_N][x_1, \ldots, x_n]$ such that

$$\mathcal{N}_{\mathcal{F}}^{(m)} = (g_1^{(m)}/h^{(m)}, \ldots, g_n^{(m)}/h^{(m)}) \tag{1}$$

and $h^{(m)}(t, \xi) \neq 0$ for every $\xi \in Z$ (see [12, Lemma 30] and [18, Subroutine 5]).

If $f_1, \ldots, f_n \in \mathbb{Q}[T_1, \ldots, T_N][x_1, \ldots, x_n]$ are polynomials of respective degrees $d_1, \ldots, d_n$ in the variables $x_1, \ldots, x_n$, given by straight-line programs of length $L_1, \ldots, L_n$, following the proof of [12, Lemma 30], one can show that straight-line programs for the numerators and the denominator of $\mathcal{N}_{\mathcal{F}}^{(m)}$ can be computed within complexity $O(m\rho^2 n^2(n^3 + L))$, where $\rho := \sum_{1\le i\le n} d_i - n + 1$ and $L := \sum_{1\le i\le n} L_i$: Observe that the i-th coordinate of the Newton operator is the rational function

$$\frac{\mathcal{J}_{\mathcal{F}}\, x_i - \sum_{1\le j\le n} a_{ij} f_j}{\mathcal{J}_{\mathcal{F}}},$$

where (a_{ij}) is the adjoint matrix of $\mathcal{D}_{\mathcal{F}}$. It is easy to see that ρ is an upper bound for the degrees of the numerator and the denominator of these rational functions, which enables us to derive the complexity bound stated above.

A basic intermediate step in our algorithms consists in the approximation of determinants of certain linear maps, which is done by means of a subroutine based on the symbolic Newton procedure described above.

Let $f_1, \ldots, f_n$ be as before. Then, the ring $A := K[x_1, \ldots, x_n]/(f_1, \ldots, f_n)$ is a finite dimensional K-algebra. Given a polynomial $f \in K[x_1, \ldots, x_n]$ we will need to compute the determinant of the linear map $m_f : A \to A$ defined by $P \mapsto f \cdot P$, which is also called the *norm* of the polynomial f. In fact, we will not compute the exact value of this determinant, but we will approximate it as a power series as, under the previous assumptions, it turns out to be an element of $\mathbb{Q}[[T_1 - t_1, \ldots, T_N - t_N]]$. To do this we will use the identity $\det(m_f) = \prod_{\xi\in Z} f(\gamma_\xi)$ (see [6, Chapter 4, Proposition 2.7]), which enables us to approximate the norm by means of Newton's algorithm:

$$\det(m_f) \equiv \prod_{\xi\in Z} f(\mathcal{N}_{\mathcal{F}}^{(m)}(\xi)) \mod (T_1 - t_1, \ldots, T_N - t_N)^{2^m}.$$

Algorithmically, we compute this approximation from $f_1, \ldots, f_n, f$, the coordinates of the points $\xi \in Z$, and the precision needed as follows: In a first step we apply procedure `NumDenNewton` to obtain a straight-line program of length $\mathcal{L}_m := O(m\rho^2 n^2(n^3 + L))$ encoding a family of polynomials $g_1^{(m)}, \ldots, g_n^{(m)}, h^{(m)}$ satisfying (1). In order to avoid divisions, we consider the homogeneization F of the polynomial f, which we assume to be encoded by a straight-line program of length $\mathcal{L}'$. Then, we obtain a straight-line program of length $\mathcal{L}_m + \mathcal{L}'$

encoding the polynomial $\widetilde{F} := F(h^{(m)}, g_1^{(m)}, \ldots, g_n^{(m)})$. Now we compute the products

$$g := \prod_{\xi \in Z} \widetilde{F}(\xi) \quad \text{and} \quad h := \prod_{\xi \in Z} \left(h^{(m)}(\xi)\right)^{\deg f},$$

and the rational function g/h approximates $\det(m_f)$ in the power series ring $\mathbb{Q}[[T_1 - t_1, \ldots, T_N - t_N]]$ with precision 2^m. (Observe that g/h can be seen as a power series in $\mathbb{Q}[[T_1 - t_1, \ldots, T_N - t_N]]$ since $h(t) \neq 0$.) The complexity of the algorithm and the length of the straight-line programs encoding g and h are of order $O(\delta(\mathcal{L}_m + \mathcal{L}'))$. In the sequel, this procedure will be denoted by `ApproxNorm`.

Finally, we will apply an effective division procedure to approximate rational functions in appropriate power series rings. This procedure is based on the well-known Strassen's algorithm for Vermeidung von Divisionen (see [24]) for the computation of quotients of polynomials. More precisely, given polynomials g and h in $\mathbb{Q}[T_1, \ldots, T_N]$ and a point $t := (t_1, \ldots, t_N)$ such that $h(t) \neq 0$, the rational function g/h can be regarded as an element of $\mathbb{Q}[[T_1 - t_1, \ldots, T_N - t_N]]$. There is an algorithm, which we will denote by `GradedParts`, that computes all the graded parts (centered at t) of g/h of degrees bounded by D within complexity $O(D^2(D + L))$ for a fixed $D \in \mathbb{N}$ from straight-line programs of length bounded by L encoding g and h. For a description of this algorithm and a proof of the estimates for its complexity we refer to [18, Section 1.4].

3 The Multihomogeneous Setting

This section deals with systems of multihomogeneous polynomials, that is, polynomials in several groups of variables which are homogeneous in the variables of each group.

First, certain properties of multihomogenous polynomial equation systems are discussed. Then, we give the precise definition of multihomogeneous resultant. Finally, we prove an analogue of the classical Poisson formula (see for instance [20], [19, Proposition 2.7]) in the multihomogeneous setting.

3.1 Notation

Here we are going to fix some notation related to multihomogeneous polynomial systems that will be used in the sequel.

Let K be a field of characteristic 0. Let $n_1, \ldots, n_r \in \mathbb{N}$ and let $X_1, \ldots, X_r$ be

r groups of indeterminates over the field K such that $X_j := (x_{j0}, \ldots, x_{j\,n_j})$ for every $1 \leq j \leq r$. Set $n := n_1 + \cdots + n_r$.

Given a vector $v = (v_1, \ldots, v_r) \in \mathbb{N}_0^r$ we denote by

$$M(v) := \{(\alpha_1, \ldots, \alpha_r) \in \mathbb{N}_0^{n_1+1} \times \cdots \times \mathbb{N}_0^{n_r+1} : |\alpha_j| = v_j\}$$

the set of exponents of all the monomials of multi-degree v in the groups of variables $X_1, \ldots, X_r$.

