Computing the Equidimensional Decomposition of an Algebraic Closed Set by means of Lifting Fibers

Grégoire Lecerf

Laboratoire de Mathématiques, UMR 8100 CNRS
Université de Versailles St-Quentin-en-Yvelines
45, avenue des États-Unis, Bâtiment Fermat
78035 Versailles, France
Gregoire.Lecerf@math.uvsq.fr

April 23, 2003

We present a new probabilistic method for solving systems of polynomial equations and inequations. Our algorithm computes the equidimensional decomposition of the Zariski closure of the solution set of such systems. Each equidimensional component is encoded by a generic fiber, that is a finite set of points obtained from the intersection of the component with a generic transverse affine subspace. Our algorithm is incremental in the number of equations to be solved. Its complexity is mainly cubic in the maximum of the degrees of the solution sets of the intermediate systems counting multiplicities.

Our method is designed for coefficient fields having characteristic zero or big enough with respect to the number of solutions. If the base field is the field of the rational numbers then the resolution is first performed modulo a random prime number after we have applied a random change of coordinates. Then we search for coordinates with small integers and lift the solutions up to the rational numbers. Our implementation is available within our package Kronecker from version 0.166, which is written in the Magma computer algebra system.

1 Introduction

Introduced by H. Hironaka in the middle of the sixties, the concept of a standard basis of an ideal in a polynomial ring has become a topic of particular interest in mathematics and computer science since B. Buchberger's work. Nowadays the effective construction of such a basis is an essential functionality in all computer algebra systems. The subjacent algorithms are unceasingly improved and make it possible to deal with concrete problems inaccessible to numerical methods. And yet the complexity of these algorithms is doubly exponential in the worst case. In the nineties, M. Giusti and J. Heintz showed that elimination problems can be brought back in a polynomial complexity class by representing the eliminating polynomials by straight-line programs. On the basis of their

work, this paper leads to a probabilistic algorithm to compute the decomposition into equidimensional components of the solution set of a system of polynomial equations and inequations. Turning into practice the concept of generic point of an algebraic variety is the key of this new algorithm.

1.1 Main Result

Let k be a field of characteristic zero, we denote by $\overline{k}$ its algebraic closure. Let $f_1, \ldots, f_s, g$ be polynomial functions in $k[x_1, \ldots, x_n]$ given by a *straight-line program* of size L. The solution set in $\overline{k}^n$ of the system

$$f_1 = \cdots = f_s = 0, \quad g \neq 0$$

is a constructible set, we are interested in describing the k-algebraic variety $\mathcal{V}$ defined as the closure of this set with respect to the Zariski topology. More precisely the algorithm we present here computes the equidimensional decomposition of $\mathcal{V}$.

Our method is incremental in the number of equations to be solved. Therefore its complexity depends on the number of solutions of the intermediate systems. For i from 0 to s, we introduce the ith *intermediate system* as the system:

$$f_1 = \cdots = f_i = 0, \quad g \neq 0.$$

We denote by $\mathcal{V}_i$ the k-algebraic variety obtained from the closure (for the Zariski topology) of the set of roots of the ith intermediate system in $\overline{k}^n$:

$$\mathcal{V}_i := \overline{\{z \in \overline{k}^n |\ f_1(z) = \cdots = f_i(z) = 0,\ g(z) \neq 0\}}, \quad i = 0, \ldots, s.$$

Note that if g is not the zero polynomial then $\mathcal{V}_0$ is $\overline{k}^n$. Our algorithm computes the equidimensional decompositions of the $\mathcal{V}_i$ in sequence for $i = 0, \ldots, s$. Each equidimensional component is represented by a set of *lifting fibers* (see definitions in §2.3 and §3.2). Each lifting fiber encodes an equidimensional variety. Our representation is not redundant in the following sense: an irreducible component of a variety represented by a lifting fiber can not be included in a variety represented by another fiber.

If $\mathcal{W}$ is an irreducible component of $\mathcal{V}_i$ we write $\mathrm{mul}(\mathcal{W}; f_1, \ldots, f_i)$ the multiplicity of the generic point of $\mathcal{W}$ as a solution of the system $f_1 = \cdots = f_i = 0$ (cf. §2.2). We denote by $\deg^{\mathrm{a}}(\mathcal{W}; f_1, \ldots, f_i)$, and call it the *algebraic degree* of $\mathcal{W}$ with respect to $f_1, \ldots, f_i$, the product $\mathrm{mul}(\mathcal{W}; f_1, \ldots, f_i)\deg(\mathcal{W})$, where $\deg(\mathcal{W})$ stands for the classical geometric degree of $\mathcal{W}$. By extension we define $\deg^{\mathrm{a}}(\mathcal{V}_i; f_1, \ldots, f_i)$ as the sum of the algebraic degrees of the irreducible components of $\mathcal{V}_i$. We denote by δ_i^a this last quantity:

$$\delta_i^a := \deg^{\mathrm{a}}(\mathcal{V}_i; f_1, \ldots, f_i).$$

The crucial quantity appearing in the complexity of our algorithm is the maximum δ^a of the δ_i^a, namely: $\delta^a := \max_{i=0,\ldots,s} \delta_i^a$. We introduce the function $\mathcal{U}$, that is used for complexity estimates from §2.1 and that dominates the complexity of the basic arithmetic

operations for univariate polynomials in degree at most z (multiplication, division and greatest common divisor):

$$\mathcal{U}(z) := z \log^2(z) \log\log(z).$$

We also use the constant Ω that is a positive real number bigger than 3 and smaller than 4: it is related to the complexity of the linear algebra over a ring as recalled in §2.1. We let d be the maximum of the degrees of the f_i and recall that L is the evaluation complexity of $f_1, \ldots, f_s, g$. The aim of this paper is the description of an algorithm that yields the following complexity result:

Theorem 1 *Let k be a field of characteristic zero. There exists a probabilistic algorithm taking as input a sequence $f_1, \ldots, f_n, g$ of polynomials in $k[x_1, \ldots, x_n]$ of degree at most d and given by a straight-line program of size at most L. The output is the equidimensional decomposition of the Zariski closure of the system*

$$f_1 = \cdots = f_s = 0, \quad g \neq 0.$$

In case of success, the procedure requires

$$\mathcal{O}\Big(s \log(d) n^4 (nL + n^\Omega) \mathcal{U}(d\delta^a)^3\Big),$$

arithmetic operations in k. Equidimensional components are encoded by a set of lifting fibers. The probability of success of the algorithm depends on the choice of a point in $k^{n^{\mathcal{O}(1)}}$: there exists a Zariski open set of points that yield a correct answer.

The above statement is subject to the following observations. First the multiplicities of the components are not computed. The algorithm does not need to know them either. Eventually it provides lower bounds on them. Moreover we do not know how to detect the cases of failure. For the sake of simplicity we chose a presentation in which the algorithm may not stop in very bad situations. We could remedy this using Bézout's inequality to compute upper bounds on the number of solutions of each intermediate system. In case of failure the worst case complexity may be higher than in case of success but we do not address this problem in this paper.

Since bad choices of points in $k^{n^{\mathcal{O}(1)}}$ are enclosed in an algebraic hypersurface the probability of failure is very low, what is confirmed in practice by our implementation.

1.2 Related Results

Let k be a field of characteristic zero and $f_1, \ldots, f_s$ be polynomials in $k[x_1, \ldots, x_n]$. We denote by $\mathcal{V}$ the $\overline{k}$-algebraic variety solution of $f_1 = \cdots = f_s = 0$. We discuss the main results concerning the computation of equidimensional decompositions.

Equidimensional decomposition For computing the equidimensional decomposition of $\mathcal{V}$, the best known deterministic algorithms have a complexity upper bound asymptotically polynomial in sd^{n^2} in terms of the number of arithmetic operations in k: in [CG83, Chi96, Chi97] Chistov and Grigoriev present algorithms with such a complexity for computing a decomposition of $\mathcal{V}$ into irreducible components; Giusti and Heintz give in [GH91] another method with the same complexity, moreover their decomposition into equidimensional components is well-parallelizable; We refer to [Vor99] for a more detailed historical presentation. Elkadi and Mourrain propose in [EM99] a method based on Bézoutian matrices, their algorithm is probabilistic and has complexity also polynomial in sd^{n^2}.

Roughly speaking sd^{n^2} is a lower bound for this problem if we represent multivariate polynomials of the output by vectors of their coefficients (dense or sparse representation as defined in [DST87]). This fact comes from the following observation: take s as the integer part of $n/2$ and consider random enough polynomials of degree $d \geq 2$. Then any eliminant polynomial in the ideal $(f_1, \ldots, f_s) \cap k[x_1, \ldots, x_{n-s+1}]$ has degree d^s (by Bézout inequality and the choice of random polynomials) and therefore the number of monomials of such an eliminant polynomial is in $d^{\mathcal{O}(n^2)}$ (for fixed d and when n tends to infinity).

From a numerical point of view, Sommese, Verschelde and Wampler propose in [SV00, SVW01a, SVW01b, SVW01c, SVW02] numerical equidimensional and irreducible decomposition algorithms based on homotopy continuation. Some of their techniques are very similar to ours but in an archimedian framework. Nowadays the complexity of their full solver is still unknown. In this vein, for the first time in [CHMP01, Cas01] Castro, Hägele, Morais and Pardo compare the efficiency of symbolic and numerical analysis procedures for polynomial system solving using the *approximate zero* theory.

In [Lec00] we propose a first breakthrough: eliminant polynomials are encoded by means of straight-line programs and algebraic varieties by *geometric resolutions*, we provide a complexity mainly polynomial in d^n. The algorithm is probabilistic with a uniform bounded error probability. Another independent approach is proposed by Jeronimo, Puddu and Sabia in [JS00, JPS01, JS02]: the main difference is that each output component is described as the set of roots of $n+1$ polynomials encoded by straight-line programs. Another algorithm for computing Chow forms of the components is proposed in [JKSS02]: it still relies on incremental solving and it is based on a new process for computing Chow forms from geometric resolutions.

A by-product of an equidimensional decomposition is the dimension but there exist direct algorithms: the dimension can be computed in cost sd^n. In [GH93] Giusti and Heintz propose a well-parallelizable algorithm polynomial in sd^n, deterministic in a non-uniform complexity model and probabilistic in a uniform one; in 1996 Chistov [Chi96, Chi97] performs the same computation within the same complexity but deterministically for a uniform complexity model. On the basis of Koiran's derandomization methods [Koi97] Rojas proposes in [Roj00] a deterministic algorithm for computing the dimension using toric resultant [Stu94, GKZ94] within a complexity polynomial in a certain mixed volume (therefore polynomial in d^n).

Evaluation techniques and lower bounds Our method is in the continuation of a series of papers initiated by Giusti, Hägele, Heintz, Krick, Matera, Montaña, Morais, Morgenstern and Pardo. The notion of geometric resolution has been introduced in [GHMP95, Par95] and the first resolution procedure for reduced regular sequences appears in 1997: In [GHH$^+$97, GHMP97, GHM$^+$98] it is presented an algorithm for computing the roots of a system of polynomial equations with a complexity polynomial in a certain degree that is intrinsic to incremental methods. In [KP96, Mor97, HMPS00] the method is generalized in order to compute the isolated roots of any system of polynomial equations and inequations.

