Sparse resultants and straight-line programs*

Gabriela Jeronimo[♯,†,⋄], Juan Sabia[†,⋄]

♯ Universidad de Buenos Aires. Facultad de Ciencias Exactas y Naturales.
Departamento de Matemática. Buenos Aires, Argentina.

† Universidad de Buenos Aires. Ciclo Básico Común.
Departamento de Ciencias Exactas. Buenos Aires, Argentina.

⋄ CONICET–Universidad de Buenos Aires.
Instituto de Investigaciones Matemáticas Luis A. Santaló (IMAS).
Buenos Aires, Argentina.

May 11, 2017

Abstract

We prove that the sparse resultant, redefined by D'Andrea and Sombra and by Esterov as a power of the classical sparse resultant, can be evaluated in a number of steps which is polynomial in its degree, its number of variables and the size of the exponents of the monomials in the Laurent polynomials involved in its definition. Moreover, we design a probabilistic algorithm of this order of complexity to compute a straight-line program that evaluates it within this number of steps.

Keywords: Sparse resultants, straight-line programs, algorithms

1 Introduction

Resultants are considered a key tool in the resolution of polynomial equation systems, mainly because of their role as eliminating polynomials. In the last decades, the practical utility of resultants has aroused interest in their effective computation.

The study of classical homogeneous resultants goes back to Bézout, Cayley and Sylvester (see [2], [6] and [41]). In [31], Macaulay obtained explicit formulae for the homogeneous resultant as a quotient of two determinants and, from then on, several effective procedures to compute these resultants have been proposed (see, for example, [9] and the references therein). More recently, Gelfand, Kapranov and Zelevinski generalized the classical notion to the sparse setting (see [17]). The first effective method for computing sparse resultants was given in [39]. In [4] (see also [5]) and [40], the authors provided an algorithm for computing a square Sylvester style matrix with determinant equal to a nonzero multiple of the resultant. A survey of matrix constructions for the computation of resultants can be

*Partially supported by the following Argentinian grants: PIP 11220130100527CO CONICET (2014-2016) and UBACYT 20020120100133 (2013-2016).

found in [13]. In [8], it was shown that the sparse resultant is a quotient of the determinant of a Sylvester style matrix by one of its minors, extending Macaulay's formulation to the sparse setting.

When dealing with the computation of the resultant of a particular system in order to know whether it vanishes or not, its representation as a quotient of determinants may not be enough because the denominator may vanish. Classical methods to solve this problem consist in making a symbolic perturbation to the system (see, for instance, [4]), but they require further computations for each particular system. This motivates the search for a division-free representation of the resultant. A possible approach to do this is the classical Strassen's method to eliminate divisions described in [38]. Using this method, in [29], it is shown how to express a quotient of two determinants that is a polynomial as a single determinant of a matrix of size polynomial in the sizes of the original matrices.

All the previously mentioned procedures for the computation of sparse resultants deal with matrices of exponential size: for $n+1$ Laurent polynomials in n variables with n-dimensional Newton polytopes, the number of rows and columns of the matrices involved in the computation of the associated resultant is of order $O(k^n n^{-3/2} D)$ (see [5, Theorem 3.10]), where k is a positive constant and D is the total degree of the resultant as a polynomial in the coefficients of the input. This implies that the algebraic complexity of any algorithm using these matrices is necessarily exponential in the number of variables n of the input polynomials. For the classical homogeneous resultant, the complexity of testing it for zero and of the algorithms to compute it via Macaulay matrices was studied in [22].

Due to the well-known estimates for the degree of the sparse resultant in terms of mixed volumes (see, for instance, [34, Corollary 2.4] and [11, Proposition 3.4]), any algorithm for its computation which encodes it 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 this polynomial 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, [18] and [20], where this data structure allowed the design of the first algorithms for solving zero-dimensional polynomial systems within complexity polynomial in the output size). 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 [25]. 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. Afterwards, in [27], an algorithm for the computation of both mixed and unmixed multihomogeneous resultants by means of straight-line programs was given. The algorithm relies on Poisson's product formula for the multihomogeneous resultant and its complexity is polynomial in the degree and the number of variables of the computed resultant.

The definition of the general sparse resultant for Laurent polynomials as an irreducible polynomial defining the corresponding incidence variety also implied a Poisson-type formula proved in [34], but this formula does not hold for arbitrary supports. A restatement

of this formula, valid in a more general setting, was given in [32]. In [11] (see also [14, Definition 3.1]), the notion of sparse resultant was redefined and studied using multiprojective elimination theory. The new sparse resultant is a power of the previous one. It has better properties and produces more uniform statements, in particular, a nicer Poisson-type formula which holds for *any* family of supports (see [11, Theorem 1.1]). Further properties of this resultant have been studied in [10], where the Macaulay-style formulas in [8] are simplified and generalized to compute this new sparse resultant as a quotient of two determinants of Sylvester-type matrices.

In this paper, we deal with the computation of the new sparse resultant defined in [11] and [14]. We prove that it can be evaluated in a number of steps which is polynomial in its number of variables, its total degree, and the size of the exponents of the monomials in the Laurent polynomials involved in its definition. This theoretical result shows that, in spite of the *large* number of terms that can be present in the monomial expansion of the sparse resultant, this polynomial has a *short* encoding, thus extending previous results in [25]. A fundamental tool in our proof is the already mentioned Poisson product formula established in [11] for the new sparse resultant.

Our main result is the following:

Theorem. *Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_n)$, where $\mathcal{A}_i \subset \mathbb{Z}^n$ is a non-empty finite set for every $0 \leq i \leq n$. The sparse resultant $\mathrm{Res}_{\mathcal{A}}$ can be evaluated by a straight-line program of length polynomial in its number of variables $N = \sum_{0\leq i\leq n} |\mathcal{A}_i|$, its total degree $D = \sum_{0\leq i\leq n} MV_n(\mathcal{A}_0, \ldots, \mathcal{A}_{i-1}, \mathcal{A}_{i+1}, \ldots, \mathcal{A}_n)$ (where MV_n denotes the standard n-dimensional mixed volume), and $\mathcal{Q} = \max\{||a|| : a \in \mathcal{A}_i,\, 0 \leq i \leq n\}$.*

In addition, we design a probabilistic algorithm to compute, from a family of $n+1$ finite subsets of $\mathbb{Z}^n$, a straight-line program for the sparse resultant associated to $n+1$ Laurent polynomials with these support sets within a number of operations of the same order as for the computation of the resultant (see Theorem 10 for a precise statement). Although we have not implemented the algorithm, we think that an implementation in the MATHEMAGIX language could be done building on the GEOMSOLVEX library ([30]).

Our approach improves previous results in the sense that our complexity bounds are not exponential in the number of variables n of the input Laurent polynomials. Even though the polynomial we compute is a power of the classical sparse resultant, the former encodes all the relevant information provided by the latter (for instance, they both vanish at the coefficients of the same Laurent polynomial systems).

Throughout the paper, to avoid confusion, following [11], we will call sparse *eliminant* to the classical (irreducible) sparse resultant, and sparse *resultant* to the new object defined as a power of the sparse eliminant.

The paper is organized as follows. In Section 2, we introduce the basic definitions and the notation used throughout the paper, we present the notion of sparse resultant and discuss our algorithmic framework. In Section 3, we first show how to reduce the problem to a special case (essential family of supports) and then, we present some auxiliary algorithmic tools we will apply. Finally, Section 4 is devoted to proving our main result.

2 Preliminaries

2.1 Basic definitions and notation

Let $x = (x_1, \ldots, x_n)$ be indeterminates over $\mathbb{Q}$. For $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}^n$, we denote by $x^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ the monomial in the ring of Laurent polynomials $\mathbb{Q}[x^{\pm 1}] := \mathbb{Q}[x_1, x_1^{-1}, \ldots, x_n, x_n^{-1}]$.

Given a family of finite subsets $\mathcal{A}_1, \ldots, \mathcal{A}_r$ of $\mathbb{Z}^n$, *a sparse system with supports* $\mathcal{A}_1, \ldots, \mathcal{A}_r$ is given by Laurent polynomials

$$h_i = \sum_{a_{i,j} \in \mathcal{A}_i} c_{a_{i,j}} x^{a_{i,j}}, \qquad \text{for } i = 1, \ldots, r,$$

with $c_{a_{i,j}} \neq 0$ for every $a_{i,j} \in \mathcal{A}_i$.

When dealing with sparse polynomial systems, the affine lattice generated by the exponents of their monomials and the convex hulls of these exponent sets play an important role.