Fix vectors $d_0, \ldots, d_n \in \mathbb{N}_0^r$ with $d_i := (d_{i1}, \ldots, d_{ir})$ for every $0 \leq i \leq n$. We introduce $n+1$ groups of new indeterminates $U_0, \ldots, U_n$ over $K[X_1, \ldots, X_r]$, where, for every $0 \leq i \leq n$, $U_i := (U_{i,\alpha})_{\alpha \in M(d_i)}$ is a vector of $N_i := \#M(d_i)$ coordinates. We denote by $F_0, \ldots, F_n$ the following family of $n+1$ generic multihomogeneous polynomials of multi-degrees $d_0, \ldots, d_n$ respectively:

$$F_i := \sum_{\alpha \in M(d_i)} U_{i,\alpha}\, X^\alpha \qquad i = 0, \ldots, n. \tag{2}$$

3.2 Multihomogeneous Polynomial Systems

The classical Multihomogeneous Bézout Theorem, which follows from the intersection theory for divisors (see for instance [23, Chapter 4]), states that the set of common zeroes of n generic multihomogeneous polynomials $F_1, \ldots, F_n$ as in (2) in the projective variety $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$ over an algebraic closure of the field $K(U_1, \ldots, U_n)$ is a zero-dimensional variety with

$$\mathrm{Bez}_{n_1,\ldots,n_r}(d_1, \ldots, d_n) := \sum \prod_{1 \leq j \leq r} d_{i_1^{(j)} j} \cdots d_{i_{n_j}^{(j)} j} \tag{3}$$

points, where the sum is taken over all those families of indices such that

- $1 \leq i_1^{(j)} < \cdots < i_{n_j}^{(j)} \leq n$ for every $1 \leq j \leq r$,
- $\#\Big(\bigcup_{1 \leq j \leq r} \{i_1^{(j)}, \ldots, i_{n_j}^{(j)}\} \Big) = n.$

From the algorithmic point of view it will be useful to consider the coordinates of these points as power series in an appropriate ring:

Proposition 1 *Under the previous assumptions, there exists $(u_1, \ldots, u_n) \in K^{N_1 + \cdots + N_n}$ such that every common zero of $F_1, \ldots, F_n$ over an algebraic closure of $K(U_1, \ldots, U_n)$ is a vector of power series in $K[[U_1 - u_1, \ldots, U_n - u_n]]$.*

PROOF. The idea is to apply the implicit function theorem in the same way as we did in Section 2.3.

For every $1 \leq j \leq r$, take a family of elements $a_{ik}^{(j)} \in K - \{0\}$, for $1 \leq i \leq n$ and $1 \leq k \leq d_{ij}$, such that $a_{i_1k_1}^{(j)} \neq a_{i_2k_2}^{(j)}$ if $i_1 \neq i_2$ or $k_1 \neq k_2$. For each $a_{ik}^{(j)}$ consider the associated linear form in the variables X_j:

$$L_{ik}^{(j)} := x_{j0} + a_{ik}^{(j)} x_{j1} + (a_{ik}^{(j)})^2 x_{j2} + \cdots + (a_{ik}^{(j)})^{n_j} x_{jn_j}.$$

For each index i, $1 \leq i \leq n$, we consider the multihomogeneous polynomial of multi-degree $d_i = (d_{i1}, \ldots, d_{ir})$

$$\prod_{1 \leq j \leq r} \prod_{1 \leq k \leq d_{ij}} L_{ik}^{(j)} \tag{4}$$

and we denote by $u_i \in K^{N_i}$ the vector of coefficients of its monomials of multi-degree d_i in a certain prefixed order.

We have the identity:

$$F_i(u_i, X_1, \ldots, X_r) = \prod_{1 \leq j \leq r} \prod_{1 \leq k \leq d_{ij}} L_{ik}^{(j)}. \tag{5}$$

The hypothesis on the choice of the elements $a_{ik}^{(j)}$ implies that for a fixed j, $1 \leq j \leq r$, every subset of n_j many linear forms $L_{ik}^{(j)}$ is a linearly independent set and so, it has a unique solution in $\mathbb{P}^{n_j}$. Moreover, any subset with more than n_j of these linear forms does not have a common solution in $\mathbb{P}^{n_j}$. We conclude that the system

$$F_1(u_1, X_1, \ldots, X_r) = 0, \ldots, F_n(u_n, X_1, \ldots, X_n) = 0 \tag{6}$$

has exactly $\mathrm{Bez}_{n_1,\ldots,n_r}(d_1, \ldots, d_n)$ solutions in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$, which are precisely the solutions to the linear systems

$$L^{(1)}_{i_1^{(1)}\,k_1^{(1)}} = 0, \ldots, L^{(1)}_{i_{n_1}^{(1)}\,k_{n_1}^{(1)}} = 0, \;\ldots,\; L^{(r)}_{i_1^{(r)}\,k_1^{(r)}} = 0, \ldots, L^{(r)}_{i_{n_r}^{(r)}\,k_{n_r}^{(r)}} = 0,$$

where

- $1 \leq i_1^{(j)} < \cdots < i_{n_j}^{(j)} \leq n$ for every $1 \leq j \leq r$,
- $\#\Big(\bigcup_{1 \leq j \leq r} \{i_1^{(j)}, \ldots, i_{n_j}^{(j)}\}\Big) = n$,
- $1 \leq k_l^{(j)} \leq d_{i_l^{(j)} j}$.

Since every solution to this system satisfies $x_{j\,n_j} \neq 0$ for every $1 \leq j \leq r$, we will deal with the dehomogenized polynomials (setting $x_{jn_j} = 1$ for every $1 \leq j \leq r$) and their common zero locus in the affine space $\mathbb{A}^n$.

For every $1 \leq j \leq r$, let $X'_j := (x_{j0}, \ldots, x_{j\,n_j - 1})$, and let $X' := (X'_1, \ldots, X'_r)$. We denote by $\mathcal{F} := (f_1, \ldots, f_n)$ the system of generic dehomogenized polynomials

$$f_i := F_i((x_{10}, \dots, x_{1\,n_1-1}, 1), \dots, (x_{r0}, \dots, x_{r\,n_r-1}, 1)) \qquad i = 1, \dots, n.$$

Consider the incidence variety

$$W := \{(\nu_1, \dots, \nu_n, x) : f_1(\nu_1, x) = 0, \dots, f_n(\nu_n, x) = 0\} \subset \mathbb{A}^{N_1 + \cdots N_n} \times \mathbb{A}^n$$

and the projection $\pi : (\nu_1, \dots, \nu_n, x) \mapsto (\nu_1, \dots, \nu_n)$, which is a dominant map of degree $\mathrm{Bez}_{n_1,\dots,n_r}(d_1, \dots, d_n)$ due to the multihomogeneous Bézout theorem. Let $\mathcal{J}_{\mathcal{F}} \in K[U_1, \dots, U_n][X']$ be the Jacobian determinant of the system $\mathcal{F}$ with respect to the variables in X'.

As a consequence of the construction of the polynomials considered in (5), the specialized system $f_1(u_1, X') = 0, \dots, f_n(u_n, X') = 0$ of dehomogenized polynomials has maximal number of solutions, and it is not difficult to see that for every solution $\xi \in \mathbb{A}^n$ to this system we have

$$\mathcal{J}_{\mathcal{F}}(u_1, \dots, u_n, \xi) \neq 0.$$

Therefore, $\pi^{-1}(u_1, \dots, u_n)$ satisfies the hypotheses stated in Section 2.3.

Then, for every solution ξ to the particular system there exists a solution γ_ξ to the generic system $\mathcal{F}$ which is a vector whose coordinates are well defined power series in $K[[U_1 - u_1, \dots, U_n - u_n]]$ and satisfies $\gamma_\xi(u_1, \dots, u_n) = \xi$. Finally, let us observe that the points γ_ξ are *all* the solutions to the system (2). $\square$

From the previous proof and the arguments in Section 2.3 we deduce:

Remark 2 *The coordinates of the solutions to the system (2) can be approximated in $K[[U_1 - u_1, \dots, U_n - u_n]]$ from the solutions of the particular system (6) by means of the Newton operator.*

3.3 The Multihomogeneous Resultant

The multihomogeneous resultant extends the classical notion of resultant (associated with a system of homogeneous polynomials) to the multihomogeneous setting. It can also be regarded as a particular case of the well-known sparse resultant (see for instance [11]).