In [Mat99, HMW01] the algorithm is revisited and improved, the exponents in the complexity estimates are detailed. In [GLS01] we simplify, redesign, improve more the algorithm, and we also explain how to implement it. The purpose of this paper is to extend this last algorithm in a natural way in order to compute not only the isolated roots but a description of all the equidimensional components. A good historical and extended presentation of these works can be found in [CGH$^+$].

In [GH01, CGH$^+$], Castro, Giusti, Heintz, Matera and Pardo explain why *universal elimination procedures* require exponential running time in worst case. Informally speaking, it follows from these results that resolution algorithms based on evaluation techniques are polynomially optimal in worst case.

1.3 Contributions

Let us recall that the algorithm presented in [GLS01] can only solve the very particular but generic situation called *reduced regular*: according to the notation of the beginning, this corresponds to the case when $s = n$, $\mathcal{V}_i$ is equidimensional of codimension i (*regularity hypothesis*) and the Jacobian matrix of $f_1, \ldots, f_i$ has full rank when evaluated at the generic points of $\mathcal{V}_i$ (*reduction hypothesis*), for $i = 1, \ldots, n$. In this paper we stay stick to the same approach: incremental solving and encoding of equidimensional varieties by means of lifting fibers.

The first difficulty we overcame is the following: if the reduction hypothesis fails then a certain Jacobian matrix is degenerated and a certain associated Newton operator is not applicable. The solution we propose comes from [Lec02], where we provide a generalization of the Newton operator that is well suited to our solver. The regularity hypothesis is easier to remove, this is the purpose of §4: we present a minimization process which ensures that our representations of equidimensional decompositions are not redundant.

In [JS00, Lec00, JKSS02] the decomposition algorithms rely on Bertini's first theorem, which demands to replace the original polynomials by generic linear combinations of them. The main drawback is that the resulting solver is not incremental. Moreover if only one polynomial is difficult to evaluate then the combination spoils the complexity of each equation. Last the deforestation, in the sense of [GHL$^+$00], of these algorithms yields implementations far away from [GLS01]. These are the reasons why we were motivated by the deflation techniques of [Lec02].

In the same way as in [GLS01], if k is the field of the rational numbers the resolution is first computed modulo a small prime number p, of size about 64 bits. Then we lift the solutions in the ring of the p-adic numbers and reconstruct the rational numbers. But before lifting the integers it is important to find a fiber of small height. This is the aim of §5.4. The algorithm presented here has been implemented in our Magma [Mag, BC95, BCM94, CP96, BCP97] package called Kronecker [Lec99] from version 0.166 and available at http://kronecker.medicis.polytechnique.fr.

2 Complexity Model and Data Structure

One key feature of the geometric resolution algorithms based on evaluation techniques is an effective use of the *Noether Normalization Lemma* also called geometrically the *Noether Position*. This technique allows to represent a positive dimensional variety by a zero-dimensional one. First we explain the complexity model we use and recall a few well-known results. Then we recall the definitions of geometric resolution and a lifting fiber.

2.1 Complexity Model

Let k be an effective field, a *straight-line program* Γ encoding a set of polynomials $f_1, \ldots, f_s$ of $k[x_1, \ldots, x_n]$ is a data structure representing an evaluation scheme for $f_1, \ldots, f_s$. Such a program is composed of a sequence of elementary instructions. Each instruction performs only one basic binary arithmetic operation (addition, subtraction or multiplication). The inputs of Γ are the variables $x_1, \ldots, x_n$, the outputs are the values of $f_1, \ldots, f_s$. We denote by L the complexity of Γ, defined as the number of instructions of the program Γ. For standard terminology about straight-line program we refer to [BCS97].

As a complexity model we use the unit cost measure, i.e. each arithmetic operation (multiplication, addition, division) of the ground field is counted as one. In the sequel we use the function $\mathcal{U}(z)$ that denotes

$$\mathcal{U}(z) := z \log^2(z) \log\log(z).$$

This function dominates the complexity of the arithmetic operations (addition, multiplication, division, greatest common divisor) with polynomials of degrees at most z in terms of number of operations in the base ring or field. The quantity $\mathcal{U}(z)$ also dominates the bit-complexity of the arithmetic operations (addition, multiplication, quotient, remainder and greatest common divisor) of the integers of bit-size at most z.

The constant Ω we use along this paper is a number bigger than 3 and such that the adjoint and the determinant of a $n \times n$ matrix over a k-algebra can be done in $\mathcal{O}(n^\Omega)$ arithmetic operations in the algebra. If k has characteristic zero (or big enough with respect to n) then we can take $\Omega = 3$ (according to the results from [CW90] and [PS78]), but the underlying algorithm is difficult to implement and is not the most efficient in our range of applications (n at most 15). Hence Ω is about 4 in practice.

Many authors have contributed to these topics. Some very good historical presentations can be found in the books by Aho, Hopcroft, Ullman [AHU74], Bürgisser, Clausen, Shokrolahi [BCS97], Bini, Pan [BP94], von zur Gathen and Gerhard [GG99], among others. A short presentation can also be found in [GLS01, §3.5].

2.2 Geometric resolutions

In this section k denotes a field of characteristic 0. Let $x_1, \ldots, x_n$ be indeterminates over k and $\mathcal{W}$ be a r-equidimensional k-variety in $\overline{k}^n$, where $\overline{k}$ denotes the algebraic closure of k. We call $\mathfrak{I}$ the annihilating ideal of $\mathcal{W}$ in $k[x_1, \ldots, x_n]$:

$$\mathfrak{I} := \{f \in k[x_1, \ldots, x_n], \quad f(z) = 0, \ \forall z \in \mathcal{W}\},$$

and we denote by $k[\mathcal{W}]$ the coordinate ring $k[x_1, \ldots, x_n]/\mathfrak{I}$. The definition of the **degree** $\deg(\mathcal{W})$ of $\mathcal{W}$ that suits best our purpose is the geometric one (see [Hei83, Ful84, Mum95] for instance):

$$\begin{aligned}\deg(\mathcal{W}) := \sup \{ \#\mathcal{V}(y_1 - p_1, \ldots, y_r - p_r) \cap \mathcal{W} \subseteq \overline{k}^n \ ; \\ y_1, \ldots, y_r \text{ are k} - \text{linear forms}, \ (p_1, \ldots, p_r) \in k^r, \\ \#\mathcal{V}(y_1 - p_1, \ldots, y_r - p_r) \cap \mathcal{W} < +\infty \},\end{aligned}$$

where $\#$ denotes the cardinal function and $\mathcal{V}(y_1 - p_1, \ldots, y_r - p_r)$ is the variety solution of $y_1 = p_1, \ldots, y_r = p_r$. If $\mathcal{W}$ is not equidimensional then its degree is defined as the sum of the degrees of its equidimensional components.

Let $\mathcal{W}$ be r-equidimensional. We say that a subset of variables $Z = \{x_{i_1}, \ldots, x_{i_k}\}$ is **free** with respect to $\mathcal{W}$ when $\mathfrak{I} \cap k[x_{i_1}, \ldots, x_{i_k}] = (0)$. A variable is **integral** with respect to a subset of variables Z if there exists in $\mathfrak{I}$ a monic polynomial annihilating it and whose coefficients are polynomial in the variables of Z only.

A Noether normalization of $\mathcal{W}$ consists of a k-linear change of variables, transforming the variables $\mathbf{x} := (x_1, \ldots, x_n)$ into new ones $\mathbf{y} := (y_1, \ldots, y_n)$, such that the linear map from $\overline{k}^n$ to $\overline{k}^r$ $(r \leq n)$ that sends $(y_1, \ldots, y_n)$ to $(y_1, \ldots, y_r)$ induces a finite surjective morphism of affine varieties $\pi : \mathcal{W} \to \overline{k}^r$. This is equivalent to the fact that the variables $y_1, \ldots, y_r$ are free and $y_{r+1}, \ldots, y_n$ integral with respect to the free ones. In this situation we say that the variables are in **Noether position** with respect to $\mathcal{W}$.

Let B denote the coordinate ring $k[\mathcal{W}]$, and $R := k[y_1, \ldots, y_r]$, then a Noether normalization induces an integral ring extension $R \to B$. Let K be the field of fractions of R and B' denote $K \otimes_R B$, then B' is a finite-dimensional K-vector space of dimension bounded by the degree of $\mathcal{W}$.

We say that the variables $y_1, \ldots, y_n$ are in **projective Noether position** if they define a Noether position for the projective Zariski closure of $\mathcal{W}$. More precisely, let x_0 be a new variable, to any polynomial f of $k[x_1, \ldots, x_n]$, we associate $f^h(x_0, \ldots, x_n)$ the homogenization of f with respect to x_0; let $\mathfrak{I}^h \subset k[x_0, \ldots, x_n]$ denote the ideal generated by all the homogenized polynomials of $\mathfrak{I}$ and $\mathcal{W}^h \subseteq \overline{k}^{n+1}$ the k-variety associated to $\mathfrak{I}^h$: $\mathcal{W}^h$ is the projective Zariski closure of $\mathcal{W}$. We say that the variables $y_1, \ldots, y_n$ are in

projective Noether position with respect to $\mathcal{W}$ when $x_0, y_1, \ldots, y_n$ are in Noether position with respect to $\mathcal{W}^h$, that is $k[x_0, y_1, \ldots, y_r] \to k[\mathcal{W}^h]$ is an integral ring extension.

From now on we assume that $y_1, \ldots, y_n$ are variables in projective Noether position for $\mathcal{W}$. In this situation the dimension of B' is exactly the degree of π, that equals $\deg(\mathcal{W})$. We are interested in some particular bases of B': A k-linear form $u = \lambda_{r+1}y_{r+1} + \cdots + \lambda_n y_n$, with $(\lambda_{r+1}, \ldots, \lambda_n) \in k^{n-r}$, such that the set of powers $1, u, \ldots, u^{\deg(\mathcal{W})-1}$ forms a basis of the vector space B' is called a **primitive element** of $\mathcal{W}$.

A **geometric resolution** of $\mathcal{W}$ is a data structure to store and manipulate $\mathcal{W}$ from a computational point of view. This record contains the following fields:

- An invertible $n \times n$ square matrix M with entries in k such that the new coordinates $\mathbf{y} = M^{-1}\mathbf{x}$ are in projective Noether position with respect to $\mathcal{W}$;
- A primitive element $u = \lambda_{r+1}y_{r+1} + \cdots + \lambda_n y_n$ of $\mathcal{W}$;
- The minimal polynomial $q(T) \in K[T]$ of u in B', monic in T, and
- The **parametrization** of $\mathcal{W}$ by the zeros of q, given by polynomials

$$v_{r+1}(T), \ldots, v_n(T) \in K[T],$$

such that $y_j = v_j(u)$ in B', for $r+1 \leq j \leq n$ and $\deg_T(v_j) < \deg_T(q)$.

Given a primitive element u, its monic minimal polynomial q is uniquely determined. But the parametrization can be expressed in several ways. In the above definition the parametrization of the algebraic coordinates has the form

$$y_j = v_j(T), \qquad r+1 \leq j \leq n.$$

However, given any polynomial p in $K[T]$ relatively prime with q another parametrization can be deduced:

$$p(T)y_j = v_j(T)p(T), \qquad r+1 \leq j \leq n.$$

One interesting choice is to express the parametrization in the following way:

$$\frac{\partial q}{\partial T}(T)y_j = w_j(T), \qquad r+1 \leq j \leq n, \tag{1}$$

with $\deg_T w_j < \deg_T q$. We call a parametrization in the form of Equation (1) a **Kronecker parametrization**. This special form has a long history and we refer to [GLS01, §1] for more details about this subject.