Let M be a r-dimensional lattice in $\mathbb{R}^n$, and let $M_\mathbb{R}$ be the linear subspace it generates. We consider the volume form vol_M defined on $M_\mathbb{R}$ from the r-dimensional Euclidean volume normalized so that the fundamental domain P of M satisfies that $\mathrm{vol}_M(P) = 1$. The *mixed volume* of a family of lattice polytopes $Q_1, \ldots, Q_r \subset M_\mathbb{R}$ is defined as

$$MV_M(Q_1, \ldots, Q_r) = \sum_{j=1}^{r} (-1)^{r-j} \sum_{1 \leq i_1 < \cdots < i_j \leq r} \mathrm{vol}_M(Q_{i_1} + \cdots + Q_{i_j}).$$

We refer the reader to [7, Chapter 7 §4] for basic properties of the mixed volume.

For a family $\mathcal{A}_1, \ldots, \mathcal{A}_n$ of finite subsets of $\mathbb{Z}^n$, we denote by $MV_n(\mathcal{A}_1, \ldots, \mathcal{A}_n)$ the mixed volume of the convex hulls of $\mathcal{A}_1, \ldots, \mathcal{A}_n$ in $\mathbb{R}^n$. Bernstein's theorem (see [1]) states that the number of isolated zeros in $(\mathbb{C}^*)^n$ (where $\mathbb{C}^* = \mathbb{C} \setminus \{0\}$) of a sparse system with supports contained in $\mathcal{A}_1, \ldots, \mathcal{A}_n$ is at most $MV_n(\mathcal{A}_1, \ldots, \mathcal{A}_n)$.

2.2 Sparse resultant

Here, we recall the notion of sparse resultant introduced in [11] (see also [14, Definition 3.1]). We state it for Laurent polynomials with exponents in $\mathbb{Z}^n$.

For $i = 0, \ldots, n$, consider a non-empty finite subset $\mathcal{A}_i = \{a_{i,0}, \ldots, a_{i,N_i}\} \subset \mathbb{Z}^n$, a set of $N_i + 1$ variables $U_i = (U_{i0}, \ldots, U_{iN_i})$, and the general Laurent polynomial with support $\mathcal{A}_i$ in the variables $x = (x_1, \ldots, x_n)$:

$$f_i = \sum_{k=0}^{N_i} U_{ik}\, x^{a_{i,k}} \in \mathbb{Q}[U_i][x_1^{\pm 1}, \ldots, x_n^{\pm 1}]. \tag{1}$$

Set $\mathcal{A} = (\mathcal{A}_0, \ldots, \mathcal{A}_n)$. Let

$$\Gamma_\mathcal{A} = \{(\xi, u) \in (\mathbb{C}^*)^n \times \prod_{i=0}^{n} \mathbb{P}^{N_i} \mid f_0(u_0, \xi) = 0, \ldots, f_n(u_n, \xi) = 0\}$$

be the associated incidence variety.

The $\mathcal{A}$-*resultant* or *sparse resultant associated to* $\mathcal{A}$, which will be denoted by $\mathrm{Res}_{\mathcal{A}}$, is defined as the unique (up to sign) primitive polynomial in $\mathbb{Z}[U_0, \ldots, U_n]$ giving an equation for the direct image $\pi_*\Gamma_{\mathcal{A}} := d_{\mathcal{A}}\pi(\Gamma_{\mathcal{A}})$, where $\pi : (\mathbb{C}^*)^n \times \prod_{i=0}^n \mathbb{P}^{N_i} \to \prod_{i=0}^n \mathbb{P}^{N_i}$ is the projection to the second factor, $d_{\mathcal{A}}$ is the degree of the restriction of π to $\Gamma_{\mathcal{A}}$ (which is defined as 0 if $\dim(\pi(\Gamma_{\mathcal{A}})) < \dim(\Omega_{\mathcal{A}})$), and $d_{\mathcal{A}}\pi(\Gamma_{\mathcal{A}})$ is the corresponding cycle.

The sparse resultant is a homogeneous polynomial in each group of variables U_i, for $i = 0, \ldots, n$, and

$$\deg_{U_i}(\mathrm{Res}_{\mathcal{A}}) = MV_n(\mathcal{A}_1, \ldots, \mathcal{A}_{i-1}, \mathcal{A}_{i+1}, \ldots, \mathcal{A}_n) \tag{2}$$

(see [11, Proposition 3.4]).

We point out that the above definition of sparse resultant differs from the classical one introduced in [17] and [40] as an irreducible polynomial in $\mathbb{Z}[U_0, \ldots, U_n]$ defining the Zariski closure of $\pi(\Gamma_{\mathcal{A}})$, if this is a hypersurface, and as 1 otherwise. Following [11], we call $\mathcal{A}$-*eliminant* or *sparse eliminant associated to* $\mathcal{A}$ to this irreducible polynomial, and we denote it by $\mathrm{Elim}_{\mathcal{A}}$.

Both notions relate as follows:

$$\mathrm{Res}_{\mathcal{A}} = \pm \mathrm{Elim}_{\mathcal{A}}^{d_{\mathcal{A}}}.$$

2.3 Algorithms and codification

Since the goal of this paper is the computation of sparse resultants as multivariate polynomials with integer coefficients, it suffices for us to consider an algorithmic model over the base field $\mathbb{Q}$. The only operations allowed in our algorithms are arithmetic operations in $\mathbb{Q}$ and comparisons ($=$ or $\neq$) between two elements in $\mathbb{Q}$. We assume that the cost of each operation is 1 and so we define the complexity of the algorithm as the number of operations it performs. For a more detailed treatment of this kind of computational models, which is widely used in the symbolic polynomial system solving framework, see for instance [3].

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

- *Sparse encoding*, that is, as a list of pairs (a, u_a) where a runs over the set of exponents of monomials in a fixed set and u_a is the corresponding coefficient of the polynomial, provided that we know in advance that the coefficient of any other monomial of the polynomial is zero.

- *Dense form,* that is, as the array of all its coefficients (including zeroes) in a prefixed order of all monomials of degree at most d, where d is an upper bound for the degree of the polynomial.

- Division-free *straight-line program* (slp for short), that is, an algorithm without branchings that enables us to evaluate the polynomial at any given point. Each of the instructions in this program is an addition, subtraction or multiplication between two precalculated polynomials, or an addition or multiplication by a variable or a rational number. The number of instructions in the program is called the *length* of the straight-line program. For a precise definition of a straight-line program we refer the reader to [3, Definition 4.2] and [24].

Note that, from any of these representations of a polynomial f with rational coefficients, it is possible to evaluate f at any given point with coordinates in an effective field K of characteristic 0.

A polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ of degree at most $d > 0$ and at most N nonzero terms can be evaluated with $O(nN \log(d))$ arithmetic operations in $\mathbb{Q}$. Thus, f can be encoded by a straight-line program of this length that can be easily obtained from its sparse representation.

In this work, we will also consider multivariate *Laurent* polynomials, which will be encoded in sparse representation.

Our algorithms are probabilistic in the sense that they make random choices of points which lead to a correct computation provided the points lie outside certain proper Zariski closed sets of suitable affine spaces. Although we will not estimate the error probability of our algorithms, it can be controlled, using the Schwartz-Zippel lemma ([36], [42]), by making the needed random choices uniformly within sufficiently large finite sets of integers whose cardinalities depend on the degrees of the polynomials defining the previously mentioned Zariski closed sets. In this sense, our algorithms can be considered of Monte Carlo type.

We will use the standard O notation in our complexity estimates: for $f, g : \mathbb{Z}_{\geq 0} \to \mathbb{R}$, $f(d) = O(g(d))$ if $|f(d)| \leq c|g(d)|$ for a positive constant c. We will also use the notation $M(d)$ for the number of arithmetic operations in a commutative ring R of characteristic 0 needed for the multiplication of two univariate polynomials of degree at most d with coefficients in R. According to [16, Chapter 8], $M(d) = O(d \log(d) \log(\log(d)))$, where log denotes logarithm to base 2. We recall that multipoint evaluation and interpolation of univariate polynomials of degree d with coefficients in R can be performed within $O(M(d) \log(d))$ operations in R (see [16, Chapter 10]).

We denote by ω the exponent in the complexity estimate $O(D^\omega)$ for the multiplication of two $D \times D$ matrices with rational coefficients. It is known that $2 \leq \omega < 2.376$ (see [16, Chapter 12]). Finally, we write Ω for the exponent in the complexity $O(D^\Omega)$ of the computation without divisions of the determinant and adjoint of a matrix of size $D \times D$ with entries in a commutative ring R. By [28], we have that $\Omega < 2.7$.

3 Tools

In this section we will prove two auxiliary results we will use in order to design our algorithm for the computation of sparse resultants.

3.1 Essential families

We recall here the notion of essential subfamily of supports introduced in [40] to characterize those families $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_n)$, where $\mathcal{A}_i \subset \mathbb{Z}^n$ is a finite set, for which the associated sparse eliminant is not constant.