Let $F_0, \dots, F_n \in \mathbb{Q}(U_0, \dots, U_n)[X_1, \dots, X_r]$ be generic multihomogeneous polynomials of multi-degree $d_0, \dots, d_n$ respectively, as defined in (2) of Section 3.1.

The *multihomogeneous resultant* $\mathrm{Res}_{(n_1,\dots,n_r),(d_0,\dots,d_n)}$ of the $n+1$ polynomials $F_0,\dots,F_n$ is an irreducible polynomial with coefficients in $\mathbb{Z}$ in the variables $U_{i,\alpha}$ ($0\le i\le n$, $\alpha\in M(d_i)$) which vanishes at a point $(u_0,\dots,u_n)\in\mathbb{P}^{n_1}(k)\times\cdots\times\mathbb{P}^{n_r}(k)$ —where k is an algebraically closed field— if and only if the polynomials $F_0(u_0,X),\dots,F_n(u_n,X)$ have a common root in $\mathbb{P}^{n_1}(k)\times\cdots\times\mathbb{P}^{n_r}(k)$.

More precisely, for every $0\le i\le n$, let N_i be the number of coefficients of F_i and set $N_i':=N_i-1$. Let $W\subset\mathbb{P}^{N_0'}\times\cdots\times\mathbb{P}^{N_n'}\times\mathbb{P}^{n_1}\times\cdots\times\mathbb{P}^{n_r}$ be the incidence variety

$$W:=\{(u_0,\dots,u_n,\xi_1,\dots,\xi_r):F_i(u_i,\xi_1,\dots,\xi_r)=0\ \forall\,0\le i\le n\}.$$

The image of W under the canonical projection $\pi:W\to\mathbb{P}^{N_0'}\times\cdots\times\mathbb{P}^{N_n'}$ is an irreducible hypersurface in $\mathbb{P}^{N_0'}\times\ldots\times\mathbb{P}^{N_n'}$ and so, it is the zero locus of an irreducible polynomial. The *multihomogeneous resultant* $\mathrm{Res}_{(n_1,\dots,n_r),(d_0,\dots,d_n)}$ is defined as an irreducible equation for $\pi(W)$. This polynomial may be chosen with integer coefficients and it is uniquely defined —up to sign— by the additional requirement that it has relatively prime coefficients. Furthermore, it is homogeneous in the coefficients U_i of each polynomial F_i, for $0\le i\le n$, and its degree in the group of variables U_i is the corresponding multihomogeneous Bézout number

$$\deg_{U_i}\mathrm{Res}_{(n_1,\dots,n_r),(d_0,\dots,d_n)}=\mathrm{Bez}_{n_1,\dots,n_r}(d_0,\dots,d_{i-1},d_{i+1},\dots,d_n)\qquad(7)$$

which controls the number of solutions of a multihomogeneous polynomial equation system (see Section 3.2).

When the number of variables and degrees are clear from the context, we will denote the resultant $\mathrm{Res}_{(n_1,\dots,n_r),(d_0,\dots,d_n)}$ associated with the generic polynomials $F_0,\dots,F_n$ simply by $\mathrm{Res}(F_0,\dots,F_n)$.

3.4 A Poisson-Type Formula

Here, we present a Poisson-type product formula for the multihomogeneous resultant which generalizes the well-known Poisson formula for the homogeneous case, providing us with a recursive description of the resultant in the multihomogeneous setting. This formula can be regarded as an instance of the product formula stated by Pedersen-Sturmfels in [22]. However, the proof we give in this paper is elementary and so, we include it for the sake of completeness.

We keep the notation defined in Section 3.1.

Before stating the product formula, we introduce some extra notation that will be used throughout this section.

We denote by

$$f_i := F_i((x_{10}, \dots, x_{1\,n_1-1}, 1), \dots, (x_{r0}, \dots, x_{r\,n_r-1}, 1)) \tag{8}$$

and, for every $1 \le j \le r$,

$$\overline{F}_{ij} := F_i(X_1, \dots, X_{j-1}, (x_{j0}, \dots, x_{j\,n_j-1}, 0), X_{j+1}, \dots, X_n). \tag{9}$$

Let m_{f_n} be the linear map defined in the 0-dimensional $\mathbb{Q}(U_0, \dots, U_n)$-algebra $\mathbb{Q}(U_0, \dots, U_n)[X'_1, \dots, X'_r]/(f_0, \dots, f_{n-1})$ by multiplication by f_n, where X'_j denotes the group of variables $X'_j := (x_{j0}, \dots, x_{j\,n_j-1})$ for every $1 \le j \le r$.

Proposition 3 *Let notation and assumptions be as before. Then, the following identity holds in $\mathbb{Q}(U_0, \dots, U_n)$:*

$$\mathrm{Res}(F_0, \dots, F_n) = \det(m_{f_n}) \cdot \prod_{1 \le j \le r} \Big(\mathrm{Res}(\overline{F}_{0j}, \dots, \overline{F}_{n-1\,j})\Big)^{d_{nj}}.$$

In order to prove this proposition, we first show an auxiliary multiplicative formula for the multihomogeneous resultant (see [19, Section 5.7] for an analogous formula in the homogeneous case):

Lemma 4 *Let $F_0, \dots, F_{n-1}$ be generic multihomogeneous polynomials with multi-degrees $d_0, \dots, d_{n-1}$ respectively. Let $d_n := (d_{n1}, \dots, d_{nr})$ be a vector of non-negative integers and, for $j = 1, \dots, r$, let $H_j(X_j)$ be a generic homogeneous polynomial of degree d_{nj} in the variables X_j. Then, the following identity holds:*

$$\mathrm{Res}\Big(F_0, \dots, F_{n-1}, \prod_{1 \le j \le r} H_j\Big) = \prod_{1 \le j \le r} \mathrm{Res}\Big(F_0, \dots, F_{n-1}, H_j\Big).$$

PROOF. By the definition of the resultant, $\mathrm{Res}(F_0, \dots, F_{n-1}, \prod_{1 \le j \le r} H_j)(u)$ vanishes if and only if the system

$$F_0(u) = 0, \dots, F_{n-1}(u) = 0, \prod_{1 \le j \le r} H_j(u) = 0$$

has a root in $\mathbb{X} := \mathbb{P}^{n_1} \times \dots \times \mathbb{P}^{n_r}$ or, equivalently, for some j with $1 \le j \le r$, the system $F_0(u) = 0, \dots, F_{n-1}(u) = 0, H_j(u) = 0$ has a common root in $\mathbb{X}$.