Proposition 1 [GLS01, Proposition 3] *According to the above notation, the polynomial q has its coefficients in R and in the Kronecker parametrization (1) the polynomials w_i have also their coefficients in R instead of K. The* total *degree of q and the w_i with respect to the variables $y_1, \ldots, y_r$ and T is bounded by $\deg_T(q) = deg(\mathcal{W})$. Moreover $q(u)$ and $\frac{dq}{dT}(u)y_j - w_j(u)$ belong to $\mathfrak{I}$, for $r+1 \leq j \leq n$.*

For instance, let $f_1 = x_3^2 + x_1x_2 + 1$ and $f_2 = x_2^2 + x_1x_3$, the variables x_1, x_2, x_3 are in Noether position, x_2 is a primitive element and we have the following Kronecker parametrization

$$x_2^4 + x_1^3x_2 + x_1^2 = 0,$$
$$(4x_2^3 + x_1^3)x_3 = 4x_1x_2 + 3x_1^2x_2^2.$$

Since we impose that $\deg(v_j) < \deg(q)$ it follows:

Proposition 2 *Given a Noether position and a primitive element, any equidimensional algebraic variety $\mathcal{W}$ admits a unique geometric resolution.*

As discussed in §1.2 geometric resolutions have been designed to be stored by means of straight-line programs. We will not compute such objects here. As the only geometric resolutions we need for our solver are for curves, we use dense polynomial representation. Nevertheless in §5.3 we propose an efficient scheme to compute expanded representations of geometric resolutions.

Let $\mathcal{V}$ be a variety, we say that a sub-variety $\mathcal{W}$ of $\mathcal{V}$ is **isolated** in $\mathcal{V}$ if $\mathcal{W}$ is the union of some irreducible components of $\mathcal{V}$. Let $\mathbf{e} = (e_1, \ldots, e_l)$ be a sequence of polynomials in $k[x_1, \ldots, x_n]$. We denote by $\mathcal{V}(\mathbf{e})$ the k-algebraic variety solution of $\mathbf{e} = 0$. Let $\mathcal{W}$ be an irreducible component of $\mathcal{V}(\mathbf{e})$, then its **multiplicity** $\mathrm{mul}(\mathcal{W}; \mathbf{e})$ as a solution of the system $\mathbf{e} = 0$ is given by:

$$\mathrm{mul}(\mathcal{W}; \mathbf{e}) = \dim_{K[\mathcal{W}]} K[\mathcal{W}][[y_{r+1} - v_{r+1}(T), \ldots, y_n - v_n(T)]]/(\mathbf{e} \circ M),$$

where M, u, q, v come from a geometric resolution of $\mathcal{W}$, $K[\mathcal{W}] \simeq K[T]/(q(T))$ is a field and $K[\mathcal{W}][[y_{r+1} - v_{r+1}(T), \ldots, y_n - v_n(T)]]$ denotes the power series ring in the n variables $y_{r+1} - v_{r+1}(T), \ldots, y_n - v_n(T)$.

For any irreducible component $\mathcal{W}$ of $\mathcal{V}(\mathbf{e})$ we denote by $\deg^{\mathrm{a}}(\mathcal{W}; \mathbf{e})$ the product $\mathrm{mul}(\mathcal{W}; \mathbf{e})\deg(\mathcal{W})$. For any isolated sub-variety $\mathcal{V}$ of $\mathcal{V}(\mathbf{e})$, $\deg^{\mathrm{a}}(\mathcal{V}; \mathbf{e})$ is defined as the sum of the $\deg^{\mathrm{a}}(\mathcal{W}; \mathbf{e})$ over all the irreducible components $\mathcal{W}$ of $\mathcal{V}$.

2.3 Fibers

We keep the notation of the previous section: $\mathcal{W}$ is a r-equidimensional variety, $\mathfrak{I}$ is its annihilating ideal, $y_1, \ldots, y_n$ are variables in projective Noether position and π is the projection from $\mathcal{W}$ onto the space spanned by the free variables. In this situation we call the **fiber** of $\mathcal{W}$ at point $\mathbf{p} := (p_1, \ldots, p_r)$ the finite set of points $\mathcal{W}_{\mathbf{p}} := \pi^{-1}(\mathbf{p})$. If all the points of a fiber are smooth on $\mathcal{W}$ and for π then the cardinal of the fiber equals $\deg(\mathcal{W})$. In general this property holds but one needs to count points in the fiber with their multiplicities [Sam67, p. 89]. A k-linear form $u = \lambda_{r+1}y_{r+1} + \cdots + \lambda_n y_n$ is said to be a **primitive element** of a fiber if its separates its points, in the sense that u takes two distinct values at two different points of the fiber.

We represent the **fiber** of $\mathcal{W}_{\mathbf{p}}$ by a record which has the following entries:

- An **annihilating system** $\mathbf{e} = (e_1, \ldots, e_l) \subseteq \mathfrak{I}$ of $\mathcal{W}$; $\mathbf{e}$ is encoded by a straight-line program with $x_1, \ldots, x_n$ as input.
- An invertible $n \times n$ square matrix M with entries in k such that the new coordinates $\mathbf{y} = M^{-1}\mathbf{x}$ are in **projective Noether position** with respect to $\mathcal{W}$.
- The **specialization point** $\mathbf{p}$;
- A **primitive element** $u = \lambda_{r+1}y_{r+1} + \cdots + \lambda_n y_n$ of $\mathcal{W}_\mathbf{p}$, with $(\lambda_{r+1}, \ldots, \lambda_n)$ in k^{n-r};
- The **minimal polynomial** $q(T) \in k[T]$ of u in the coordinate ring $k[\mathcal{W}_\mathbf{p}]$;
- $n-r$ polynomials $\mathbf{v} := (v_{r+1}, \ldots, v_n)$ of $k[T]$, of degrees strictly less than $\deg_T(q)$, giving the parametrization of $\mathcal{W}_\mathbf{p}$ by the zeros of q:

$$k[y_{r+1}, \ldots, y_n]/(q(u), y_{r+1} - v_{r+1}(u), \ldots, y_n - v_n(u)) \simeq k[\mathcal{W}_\mathbf{p}].$$

Observe that the following relations hold in the factor ring:

$$u(v_{r+1}(T), \ldots, v_n(T)) = T,$$

$$e_j \circ M(p_1, \ldots, p_r, v_{r+1}(T), \ldots, v_n(T)) \equiv 0 \bmod q(T), \quad 1 \leq j \leq l.$$

If the Jacobian matrix of $\mathbf{e} \circ M$ with respect to the variables $y_{r+1}, \ldots, y_n$ has rank $n-r$ when evaluated at each point of the fiber then we call p a **lifting point**, and the fiber is said to be a **lifting fiber**. In this case note that we have $\deg(\mathcal{W}_\mathbf{p}) = \deg(\mathcal{W})$. Such a fiber represents $\mathcal{W}$ in the sense that it is possible to recover the geometric resolution with primitive element u **lying over** this fiber: the minimal polynomial and the parametrization of this geometric resolution of $\mathcal{W}$ specialize to the ones of the fiber [GLS01, Proposition 5].

When $\mathcal{W}$ is isolated in $\mathcal{V}(\mathbf{e})$ we will say for short that the fiber is **isolated**. If a fiber is not isolated then one can not find lifting points, this is a consequence of the Jacobian criterion [Mat86, §30]. If a fiber is isolated, lifting points exist if and only if $\mathrm{mul}(\mathcal{W}; \mathbf{e}) = 1$. In this case the specialization points that are not lifting points are enclosed in an algebraic hypersurface of k^r [GLS01, Lemma 1]. The purpose of the next section is to generalize the notion of lifting points for isolated fibers of components featuring multiplicities. In such cases we make use of the deflation algorithm introduced in [Lec02].

Notation for the pseudo-code For the pseudo-code of the algorithms we use the following notation. If F denotes a fiber: $F_{ChangeOfVariables}$ is M, $F_{PrimitiveElement}$ is u, $F_{SpecializationPoint}$ is $\mathbf{p}$, $F_{MinimalPolynomial}$ is q, $F_{Parametrization}$ is $\mathbf{v}$ and $F_{AnnihilatingSystem}$ is $\mathbf{e}$.

Generic Fibers Let $\mathfrak{P}(P)$ be a property depending on a point $P \in k^N$ for a given integer N, we say that $\mathfrak{P}(P)$ is true for **almost all** points P if there exists a nonzero polynomial $H \in k[z_1, \ldots, z_N]$ such that $H(P) \neq 0$ implies $\mathfrak{P}(P)$.

Let $\mathcal{W}$ be an equidimensional variety and $\mathbf{e}$ a sequence of polynomials in $k[x_1, \ldots, x_n]$ such that $\mathcal{W} \subseteq \mathcal{V}(\mathbf{e})$. A fiber of $\mathcal{W}$ with annihilating system $\mathbf{e}$ corresponds to the choice of a $n \times n$ matrix over k, a linear form depending on at most n variables and a specialization point of size at most n. In other words, such a fiber corresponds to a choice of a certain point P in k^N with $N = n^2 + 2n$. If $\mathfrak{P}$ is a property depending on a fiber of $\mathcal{W}$ for the annihilating system $\mathbf{e}$ then we say that $\mathfrak{P}$ is true for almost all fibers of $\mathcal{W}$ if $\mathfrak{P}$ is true for almost all values of the point P associated to the fiber.

3 Lifting algorithms

In this section we import the main results of [Lec02], the deflation process described therein is the core of our solver: the deflation process is defined in §3, the generic trace in §3.5, the nested coordinates in §3.4, the functions `NestedCoordinatesWithTrace` and `LiftNestedCoordinates` are presented in §4.

3.1 Fast deflation

The following framework is similar to and generalizes [GLS01, §4]. Let $\mathfrak{o}$ be a Noetherian domain, $\mathfrak{m}$ one of its maximal ideals, K its field of fractions and $\hat{\mathfrak{o}}$ its completion with respect to the $\mathfrak{m}$-adic topology. We assume that K has characteristic zero. The main applications we have in mind are: $(\mathfrak{o}, \mathfrak{m}) = (k[t], (t))$, $(\mathfrak{o}, \mathfrak{m}) = (k[y_1, \ldots, y_r], (y_1 - p_1, \ldots, y_r - p_r))$, where k is a field of characteristic zero and $(p_1, \ldots, p_r) \in k^r$ and $(\mathfrak{o}, \mathfrak{m}) = (\mathbb{Z}, (p))$, where p is a prime number.

Roughly speaking we are given a set of isolated roots of a system $\mathbf{e} = 0$, at precision $\mathfrak{m}$, our aim is to recover these roots at precision $\mathfrak{m}^\kappa$ for any values of κ. The method we propose below works for almost all maximal ideal $\mathfrak{m}$. More formally, our lifting algorithm takes as input:

(I1) A sequence $\mathbf{e} = (e_1, \ldots, e_l)$ of polynomials in $K[x_1, \ldots, x_n]$ encoded by a straight-line program;

(I2) A linear form $u = \lambda_1 x_1 + \cdots + \lambda_n x_n$, with $\lambda_i \in K$;

(I3) A monic squarefree polynomial q in $\mathfrak{o}[T]$;

(I4) $\mathbf{v} = (v_1, \ldots, v_n)$, n polynomials in $\mathfrak{o}[T]$ of degrees strictly less than $\deg(q)$;

(I5) $\mathcal{T}$, the *generic trace* of the deflation process described in the next paragraphs.