For $i = 0, \ldots, n$, if $\mathcal{A}_i = \{a_{i,0}, \ldots, a_{i,N_i}\} \subset \mathbb{Z}^n$, let $L_{\mathcal{A}_i} := \sum_{k=1}^{N_i} (a_{i,k} - a_{i,0})\mathbb{Z}$. For $I \subset \{0, \ldots, n\}$, we write $\mathcal{A}_I := (\mathcal{A}_i)_{i \in I}$ and $L_{\mathcal{A}_I} := \sum_{i \in I} L_{\mathcal{A}_i}$. As usual, $|I|$ denotes the cardinality of the set I and, for a lattice L, $\operatorname{rank}(L)$ denotes its rank.

Definition 1 *Let $I \subset \{0,\dots,n\}$. The subfamily $\mathcal{A}_I$ is said to be* essential *if the following conditions hold:*

- $|I| = \operatorname{rank}(L_{\mathcal{A}_I}) + 1$*;*
- *for every $I' \subsetneq I$, $|I'| \leq \operatorname{rank}(L_{\mathcal{A}_{I'}})$.*

As proved in [40, Corollary 1.1], the sparse resultant associated to a family of supports $\mathcal{A}$ is not constant if and only if $\mathcal{A}$ has a unique essential subfamily $\mathcal{A}_I$ and, if this is the case, $\operatorname{Res}_{\mathcal{A}}$ depends only on the coefficients of Laurent polynomials with supports $\mathcal{A}_I$. We start proving a result on the existence of a unique essential subfamily.

For every $0 \leq i \leq n$, we denote $M_i(\mathcal{A}) = MV_n(\mathcal{A}_0,\dots,\mathcal{A}_{i-1},\mathcal{A}_{i+1},\dots,\mathcal{A}_n)$.

Proposition 2 *Let $\mathcal{A} = (\mathcal{A}_0,\mathcal{A}_1,\dots,\mathcal{A}_n)$, where $\mathcal{A}_i \subset \mathbb{Z}^n$ is a finite set for every $0 \leq i \leq n$. Let $\mathcal{I} := \{i \mid 0 \leq i \leq n,\ M_i(\mathcal{A}) > 0\}$. The family $\mathcal{A}$ has a unique essential subfamily if and only if $\mathcal{A}_{\mathcal{I}}$ is essential.*

Proof. Recall that for a family $\mathcal{B} = (\mathcal{B}_1,\dots,\mathcal{B}_n)$ of finite sets of $\mathbb{Z}^n$, $MV_n(\mathcal{B}_1,\dots,\mathcal{B}_n) > 0$ if and only if, for every $J \subset \{1,\dots,n\}$, $|J| \leq \operatorname{rank}(L_{\mathcal{B}_J})$ ([15, Chapter IV, Theorem 4.13]). Moreover, if $MV_n(\mathcal{B}_1,\dots,\mathcal{B}_n) = 0$, it is easy to see that for a minimal subset $J \subset \{1,\dots,n\}$ such that $\operatorname{rank}(L_{\mathcal{B}_J}) < |J|$, the family $\mathcal{B}_J$ is essential.

In order to prove the proposition, first, note that, if $M_i(\mathcal{A}) > 0$ for every $0 \leq i \leq n$, then $\mathcal{A}$ is the unique essential subfamily. Otherwise, if $M_i(\mathcal{A}) = 0$ for some index i, there exists $J \subset \{0,\dots,n\} \setminus \{i\}$ such that $\mathcal{A}_J$ is essential. It follows that there always exists an essential subfamily of $\mathcal{A}$.

We have that $\mathcal{I} \subset I$ for every $I \subset \{0,\dots,n\}$ such that $\mathcal{A}_I$ is essential: for $i \in \mathcal{I}$, we have that $M_i(\mathcal{A}) > 0$ and so, no family $\mathcal{A}_J$ with $J \subset \{0,\dots,n\} \setminus \{i\}$ is essential; therefore, $i \in I$.

If $I_1 \neq I_2$ are two subsets of $\{0,\dots,n\}$ such that $\mathcal{A}_{I_1}$ and $\mathcal{A}_{I_2}$ are essential, then $\mathcal{I} \subset I_1 \cap I_2 \subsetneq I_1$, and therefore, $\mathcal{A}_{\mathcal{I}}$ is not essential.

Assume now that $\mathcal{A}$ has a unique essential subfamily $\mathcal{A}_I$. If $j \notin \mathcal{I}$, then $M_j(\mathcal{A}) = 0$ and therefore, $I \subset \{0,\dots,n\} \setminus \{j\}$, which implies that $j \notin I$. As we already know that $\mathcal{I} \subset I$, we conclude that $I = \mathcal{I}$. □

Remark 3 *The subset $\mathcal{I} = \{i \mid 0 \leq i \leq n,\ M_i(\mathcal{A}) > 0\}$ satisfies that $\mathcal{A}_{\mathcal{I}}$ is essential if and only if rank$(L_{\mathcal{A}_{\mathcal{I}}}) = |\mathcal{I}| - 1$. This is a direct consequence of the fact that if $\mathcal{A}_{\mathcal{I}}$ is not essential, but $I \subset \{0,\dots,n\}$ satisfies that $\mathcal{A}_I$ is essential, we have that $\mathcal{I} \subsetneq I$; then, rank$(L_{\mathcal{A}_{\mathcal{I}}}) \geq |\mathcal{I}|$.*

As shown in [12, Theorem 8], there is a polynomial time algorithm to determine whether the mixed volume of n convex polytopes in $\mathbb{R}^n$ is zero or not. Therefore, we can compute the set $\mathcal{I}$ and decide if $\mathcal{A}_{\mathcal{I}}$ is the unique essential subfamily of $\mathcal{A}$ in polynomial time.

3.2 Geometric resolutions and Newton-Hensel lifting

A common way to describe zero-dimensional affine varieties defined by polynomials over $\mathbb{Q}$ is a *geometric resolution* (see, for instance, [21] and the references therein). The precise definition we are going to use in our algorithm is the following.

Let $V \subset \mathbb{C}^n$ be a zero-dimensional variety defined by rational polynomials consisting of δ points. Given a linear form $\ell = \ell_1 x_1 + \cdots + \ell_n x_n$ in $\mathbb{Q}[x]$ such that $\ell(\xi) \neq \ell(\xi')$ if $\xi \neq \xi'$, the following polynomials completely characterize V:

- the minimal polynomial $q = \prod_{\xi \in V}(Y - \ell(\xi)) \in \mathbb{Q}[Y]$ of ℓ over the variety V (where Y is a new variable),
- polynomials $v_1, \ldots, v_n \in \mathbb{Q}[Y]$ with $\deg(v_j) < \delta$ for every $1 \leq j \leq n$ satisfying $V = \{(v_1(\eta), \ldots, v_n(\eta)) \in \mathbb{C}^n \mid \eta \in \mathbb{C}, q(\eta) = 0\}$.

The family of univariate polynomials $(q, v_1, \ldots, v_n) \in \mathbb{Q}[Y]^{n+1}$ is called a geometric resolution of V (associated with the linear form ℓ).

A geometric resolution of a zero-dimensional variety can be obtained algorithmically from a finite set of polynomials defining it. We will use a subroutine from [26] which computes geometric resolutions in the sparse setting (we recall that we use the standard notation $M(d) = d\log(d)\log\log(d)$, but the notation in [26] is $M(d) = d\log^2(d)\log\log(d)$):

Lemma 4 *([26, Proposition 5.13]) Let $h_1, \ldots, h_n \in \mathbb{Q}[x_1, \ldots, x_n]$ be generic sparse polynomials with supports $\mathcal{A}_1, \ldots, \mathcal{A}_n \subset (\mathbb{Z}_{\geq 0})^n$. There is a probabilistic algorithm which computes a geometric resolution of the set of common zeros of $h_1, \ldots, h_n$ in $(\mathbb{C}^*)^n$ within complexity*

$$O(n^3 N \log(\mathcal{Q}) M(\delta)\log(\delta)(M(\delta)\log(\delta) + M(\delta')\log(\delta'))),$$

with $N := \sum_{1 \leq i \leq n} |\mathcal{A}_i|$, $\mathcal{Q} := \max\{||a_{i,k}|| : 1 \leq i \leq n,\ 1 \leq k \leq |\mathcal{A}_i|\}$, $\delta := MV_n(\mathcal{A}_1, \ldots, \mathcal{A}_n)$, and $\delta' := \sum_{1 \leq i \leq n} MV_n(\Delta, \mathcal{A}_1, \ldots, \mathcal{A}_{i-1}, \mathcal{A}_{i+1}, \ldots, \mathcal{A}_n)$, where Δ denotes the standard n-dimensional simplex.

Another tool we will use is an algorithmic version of the Newton-Hensel lifting (see [19, 23]).