But the condition that $F_0(u), \dots, F_{n-1}(u), H_j(u)$ have a common root in $\mathbb{X}$ is given by the vanishing of the resultant $\mathrm{Res}(F_0, \dots, F_{n-1}, H_j)$ in u. Since these resultants are irreducible polynomials for $1 \leq j \leq r$, we conclude that the irreducible factors of $\mathrm{Res}(F_0, \ldots, F_{n-1}, \prod_{1\leq j\leq r} H_j)$ are exactly the r multihomogeneous resultants $\mathrm{Res}(F_0, \ldots, F_{n-1}, H_j)$ for $1 \leq j \leq r$, and so, there exist $a_1, \ldots, a_r \in \mathbb{N}$ such that

$$\mathrm{Res}\Big(F_0, \ldots, F_{n-1}, \prod_{1\leq j\leq r} H_j\Big) = \prod_{1\leq j\leq r} \mathrm{Res}(F_0, \ldots, F_{n-1}, H_j)^{a_j}. \qquad (10)$$

It remains to be shown that $a_j = 1$ for $1 \leq j \leq r$. This follows easily by comparing the degrees in the variable coefficients of $H_1, \ldots, H_r$ of the polynomials involved in both sides of identity (10): the degree of the resultant $\mathrm{Res}(F_0, \ldots, F_{n-1}, F_n)$ in the coefficients of the generic polynomial F_n of multidegree d_n is the Bézout number $\mathrm{Bez}_{n_1,\ldots,n_r}(d_0, \ldots, d_{n-1})$. Then, the polynomial $\mathrm{Res}(F_0, \ldots, F_{n-1}, \prod_{1\leq j\leq r} H_j)$ has degree $r\,\mathrm{Bez}_{n_1,\ldots,n_r}(d_0, \ldots, d_{n-1})$ in the coefficients of the polynomials $H_1, \ldots, H_r$, for each coefficient of $\prod_{1\leq j\leq r} H_j$ is a product of r variables. But this degree coincides with the sum of the degrees of all the irreducible factors $\mathrm{Res}(F_0, \ldots, F_{n-1}, H_j)$, $1 \leq j \leq r$, which implies that the exponent a_j equals 1 for every $1 \leq j \leq r$. $\square$

Now we are ready to prove Proposition 3:

PROOF. (Proof of Proposition 3). Let $f_0, \ldots, f_n$ be the generic polynomials defined in (8) and set N for the number of their coefficients. Consider the incidence variety associated with these polynomials

$$W_{\mathrm{af}} := \{(u_0, \ldots, u_n, \xi) \in \mathbb{A}^N \times \mathbb{A}^n : f_i(u_i, \xi) = 0 \;\forall\, 0 \leq i \leq n\}$$

and the canonical projection $\pi : \mathbb{A}^N \times \mathbb{A}^n \to \mathbb{A}^N$ into the first coordinates. Then, the multihomogeneous resultant $\mathrm{Res}(F_0, \ldots, F_n)$ can be alternatively defined as the unique —up to scalar factors— polynomial defining the Zariski closure $\overline{\pi(W_{\mathrm{af}})}$, which is an irreducible hypersurface in $\mathbb{A}^N$. Therefore, by elementary elimination theory, the following identity of ideals holds:

$$(\mathrm{Res}(F_0, \ldots, F_n)) = (f_0, \ldots, f_n) \cap \mathbb{Q}[U_0, \ldots, U_n].$$

Therefore,

$$(\mathrm{Res}(F_0, \ldots, F_n))\,.K[U_n] = \Big((f_0, \ldots, f_n).K[U_n][X_1', \ldots, X_r']\Big) \cap K[U_n], \quad (11)$$

where $K := \mathbb{Q}(U_0, \ldots, U_{n-1})$.

The ideal appearing on the right hand side of identity (11) can also be regarded as an eliminating ideal: Let N_n be the number of coefficients of f_n and let

$$W_{\mathrm{af}}^e := \{(u_n, \xi) \in \mathbb{A}^{N_n}(\overline{K}) \times \mathbb{A}^n(\overline{K}) : f_i(\xi) = 0 \;\forall\, 0 \leq i \leq n-1,\; f_n(u_n, \xi) = 0\}.$$

Let π^e be the canonical projection into the first N_n coordinates. As before, the defining ideal of $\overline{\pi^e(W^e_{\mathrm{af}})}$ is $((f_0,\dots,f_n).K[U_n][X'_1,\dots,X'_r]) \cap K[U_n]$, which is the one appearing in the right hand side of (11).

On the other hand, we have that $V(f_0,\dots,f_{n-1}) := \{\xi \in \mathbb{A}^n : f_i(\xi) = 0 \ \forall\, 0 \le i \le n-1\}$ is a zero-dimensional variety and, therefore, the ideal of $\overline{\pi^e(W^e_{\mathrm{af}})}$ is generated by the polynomial $\prod_{\xi \in V(f_0,\dots,f_{n-1})} f_n(U_n,\xi) \in K[U_n]$, which under our generic conditions equals the determinant $\det(m_{f_n})$ of the multiplication by f_n in $K(U_n)[X'_1,\dots,X'_r]/(f_0,\dots,f_{n-1})$.

Then, it follows that there exists an element $\lambda \in \mathbb{Q}(U_0,\dots,U_{n-1}) - \{0\}$ such that

$$\mathrm{Res}(F_0,\dots,F_n) = \det(m_{f_n}) \cdot \lambda. \tag{12}$$

In particular, specializing the variables U_n into the coefficients of the polynomial $x_{1n_1}^{d_{n1}} \cdots x_{rn_r}^{d_{nr}}$ we obtain the identity $\lambda = \mathrm{Res}(F_0,\dots,F_{n-1}, x_{1n_1}^{d_{n1}} \cdots x_{rn_r}^{d_{nr}})$ and we deduce that $\lambda \in \mathbb{Q}[U_0,\dots,U_{n-1}]$ is a polynomial.

Applying Lemma 4, we conclude that λ factors as the following product of specialized resultants:

$$\lambda = \prod_{1 \le j \le r} \mathrm{Res}(F_0,\dots,F_{n-1}, x_{jn_j}^{d_{nj}}).$$

Adapting the proof of Lemma 4, we can easily obtain that, for every $1 \le j \le r$,

$$\mathrm{Res}(F_0,\dots,F_{n-1}, x_{jn_j}^{d_{nj}}) = \mathrm{Res}(\overline{F}_{0j},\dots,\overline{F}_{n-1\,j})^{d_{nj}}$$

and so,

$$\lambda = \prod_{1 \le j \le r} \mathrm{Res}(\overline{F}_{0j},\dots,\overline{F}_{n-1\,j})^{d_{nj}}. \tag{13}$$

The Poisson formula stated in the Proposition follows from (12) and (13). $\square$

4 Computing Multihomogeneous Resultants

This section is devoted to the description and complexity analysis of our algorithm for the computation of multihomogeneous resultants. In order to construct this algorithm, we are going to use the formula stated in Proposition 3 recursively.

Our main result is the following:

Theorem 5 *Let $n_1, \dots, n_r \in \mathbb{N}$ and set $n := n_1 + \dots + n_r$. Fix vectors $d_0, \dots, d_n \in \mathbb{N}_0^r$. Let*

$$D := \sum_{0 \le i \le n} \mathrm{Bez}_{n_1,\dots,n_r}(d_0, \dots, \hat{d_i}, \dots, d_n),$$

$$\delta := \mathrm{Bez}_{n_1,\dots,n_r}(d_0, \dots, d_{n-1}),$$

$$\rho := \sum_{0 \le i \le n-1} |d_i| - n + 1,$$

$$N := \sum_{0 \le i \le n} \prod_{1 \le j \le r} \binom{n_j + d_{ij}}{d_{ij}}.$$

Then, there exists a straight-line program of length

$$O(D^2(D + n_1 \dots n_r \delta \log(D) \rho^2 n^2 (n^3 + rN)))$$

which encodes (a scalar multiple of) $\mathrm{Res}_{(n_1,\dots,n_r),(d_0,\dots,d_n)}$, the multihomogeneous resultant of $n+1$ multihomogeneous polynomials of respective multi-degrees $d_0, \dots, d_n$ in r groups of $n_1 + 1, \dots, n_r + 1$ variables respectively. Moreover, this straight-line program can be obtained algorithmically within complexity $O(D^2(D + n_1 \dots n_r \delta \log(D) \rho^2 n^2 (n^3 + rN)))$.