Let $A := \hat{\mathfrak{o}}[T]/(q)$. We say that an element in K (given by its numerator and denominator) is **well-defined** in $\hat{\mathfrak{o}}$ if its denominator is invertible modulo $\mathfrak{m}$. We assume that:

ALGORITHM 1: **Global Newton Iterator**

`GlobalNewton`($\mathbf{e}, u, q, \mathbf{v}$,StopCriterion, $\mathcal{T}$)

- $\mathbf{e}$ is a sequence of polynomials in $K[x_1, \ldots, x_n]$.
- u is a linear form in the variables $x_1, \ldots, x_n$.
- q is a monic polynomial in $\mathfrak{o}[T]$.
- $\mathbf{v}$ is a sequence of polynomials in $\mathfrak{o}[T]$.
- StopCriterion is a function returning a Boolean value. Its arguments are taken from the local variables of this procedure. It returns whether the lifted parametrization $(V_1, \ldots, V_n)$ at precision $\mathfrak{m}^\kappa$ is sufficient or not.
- $\mathcal{T}$ is the generic trace.

If (H1),...,(H7) are satisfied then for almost all matrices N introduced below the procedure returns Q and $(V_1, \ldots, V_n)$ as in (O1), (O2), where the precision κ is implicitly fixed by StopCriterion.

Computations are performed in $A/(\mathfrak{m}^\kappa A)$, for increasing values of κ.
    $N \leftarrow$ a random $n \times n$ invertible matrix with entries in K.
    # Change the coordinates to N
    $\mathbf{e} \leftarrow \mathbf{e} \circ N$; $\mathbf{v} \leftarrow N^{-1}\mathbf{v}$; $u \leftarrow u \circ N$;
    $\kappa \leftarrow 1$;
    $\mathbf{Y} \leftarrow$ `NestedCoordinatesWithTrace`$(\mathbf{e}, q, \mathbf{v}, \mathcal{T})$;
    $\mathbf{V} \leftarrow \mathbf{v}$; $Q \leftarrow q$;
    while not StopCriterion$(\mathbf{e}, u, Q, \mathbf{V}, \kappa, N)$ **do**
        $\kappa \leftarrow 2\kappa$;
        $\mathbf{Y} \leftarrow$ `LiftNestedCoordinates`$(\mathbf{e}, Q, \mathbf{Y}, \mathcal{T})$;
        # $\mathbf{Y}$ is a n-uple of multivariate power series in n variables.
        $\mathbf{V} \leftarrow$ constant coefficient of $\mathbf{Y}$;
        $\Delta \leftarrow u(\mathbf{V}) - T$;
        $\mathbf{Y} \leftarrow \mathbf{Y} - \left(\Delta \frac{\partial \mathbf{Y}}{\partial T} \bmod Q\right)$;
        $Q \leftarrow Q - \left(\Delta \frac{\partial Q}{\partial T} \bmod Q\right)$;
    od;
    $\mathbf{V} \leftarrow$ constant coefficient of $\mathbf{Y}$;
    $\mathbf{V} \leftarrow N\mathbf{V}$; # Change the coordinates back
    return$(Q, \mathbf{V})$;

(H1) There exist polynomials $\hat{Q}, \hat{V}_1, \ldots, \hat{V}_n$ in $K[T]$ such that $x^* = (\hat{V}_1, \ldots, \hat{V}_n)$ in $(K[T]/\hat{Q}(T))^n$ represents a set of isolated roots with respect the Zariski topology of the system $\mathbf{e} = 0$.

Let $\hat{Q}_i$, for $i = 1, \ldots, c$, be the irreducible factors of $\hat{Q}$, then we denote by m_i the multiplicity of $(\hat{V}_1, \ldots, \hat{V}_n)$ as a root of $\mathbf{e} = 0$ in $K[T]/\hat{Q}_i(T)$. By extension we say that an element of $K[T]/\hat{Q}(T)$ is well-defined in A if all its coefficients are well-defined in $\hat{\mathfrak{o}}$. Let m denote the minimum of the multiplicities of the points represented by x^*: $m := \min_{i=1}^{c} m_i$. We assume more:

(H2) $\deg(\hat{Q}) = \deg(q)$, $\hat{Q}$ is monic, well-defined in $\hat{\mathfrak{o}}$ and coincides with q modulo $\mathfrak{m}$.

(H3) $\deg(\hat{V}_j) < \deg(q)$, for $j = 1, \ldots, n$, x^* is well-defined in A^n and coincides with $(v_1, \ldots, v_n)$ modulo $\mathfrak{m}$.

(H4) $u(\hat{V}_1, \ldots, \hat{V}_n) = T$.

(H5) $\mathbf{e}$ is well-defined over A.

Let us now describe the main construction which yields to the fast deflation algorithm. We intensively use Gantmacher's notation $z_{i:j}$ to denote the sub-sequence $z_i, \ldots, z_j$ of a vector or sequence z. We construct sequences $(\mathbf{e}_k)_{k\geq 1}$ and $(n_k)_{k\geq 1}$ incrementally as follows. We start with $n_1 := 1$, $\mathbf{e}_1 := \mathbf{e}$ and for $k \geq 2$:

1. We introduce the power series ring $S_k := K[T]/\hat{Q}(T)[[x_{n_k:n} - \hat{V}_{n_k:n}]]$, $\mathbf{e}_k$ is a finite subset of S_k.

2. We let $\mu_k := \mathrm{val}(\mathbf{e}_k)$ be the minimum of the valuations of the elements of $\mathbf{e}_k$.

3. We define
$$\widetilde{\mathbf{e}}_k := \bigcup_{j=0}^{\mu_k - 1} \left\{ \frac{\partial^j e}{\partial {x_{n_k}}^j},\ e \in \mathbf{e}_k \right\}.$$

4. For the sake of simplicity we assume that there exists an element in $\mathbf{e}_k$ having a nonzero coefficient with respect to the monomial $x_{n_k}^{\mu_k}$. In this case we say that x_{n_k} is in **Weierstraß position** with respect to $\mathbf{e}_k$. When entering Algorithm 1 this hypothesis may not hold, this is why we change the coordinates: almost all matrices N involved in Algorithm 1 ensure Weierstraß positions.

5. By construction the Jacobian matrix of $\widetilde{\mathbf{e}}_k$ at $\hat{V}_{n_k:n}$ is nonzero and has rank $\rho_k \geq 1$. We define $n_{k+1} := n_k + \rho_k$ and try to extract a subset Σ_k of ρ_k elements in $\widetilde{\mathbf{e}}_k$ such that the Jacobian matrix of Σ_k with respect to the variables $x_{n_k:n_{k+1}-1}$ at $\hat{V}_{n_k:n}$ is invertible. We will say that x^* is **DA-irreducible at order** k if the algorithm used to compute the rank ρ_k and extract Σ_k does not fail at inverting zero-divisors (we refer to [Lec02] for more details about this algorithm). In order to continue we need to assume that this kth DA-irreducibility holds.

6. Conditions required by the implicit function theorem are satisfied: it is possible to define $y_{n_k:n_{k+1}-1}$ as the unique power series in S_{k+1} satisfying $\Sigma_k(y_{n_k:n_{k+1}-1}, x_{n_{k+1}:n}) = 0$ in the neighbourhood of $\hat{V}_{n_k:n}$. Last we define $\mathbf{e}_{k+1} := \widetilde{\mathbf{e}}_k(y_{n_k:n_{k+1}-1}, x_{n_{k+1}:n}) \subset S_{k+1}$.

This construction stops once we have exhausted all the variables, that is when $n_{k+1} = n+1$. We let ν be such that $n_{\nu+1} = n+1$ and call it the **depth** of the deflation. Gathering the necessary DA-irreducibility conditions we must assume:

(H6) x^* is DA-irreducible at order k, for all $k \in \{1, \ldots, \nu\}$. In particular this implies that x^* is **DA-irreducible** as defined in [Lec02, §5].

Observe that the concept of DA-irreducibility strongly depends on the algorithm involved in step 5, which chooses of the subsets Σ_k: different choices yield different DA-irreduciblities. The underlying idea corresponds to the fact that the construction of the deflation sequence can be done without splitting $\hat{Q}$. Last, in order to ensure the well-definition of all the Σ_k over A we need to add one last hypothesis:

(H7) All the quantities involved in the calculations of step 5 are well-defined over A.

Before studying Algorithm 1 it remains to define the subfunctions it uses. These subfunctions compute and lift what are called the **nested coordinates** $\mathbf{Y} = (Y_1, \ldots, Y_n)$ associated to the deflation process and defined as power series in $K[T]/\hat{Q}(T)[[\epsilon_1, \ldots, \epsilon_n]]$ by:

$$\begin{aligned}
Y_{n_\nu:n_{\nu+1}-1} &:= y_{n_\nu:n_{\nu+1}-1}(), \\
Y_{n_{\nu-1}:n_\nu-1} &:= y_{n_{\nu-1}:n_\nu-1}(Y_{n_\nu:n_{\nu+1}-1} + \epsilon_{n_\nu:n_{\nu+1}-1}), \\
&\ldots \\
Y_{n_1:n_2-1} &:= y_{n_1:n_2-1}(Y_{n_2:n_3-1} + \epsilon_{n_2:n_3-1}, \ldots, Y_{n_\nu:n_{\nu+1}-1} + \epsilon_{n_\nu:n_{\nu+1}-1}).
\end{aligned}$$

According to (H7) the nested coordinates are also well-defined over A. The **generic trace** $\mathcal{T}$ of the deflation is a data structure that stores all the choices performed during the construction and for generic coordinates. We refer to [Lec02, §3.5] for a precise definition.

The first subfunction `NestedCoordinatesWithTrace` we use takes as input the polynomial system, the approximation of the root mod $\mathfrak{m}$ and computes the nested coordinates $\mathbf{Y}$ at precision $\mathfrak{m}$, but the generic trace of the root is supposed to be known. The second subfunction `LiftNestedCoordinates` takes as input the value of $\mathbf{Y}$ at precision κ and returns the lifted value at precision 2κ. This lifting operates only if the generic trace is known.

We are now able to describe how our lifting algorithm computes approximations of x^* in A at any arbitrary precision. More precisely, for any given $\kappa > 0$ we are able to compute:

(O1) Q a monic polynomial in $\mathfrak{o}[T]$ of degree $\deg(q)$ which coincides with $\hat{Q}$ modulo $\mathfrak{m}^\kappa$.

(O2) $\mathbf{V} = (V_1, \ldots, V_n)$, n polynomials in $\mathfrak{o}[T]$ of degrees strictly less than $\deg(q)$ which coincides with $(\hat{V}_1, \ldots, \hat{V}_n)$ modulo $\mathfrak{m}^\kappa$ and $u(\mathbf{V}) = T$ modulo $\mathfrak{m}^\kappa$.

The lifting algorithm is the combination of the deflation algorithm of [Lec02] and the globalization trick of [GLS01, §4]. It is summarized in Algorithm 1. We recall that the algorithm is probabilistic: in order to work properly the coordinates must be generic enough in order to satisfy the Weierstraß positions required by the deflation [Lec02, §2.2]. This is why the procedure starts with picking up a random matrix N. Then we change $\mathbf{e}$ to $\mathbf{e} \circ N$, $\mathbf{v}$ to $N^{-1}(\mathbf{v})$ and u to $u \circ N$ before entering the deflation routines. At the end of the lifting we change back the coordinates. If N is not generic enough then the `NestedCoordinatesWithTrace` may either raise a "division by zero" error, return a bad result or may never stop. We denote by $\mathtt{a}(h)$ the cost of the arithmetic operations in $\mathfrak{o}/\mathfrak{m}^h$: binary arithmetic operations (addition, multiplication, inversion) and projections from $\mathfrak{o}/\mathfrak{m}^{h'}$ to $\mathfrak{o}/\mathfrak{m}^h$, for any $h' \leq h$.