Let $F := (f_1(U, x), \ldots, f_n(U, x))$ be polynomials in $\mathbb{Q}[U, x]$, where $x = (x_1, \ldots, x_n)$, and U is a family of indeterminates over $\mathbb{Q}[x]$. Assume the Jacobian matrix $DF(x)$ of the polynomials F with respect to the variables x is non-singular. The Newton operator associated to F is defined as:

$$\mathcal{N}_F(x)^t := x^t - DF(x)^{-1}.F(x)^t.$$

For $\kappa \in \mathbb{Z}_{>0}$, the κth iteration of the Newton operator is given by a vector of rational functions:

$$\mathcal{N}_F^\kappa(x) = \left(\frac{g_1^{(\kappa)}}{h^{(\kappa)}}, \ldots, \frac{g_n^{(\kappa)}}{h^{(\kappa)}}\right)$$

where $g_1^{(\kappa)}, \ldots, g_n^{(\kappa)}, h^{(\kappa)} \in \mathbb{Q}[U, x]$. The following lemma states the complexity of the computation of these polynomials.

Lemma 5 *([19, Lemma 30]) Let notations and assumptions be as before. If the polynomials F have degrees bounded by d in the variables x and are given by a straight-line program of length L, for a given $\kappa \in \mathbb{Z}_{>0}$, there is a straight-line program of length $O(\kappa d^2 n^7 L)$ which evaluates $g_1^{(\kappa)}, \ldots, g_n^{(\kappa)}, h^{(\kappa)}$. This straight-line program can be obtained within complexity of the same order.*

3.3 Padé approximation

Our algorithm for the computation of the sparse resultant will compute it as the numerator of a rational function that will be approximated up to a prescribed order. To recover the numerator and the denominator of a rational function from a suitable power series expansion we will apply the well-known technique of Padé approximation (see, for instance, [16, Section 5.9]). The following result provides a complexity estimate for the procedure in our setting.

Lemma 6 *Let $Z = (Z_1, \ldots, Z_m)$ be indeterminates over $\mathbb{Q}$. Let $\varphi = p/q \in \mathbb{Q}(Z)$ be a rational function such that p and q are relatively prime polynomials in $\mathbb{Q}[Z]$ with $\deg(p), \deg(q) \leq D$. Assume that $q(z) \neq 0$ for a given $z = (z_1, \ldots, z_m) \in \mathbb{Q}^m$ and let $\Phi = \sum_{i=0}^{2D} \varphi_i$ be the Taylor expansion of order $2D$ of φ centered at z, where φ_i is a homogeneous polynomial of degree i in $Z - z = (Z_1 - z_1, \ldots, Z_m - z_m)$. There is a probabilistic algorithm that, from a straight-line program of length L encoding $\varphi_0, \ldots, \varphi_{2D}$, computes a straight-line program encoding p and q (up to a factor in $\mathbb{Q}$) within complexity $O(D^2(D^\Omega + L))$.*

Proof. The algorithm is based on Padé approximation for univariate power series. In order to reduce the problem to the univariate case, we follow the strategy in [35, Section 4.3], adapting the procedure to work with (division-free) straight-line programs and to estimate the complexity in our computational model.

First, we introduce a new variable s. Let $\widetilde{\Phi} := \sum_{i=0}^{2D} \varphi_i s^i$. Note that $\widetilde{\Phi}$ is a polynomial of degree at most $2D$ in the variable s and that $(p(s(Z_1 - z_1) + z_1, \ldots, s(Z_m - z_m) + z_m), q(s(Z_1 - z_1) + z_1, \ldots, s(Z_m - z_m) + z_m))$ in $\mathbb{Q}(Z)[s]^2$ is a $(D+1, D)$ Padé approximant to $\widetilde{\Phi}$.

Let $(\overline{P}, \overline{Q}) \in \mathbb{Q}(Z)[s]^2$ be the Padé approximant to $\widetilde{\Phi}$ obtained by applying the Extended Euclidean Algorithm to s^{2D+1} and $\widetilde{\Phi}$ ([16, Corollary 5.21]). Due to the uniqueness of the $(D+1, D)$-Padé approximant, it follows that $\overline{Q}(s)/\overline{Q}(0) = q(s(Z_1 - z_1) + z_1, \ldots, s(Z_m - z_m) + z_m))/q(z)$ and $\overline{P}(s)/\overline{Q}(0) = p(s(Z_1 - z_1) + z_1, \ldots, s(Z_m - z_m) + z_m))/q(z)$.

So, in order to obtain (a scalar multiple of) p and q, it suffices to compute $\overline{P}(1)/\overline{Q}(0)$ and $\overline{Q}(1)/\overline{Q}(0)$.

Let $r_j, s_j, t_j \in \mathbb{Q}(Z)[s]$ be the polynomials appearing in the jth row in the Extended Euclidean Algorithm for s^{2D+1} and $\widetilde{\Phi}$. If j is minimal such that $\deg(r_j) \leq D$, by [16, Corollary 5.21], we have that $(r_j, t_j) \in \mathbb{Q}(Z)[s]^2$ is a $(D+1, D)$-Padé approximant to $\widetilde{\Phi}$. Following [16, Corollaries 6.48 and 6.49], we can compute without divisions multiples in $\mathbb{Q}[Z][s]$ of these polynomials by the same factor in $\mathbb{Q}[Z]$. We now explain briefly the procedure in order to estimate the complexity in our framework.

Using the notation in [16, Section 6.10], if S_k is the kth subresultant matrix of s^{2D+1} and $\widetilde{\Phi}$, in the first step we determine $k_0 := \max\{0 \leq k \leq D : \det(S_k) \neq 0\}$, which is the degree of r_j. The entries of the matrices S_k are polynomials in $\mathbb{Q}[Z]$ encoded by an slp; then, in order to decide if the determinants $\det(S_k)$ are zero or not, we evaluate them at a randomly chosen point in $\mathbb{Q}^m$. Since each S_k is a submatrix of the Sylvester matrix of s^{2D+1} and $\widetilde{\Phi}$, which is a square matrix of size $(4D+1) \times (4D+1)$, the complexity of this step is of order $O(L + D^{\omega+1})$.

In a second step, we solve the linear system $S_{k_0}(\alpha,\beta)^t = \det(S_{k_0})(0,\ldots,0,1)^t$, where $\alpha = (\alpha_{2D-k_0-1},\ldots,\alpha_0) \in \mathbb{Q}[Z]^{2D-k_0}$ and $\beta = (\beta_{2D-k_0},\ldots,\beta_0) \in \mathbb{Q}[Z]^{2D-k_0+1}$ give the coefficients for polynomials $V := \sum_{i=0}^{2D-k_0-1} \alpha_i s^i$ and $\overline{Q} := \sum_{i=0}^{2D-k_0} \beta_i s^i$ such that if $\overline{P} := Vs^{2D+1} + \overline{Q}\,\widetilde{\Phi}$, then $(\overline{P},\overline{Q})$ is a $(D+1,D)$-Padé approximant of $\widetilde{\Phi}$ in $\mathbb{Q}[Z][s]$. By using Cramer's rule and division-free computation of the adjoint matrix of S_{k_0}, the complexity of this step is of order $O(D^\Omega)$ and it produces an slp of length $O(L+D^\Omega)$ for the coordinates of α and β.

The polynomials $\overline{Q}(0), \overline{P}(1)$ and $\overline{Q}(1) \in \mathbb{Q}[Z]$ can be obtained by specialization of s within $O(D)$ additional operations, which does not modify the complexity order or the slp length. Finally, we compute the exact quotients $\overline{P}(1)/\overline{Q}(0)$ and $\overline{Q}(1)/\overline{Q}(0)$ by means of the well-known Strassen Vermeidung von Divisionen (division avoiding) algorithm from [38]. In order to apply it, we choose randomly a point $\zeta \in \mathbb{Q}^m$ so that $\overline{Q}(0)(\zeta) \neq 0$. Following [25, Lemma 1.7], the complexity of this step is of order $O(D^2(D^\Omega + L))$. □

4 A short slp for the sparse resultant

This section is devoted to describing our main algorithm, which computes an slp for the sparse resultant with length polynomial in the number of variables of the resultant, its total degree, and an upper bound for the size of the points in the input supports.