In particular, this theorem provides an algorithm for the computation of classical resultants of homogeneous polynomial systems:

Remark 6 *A straight-line program for the resultant $\mathrm{Res}_{d_0,\dots,d_n}$ of $n+1$ homogeneous polynomials in $n+1$ variables of respective degrees $d_0, \dots, d_n$ can be computed within complexity*

$$O(D^2(D + \delta \log(D) \rho^2 n^3 (n^3 + N))),$$

where $D =: \sum_{0 \le i \le n} d_0 \dots \hat{d_i} \dots d_n$, $\delta := d_0 \dots d_n$, $\rho := \sum_{0 \le i \le n} d_i - n + 1$ and $N := \sum_{0 \le i \le n} \binom{d_i + n}{n}$. The length of this straight-line program is of order $O(D^2(D + \delta \log(D) \rho^2 n^3 (n^3 + N)))$.

Now we prove the theorem.

PROOF. (Proof of Theorem 5.) Before stating the formula that will allow us to compute the desired resultant, we are going to fix some notation.

Let $F_0, \dots, F_n \in \mathbb{Q}(U_0, \dots, U_n)[X_1, \dots, X_r]$ be generic multihomogeneous polynomials as in (2).

For an integer vector $(k_1, \dots, k_r) \in \mathbb{N}_0^r$ such that $0 \le k_j \le n_j$ for every $1 \le j \le r$, given any multihomogeneous polynomial H in the groups of variables

$X_1, \ldots, X_r$, we define the associated polynomial $h^{(k_1,\ldots,k_r)}$ as the one we obtain by specializing in H the variables $x_{j\ell} = 0$ for $1 \leq j \leq r$ and $n_j - k_j + 1 \leq \ell \leq n_j$, and the variables $x_{j\, n_j - k_j} = 1$ for $1 \leq j \leq r$ (note that this specialization is denoted both by the vector superindex and by the change from capital to lower case letter).

We also introduce the following notation for sets of variables, where $\kappa := n - |(k_1, \ldots, k_r)|$:

$$\begin{aligned}
U^{(k_1,\ldots,k_r)} &:= \bigcup_{0 \leq i \leq \kappa - 1} \{U_{i,\alpha} : |\alpha_j| = d_{ij}, \alpha_{j\ell} = 0 \text{ for } \ell = n_j - k_j + 1, \ldots, n_j; 1 \leq j \leq r\}, \\
\widehat{U}^{(k_1,\ldots,k_r)} &:= \bigcup_{0 \leq i \leq \kappa} \{U_{i,\alpha} : |\alpha_j| = d_{ij}, \alpha_{j\ell} = 0 \text{ for } \ell = n_j - k_j + 1, \ldots, n_j; 1 \leq j \leq r\}, \\
X^{(k_1,\ldots,k_r)} &:= \bigcup_{1 \leq j \leq r} \{x_{j\ell} : 0 \leq \ell \leq n_j - k_j - 1\}.
\end{aligned}$$

Finally, we consider the polynomials $f_0^{(k_1,\ldots,k_r)}, \ldots, f_{\kappa-1}^{(k_1,\ldots,k_r)}$ obtained after the polynomials $F_0, \ldots, F_{\kappa-1}$ according to our notation. Let

$$A^{(k_1,\ldots,k_r)} := \mathbb{Q}(\widehat{U}^{(k_1,\ldots,k_r)})[X^{(k_1,\ldots,k_r)}]/(f_0^{(k_1,\ldots,k_r)}, \ldots, f_{\kappa-1}^{(k_1,\ldots,k_r)})$$

and let

$$m_{f_\kappa^{(k_1,\ldots,k_r)}} : A^{(k_1,\ldots,k_r)} \to A^{(k_1,\ldots,k_r)} \tag{14}$$

be the linear map given by multiplication by $f_\kappa^{(k_1,\ldots,k_r)}$.

Applying Proposition 3 recursively, we obtain a formula for the multihomogeneous resultant involving the determinants of the linear maps defined in (14):

$$\mathrm{Res}_{(n_1,\ldots,n_r),(d_0,\ldots,d_n)} = U_{0,\alpha(0)}^{e(n_1,\ldots,n_r)} \prod_{\substack{1 \leq \kappa \leq n \\ |(k_1,\ldots,k_r)| = n - \kappa,\, 0 \leq k_j \leq n_j}} \Big(\det(m_{f_\kappa{}^{(k_1,\ldots,k_r)}})\Big)^{e(k_1,\ldots,k_r)}.$$

Here, $\alpha(0) := ((d_{01}, 0, \ldots, 0), \ldots, (d_{0r}, 0, \ldots, 0))$, and for every $(k_1, \ldots, k_r)$ with $0 \leq k_j \leq n_j$ $(1 \leq j \leq r)$, if $|(k_1, \ldots, k_r)| = n - \kappa$,

$$e(k_1, \ldots, k_r) := \sum \prod_{1 \leq l \leq n - \kappa} d_{n-l+1\, j_l}, \tag{15}$$

where the sum runs over the vectors $(j_1, \ldots, j_{n-\kappa})$ satisfying $\#\{t / j_t = j\} = k_j$ for every $1 \leq j \leq r$.

So, to compute the desired resultant it would suffice to compute the exponents and the determinants involved in the previous formula.

The first step of the algorithm consists in the computation of straight-line programs for approximations to these determinants in a suitable power series ring.

For every $1 \leq i \leq n$ let

$$G_{i-1} := \prod_{1\leq j\leq r} \prod_{1\leq k\leq d_{ij}} L_{ik}^{(j)} \in \mathbb{Q}[X_1, \ldots, X_r] \tag{16}$$

as defined in (4).

Let $(k_1, \ldots, k_r) \in \mathbb{N}_0^r$ be such that $0 \leq k_j \leq n_j$ $(1 \leq j \leq r)$. Consider the polynomials $f_0^{(k_1,\ldots,k_r)}, \ldots, f_{\kappa-1}^{(k_1,\ldots,k_r)}$ in $\mathbb{Q}[U^{(k_1,\ldots,k_r)}][X^{(k_1,\ldots,k_r)}]$ where $\kappa = n - |(k_1, \ldots, k_r)|$ and the variety

$$W^{(k_1,\ldots,k_r)} := \{f_0^{(k_1,\ldots,k_r)} = 0, \ldots, f_{\kappa-1}^{(k_1,\ldots,k_r)} = 0\} \subset \mathbb{A}^{N^{(k_1,\ldots,k_r)}} \times \mathbb{A}^{\kappa}$$

where $N^{(k_1,\ldots,k_r)}$ is the number of variables in $U^{(k_1,\ldots,k_r)}$.