Proposition 3 *According to the above notation, under hypotheses* (H1),...,(H7) *for almost all matrices N of Algorithm 1, which is well-defined over $\mathfrak{o}$ and invertible modulo $\mathfrak{m}$, Algorithm 1 returns a correct answer, with a complexity in*

$$\mathcal{O}\Big(\log(d) n^4 (nL + n^\Omega) m^2 \mathcal{U}(\deg(q)) \sum_{j=0}^{\log_2(\kappa)+1} \mathtt{a}(2^j) \Big),$$

where d is an upper bound on the degrees of the elements of $\mathbf{e}$ and L denotes the evaluation complexity of the straight-line program encoding $\mathbf{e}$.

Proof. This is a corollary of [Lec02, Theorem 1]. The cost of the changes of variables is negligible. The complexity is essentially the sum of the complexities of the computation of the nested coordinates and the successive liftings. The deformation for updating $\mathbf{Y}$ and Q are done within $\mathcal{O}(n^3 m)$ arithmetic operations in $A/(\mathfrak{m}^\kappa A)$, since the support of the multivariate power series for $\mathbf{Y}$ is in $\mathcal{O}(nm)$ (combine Proposition 3, Proposition 6 and Lemma 7 of [Lec02]). The value m is bounded by d^n, hence $\log(nm)$ is bounded by $\log(n) + n\log(d) \in \mathcal{O}(n \log(d))$. □

3.2 Lifting Fibers for Multiple Components

We come back to the notation of §2.2: $\mathcal{W}$ is a r-equidimensional variety, M, $\mathbf{y}$, u, q, $\mathbf{v}$ constitute a geometric resolution of $\mathcal{W}$. We recall that $K := k(y_1, \ldots, y_r)$. Let $\mathbf{e}$ be a sequence of polynomials in $k[x_1, \ldots, x_n]$. We assume that $\mathcal{W}$ is isolated in $\mathcal{V}(\mathbf{e})$.

Let $K[a] := K[T]/q(T)$ and $y^* = (v_{r+1}(a), \ldots, v_n(a))$ in $K[a]^n$. Then the vector y^* represents a set of isolated roots of the system $\mathbf{e} \circ M = 0$ seen as polynomials in $K[y_{r+1}, \ldots, y_n]$. We are in the frame of the deflation process of the previous section: we say that the geometric resolution is **DA-irreducible** with respect to the sequence $\mathbf{e} = 0$ if Ny^* is DA-irreducible as a root of $\mathbf{e} \circ M \circ N^{-1} = 0$ for almost all matrices N in $GL_n(k)$. This property is independent of the geometric resolution chosen. Therefore

we say for short that $\mathcal{W}$ is DA-irreducible in $\mathcal{V}(\mathbf{e})$. By extension, the **generic trace** $\mathcal{T}$ of $\mathcal{W}$ with respect to $\mathbf{e}$ is defined to be the generic trace of y^* (and is also independent of the geometric resolution).

We introduce the function `DaSplit` of [Lec02, §5]. If given as input the polynomials $\mathbf{e}$ and the parametrisation of the roots of $\mathbf{e} = 0$ over K coming from the geometric resolution then it returns a partition of these roots into DA-irreducible subsets together with their corresponding generic traces. To this partition corresponds a decomposition of $\mathcal{W}$. We call a **DA-lifting point** $\mathbf{p}$ a point in k^r satisfying the following property: the function `DaSplit` of [Lec02, §5] called on Ny^* commutes with the specialization of the free variables at $\mathbf{p}$, for almost all matrices N in $GL_{n-r}(k)$. By this commutation we mean that all the branchings performed during the execution of `DaSplit` are preserved. In other words the deflation algorithm executed over K or over k after the specialization $y_1 = p_1, \ldots, y_r = p_r$ keeps the same branchings. A fiber with a DA-lifting point is called a **DA-lifting fiber**. In this case, if $\mathcal{W}$ is DA-irreducible then we say that the fiber is a **DA-irreducible lifting fiber**.

This definition is technical and the lack of geometric interpretation makes it difficult to manipulate. The only useful result we shall use is:

Lemma 1 *If $\mathcal{W}$ is isolated in $\mathcal{V}(\mathbf{e})$ then almost all fibers of $\mathcal{W}$ with respect to $\mathbf{e}$ are DA-lifting fibers.*

Proof. This is a consequence of Propositions 7, 8 and 20 of [Lec02]. □

This terminology is used in §5.3: once the generic trace is known we are able to test if a fiber is a DA-lifting fiber. Last we need to revisit the definition of a fiber given in §2.3: we need to add a new field to store the generic trace. For the pseudo-code we introduce $F_{GenericTrace}$ to denote the generic trace of the fiber F. It is important to notice that this trace is only known for DA-irreducible lifting fibers.

3.3 Lifted Curves

Let $\mathcal{W}$ be a r-equidimensional variety. Following the notation of §2.2, π denotes the projection map from $\mathcal{W}$ onto the space spanned by the free variables $y_1, \ldots, y_r$. Let F be a DA-irreducible lifting fiber of $\mathcal{W}$, $\mathbf{p}$ its specialization point and $\mathbf{p}' \in k^r$ be a point different from $\mathbf{p}$. We denote by D the line spanned by $\mathbf{p}$ and $\mathbf{p}'$. The inverse image $\mathcal{W}_D := \pi^{-1}(D)$ is a one equidimensional variety called a **lifted curve** [GLS01, §4.5]. We are interested in computing a geometric resolution of it.

This lifted curve computation is achieved by replacing the function `GlobalNewton` of [GLS01, §4.3] by the one of Algorithm 1 in the function `LiftCurve` of [GLS01, §4.5]. One slight modification to notice is that `GlobalNewton` takes the trace of the fiber as a new argument. It is also important to observe that this new `LiftCurve` procedure is now probabilistic because of the random choice of N introduced in the new `GlobalNewton`.

The lifted curve algorithm is summarized in Algorithm 2. Let $\mathbf{e}$, M, $\mathbf{p}$, u, q, $\mathbf{v}$ compose the fiber F. First we express the equations in the variables $y_1, \ldots, y_n$: $\mathbf{g} := \mathbf{e} \circ M$. Then we introduce a new variable t and compute $\mathbf{h}$ from $\mathbf{g}$ by substituting y_i by

ALGORITHM 2: **Lift Curve**

`LiftCurve`(F,$\mathbf{p}'$)

- F is a DA-irreducible lifting fiber of $\mathcal{W}$.
- $\mathbf{p}'$ is a point in k^r different from the lifting point of F.

For almost all choices of the matrix N involved in the function `GlobalNewton`, the procedure returns the Kronecker parametrization $q, \mathbf{w}$ of the geometric resolution of the lifted curve for the line (pp').

$r \leftarrow \dim(F)$;
$\delta \leftarrow \deg(F)$;
$\mathbf{p} \leftarrow F_{SpecializationPoint}$;
$\mathbf{g} \leftarrow F_{AnnihilatingSystem} \circ F_{ChangeOfVariables}$;
$\mathbf{h} \leftarrow \mathbf{g}((p'_1 - p_1)t + p_1, \ldots, (p'_r - p_r)t + p_r, y_{r+1}, \ldots, y_n)$;
StopCriterion $\leftarrow ((k) \mapsto k > \delta)$;
q,$\mathbf{v}$:=`GlobalNewton`($\mathbf{h}, F_{PrimitiveElement}, F_{MinimalPolynomial}$,
$F_{Parametrization}$,StopCriterion, $F_{GenericTrace}$);
$\mathbf{w} \leftarrow [z\frac{\partial q}{\partial T} \bmod q : \ z \in \mathbf{v}]$;
$q \leftarrow \text{Truncate}(q, t^{\delta+1})$;
$\mathbf{w} \leftarrow [\text{Truncate}(z, t^{\delta+1}) : \ z \in \mathbf{w}]$;
return$(q, \mathbf{w})$;

$p_i + (p'_i - p_i)t$ for $i = 1, \ldots, r$. Last we call the deflation process at the points of F with the system $\mathbf{h} = 0$ and with $(\mathfrak{o}, \mathfrak{m}) = (k[[t]], (t))$ (with respect to the notation of §3). The function StopCriterion forces the lifting to stop once the precision $t^{\deg(\mathcal{W})+1}$ is reached. At the end, denoting $\mathbf{w} = (w_{r+1}, \ldots, w_n)$, the parametrization of $\mathcal{W}_D$ becomes:

$$q(t,u) = 0, \quad \begin{cases} \frac{\partial q(t,T)}{\partial T}(t,u)y_{r+1} &= w_{r+1}(t,u), \\ & \vdots \\ \frac{\partial q(t,T)}{\partial T}(t,u)y_n &= w_n(t,u), \end{cases}$$

in other words:

$$k(t) \otimes k[\mathcal{W}_D] \simeq k(t)[T]/\Big(q(t,T), \frac{\partial q(t,T)}{\partial T}y_{r+1} - w_{r+1}(t,u), \ldots, \frac{\partial q(t,T)}{\partial T}y_n - w_n(t,u)\Big).$$

Let us recall that the total degree of q and the w_i is bounded by $\deg(\mathcal{W})$ [GLS01, Proposition 3], which justifies our stop criterion. At the end of the procedure the function Truncate is used to recover bivariate polynomials of degrees at most δ from series known at precision $\delta + 1$.

ALGORITHM 3: **Splittings due to the deflation**

`DaSplitFiber`(F)

- F is a DA-lifting fiber of $\mathcal{W}$.

For almost all matrix N the function returns a set $\mathbf{F}$ of DA-irreducible lifting fibers representing $\mathcal{W}$.

$r \leftarrow \dim(F)$;
$N \leftarrow (n-r) \times (n-r)$ invertible random matrix over k;
$\mathbf{g} \leftarrow F_{AnnihilatingSystem} \circ F_{ChangeOfVariables}$;
$\mathbf{g} \leftarrow \mathbf{g}(F_{SpecializationPoint}, y_{r+1}, \ldots, y_n)$;
$\mathbf{g} \leftarrow \mathbf{g} \circ N$;
$\mathbf{v} \leftarrow N^{-1} F_{Parametrization}$;
$(Q_i, \mathcal{T}_i)_{i=1,\ldots,t} \leftarrow$ `DaSplit`$(\mathbf{g}, F_{MinimalPolynomial}, \mathbf{v})$;
$\mathbf{F} \leftarrow \{\}$;
for i **from** 1 to t **do**
    $F' \leftarrow F$;
    $F'_{MinimalPolynomial} \leftarrow Q_i$;
    $F'_{Parametrization} \leftarrow F_{Parametrization} \bmod Q_i$;
    $F'_{GenericTrace} \leftarrow \mathcal{T}_i$;
    $\mathbf{F} \leftarrow \mathbf{F} \cup \{F'\}$;
return$(\mathbf{F})$;

Proposition 4 *According to the above notation, for almost all fibers F of $\mathcal{W}$ and for almost all choices of N in* `GlobalNewton` *the complexity of Algorithm 2 is in*

$$\mathcal{O}\Big(\log(d) n^4 (nL + n^\Omega) \mathcal{U}(\deg^{\mathrm{a}}(\mathcal{W}; F_{AnnihilatingSystem}))^2\Big),$$

in terms of arithmetic operations in k, where d is an upper bound on the degrees of the elements $F_{AnnihilatingSystem}$ and L denotes the evaluation complexity of the straight-line program encoding $F_{AnnihilatingSystem}$.