Given an arbitrary family of $n+1$ supports in $\mathbb{Z}^n$, we first reduce the problem to the computation of a power of a resultant associated to a family of $k+1$ supports in $\mathbb{Z}^k$ which is essential. More specifically, given $\mathcal{A} = (\mathcal{A}_0,\mathcal{A}_1,\ldots,\mathcal{A}_n)$, where $\mathcal{A}_i \subset \mathbb{Z}^n$ is a finite set for every $0 \leq i \leq n$, we first determine the set $\mathcal{I} := \{i \mid 0 \leq i \leq n,\ M_i(\mathcal{A}) > 0\}$. By Proposition 2, Remark 3 and [40, Corollary 1.1], if $\operatorname{rank}(L_{\mathcal{A}_\mathcal{I}}) \neq |\mathcal{I}|-1$, then $\operatorname{Res}_\mathcal{A} = 1$; otherwise, $\mathcal{A}_\mathcal{I}$ is the only essential subfamily of $\mathcal{A}$ and

$$\operatorname{Res}_\mathcal{A} = (\operatorname{Res}_{\mathcal{A}_\mathcal{I}})^{e_\mathcal{I}} \ \text{ with } e_\mathcal{I} = MV_{\mathbb{Z}^n/L^{sat}_{\mathcal{A}_\mathcal{I}}}(\{\varpi(\mathcal{A}_j) : j \notin \mathcal{I}\}),$$

where $L^{sat}_{\mathcal{A}_\mathcal{I}} = (L_{\mathcal{A}_\mathcal{I}} \otimes \mathbb{Q}) \cap \mathbb{Z}^n$ and ϖ is the projection $\varpi : \mathbb{Z}^n \to \mathbb{Z}^n/L^{sat}_{\mathcal{A}_\mathcal{I}}$ (see [11, Proposition 3.13]). In this case, if $f_0,\ldots,f_n$ are the polynomials defined in (1), by means of a suitable change of variables, the polynomials $\{f_i\}_{i\in\mathcal{I}}$ can be written as polynomials in $|\mathcal{I}|-1$ new variables. This can be done by computing the Smith normal form of an integer matrix (see, for instance, [37] for algorithms and complexity estimates). Thus, without loss of generality, from now on we will assume that $\mathcal{I} = \{0,\ldots,n\}$.

The algorithm relies on the Poisson formula for the sparse resultant from [11, Theorem 1.1]. Before stating the formula we introduce some notation.

Let $\mathcal{B} \subset \mathbb{Z}^n$ be a nonempty finite set and $f = \sum_{b\in\mathcal{B}} c_b x^b$ be a Laurent polynomial in the variables $x = (x_1,\ldots,x_n)$. For $v \in \mathbb{Z}^n$, we set

$$h_\mathcal{B}(v) = \min_{b\in\mathcal{B}} \langle b, v\rangle, \quad \mathcal{B}_v = \{b \in \mathcal{B} \mid \langle b, v\rangle = h_\mathcal{B}(v)\} \ \text{ and } \ f_v = \sum_{b\in\mathcal{B}_v} c_b x^b.$$

With this notation and following [11, Definition 4.1], the Poisson formula states that

$$\operatorname{Res}_\mathcal{A}(U_0,U_1,\ldots,U_n) = \pm \prod_v \operatorname{Res}_{\mathcal{A}_{1,v},\ldots,\mathcal{A}_{n,v}}(f_{1,v},\ldots,f_{n,v})^{-h_{\mathcal{A}_0}(v)} \prod_{\xi\in V(f_1,\ldots,f_n)} f_0(\xi),$$

where the first product runs over the primitive vectors $v \in \mathbb{Z}^n$ and $V(f_1, \ldots, f_n)$ denotes the set of common zeros in $(\overline{\mathbb{Q}(U_1, \ldots, U_n)}^*)^n$ of $f_1, \ldots, f_n$ (here the overline denotes algebraic closure). Note that all points in $V(f_1, \ldots, f_n)$ have multiplicity 1, since the system is generic and the characteristic of the base field is 0 (see, for instance, [33, Chapter V, Corollary (3.2.1)]).

Set $U := (U_1, \ldots, U_n)$ and $x^{\pm 1} := (x_1^{\pm 1}, \ldots, x_n^{\pm 1})$. For $f \in \mathbb{Q}(U_0, U)[x^{\pm 1}]$ and the $\mathbb{Q}(U_0, U)$-algebra $A = \mathbb{Q}(U_0, U)[x^{\pm 1}]/(f_1, \ldots, f_n)$, we write $\overline{f}$ for the class of f in the quotient ring A and $m_f : A \to A$ for the linear map defined as

$$m_f(g) = \overline{f}.g.$$

Lemma 7 *For $f_0 \in \mathbb{Q}(U_0, U)[x^{\pm 1}]$ we have that*

$$\mathrm{Res}_{\mathcal{A}}(U_0, U_1, \ldots, U_n) = \rho(U) \det(m_{f_0})$$

with $\rho(U) = \pm \prod_v \mathrm{Res}_{\mathcal{A}_{1,v}, \cdots, \mathcal{A}_{n,v}}(f_{1,v}, \ldots, f_{n,v})^{-h_{\mathcal{A}_0}(v)} \in \mathbb{Q}(U) \setminus \{0\}$, where the product runs over the primitive vectors $v \in \mathbb{Z}^n$.

Proof. The result follows from the Poisson formula stated above and [7, Chapter 2, Theorem (4.5)]. □

Without loss of generality, by multiplying f_0 by $x^{-\alpha}$ for any $\alpha \in \mathcal{A}_0$, we may assume that $0 \in \mathcal{A}_0$ (this does not change the resultant). Then, we have:

Lemma 8 *If $0 \in \mathcal{A}_0$, the sparse resultant $\mathrm{Res}_{\mathcal{A}}$ is the numerator of a representation as an irreducible fraction of $\det(m_{f_0}) \in \mathbb{Q}(U_0, U_1, \ldots, U_n)$.*

Proof. Note that the determinant $\det(m_{f_0})$ is in $\mathbb{Q}(U)[U_0]$ and, since it is monic in the variable U_{00}, when written as an irreducible fraction $G(U_0, U)/H(U)$, the polynomial G does not have a non-constant factor in $\mathbb{Q}[U]$. On the other hand, $\mathrm{Res}_{\mathcal{A}}$ is a power of the irreducible polynomial $\mathrm{Elim}_{\mathcal{A}}$, which depends effectively on the variables U_0. Then, from Lemma 7, taking into account that the assumption $0 \in \mathcal{A}_0$ implies that $h_{\mathcal{A}_0}(v) \leq 0$ for every v, it follows that the denominator of $\det(m_{f_0})$ is a scalar multiple of $\prod_v \mathrm{Res}_{\mathcal{A}_{1,v}, \ldots, \mathcal{A}_{n,v}}(f_{1,v}, \ldots, f_{n,v})^{-h_{\mathcal{A}_0}(v)}$ and the numerator is the corresponding scalar multiple of $\mathrm{Res}_{\mathcal{A}}$. □

Our algorithm recovers a scalar multiple of the sparse resultant $\mathrm{Res}_{\mathcal{A}}$ from a suitable approximation of the rational function $\det(m_{f_0})$ as a power series. To this end, we first compute a Taylor expansion of the determinant which approximates it with an adequate precision. Finally, we reconstruct the numerator and the denominator of $\det(m_{f_0})$ from the computed Taylor series expansion by means of Padé approximation.

We first describe the approximation step.

By multiplying each of the Laurent polynomials $f_1, \ldots, f_n$ by a suitable monomial if needed, we assume that $\mathcal{A}_i \subset (\mathbb{Z}_{\geq 0})^n$ for $i = 1, \ldots, n$. In this situation, there is an isomorphism between

$$A = \mathbb{Q}(U_0, U)[x^{\pm 1}]/(f_1, \ldots, f_n)$$

and

$$B = \mathbb{Q}(U_0, U)[x]/(f_1, \ldots, f_n) : (x_1 \cdots x_n)^\infty,$$

where $(f_1, \ldots, f_n) : (x_1 \cdots x_n)^\infty = \{g \in \mathbb{Q}(U_0, U)[x] \mid (x_1 \ldots x_n)^N g \in (f_1, \ldots, f_n) \text{ for some } N \in \mathbb{Z}_{\geq 0}\}$. For $f \in \mathbb{Q}(U_0, U)[x]$, via this isomorphism, the map $m_f : A \to A$ can be interpreted as $m_f : B \to B$.

The Newton-Hensel lifting introduced in Section 3.2 enables us to approximate $\det(m_f)$ for any $f \in \mathbb{Q}[U_0][U, x]$ from a geometric resolution of the set of common zeros in $(\mathbb{C}^*)^n$ of a generic system with supports $\mathcal{A}_1, \ldots, \mathcal{A}_n$:

Let $u_1, \ldots, u_n$ be rational vectors such that $f_1(u_1, x), \ldots, f_n(u_n, x) \in \mathbb{Q}[x]$ have $\delta := MV_n(\mathcal{A}_1, \ldots, \mathcal{A}_n)$ simple common zeros in $(\mathbb{C}^*)^n$ (by Bernstein's theorem, this holds for a generic choice of $u_1, \ldots, u_n$). Assume a geometric resolution $q, v_1, \ldots, v_n \in \mathbb{Q}[Y]$ associated to a linear form ℓ of this set of common zeros is given. Let $M \in \mathbb{Q}^{\delta \times \delta}$ be the companion matrix of q, and, for $j = 1, \ldots, n$, consider $v_j(M)$, which is the matrix of $m_{x_j} : \mathbb{Q}[x]/(f_1(u_1, x), \ldots, f_n(u_n, x)) : (x_1 \cdots x_n)^\infty \to \mathbb{Q}[x]/(f_1(u_1, x), \ldots, f_n(u_n, x)) : (x_1 \cdots x_n)^\infty$ in the basis $\{1, \ell, \ldots, \ell^{\delta-1}\}$.