We consider the polynomials $g_0^{(k_1,\ldots,k_r)}, \ldots, g_{\kappa-1}^{(k_1,\ldots,k_r)}$ defined after $G_0, \ldots, G_{\kappa-1}$, and the zero-dimensional variety

$$Z^{(k_1,\ldots,k_r)} := \{g_0^{(k_1,\ldots,k_r)} = 0, \ldots, g_{\kappa-1}^{(k_1,\ldots,k_r)} = 0\} \subset \mathbb{A}^{\kappa}.$$

Let $u^{(k_1,\ldots,k_r)} \in \mathbb{A}^{N^{(k_1,\ldots,k_r)}}$ be the vector of coefficients of the polynomial system defining $Z^{(k_1,\ldots,k_r)}$.

We are exactly under the hypotheses stated in Section 3.2. Therefore, the determinant $\det(m_{f_\kappa^{(k_1,\ldots,k_r)}})$ is an element of $\mathbb{Q}[[U^{(k_1,\ldots,k_r)} - u^{(k_1,\ldots,k_r)}]][U_{\kappa,\alpha}]$ and Newton's algorithm applied to the system $f_0^{(k_1,\ldots,k_r)}, \ldots, f_{\kappa-1}^{(k_1,\ldots,k_r)}$ allows us to approximate $\det(m_{f_\kappa^{(k_1,\ldots,k_r)}})$ (see Proposition 1 and Remark 2). Then, we can obtain polynomials $g_{(k_1,\ldots,k_r)} \in \mathbb{Q}[U^{(k_1,\ldots,k_r)}][U_{\kappa,\alpha}]$ and $h_{(k_1,\ldots,k_r)} \in \mathbb{Q}[U^{(k_1,\ldots,k_r)}]$ with $h_{(k_1,\ldots,k_r)}(u^{(k_1,\ldots,k_r)}) \neq 0$ such that the rational function $g_{(k_1,\ldots,k_r)}/h_{(k_1,\ldots,k_r)}$ approximates the desired determinant up to degree D, which is the total degree of $\mathrm{Res}_{(n_1,\ldots,n_r),(d_0,\ldots,d_n)}$ (see (7)).

Note that all the determinants considered are in $\mathbb{Q}[[U^{(0,\ldots,0)} - u^{(0,\ldots,0)}]][U_{n,\alpha}]$.

Now we obtain straight-line programs for the polynomials

$$g := \prod_{(k_1,\ldots,k_r),\, 0\leq k_j\leq n_j} \Big(g_{(k_1,\ldots,k_r)})\Big)^{e(k_1,\ldots,k_r)} \quad \text{and} \tag{17}$$

$$h := \prod_{(k_1,\ldots,k_r),\, 0\leq k_j\leq n_j} \Big(h_{(k_1,\ldots,k_r)})\Big)^{e(k_1,\ldots,k_r)}, \tag{18}$$

where $g_{(n_1,\ldots,n_r)} := U_{0,\alpha(0)}$ and $h_{(n_1,\ldots,n_r)} := 1$.

Finally, as $h(u^{(0,\dots,0)}) \neq 0$, we can apply procedure `GradedParts` (see Section 2.3) in order to compute the homogeneous components of the quotient g/h centered at $(u^{(0,\dots,0)}, 0)$ up to degree D. The sum of these components is (a scalar multiple of) $\mathrm{Res}_{(n_1,\dots,n_r),(d_0,\dots,d_n)}$.

Now we estimate the complexity of the algorithm.

Fix $(k_1,\dots,k_r) \in \mathbb{N}_0^r$ such that $0 \leq k_j \leq n_j$ for $j = 1,\dots,r$. Set $\kappa := n - |(k_1,\dots,k_r)|$. We will denote by

$$N_i^{(k_1,\dots,k_r)} := \prod_{1\leq j\leq r} \binom{n_j - k_j + d_{ij}}{d_{ij}} \qquad i = 0,\dots,\kappa$$

$$\delta_{(k_1,\dots,k_r)} := \mathrm{Bez}_{n_1-k_1,\dots,n_r-k_r}(d_0,\dots,d_{\kappa-1})$$

the number of coefficients in $f_i^{(k_1,\dots,k_r)}$ $(0 \leq i \leq \kappa)$ and the number of solutions of the generic system $f_0^{(k_1,\dots,k_r)},\dots,f_{\kappa-1}^{(k_1,\dots,k_r)}$ respectively. Recall that $N^{(k_1,\dots,k_r)} = \sum_{0\leq i\leq\kappa-1} N_i^{(k_1,\dots,k_r)}$ is the total number of coefficients of the polynomials $f_i^{(k_1,\dots,k_r)}$ $(0 \leq i \leq \kappa - 1)$.

First, we compute straight-line programs encoding $f_0^{(k_1,\dots,k_r)},\dots,f_{\kappa-1}^{(k_1,\dots,k_r)}$ within complexity $O(rN^{(k_1,\dots,k_r)})$ (see Section 2.2). For $i = 0,\dots,\kappa-1$, the length of the straight-line program encoding $f_i^{(k_1,\dots,k_r)}$ is $O(rN_i^{(k_1,\dots,k_r)})$. Therefore, the complexity of applying procedure `NumDenNewton` using these straight-line programs is of order $O(\log(D)\rho_\kappa^2\kappa^2(\kappa^3 + rN^{(k_1,\dots,k_r)}))$ (see Section 2.3), where $\rho_\kappa := \sum_{0\leq i\leq\kappa-1} |d_i| - \kappa + 1$.

In order to compute the approximation of $\det(m_{f_\kappa^{(k_1,\dots,k_r)}})$ from the output of `NumDenNewton`, we obtain the points in $Z^{(k_1,\dots,k_r)}$, that is, the solutions to the system $g_0^{(k_1,\dots,k_r)} = 0,\dots,g_{\kappa-1}^{(k_1,\dots,k_r)} = 0$. Note that, due to the structure of the polynomials $g_i^{(k_1,\dots,k_r)}$ $(0 \leq i \leq \kappa-1)$, this can be achieved by solving $\delta_{(k_1,\dots,k_r)}$ linear systems. Each of these linear systems can be split into r linear systems in the different groups of variables (see Section 3.2): for every $1 \leq j \leq r$, we have to solve a system of $n_j - k_j$ linear equations

$$x_{j0} + a_l x_{j1} + \dots + a_l^{n_j-k_j-1} x_{j\,n_j-k_j-1} + a_l^{n_j-k_j} = 0 \qquad l = 1,\dots,n_j-k_j \quad (19)$$

for certain constants $a_1,\dots,a_{n_j-k_j}$. For a fixed j $(1 \leq j \leq r)$, the solution to (19) is the vector of coefficients of the monic univariate polynomial of degree $n_j - k_j$ whose roots are $a_1,\dots,a_{n_j-k_j}$. These coefficients can be computed from $a_1,\dots,a_{n_j-k_j}$ within complexity $(n_j-k_j)^2$. Therefore, we obtain all the points in $Z^{(k_1,\dots,k_r)}$ within complexity $\delta_{(k_1,\dots,k_r)} \sum_{1\leq j\leq r}(n_j-k_j)^2 = O(\delta_{(k_1,\dots,k_r)}\kappa^2)$.