Proof. The proof is very similar the one of Lemma 3 in [GLS01]: we take the function $\mathtt{a}$ as $\mathcal{U}$ in Proposition 3 and we observe that $m\mathcal{U}(\deg(\mathcal{W})) \leq \mathcal{U}(\deg^{\mathrm{a}}(\mathcal{W}; F_{AnnihilatingSystem}))$. □

3.4 Splittings due to the Deflation

During the resolution process of §5 we produce DA-lifting fibers that are not necessarily DA-irreducible. We need to split them into DA-irreducible ones and compute their

generic traces. The definition of DA-lifting points is exactly stated in such a way that the execution of the splitting algorithm of [Lec02, §5] leads to correct decompositions and traces, when executed on DA-lifting fibers. The resulting method is summarized in Algorithm 3. The function `DaSplit` is the one from §5 of [Lec02]. From Propositions 19 and 20 of [Lec02] we deduce:

Proposition 5 *According to the notation of Algorithm 3, for almost all matrices N in $GL_{n-r}(k)$ the function* `DaSplitFiber` *returns a correct answer, with a complexity in*

$$\mathcal{O}\Big(\log(d)n^4(nL+n^{\Omega})\deg^{\mathrm{a}}(\mathcal{W};F_{AnnihilatingSystem})^2\mathcal{U}(\deg(F))\Big),$$

in terms of arithmetic operations in k, where d is an upper bound on the degrees of the elements of $F_{AnnihilatingSystem}$ and L denotes the evaluation complexity of the straight-line program encoding $F_{AnnihilatingSystem}$.

4 Removing Redundancies

Let $\mathbf{e} = (e_1, \ldots, e_l)$ be a sequence of polynomials in $k[x_1, \ldots, x_n]$. We assume that we are given a set $\mathbf{F}$ of fibers of varieties whose union is isolated in $\mathcal{V}(\mathbf{e})$. In this section we describe a process for computing a minimal subset of isolated fibers from $\mathbf{F}$ that represents $\mathcal{V}(\mathbf{e})$. We call this stage of the solver the **minimization process**. The basic underlying operation is the inclusion test between two equidimensional varieties.

4.1 Inclusion between two varieties

Let $\mathcal{W}_1$ (resp. $\mathcal{W}_2$) be a r_1 (resp. r_2)-equidimensional variety. Let F^1 (resp. F^2) be a fiber of $\mathcal{W}_1$ (resp. $\mathcal{W}_2$). We assume that F^2 is a DA-irreducible lifting fiber. We are looking for a fiber for the reunion of the irreducible components of $\mathcal{W}_1$ that are not included in $\mathcal{W}_2$. Our method consists in computing one judicious lifted curve of F^2.

First we explain the method in terms of geometric resolutions, then we specialize the free variables and deduce the algorithm for the fibers. We assume that F^1 is generic enough so that we can consider the geometric resolutions lying over F^1 and F^2: M_i is the change of variables, u_i the primitive element, Q_i the minimal polynomial and $\mathbf{V}^i = (V^i_{r_i+1}, \ldots, V^i_n)$ the parametrization of the geometric resolution lying over F^i, for $i = 1, 2$. Let $\mathbf{y}^i := M_i^{-1}\mathbf{x}$, then the parametrization of $\mathcal{W}_i$ is given by:

$$Q_i(y^i_1, \ldots, y^i_{r_i}, u_i) = 0, \quad \begin{cases} y^i_{r_i+1} &= V^i_{r_i+1}(y^i_1, \ldots, y^i_{r_i}, u_i), \\ &\vdots \\ y^i_n &= V^i_n(y^i_1, \ldots, y^i_{r_i}, u_i). \end{cases}$$

We express the parametrization of F^1 in the coordinates of F^2:

$$(Z_1, \ldots, Z_n) := M_2^{-1}M_1(y^1_1, \ldots, y^1_{r_1}, V^1_{r_1+1}, \ldots, V^1_{n+1}).$$

ALGORITHM 4: **Difference**

`Difference`(F^1, F^2)

- F^1 is a fiber of $\mathcal{W}_1$.
- F^2 is a DA-irreducible lifting fiber of $\mathcal{W}_2$.

For almost all fibers F^1 of $\mathcal{W}$, almost all primitive elements for F^2 and almost all matrices N involved in `GlobalNewton` called from `LiftCurve`, the procedure returns F a fiber of the components of $\mathcal{W}_1$ not included in $\mathcal{W}_2$.

$r_1 \leftarrow \dim(F^1)$;
$r_2 \leftarrow \dim(F^2)$;
if $r_1 > r_2$ **then return**(F^1);
$M_1 \leftarrow F^1_{ChangeOfVariables}$;
$M_2 \leftarrow F^2_{ChangeOfVariables}$;
$(z_1, \ldots, z_n) \leftarrow M_2^{-1} M_1(F^1_{SpecializationPoint}, F^1_{Parametrization})$;
$\mathcal{C} \leftarrow$ `LiftCurve`$(F^2, (z_1, \ldots, z_{r_2}))$ in $k[a] := k[T]/(F^1_{MinimalPolynomial})$;
$\mathcal{C} \leftarrow \mathrm{subs}(t = 1, \mathcal{C})$;
$a \leftarrow F^2_{PrimitiveElement}(z_{r_2+1}, \ldots, z_n)$;
$b \leftarrow \mathcal{C}_{MinimalPolynomial}(a)$;
$b \leftarrow b$ viewed in $k[T]$;
$F \leftarrow F^1$;
$F_{MinimalPolynomial} \leftarrow F^1_{MinimalPolynomial}$ div $\gcd(F^1_{MinimalPolynomial}, b)$;
$F_{Parametrization} \leftarrow F^1_{Parametrization}$ mod $F_{MinimalPolynomial}$;
return(F);

Let $\chi_t(y^2_1, \ldots, y^2_{r_2}, T) \in k[y^2_1, \ldots, y^2_{r_2}, T]$ be the minimal polynomial of the linear form $u_\lambda = \lambda_{r_2+1} y^2_{r_2+1} + \cdots + \lambda_n y^2_n$ with respect to $\mathcal{W}_2$ where the λ_i are new parameters in k:

$$\chi_t(y^2_1, \ldots, y^2_{r_2}, u_\lambda) = 0 \text{ in } k[\mathcal{W}_2].$$

Let $A := u_\lambda(Z_{r_2+1}, \ldots, Z_n)$, we deduce that, for almost all u_λ:

$$\chi_t(Z_1, \ldots, Z_{r_2}, A) = 0 \text{ in } k(y^1_1, \ldots, y^1_{r_1})[T]/(Q_1)$$

is equivalent to the inclusion of $\mathcal{W}_1$ in $\mathcal{W}_2$ [Lec00, Appendix A]. Therefore the components of $\mathcal{W}_1$ included in $\mathcal{W}_2$ correspond to the greatest common divisor of Q_1 and $\chi_t(Z_1, \ldots, Z_{r_2}, A) = 0$, for almost all u_λ.

Let us now consider what happens in the above computations when we specialize $y^1_1, \ldots, y^1_{r_1}$ to the lifting point $\mathbf{p}_1$ of F^1. More precisely let M_1, $\mathbf{p}_1$, u_1, q_1, $\mathbf{v}^1$ be the change of coordinates, the specialization point, the primitive element, the minimal

polynomial and the parametrization of F. We recall that q_1 and $\mathbf{v}_1$ can be deduced from Q_1 and $\mathbf{V}^1$ by substituting $y_1^1, \ldots, y_{r_1}^1$ by $\mathbf{p}_1$.

First we compute

$$\mathbf{z} = (z_1, \ldots, z_n) := M_2^{-1} M_1(\mathbf{p}_1, \mathbf{v}^1).$$

Then we compute the lifted curve of F^2 for the line going through $\mathbf{p}_2$ and $(z_1, \ldots, z_{r_2})$: this computation is performed in $k[T]/(q_1)$ instead of k but no inversion is required in this algebra. From the parametrization of this curve we deduce the value $q(T) := \chi_t(z_1, \ldots, z_{r_2}, T)$ for $u_\lambda = u_2$.

Last it remains to compute $a := u_2(z_{r_2+1}, \ldots, z_n)$ and evaluate q at a: $b := q(a)$. The value b belongs to $k[T]/(q_1)$, we convert it to be the element of $k[T]$ of degree strictly less than $\deg(q_1)$.

Let $\overline{q}$ be the greatest common divisor of b and q, F the restriction of F^1 modulo $q/\overline{q}$. If the specialization point of F^1 and the primitive element of F^2 are generic enough then F represents the components of F^1 that are not included in F^2. This method is summarized in Algorithm 4 and has the following complexity:

Proposition 6 *According to the above notation, for almost all primitive elements of F^2, almost all fiber F^1 of $\mathcal{W}_1$ and almost all matrices N involved in* `LiftCurve` *then Algorithm 4 returns a correct answer within a complexity in*

$$\mathcal{O}\Big(\log(d) n^4 (nL + n^\Omega) \mathcal{U}(\deg^{\mathrm{a}}(\mathcal{W}_2; F^2_{AnnihilatingSystem}))^2 \mathcal{U}(\deg(\mathcal{W}_1)) \Big),$$

in terms of operations in k, where d is an upper bound on the degrees of the elements of $F^2_{AnnihilatingSystem}$ and L denotes the evaluation complexity of the straight-line program encoding $F^2{}_{AnnihilatingSystem}$.

Proof. The complexity is the one of `LiftCurve` over $k[T]/(q_1)$, this yields the extra factor $\mathcal{U}(\deg(\mathcal{W}_1))$. □

4.2 Minimization

We are now coming to the problem of minimizing a set of fibers $\mathbf{F}$. Our method relies on the above function `Difference` and is summarized in Algorithm 5. It works well for almost all sets of fibers.

The precise problem is the following: we are given a set of fibers $\mathbf{F} = \{F_1, \ldots, F_t\}$ where F_i is a fiber for the variety $\mathcal{W}_i$. We assume that all the F_i share $\mathbf{e}$ as annihilating system and that $\mathcal{V} := \mathcal{W}_1 \cup \cdots \cup \mathcal{W}_t$ is isolated in $\mathcal{V}(\mathbf{e})$. The output of the procedure is a set of fibers $\mathbf{F}' = \{F'_1, \ldots, F'_{t'}\}$, where F'_i is a fiber of the variety $\mathcal{W}'_i$ and satisfying:

1. $\mathcal{W}'_1 \cup \cdots \cup \mathcal{W}'_{t'} = \mathcal{V}$,
2. all the elements of $\mathbf{F}'$ are isolated and DA-irreducible,
3. for any i, $1 \leq i \leq t'$, no irreducible component of $\mathcal{W}'_i$ is included $\cup_{j \neq i} \mathcal{W}'_j$.

ALGORITHM 5: **Minimization**

`Minimize`($\mathbf{F}$)

- $\mathbf{F}$ is a set of fibers sharing the same annihilating system $\mathbf{e}$.