Lemma 9 *With the previous assumptions and notations, consider the Newton operator $\mathcal{N}_F(x)$ associated to $F := (f_1(U_1, x), \ldots, f_n(U_n, x))$ with respect to the variables x. Let $\kappa \in \mathbb{Z}_{>0}$, and $g_1, \ldots, g_n, h \in \mathbb{Q}[U, x]$ such that the κth iteration of $\mathcal{N}_F$ is given as $\mathcal{N}_F^\kappa(x) = \left(\frac{g_1}{h}, \ldots, \frac{g_n}{h}\right)$. For $j = 1, \ldots, n$, let $\mathcal{N}_j := h(U, \mathbf{v}(M))^{-1} g_j(U, \mathbf{v}(M))$, where $\mathbf{v}(M) := (v_1(M), \ldots, v_n(M))$.*

For $f \in \mathbb{Q}[U_0][U, x]$, if $\widetilde{M_f} := f(U_0, U, \mathcal{N}_1, \ldots, \mathcal{N}_n)$, then $\det(\widetilde{M_f})$ approximates $\det(m_f)$ with precision 2^κ in the ring $\mathbb{Q}[U_0][[U - u]]$ of formal power series in $U - u = (U_1 - u_1, \ldots, U_n - u_n)$ with coefficients in $\mathbb{Q}[U_0]$, that is to say, $\det(\widetilde{M_f}) - \det(m_f)$ lies in the ideal $(U - u)^{2^\kappa + 1}$.

Proof. The result follows straightforwardly from [23, Lemma 6], □

Now, we can prove our main result.

Theorem 10 *Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_n)$, where $\mathcal{A}_i \subset \mathbb{Z}^n$ is a finite set for every $0 \leq i \leq n$. Assume the family $\mathcal{A}$ is essential. There is a probabilistic algorithm which computes a straight-line program of length $O(D^4(D^{\Omega-2} + \delta^{\omega+1} \log(D) \mathcal{Q}^2 \log(\mathcal{Q}) n^{10} N))$ for a scalar multiple of the sparse resultant $\mathrm{Res}_{\mathcal{A}}$ within complexity $O(n^3 N(\log(n) + \log(\mathcal{Q}))M(\delta)\log(\delta)M(\delta')\log(\delta') + D^4(D^{\Omega-2} + \delta^{\omega+1}\log(D)\mathcal{Q}^2\log(\mathcal{Q})n^{10}N))$, where*

- $N := \sum_{0 \leq i \leq n} |\mathcal{A}_i|$,
- $\mathcal{Q} = \max\{||a|| : a \in \mathcal{A}_i,\ 0 \leq i \leq n\}$,
- $D := \sum_{0 \leq i \leq n} MV_n(\mathcal{A}_0, \ldots, \mathcal{A}_{i-1}, \mathcal{A}_{i+1}, \ldots, \mathcal{A}_n)$,
- $\delta := MV_n(\mathcal{A}_1, \ldots, \mathcal{A}_n)$,
- $\delta' := \sum_{1 \leq i \leq n} MV_n(\Delta, \mathcal{A}_1, \ldots, \mathcal{A}_{i-1}, \mathcal{A}_{i+1}, \ldots, \mathcal{A}_n)$, *where Δ denotes the standard n-dimensional simplex.*

Proof. Without loss of generality, by multiplying by suitable monomials if needed (which does not change the resultant), we may assume that $f_1, \ldots, f_n \in \mathbb{Q}[U][x]$. We may also assume that $0 \in \mathcal{A}_0$. Then, by Lemma 8, the resultant $\mathrm{Res}_{\mathcal{A}}$ is a numerator of the rational function $\det(m_{f_0})$.

To compute it, we first choose integer vectors $u_1, \ldots, u_n$ for the coefficients of the polynomials $f_1, \ldots, f_n$ at random, and approximate $\det(m_{f_0})$ as an element of $\mathbb{Q}[U_0][[U - u]]$, where $U - u = (U_1 - u_1, \ldots, U_n - u_n)$. We determine the order of approximation required to apply Lemma 6 in terms of the degrees of the numerator and the denominator.

From Lemmas 7 and 8, we have that a denominator of $\det(m_{f_0})$ is

$$\prod_v \mathrm{Res}_{\mathcal{A}_{1,v},\ldots,\mathcal{A}_{n,v}}(f_{1,v}, \ldots, f_{n,v})^{-h_{\mathcal{A}_0}(v)} \in \mathbb{Q}(U) \setminus \{0\},$$

where the product runs over the primitive vectors $v \in \mathbb{Z}^n$. By the known formula for the degrees of sparse resultants (see equation (2)), for every $1 \leq i \leq n$, this polynomial is homogeneous in the variables U_i of degree

$$-\sum_v h_{\mathcal{A}_0}(v) MV(\mathcal{A}_{1,v}, \ldots, \mathcal{A}_{i-1,v}, \mathcal{A}_{i+1,v}, \ldots, \mathcal{A}_{n,v}) = MV_n(\mathcal{A}_0, \ldots, \mathcal{A}_{i-1}, \mathcal{A}_{i+1}, \ldots, \mathcal{A}_n),$$

where the equality follows from [15, Chapter IV, Theorem 4.10]. Recalling that the numerator of $\det(m_{f_0})$ is $\mathrm{Res}_{\mathcal{A}}$, it follows that $\det(m_{f_0})$ is a quotient of two polynomials in $\mathbb{Q}[U_0, \ldots, U_n]$ of total degrees $\sum_{0 \leq i \leq n} MV_n(\mathcal{A}_0, \ldots, \mathcal{A}_{i-1}, \mathcal{A}_{i+1}, \ldots, \mathcal{A}_n) = D$ and $\sum_{1 \leq i \leq n} MV_n(\mathcal{A}_0, \ldots, \mathcal{A}_{i-1}, \mathcal{A}_{i+1}, \ldots, \mathcal{A}_n) \leq D$ respectively.

Therefore, by Lemma 6, we may recover the numerator and the denominator of $\det(m_{f_0})$ from its Taylor expansion up to order $2D$ centered at $(0, u)$, where 0 is the center corresponding to the variables U_0 and $u = (u_1, \ldots, u_n)$. We compute this Taylor expansion following the Newton-Hensel lifting approach described above. To this end, we will deal with multiplication maps associated to *polynomials* in the variables x.

If $\alpha_{0j} := -\min\{0; (a_{0,k})_j \mid a_{0,k} \in \mathcal{A}_0\}$ and $\alpha_0 := (\alpha_{01}, \ldots, \alpha_{0n})$, we have that $\widetilde{f_0} := x^{\alpha_0} f_0 \in \mathbb{Q}[U_0][x]$. From the definition of $\widetilde{f_0}$, it follows that the linear maps $m_{\widetilde{f_0}}$, $m_{x^{\alpha_0}}$ and m_{f_0} defined over $\mathbb{Q}(U_0, U)[x^{\pm 1}]/(f_1, \ldots, f_n)$ satisfy

$$m_{\widetilde{f_0}} = m_{x^{\alpha_0}} \circ m_{f_0},$$

and so,

$$\det(m_{f_0}) = \det(m_{x^{\alpha_0}})^{-1} \det(m_{\widetilde{f_0}}).$$

As $\widetilde{f_0}$ and x^{α_0} are polynomials, we will work in $\mathbb{Q}(U_0, U)[x]/(f_1, \ldots, f_n) : (x_1 \cdots x_n)^{\infty}$.