We also need a straight-line program encoding the homogenized polynomial in $\mathbb{Q}(U_\kappa)[T, X^{(k_1,\dots,k_r)}]$ of $f_\kappa^{(k_1,\dots,k_r)}$ with a new single variable T. This is obtained within complexity $O(r\kappa N_\kappa^{(k_1,\dots,k_r)})$ by computing first all the monomials in $X^{(k_1,\dots,k_r)}$ and the powers of T, then the homogeneous monomials in $T, X^{(k_1,\dots,k_r)}$ multiplied by the corresponding coefficients, and finally their sum. The length of this straight-line program is of order $O(rN_\kappa^{(k_1,\dots,k_r)})$.

This implies that the polynomials $g_{(k_1,\dots,k_r)}$ and $h_{(k_1,\dots,k_r)}$, whose quotient gives the desired approximation, can be computed from $f_0^{(k_1,\dots,k_r)}, \dots, f_{\kappa-1}^{(k_1,\dots,k_r)}$, the homogenized polynomial of $f_\kappa^{(k_1,\dots,k_r)}$ and the points in $Z^{(k_1,\dots,k_r)}$ within complexity $O\Big(\delta_{(k_1,\dots,k_r)}(\log(D)\rho_\kappa^2\kappa^2(\kappa^3+rN^{(k_1,\dots,k_r)})+rN_\kappa^{(k_1,\dots,k_r)})\Big)$ and are encoded by straight-line programs whose length are of the same order as this complexity.

The total complexity for the computation of $g_{(k_1,\dots,k_r)}$ and $h_{(k_1,\dots,k_r)}$ is of order $O\Big(\delta_{(k_1,\dots,k_r)}\kappa(\log(D)\rho_\kappa\kappa(\kappa^3 + rN^{(k_1,\dots,k_r)}) + rN_\kappa^{(k_1,\dots,k_r)})\Big)$.

The next step of the algorithm consists in the computation of the polynomials g and h defined in (17) and (18) respectively.

In order to do this, it is necessary to compute the exponents $e(k_1, \dots, k_r)$ for all vectors $(k_1, \dots, k_r)$ with $0 \le k_j \le n_j$. We compute them recursively according to the next formula which follows easily from the definition (15):

$$e(k_1, \dots, k_r) = \sum_{1\le j\le r;\ k_j>0} d_{\kappa+1\,j}\ \ e(k_1, \dots, k_j - 1, \dots, k_r) \tag{20}$$

where $\kappa := n-|(k_1, \dots, k_r)|$, starting from $e(0, \dots, 0) = 1$. As the computation of an exponent according to (20) requires at most r products and $r-1$ additions of previously computed numbers, we conclude that the computation of all the exponents $e(k_1, \dots, k_r)$ ($0 \le k_j \le n_j$, $1 \le j \le r$) can be performed within complexity $O(r\, n_1 \dots n_r)$.

Now we compute, for every $(k_1, \dots, k_r)$, the powers $\Big(g_{(k_1,\dots,k_r)}\Big)^{e(k_1,\dots,k_r)}$ and $\Big(h_{(k_1,\dots,k_r)}\Big)^{e(k_1,\dots,k_r)}$ within complexity $O(\log(e(k_1, \dots, k_r))$. Taking into account that

$$\begin{aligned} e(k_1, \dots, k_r) &\le \mathrm{Bez}_{n_1,\dots,n_r}(d_1, \dots, d_n) \le D, \\ \delta_{(k_1,\dots,k_r)} &\le \delta := \mathrm{Bez}_{n_1,\dots,n_r}(d_0, \dots, d_{n-1}), \\ \rho_\kappa &\le \rho := \sum_{0\le i\le n-1} |d_i| - n + 1, \end{aligned}$$

after computing the products in (17) and (18), we obtain straight-line programs of length $\mathcal{L} := O(n_1 \dots n_r \delta \log(D)\rho^2 n^2(n^3 + rN))$ encoding g and h.

Finally, we apply procedure `GradedParts` to g and h in order to compute a straight-line program of length

$$O(D^2(D+\mathcal{L})) = O\left(D^2(D + n_1 \dots n_r \delta \log(D)\rho^2 n^2(n^3 + rN))\right)$$

encoding the first $D+1$ homogeneous components of their quotient centered at $(u^{(0,\dots,0)}, 0)$.

The complexity of computing $u^{(0,\dots,0)}$, that is, the vector whose entries are the coefficients of the polynomials $G_0, \dots, G_{n-1}$ defined in (16), is bounded by $O(\delta n r N)$.

This implies that the total complexity of the computation of the above mentioned homogeneous components is of order

$$O(D^2(D + n_1 \dots n_r \delta \log(D)\rho^2 n^2(n^3 + rN))).$$

Adding all the homogeneous components computed to obtain the straight-line program for (a scalar factor) of $\mathrm{Res}_{(n_1,\dots,n_r),(d_1,\dots,d_r)}$ does not modify the order of the complexity or the length of the straight-line program. $\square$

All the parameters involved in the complexity of the algorithm underlying Theorem 5 can easily be bounded in terms of D and N, which leads to the following complexity result:

Remark 7 *The complexity of the computation of the multihomogeneous resultant is* polynomial *in its degree D and the number of its variables N.*

We summarize the algorithm in Procedure `MultiResultant`. Herein, we use the following notation for subroutines:

- `Vects`$(n, \lambda_1, \dots, \lambda_n)$ constructs a family of n vectors of $\lambda_1, \dots, \lambda_n$ coordinates each, with all their coordinates being different rational numbers.
- `Vars`$(n, d_0, \dots, d_n)$ produces a family of $n+1$ sets of variables indexed by the monomials of multi-degrees $d_0, \dots, d_n$.
- `Homog`(f, d) computes the homogenization of the polynomial f up to degree $d \geq \deg f$.
- For $H(X_1, \dots, X_r)$ multihomogeneous and $(k_1, \dots, k_r) \in \mathbb{N}_0^r$, $h^{(k_1,\dots,k_r)}$ denotes the output of a subroutine which computes a straight-line program for the polynomial derived from H by specializing the last k_j variables of the group X_j to 0 and setting $x_{j\, n_j - k_j} = 1$ for every $1 \leq j \leq r$.

procedure $\texttt{MultiResultant}(n, r, n_1, \ldots, n_r, d_0, \ldots, d_n)$

\# $n, r \in \mathbb{N}$
\# $n_1, \ldots, n_r \in \mathbb{N}$ such that $n_1 + \cdots + n_r = n$
\# $d_0, \ldots, d_n \in \mathbb{N}_0^r$

\# The procedure returns the resultant of $n+1$ multihomogeneous polynomials in
\# r groups of $n_1, \ldots, n_r$ variables and multi-degrees $d_0, \ldots, d_n$.