For almost all sets of fibers $\mathbf{F}$ and almost all matrices N involved in `DaSplitFiber`, if the union of the varieties represented by the elements of $\mathbf{F}$ is isolated in $\mathcal{V}(\mathbf{e})$ then the procedure returns $\mathbf{F}'$ a set of isolated fibers representing $\mathcal{V}$ without redundancy.

for i **from** -1 **to** n **do**
    $\mathbf{F}_i \leftarrow \{F \in \mathbf{F} |\ \dim(F) = n - i\}$;
$\mathbf{F}'_{-1} \leftarrow \{\}$;
for i **from** 0 **to** n **do**
    $\mathbf{F}'_i \leftarrow \{\}$;
    for F **in** $\mathbf{F}_i$ **do**
        $F' \leftarrow F$;
        for F'' **in** $\mathbf{F}'_{-1} \cup \cdots \cup \mathbf{F}'_i$ **do**
            $F' \leftarrow$ `Difference`(F', F'');
        $\mathbf{F}'_i \leftarrow \mathbf{F}'_i \cup$ `DaSplitFiber`(F');
$\mathbf{F}' \leftarrow \mathbf{F}'_0 \cup \cdots \cup \mathbf{F}'_n$;
return$(\mathbf{F}')$;

In this situation we say that $\mathbf{F}'$ **represents** $\mathcal{V}$ **without redundancy**.

The process we are to describe is valid for almost all sets of fibers $\mathbf{F}$. We denote by $\mathbf{F}_i$ (resp. $\mathbf{F}'_i$) the subset of fibers of $\mathbf{F}$ (resp. $\mathbf{F}'$) representing varieties of codimension i, for $i = -1, \ldots, n$. First observe that $\mathbf{F}'_{-1} = \mathbf{F}_{-1} = \{\}$. By induction, let us assume that we have already computed $\mathbf{F}'_{-1}$ up to $\mathbf{F}'_{i-1}$. Then we initialize $\mathbf{F}'_i$ as the empty set and for each fiber F of $\mathbf{F}_i$ we first remove from it all the components that belong to the variety represented by $\mathbf{F}'_0 \cup \cdots \cup \mathbf{F}'_i$, we get a new fiber F'. By construction the variety $\mathcal{W}'$ represented by F' is isolated in $\mathcal{V}$. We deduce that $\mathcal{W}'$ is isolated in $\mathcal{V}(\mathbf{e})$ and therefore we can apply the deflation algorithm. The new DA-irreducible fibers we obtain this way are added to $\mathbf{F}'_i$ and we carry on with another element of $\mathbf{F}_i$.

Proposition 7 *According to the above notation, for almost all fibers F_i of $\mathcal{W}_i$ sharing the same annihilating system $\mathbf{e}$ and almost all matrices N involved in* `DaSplitFiber`*, if $\mathcal{V}$ is isolated in $\mathcal{V}(\mathbf{e})$ then Algorithm 5 returns a correct result within a complexity in*

$$\mathcal{O}\Big(\log(d) n^4 (nL + n^\Omega) \mathcal{U}(\deg^{\mathrm{a}}(\mathcal{V}; \mathbf{e}))^2 \mathcal{U}(D)\Big),$$

in terms of operations in k, where d is an upper bound on the degrees of the elements of $\mathbf{e}$, L denotes the evaluation complexity of the straight-line program encoding $\mathbf{e}$ and $D := \sum_{i=1}^{t} \deg(\mathcal{W}_i)$.

Proof. The sum of the costs due to the calls to `Difference` is in

$$\mathcal{O}\left(\log(d)n^4(nL+n^\Omega)\sum_{i=1}^{t}\sum_{i'=1}^{t'}\mathcal{U}(\deg^{\mathrm{a}}(\mathcal{W}'_{i'};\mathbf{e}))^2\mathcal{U}(\deg(\mathcal{W}_i))\right)$$

$$\subseteq \mathcal{O}\left(\log(d)n^4(nL+n^\Omega)\mathcal{U}(\deg^{\mathrm{a}}(\mathcal{V};\mathbf{e}))^2\mathcal{U}(D)\right).$$

The sum of the costs of `DaSplitFiber` is in

$$\mathcal{O}\left(\log(d)n^4(nL+n^\Omega)\sum_{i'=1}^{t'}\deg^{\mathrm{a}}(\mathcal{W}'_{i'};\mathbf{e})^2\mathcal{U}(\deg(\mathcal{W}'_{i'}))\right)$$

$$\subseteq \mathcal{O}\left(\log(d)n^4(nL+n^\Omega)\deg^{\mathrm{a}}(\mathcal{V};\mathbf{e})^2\mathcal{U}(\deg(\mathcal{V})\right).$$

Last, note that $\deg(\mathcal{V}) \leq D$. □

4.3 Splittings

Let $\mathcal{W}$ be an equidimensional variety and f be a polynomial function given by a straight-line program of size at most L. We are interested in splitting $\mathcal{W}$ into $\mathcal{W}^r$ and $\mathcal{W}^i$ where $\mathcal{W}^r$ is the union of the irreducible components of $\mathcal{W}$ that are not included in $\mathcal{V}(f)$ and $\mathcal{W}^i$ is the union of the other components. In the next section we will see that $\mathcal{W}^r$ stands for the components that are *regularly* intersected by $\mathcal{V}(f)$ and $\mathcal{W}^i$ *irregularly.* As in [GLS01, §6.6] we can perform this computation with almost all fibers of $\mathcal{W}$. The method is summarized in Algorithm 6.

Proposition 8 *According to the above notation, for almost all fibers F of $\mathcal{W}$, Algorithm 6 works well within a complexity in:*

$$\mathcal{O}\Big((n^2+L)\mathcal{U}(\deg(F))\Big),$$

where L denotes the evaluation complexity of the straight-line program encoding f.

5 Resolution Algorithm

Putting together the previous algorithms we deduce the incremental step of our solver.

5.1 Incremental Solving

Let $\mathbf{F}$ be a set of isolated fibers representing $\mathcal{V}$ without redundancy, as defined in §4: we denote by $\mathbf{e}$ their common annihilating system. Let f and g be polynomials encoded by a straight-line program of size L. We are interested in computing a set of isolated fibers $\mathbf{F}'$ representing $\overline{(\mathcal{V}\cap\mathcal{V}(f))\backslash\mathcal{V}(g)}$ without redundancy.

Our method is summarized in Algorithm 7, it works well for almost all sets $\mathbf{F}$ representing $\mathcal{V}$. Let us describe how it works. For any F in $\mathbf{F}$ we denote by F^r the sub-fiber of F composed of the points which are not contained in $\mathcal{V}(f)$ and F^i for the other

ALGORITHM 6: **Splitting a Lifting Fiber**

`Split`(F, f)

- F is a fiber of $\mathcal{W}$.
- f is a polynomial in $k[x_1, \ldots, x_n]$.

For almost all fibers F of $\mathcal{W}$, the function returns (F^i, F^r) a couple of fibers such that F^i is a fiber for the components of $\mathcal{W}$ included in $\mathcal{V}(f)$ and F^r is a fiber for the other components.

$q \leftarrow F_{MinimalPolynomial}$;
$\mathbf{v} \leftarrow F_{Parametrization}$;
$e \leftarrow \gcd(q, f \circ F_{ChangeOfVariables}(F_{SpecializationPoint}, \mathbf{v}))$;
$q \leftarrow q/e$;
$F^i \leftarrow F$; $F^r \leftarrow F$;
$F^i_{MinimalPolynomial} \leftarrow e$; $F^i_{Parametrization} \leftarrow \mathbf{v} \bmod e$;
$F^r_{MinimalPolynomial} \leftarrow q$; $F^r_{Parametrization} \leftarrow \mathbf{v} \bmod q$;
return(F^i, F^r);

points (the exponent r stands for *regular* and i for *irregular*). This step is achieved using Algorithm 6. Then for each F^r we compute a fiber F' of the Zariski closure of $(\mathcal{W} \cap \mathcal{V}(f))\backslash\mathcal{V}(g)$, where $\mathcal{W}$ denotes the algebraic variety represented by F^r. For this purpose we call the function `OneDimensionalIntersect` of [GLS01, §6]: according the genericity of the fibers we take 0 for the Liouville point.

At the end of the main loop over the elements of $\mathbf{F}$ we obtain a set $\mathbf{F}'$ containing all the F^i and F' computed previously. We are in the frame of §4 and we call the function `Minimize` that produces a set of isolated fibers representing $\overline{(\mathcal{V} \cap \mathcal{V}(f))\backslash\mathcal{V}(g)}$.

Proposition 9 *Let* $D_1 := \deg^{\mathrm{a}}(\mathcal{V}; \mathbf{e})$ *and* $D_2 := \deg^{\mathrm{a}}(\overline{(\mathcal{V} \cap \mathcal{V}(f))\backslash\mathcal{V}(g)}; \mathbf{e}, f)$. *For almost all set of fibers* $\mathbf{F}$ *representing* $\mathcal{V}$ *without redundancy, and almost all choices of the matrices* N *involved in the subfunctions of* `Intersect`*, Algorithm 7 returns a correct answer within a complexity in*

$$\mathcal{O}\left(\log(d) n^4 (nL + n^\Omega) \mathcal{U}(\deg(f)\deg(\mathcal{V}))\Big(\mathcal{U}(D_1)^2 + \mathcal{U}(D_2)^2\Big)\right),$$

in terms of operations in k*, where* d *denotes the maximum of the degrees of the elements of* $\mathbf{e}, f$ *and* L *the evaluation complexity of the straight-line program encoding* $\mathbf{e}, f, g$.

Proof. We denote by $\mathcal{W}_1, \ldots, \mathcal{W}_t$ the varieties represented by the elements of $\mathbf{F}$ and by $\mathcal{W}'_1, \ldots, \mathcal{W}'_{t'}$ the ones of $\mathbf{F}'$. From Proposition 8 the costs of the function `Split` is not significant. The total cost due to `LiftCurve` is in

$$\mathcal{O}\left(\log(d) n^4 (nL + n^\Omega) \mathcal{U}(\deg^{\mathrm{a}}(\mathcal{V}; \mathbf{e}))^2\right).$$

ALGORITHM 7: **Intersection**

`Intersect`$(\mathbf{F}, f, g)$

- $\mathbf{F}$ is a set of isolated fibers representing $\mathcal{V}$ without redundancy.
- f, g are polynomial functions in $k[x_1, \ldots, x_n]$.

For almost all sets of fibers $\mathbf{F}$ representing $\mathcal{V}$, and almost all choices of the matrices N involved in the subfunctions, the algorithm returns a set of isolated fibers representing $\overline{(\mathcal{V} \cap \mathcal{V}(f)) \backslash \mathcal{V}(g)}$ without redundancy.