The algorithm underlying Lemma 4 allows us to compute a geometric resolution $q, v_1, \ldots, v_n$ associated with a linear form $\ell \in \mathbb{Q}[x]$ of the set V of common zeros in $(\mathbb{C}^*)^n$ of the system $f_1(u_1, x), \ldots, f_n(u_n, x)$. Then, by Lemma 9 applied to the polynomials $\widetilde{f_0}$ and x^{α_0} for $\kappa = \lceil \log(2D + 1) \rceil$, we can obtain approximations of $\det(m_{\widetilde{f_0}})$ and $\det(m_{x^{\alpha_0}})$ with precision $2D$ in $\mathbb{Q}[U_0][[U - u]]$:

Let $g_1, \ldots, g_n, h \in \mathbb{Q}[U, x]$ be the polynomials appearing in the κth iteration of the Newton operator $\mathcal{N}_F(x)$ associated with $F = (f_1(U_1, x), \ldots, f_n(U_n, x))$. Let $M \in \mathbb{Q}^{\delta \times \delta}$ be the companion matrix of the polynomial q and $\mathbf{v}(M) = (v_1(M), \ldots, v_n(M))$. Consider the matrices

$$H := h(U, \mathbf{v}(M)) \quad \text{and} \quad G_j := g_j(U, \mathbf{v}(M)), \text{ for } 1 \leq j \leq n.$$

Then, as consequence of Lemma 9, we have that

$$\det(H)^{-|\alpha_0|} \prod_{1\leq j\leq n} \det(G_j)^{\alpha_{0j}}$$

approximates $\det(m_{x^{\alpha_0}})$ and also that, if $M_0 = \widetilde{f}_0^{hom}(U_0, H, G_1, \ldots, G_n)$, where $\widetilde{f}_0^{hom} \in \mathbb{Q}[U_0, x_0, x]$ is the polynomial obtained by homogeneizing $\widetilde{f}_0$ up to degree $\delta_0 := \deg(\widetilde{f}_0)$ with a new variable x_0,

$$\det(H)^{-\delta_0} \det(M_0)$$

approximates $\det(m_{\widetilde{f}_0})$. Therefore, we approximate $\det(m_{f_0})$ with

$$\det(H)^{|\alpha_0|-\delta_0} \prod_{1\leq j\leq n} \det(G_j)^{-\alpha_{0j}} \det(M_0). \tag{3}$$

Finally, we recover the numerator and the denominator of $\det(m_{f_0})$ and, therefore, a scalar multiple of the desired resultant, applying Lemma 6. To this end, we first obtain the homogeneous components of the Taylor expansion of order $2D$ centered at $(0, u)$ of the rational function in (3). Depending on whether $|\alpha_0| - \delta_0 \geq 0$ or not, we regard this rational function as the quotient of the polynomials

$$\det(H)^{|\alpha_0|-\delta_0} \det(M_0) \quad \text{and} \quad \prod_{1\leq j\leq n} \det(G_j)^{\alpha_{0j}},$$

or of the polynomials

$$\det(M_0) \quad \text{and} \quad \det(H)^{\delta_0-|\alpha_0|} \prod_{1\leq j\leq n} \det(G_j)^{\alpha_{0j}},$$

and, in order to obtain a polynomial that approximates the inverse of the denominator with precision $2D$ in $\mathbb{Q}[[U - u]]$, we use the formula in [23, p. 99]. This completes the description of the algorithm and the proof of its correctness.

Algorithm `SparseResultant` below summarizes the procedure. Now, we estimate its complexity.

Step 1 involves computing $a_{0,k} - a_{0,0}$ for $k = 1, \ldots, N_0$, which can be done with nN_0 operations.

For $i = 0, \ldots, n$, the vector α_i in Step 2 can be computed within complexity $O(nN_i)$.

After the multiplication by monomials in Step 3, we have that $\max\{||a_{i,k} + \alpha_i|| : 1 \leq i \leq n, 0 \leq k \leq N_i\} \leq (n+1)\mathcal{Q}$. Then, the computation of the geometric resolution $q, v_1, \ldots, v_n$ in Step 5 can be done, by Lemma 4, within complexity

$$O(n^3 N(\log(n) + \log(\mathcal{Q}))M(\delta)\log(\delta)(M(\delta)\log(\delta) + M(\delta')\log(\delta'))).$$

As $M \in \mathbb{Q}^{\delta\times\delta}$ and $\deg(v_j) \leq \delta$ for $j = 1, \ldots, n$, then the matrices $\mathbf{v}(M)$ in Step 7 can be computed by means of an slp of length $O(n\delta^{\omega+1})$.

Since the degree in each variable x_j of the polynomials f_i defined in Step 3 is bounded by $2\mathcal{Q}$, we can obtain an slp of length $O(nN\log(\mathcal{Q}))$ encoding $f_1, \ldots, f_n \in \mathbb{Q}[U, x]$. Then, taking into account that the total degree in x of the polynomials f_i is at most $(n+1)Q$,

by Lemma 5, we compute an slp of length $O(\log(D)\mathcal{Q}^2\log(\mathcal{Q})n^{10}N)$ for the polynomials $g_1,\ldots,g_n,h$ appearing in the κth iteration of the Newton operator associated with $f_1,\ldots,f_n$, for $\kappa=\lceil\log(2D+1)\rceil$ (Step 8), within a complexity of the same order as the slp length.

From the slp computing $\mathbf{v}(M)$ and the slp for $g_1,\ldots,g_n,h$, we easily obtain an slp of length $O(n\delta^{\omega+1}+\delta^3\log(D)\mathcal{Q}^2\log(\mathcal{Q})n^{10}N)$ evaluating the entries of the matrices $G_1,\ldots,G_n$, H in Step 9.

Now, we obtain an slp of length $O(nN_0\log(\mathcal{Q}))$ evaluating the polynomial $\widetilde{f}_0^{hom}$ (Step 11) and, from this slp and the slp computing the matrices $G_1,\ldots,G_n,H$, we get an slp for the entries of the matrix M_0 in Step 12. The total length of this slp is $O(n\delta^{\omega+1}+\delta^3\log(D)\mathcal{Q}^2\log(\mathcal{Q})n^{10}N)$.

The determinant of each of the matrices $G_1,\ldots,G_n,H$ and M_0 with polynomial entries can be computed without divisions by means of an slp of length $O(\delta^\Omega)$ from an slp encoding their entries. Then, as $\Omega\leq\omega+1$ (see Section 2.3), we obtain an slp of length $O(n\delta^{\omega+1}+\delta^3\log(D)\mathcal{Q}^2\log(\mathcal{Q})n^{10}N)$ evaluating Φ_{Num} and Φ_{Den} in Step 14 or Step 16. Computing a polynomial that approximates the inverse of Φ_{Den} with precision $2D$ in $\mathbb{Q}[[U-u]]$ following [23, p. 99] (Step 17) increases the slp length in $O(\log(D))$.

By applying a standard procedure which from an slp evaluating a polynomial computes an slp evaluating its homogeneous components up to a given degree (see, for instance, [3, Lemma 21.25]), in Step 18 we obtain an slp of length $O(D^2(n\delta^{\omega+1}+\delta^3\log(D)\mathcal{Q}^2\log(\mathcal{Q})n^{10}N))$. Finally, following Lemma 6, in Step 19, we obtain an slp of length

$$O(D^2(D^\Omega+D^2(n\delta^{\omega+1}+\delta^3\log(D)\mathcal{Q}^2\log(\mathcal{Q})n^{10}N)))$$

encoding the numerator and denominator of $\det(m_{f_0})$ and, therefore, the desired resultant.

The overall complexity of the algorithm follows by adding the complexity of its successive steps. □

The notation and subroutines involved in Algorithm `SparseResultant` are the following:

- `Vects`$(n;K_1,\ldots,K_n)$ stands for n randomly chosen integer vectors of K_i coordinates for $i=1,\ldots,n$.
- `GeometricResolution`$(h_1,\ldots,h_n)$ computes a geometric resolution of the zero set in $(\mathbb{C}^*)^n$ of the sparse system $h_1,\ldots,h_n\in\mathbb{Q}[x]$ in n variables.
- `NumDenNewton`$(f_1,\ldots,f_n;x,\kappa)$ computes numerators and a denominator for the κth iteration of the Newton operator associated to $f_1,\ldots,f_n$ with respect to the variables x.
- `Inverse`$(\Psi;z,\nu)$ computes an approximation with precision ν in $\mathbb{Q}[[Z-z]]$ of the inverse of $\Psi\in\mathbb{Q}[Z]$ provided that $\Psi(z)\neq 0$.
- `GradedParts`$(\Psi;z,\nu)$ computes the graded parts of $\Psi\in\mathbb{Q}[Z]$ centered at z up to degree ν.
- `Pade`$(\Phi;z,2\nu)$ computes the $(\nu+1,\nu)$-Padé approximant of the function whose Taylor expansion centered at z has graded parts Φ.

Algorithm `SparseResultant`$(n, \mathcal{A}, D)$

\# $n \in \mathbb{Z}_{>0}$
\# $\mathcal{A} = (\mathcal{A}_0, \ldots, \mathcal{A}_n) \subset (\mathbb{Z}^n)^{n+1}$ an essential family, where $\mathcal{A}_i = \{a_{i,0}, \ldots, a_{i,N_i}\}$
\# $D = \sum_{0 \le i \le n} MV_n(\mathcal{A}_0, \ldots, \mathcal{A}_{i-1}, \mathcal{A}_{i+1}, \ldots, \mathcal{A}_n)$
\# The procedure returns a scalar multiple of the $\mathcal{A}$-resultant.