$D := \sum_{0 \le i \le n} \mathrm{Bez}_{n_1,\ldots,n_r}(d_0, \ldots, \hat{d_i}, \ldots, d_n)$;
$(a^{(1)}, \ldots, a^{(r)}) := (\texttt{Vects}(n, d_{01}, \ldots, d_{n-1\,1}), \ldots, \texttt{Vects}(n, d_{0r}, \ldots, d_{n-1\,r}))$;
$(U_0, \ldots, U_n) := \texttt{Vars}(n+1, d_0, \ldots, d_n)$;
for $i = 0, \ldots, n$ **do**
    $F_i := \sum_\alpha U_{i,\alpha} X^\alpha$;
od;
for $i = 0, \ldots, n-1$ **do**
    $G_i := \prod_{1 \le j \le r} \prod_{1 \le k \le d_{ij}} x_{j0} + a_{ik}^{(j)} x_{j1} + (a_{ik}^{(j)})^2 x_{j2} + \cdots + (a_{ik}^{(j)})^{n_j} x_{jn_j}$;
od;
for $\kappa = n, \ldots, 0$ **do**
    $S_\kappa := \{(k_1, \ldots, k_r) \in \mathbb{N}_0^r : 0 \le k_j \le n_j,\ 1 \le j \le r,\ k_1 + \cdots + k_r = n - \kappa\}$;
    for $(k_1, \ldots, k_r) \in S_\kappa$ **do**
        $F := \texttt{Homog}(f_\kappa^{(k_1,\ldots,k_r)}, d_{\kappa 1} + \cdots + d_{\kappa r})$;
        $Z := \texttt{Solve}(g_0^{(k_1,\ldots,k_r)}, \ldots, g_{\kappa-1}^{(k_1,\ldots,k_r)})$;
        $(g_{(k_1,\ldots,k_r)}, h_{(k_1,\ldots,k_r)}) := \texttt{ApproxNorm}(f_0^{(k_1,\ldots,k_r)}, \ldots, f_{\kappa-1}^{(k_1,\ldots,k_r)}, F, Z, D)$;
        $e(k_1, \ldots, k_r) := \sum_{1 \le j \le r;\, k_j > 0} d_{\kappa+1\,j}\, e(k_1, \ldots, k_j - 1, \ldots, k_r)$;
    od;
od;
$g := \prod_{(k_1,\ldots,k_r) \in \bigcup_{0 \le \kappa \le n} S_\kappa} g_{(k_1,\ldots,k_r)}^{e(k_1,\ldots,k_r)}$;
$h := \prod_{(k_1,\ldots,k_r) \in \bigcup_{0 \le \kappa \le n} S_\kappa} h_{(k_1,\ldots,k_r)}^{e(k_1,\ldots,k_r)}$;
$u^{(0,\ldots,0)} := \texttt{Coeffs}(G_0, \ldots, G_{n-1})$;
$(R_0, \ldots, R_D) := \texttt{GradedParts}(g, h, (u^{(0,\ldots,0)}, 0), D)$;
$\mathrm{Res} := \sum_{0 \le t \le D} R_t$;
return(Res)

end

References

[1] E. Bézout, Théorie Générale des Équations Algébriques, Paris, 1779.

[2] P. Bürgisser, M. Clausen, M.A. Shokrollahi, Algebraic complexity theory, Springer, 1997.

[3] J.F. Canny, I.Z. Emiris, An efficient algorithm for the sparse mixed resultant, In Cohen, G.; Mora, T.; Moreno, O.; eds. Proc. Int. Symp. on Appl. Algebra, Algebraic Algorithms and Error-Corr. Codes, Puerto Rico, LNCS **263** (1993) 89-104.

[4] J.F. Canny, I.Z. Emiris, A subdivision-based algorithm for the sparse resultant, J. ACM **47** (3) (2000) 417-451.

[5] A. Cayley, On the theory of elimination, Cambridge and Dublin Math. J. **3** (1848) 116-120.

[6] D. Cox, J. Little, D. O'Shea, Using algebraic geometry, Grad. Texts in Math. **185**, Springer-Verlag, 1998.

[7] C. D'Andrea, Macaulay style formulas for sparse resultants, Trans. Amer. Math. Soc. **354**, No. 7 (2002) 2595-2629.

[8] C. D'Andrea, A. Dickenstein, Explicit formulas for the multivariate resultant, J. Pure Appl. Algebra **164**, No.1-2 (2001) 59-86.

[9] A. Dickenstein, I.Z. Emiris, Multihomogeneous resultant formulae by means of complexes, J. Symbolic Comput. 36 (2003), No. 3-4, 317–342.

[10] I.Z. Emiris, B. Mourrain, Matrices in elimination theory, J. Symbolic Comput. **28**, No. 1-2 (1999) 3-44.

[11] I.M. Gelfand, M.M. Kapranov, A.V. Zelevinsky, Discriminants, resultants, and multidimensional determinants, Birkhäuser, 1994.

[12] M. Giusti, K. Hägele, J. Heintz, J.L. Montaña, L.M. Pardo, J.E. Morais, Lower bounds for Diophantine approximation, J. Pure Appl. Algebra **117 & 118** (1997) 277-317.

[13] M. Giusti, J. Heintz, La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial, Computational algebraic geometry and commutative algebra (Cortona, 1991), 216-256, Sympos. Math. XXXIV, Cambridge Univ. Press, Cambridge, 1993.

[14] M. Giusti, J. Heintz, J.E. Morais, J. Morgenstern, L.M. Pardo, Straight-line programs in geometric elimination theory, J. Pure Appl. Algebra **124** (1998), no. 1-3, 101-146.

[15] J. Heintz, Definability and fast quantifier elimination in algebraically closed fields, Theoret. Comput. Sci. **24** (1983) 239-277.

[16] J. Heintz, T. Krick, S. Puddu, J. Sabia, A. Waissbein, Deformation techniques for efficient polynomial equation solving, J. Complexity **16** (2000) 70-109.

[17] J. Heintz, C.-P. Schnorr, Testing polynomials which are easy to compute, Monographie 30 de l'Enseignement Mathématique (1982) 237-254.

[18] G. Jeronimo, T. Krick, M. Sombra, J. Sabia, The computational complexity of the Chow form, Found. Comput. Math. **4** (2004), No. 1, pp. 41-117.

[19] J.P. Jouanolou, Le formalisme du résultant, Advances in Mathematics Vol. 90, No. 2 (1991) 117-263.

[20] F. Macaulay, Some formulae in elimination, Proc. London Math. Soc. 1 **33** (1902) 3-27.

[21] N. McCoy, On the resultant of a system of forms homogeneous in each of several sets of variables, Trans. Amer. Math. Soc. **35** (1933), no. 1, 215-233.

[22] P. Pedersen, B. Sturmfels, Product formulas for resultants and Chow forms, Math. Z. **214** (1993) 377-396.

[23] I. Shafarevich, Basic algebraic geometry, Springer-Verlag, 1972.

[24] V. Strassen, Vermeidung von Divisionen, J. Reine Angew. Math. **264** (1973) 182-202.

[25] B. Sturmfels, Sparse elimination theory, In D. Eisenbud and L. Robbbiano, eds. Computational algebraic geometry and commutative algebra (Cortona, 1991), Sympos. Math. XXXIV, 264-298, Cambridge Univ. Press, 1993.

[26] B. Sturmfels, On the Newton polytope of the resultant, J. Algebraic Combin. 3 (1994), no. 2, 207-236.

[27] J.J. Sylvester, On a theory of syzygetic relations of two rational integral functions. Comprising an Application to the theory of Sturm's functions, and that of the greatest algebraic common measure, Philosophical Trans. **143** (1853) 407-548.

[28] J. von zur Gathen, Parallel arithmetic computations: a survey, In Proc. 12th FOCS, Bratislava, 1986. LNCS **33** (1986) 93-112.

[29] J. Weyman, A. Zelevinsky, Determinantal formulas for multigraded resultants, J. Algebraic Geom. **3** (1994), no. 4, 569-597.