1. $f_0 := x^{-a_{0,0}} \sum_{k=0}^{N_0} U_{0,k}\, x^{a_{0,k}}$; \# 0 **is in the support of** f_0
2. for $i = 0, \ldots, n$, $j = 1, \ldots, n$,
   $\alpha_{ij} := -\min\{0; (a_{i,k})_j \mid a_{i,k} \in \mathcal{A}_i\}$, and $\alpha_i := (\alpha_{i1}, \ldots, \alpha_{in})$;
3. for $i = 1, \ldots, n$, $f_i := x^{\alpha_i} \sum_{k=0}^{N_i} U_{i,k}\, x^{a_{i,k}}$; \# f_i **is a polynomial in** x
4. $(u_1, \ldots, u_n) :=$ `Vects`$(n; N_1 + 1, \ldots, N_n + 1)$;
5. $(q, v_1, \ldots, v_n) :=$ `GeometricResolution`$(f_1(u_1, x), \ldots, f_n(u_n, x))$;
6. $M :=$ `CompanionMatrix`(q);
7. $\mathbf{v}(M) := (v_1(M), \ldots, v_n(M))$;
8. $(g_1, \ldots, g_n, h) :=$ `NumDenNewton`$(f_1, \ldots, f_n; x, \lceil \log(2D+1) \rceil)$;
9. $(G_1, \ldots, G_n, H) := (g_1(U, \mathbf{v}(M)), \ldots, g_n(U, \mathbf{v}(M)), h(U, \mathbf{v}(M)))$;
10. $\delta_0 := \max\{|a_{0,k} + \alpha_0| : 0 \le k \le N_0\}$;
11. $\widetilde{f}_0^{hom}(U_0, x_0, x) := \sum_{k=0}^{N_0} U_{0,k}\, x_0^{\delta_0 - |a_{0,k}+\alpha_0|}\, x^{a_{0,k}+\alpha_0}$;
12. $M_0 := \widetilde{f}_0^{hom}(U_0, H, G_1, \ldots, G_n)$;
13. if $|\alpha_0| \ge \delta_0$ then
14. $\quad (\Phi_{Num}, \Phi_{Den}) := (\det(H)^{|\alpha_0| - \delta_0} \det(M_0), \prod_{1 \le j \le n} \det(G_j)^{\alpha_{0j}})$;
15. else
16. $\quad (\Phi_{Num}, \Phi_{Den}) := (\det(M_0), \det(H)^{\delta_0 - |\alpha_0|} \prod_{1 \le j \le n} \det(G_j)^{\alpha_{0j}})$;
17. $\Phi_{Den}^{inv} :=$ `Inverse`$(\Phi_{Den}; u, 2D)$;
18. $\Phi :=$ `GradedParts`$(\Phi_{Num}\Phi_{Den}^{inv}; (0, u), 2D)$;
19. $(P, Q) :=$ `Pade`$(\Phi; (0, u), 2D)$;
20. return(P);

Acknowledgement. The authors wish to thank the referees for their careful reading of the paper and helpful suggestions.

References

[1] D. N. Bernstein, The number of roots of a system of equations. Functional Anal. Appl. 9 (1975), no. 3, 183–185.

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

[3] B. Bürgisser, M. Clausen, M.A. Shokrollahi, Algebraic Complexity Theory. Springer-Verlag, 1997.

[4] J. F. Canny, I. Z. Emiris, An efficient algorithm for the sparse mixed resultant. In: Cohen, G., Mora, T., Moreno, O. 35 (Eds.), Proc. Int. Symp. on Appl. Algebra, Algebraic Algorithms and Error-Corr. Codes. Puerto Rico. In: LNCS, vol. 36, 263 (1993), 89–104.

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

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

[7] D. Cox, J. Little, D. O'Shea, Using algebraic geometry. Graduate Texts in Mathematics 185. Second Edition. Springer-Verlag, New York, 2005.

[8] C. D'Andrea, Macaulay style formulas for sparse resultants. Trans. Amer. Math. Soc. 354 (7) (2002), 2595–2629.

[9] C. D'Andrea, A. Dickenstein, Explicit formulas for the multivariate resultant. J. Pure Appl. Algebra 164 (12) (2001), 59–86.

[10] C. D'Andrea, G. Jeronimo, M. Sombra, Sparse resultants: combinatorial properties and Macaulay style formulas, in preparation.

[11] C. D'Andrea, M. Sombra, A Poisson formula for the sparse resultant. Proc. Lond. Math. Soc. (3) 110 (2015), no. 4, 932–964.

[12] M. Dyer, P. Gritzmann, A. Hufnagel, On the complexity of computing mixed volumes. SIAM J. Comput. 27 (1998) (2), 356–400.

[13] I.Z. Emiris, B. Mourrain, Matrices in elimination theory. J. Symbolic Comput. 28 (1999), 3–44.

[14] A. Esterov, Newton polyhedra of discriminants of projections. Discrete Comput. Geom. 44 (2010), no. 1, 96–148.

[15] G. Ewald, Combinatorial Convexity and Algebraic Geometry. Grad. Texts in Math., vol. 168, Springer, New York, 1996.

[16] J. von zur Gathen, J. Gerhard, Modern computer algebra. Cambridge University Press, New York, 1999.

[17] I. M. Gelfand, M. M. Kapranov, A. V. Zelevinsky, Discriminants, resultants, and multidimensional determinants. Mathematics: Theory & Applications. Birkhäuser Boston, Inc., Boston, MA, 1994.

[18] M. Giusti, J. Heintz, La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial. In: Computational Algebraic Geometry and Commutative Algebra (Cortona, 1991). Sympos. Math. XXXIV, Cambridge Univ. Press, Cambridge (1993), 216–256.

[19] M. Giusti, J. Heintz, K. Hägele, J. E. Morais, L. M. Pardo, J. L. Montaña, Lower bounds for Diophantine approximations. Algorithms for algebra (Eindhoven, 1996). J. Pure Appl. Algebra 117/118 (1997), 277–317.

[20] M. Giusti, J. Heintz, J. E. Morais, J. Morgenstern, L. M. Pardo, Straight-line programs in geometric elimination theory. J. Pure Appl. Algebra 124 (13) (1998), 101–146.

[21] M. Giusti, G. Lecerf, B. Salvy, A Gröbner free alternative for polynomial system solving. J. Complexity 17 (2001), no. 1, 154-211.

[22] B. Grenet, P. Koiran, N. Portier, On the complexity of the multivariate resultant. J. Complexity 29 (2013), 142–157.

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

[24] J. Heintz, C. P. Schnorr, 1982. Testing polynomials which are easy to compute. In: Monographie 30 de l'Enseignement Mathématique, 237–254.

[25] G. Jeronimo, T. Krick, J. Sabia, M. Sombra, The computational complexity of the Chow form. Found. Comput. Math. 4 (2004), no. 1, 41–117.

[26] G. Jeronimo, G. Matera, P. Solernó, A. Waissbein, Deformation techniques for sparse systems. Found. Comput. Math. 9 (2009), no. 1, 1–50.

[27] G. Jeronimo, J. Sabia, Computing multihomogeneous resultants using straight-line programs. J. Symbolic Comput. 42 (2007), 218–235.

[28] E. Kaltofen, G. Villard, On the complexity of computing determinants. Comput. Complexity 13 (2004), no. 3-4, 91–130.

[29] E. Kaltofen, P. Koiran, Expressing a fraction of two determinants as a determinant. Proceedings ISSAC 2008, 141–146.

[30] G. Lecerf, GEOMSOLVEX, a MATHEMAGIX library for geometric polynomial system solving, 2012,
`http://www.mathemagix.org/www/geomsolvex/doc/html/index.en.html`.

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

[32] M. Minimair, Sparse resultant under vanishing coefficients. J. Algebraic Combin. 18 (2003), 53–73.

[33] M. Oka, Non-degenerate complete intersection singularity. Actualités Mathématiques. Hermann, Paris, 1997.

[34] P. Pedersen, B. Sturmfels, Product formulas for resultants and Chow forms. Math. Z. 214 (1993), no. 3, 377–396.

[35] E. Schost, Computing parametric geometric resolutions. Appl. Algebra Engrg. Comm. Comput. 13 (2003), no. 5, 349–393.

[36] J. Schwartz, Fast probabilistic algorithms for verification of polynomial identities. J. ACM 27 (1980), 701–717.

[37] A. Storjohann, Algorithms for matrix canonical forms, Ph.D. thesis, ETH, Zürich, Switzerland, 2000.

[38] V. Strassen, Vermeidung von Divisionen. J. Reine Angew. Math. 264 (1973), 184–202.

[39] B. Sturmfels, Sparse elimination theory. In: Eisenbud, D., Robbbiano, L. (Eds.), Computational Algebraic 40 Geometry and Commutative Algebra (Cortona, 1991). In: Sympos. Math. XXXIV, Cambridge Univ. Press, 41 (1993), 264–298.

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

[41] 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. Philos. Trans. 143 (1853), 407–548.

[42] R. Zippel, Effective Polynomial Computation. Kluwer Int. Ser. Eng. Comput. Sci., vol. 241 (1993), Kluwer, Dordrecht.