$\mathbf{F}' \leftarrow \{\}$;
for F **in** $\mathbf{F}$ **do**
  $F^i, F^r \leftarrow$ `Split`(F, f);
  $F^i_{AnnihilatingSystem} \leftarrow F^i_{AnnihilatingSystem} \cup \{f\}$;
  $\mathcal{C} \leftarrow \mathrm{subs}(t = y_r - p_r,$`LiftCurve`$(F^r, F_{SpecializationPoint} + (0, \ldots, 0, 1)))$;
  $F' \leftarrow$ `OneDimensionalIntersect`$(\mathcal{C}, f, 0, g)$;
  $F'_{AnnihilatingSystem} \leftarrow F'_{AnnihilatingSystem} \cup \{f\}$;
  $\mathbf{F}' \leftarrow \mathbf{F}' \cup \{F', F^i\}$;
return(`Minimize`$(\mathbf{F}')$);

From Lemma 16 of [GLS01] the total cost of the function `OneDimensionalIntersect` is in

$$\mathcal{O}\left(n(L + n^2)\mathcal{U}(\deg(\mathcal{V}))\mathcal{U}(\deg(f)\deg(\mathcal{V}))\right).$$

Last, noticing that $\sum_{i=1}^{t'} \deg(\mathcal{W}'_i) \leq \deg(f)\deg(\mathcal{V})$, we deduce from Proposition 7 the cost of the minimization of $\mathbf{F}'$:

$$\mathcal{O}\left(\log(d)n^4(nL + n^\Omega)\mathcal{U}(\deg(f)\deg(\mathcal{V}))\mathcal{U}(D_2)^2\right).$$

□

5.2 Main Function

We come back to the notation from the beginning of §1.1: we recall that we are given a polynomial system $f_1 = \cdots = f_s = 0,\ g \neq 0$ to solve. We let $\mathcal{V}_i := \overline{\mathcal{V}(f_1, \ldots, f_i) \backslash \mathcal{V}(g)}$, for $i = 0, \ldots, s$. By induction we assume that we have already computed a set of DA-irreducible lifting fibers $\mathbf{F}$ representing $\mathcal{V}_i$ without redundancy and sharing all $f_1, \ldots, f_i$ as annihilating systems for $i \geq 0$. We want to compute such a representation for $\mathcal{V}_{i+1}$. Concerning the initialization of the induction we must distinguish two cases: if $\mathcal{V}_0$ is empty (that is $g = 0$) then we set $\mathbf{F} = \{\}$ and the algorithm stops; otherwise $\mathcal{V}_0$ is $\overline{k}^n$: we take 0 as primitive element and T as minimal polynomial. It is now straightforward

ALGORITHM 8: **Equidimensional Decomposition**

`GeometricSolve`$(\mathbf{f}, g)$

- $\mathbf{f}$ is a sequence of polynomials in $k[x_1, \ldots, x_n]$.
- g is a polynomial in $k[x_1, \ldots, x_n]$.

For almost all choices of M, u, $\mathbf{p}$ and almost all matrices N involved in the subfunctions, the function returns the equidimensional decomposition of the solution set of the system $\mathbf{f} = 0$, $g \neq 0$, encoded by a set of DA-irreducible lifting fibers without redundancy.

$M \leftarrow$ random $n \times n$ matrix over k;
$u \leftarrow$ random n-linear form over k;
$\mathbf{p} \leftarrow$ random point in k^n;
if $g(\mathbf{p}) = 0$ then **return**$(\{\})$;
$F_{AnnihilatingSystem} \leftarrow \{\}$;
$F_{ChangeOfVariables} \leftarrow M$;
$F_{PrimitiveElement} \leftarrow u$;
$F_{SpecializationPoint} \leftarrow \mathbf{p}$;
$F_{MinimalPolynomial} \leftarrow T$;
$F_{Parametrization} \leftarrow []$;
$F_{GenericTrace} \leftarrow []$;
for f **in** $\mathbf{f}$ **do**
    $\mathbf{F} \leftarrow$ `Intersect`$(\mathbf{F}, f, g)$;
return$(\mathbf{F})$;

to deduce the resolution algorithm from the previous function. This is detailed in Algorithm 8. We obtain the following complexity result as a corollary of Proposition 9; Hence the proof of Theorem 1.

Corollary 1 *According the above notation, for almost all choices of M, u, p and almost all matrices N involved in the subfunctions Algorithm 8 returns a correct answer within a complexity in*

$$\mathcal{O}\Big(s \log(d) n^4 (nL + n^{\Omega}) \mathcal{U}(d\delta^a)^3\Big),$$

in terms of operations in k, where d is an upper bound on the degrees of the f_i, L denotes the evaluation complexity of the straight-line program encoding $f_1, \ldots, f_s, g$ and $\delta^a := \max_{i=1,\ldots,s} \deg^{\mathrm{a}}(\mathcal{V}_i; f_1, \ldots, f_i)$,

Observe that because of the choice of the random matrix M in Algorithm 8 we could have removed all the choices of the random matrices N in the previous algorithms. But the use of the matrices N is justified in the next paragraphs.

5.3 Post Processing

Let F denote a DA-irreducible lifting fiber of a r-equidimensional variety $\mathcal{W}$. Following the notation of §2.3, F is composed of $\mathbf{e}$, M, $\mathbf{p}$, u, q, $\mathbf{v}$. As described in [GLS01, §5] it is possible to compute another fiber F' from F with a different specialization point, or a different primitive element. It is also easy to perform a change of variables concerning the free variables only or the dependent variables only. But in order to deal with positive dimensional varieties it is also necessary to be able to change the Noether position M. This task can be solved by means of a judicious use of the two fundamental functions `LiftCurve` and `OneDimensionalIntersect`. Before all let us simplify the Noether position.

Simplifying the coordinates Let us divide the matrix M into four blocks:

$$M = \begin{pmatrix} M_{1,1} & M_{1,2} \\ M_{2,1} & M_{2,2} \end{pmatrix},$$

such that $M_{1,1}$ is a $r \times r$ matrix. We assume that $M_{1,1}$ and $M_{2,2}$ are invertible, which is true for almost all matrices M. This hypothesis is not very restrictive: we can enforce all the Noether positions occurring in the resolution to have this property; Note that this is already the case in [GLS01].

We are looking for an invertible matrix R such that, written with the same block pattern we have:

$$R := \begin{pmatrix} R_{1,1} & 0 \\ R_{2,1} & R_{2,2} \end{pmatrix} \quad \text{and} \quad MR = \begin{pmatrix} Id_r & ? \\ 0 & Id_{n-r} \end{pmatrix} \text{ holds,}$$

where Id represents identity matrices. This problem admits a unique solution R given by:

$$\begin{cases} R_{1,1} = (M_{1,1} - M_{1,2}M_{2,2}^{-1}M_{2,1})^{-1}, \\ R_{2,1} = -M_{2,2}^{-1}M_{2,1}R_{1,1}, \\ R_{2,2} = M_{2,2}^{-1}. \end{cases}$$

We build the fiber F' of $\mathcal{W}$ from F by changing the Noether position to MR, the specialization point to $\mathbf{p}' = R_{1,1}^{-1}\mathbf{p}$, the primitive element $u' = u \circ N$ and the parametrization $\mathbf{v}' = R_{2,2}^{-1}\mathbf{v} - R_{2,1}\mathbf{p}$, then F' is a DA-irreducible lifting fiber of $\mathcal{W}$.

Changing the Noether Position According to the previous paragraph we assume that the Noether position given by M has the following form, with the same block pattern as above:

$$\begin{pmatrix} Id_r & ? \\ 0 & Id_{n-r} \end{pmatrix}.$$

In order the change the Noether position it suffices to be able to perform the following change of coordinates for any fixed i, $1 \leq i \leq r$:

$$\begin{aligned} Y_i &:= y_i + \sum_{j=r+1}^{n} a_j y_j, \quad a_j \in k, \text{ for } r+1 \leq j \leq n, \\ Y_j &:= y_j, \quad \text{for all } j \neq i. \end{aligned}$$

Let $\mathbf{Y} := (Y_1, \ldots, Y_n)$ and R be the matrix such that $\mathbf{y} = R\mathbf{Y}$. We denote by f_y the polynomial $y_i + \sum_{j=r+1}^{n} a_j y_j - p_i$ and by f_x the polynomial $f_y \circ M^{-1}$.

We construct the following point $\mathbf{p}'$ in k^r: $p'_i := p_i + 1$, $p'_j := p_j$, for $j \neq i$. Then we call the function `LiftCurve` with F and $\mathbf{p}'$ as arguments in order to recover the curve $\mathcal{C}$ parametrized by $t = y_i - p_i$. Then we call the function `OneDimensionalIntersect` with $\mathcal{C}$, f_x, 0 and 1 as arguments. If $V(f_x)$ intersects $\mathcal{C}$ regularly then we obtain a description of the following set of points as output:

$$\mathcal{W} \cap \mathcal{V}(y_1 - p_1, \ldots, y_{i-1} - p_{i-1}, f_y, y_{i+1} - p_{i+1}, \ldots, y_r - p_r).$$

Rewriting this variety in terms of the new coordinates $\mathbf{Y}$ yields:

$$\mathcal{W} \cap \mathcal{V}(Y_1 - p_1, \ldots, Y_r - p_r),$$

hence we deduce the following fiber F' differing from F by replacing the change of coordinates by $M \circ R$.

If the degree of F' equals $\deg(\mathcal{W})$ then F' is a good candidate to be a DA-lifting fiber, what remains to be checked. Since we know the generic trace of $\mathcal{W}$, this test can be achieved in a probabilistic way using Proposition 17 of [Lec02]. No more code is actually needed: one can re-use `LiftCurve` but the execution can be stopped just after `NestedCoordinatesWithTrace`. If this piece of code runs without raising any division by zero error then this proves that the new fiber F' is a DA-lifting fiber.

Performing this process for all the values of i ranging from 1 to r, one can compute from F almost all fibers of $\mathcal{W}$. The cost of this process is dominated by $\mathcal{O}(r)$ times the cost of `LiftCurve` (because the cost of the intersection is negligible).

Recovering the Geometric Resolution In case one would be tempted to lift the free variables and write down the geometric resolution lying over a lifting fiber using dense polynomial representations, we give a reasonable strategy. We can use the global lifting procedure in $k[[y_1 - p_1, \ldots, y_r - p_r]]$ with respect to the maximal ideal $(y_1 - p_1, \ldots, y_r - p_r)$. The complexity of the multiplication of multivariate power series is addressed in [LS01] and is mainly linear in the size of the series (up to logarithmic factors). This yields an algorithm for recovering the geometric resolution mainly linear in the size of the output. More general algorithms are proposed in [Sch00, Sch03, Sch02].

5.4 Special Case of the Integers

Our geometric resolution algorithm works well over a field k of characteristic zero. If k is the field $\mathbb{Q}$ of the rational numbers then we pick up a random prime p and first solve the system modulo p. If p is lucky then the algorithm computes a correct resolution of the system modulo p. The luckiness of p corresponds to the commutation of the computations over $\mathbb{Q}$ and modulo p. This is why the output is correct except for a finite number of primes. Once the modular resolution is completed then each returned fiber has a change of variables, a primitive element and a specialization point with entries in the range $0, \ldots, p-1$. Before recovering fibers over $\mathbb{Q}$ it is important to find changes of

variables, specialization points and primitive elements of small height. For this purpose we perform trials with small random integers. The height of the integers is increased after a certain number of failures.

From §5.3 we know how to change the Noether position and from [GLS01, §5] we know how to change the specialization point and the primitive element. Moreover as explained in §5.3 it is always possible to check whether the resulting fibers are still DA-lifting fibers or not.

Once we have found a fiber of small height then the lifting of the integers is exactly the same as in [GLS01, §4.6] except that we must use the new `GlobalNewton` from §3 based on the deflation algorithm.

6 Examples

The algorithm presented in this paper has been implemented within the Magma computer algebra system. The package is called Kronecker [Lec99] and the current version is 0.166. The timings we give in this section concern a 1 GHz Pentium III based computer with 512 MB of internal memory and running Linux 2.4; We use Magma 2.9. More examples are given in [Lec01].

The family of examples we have chosen to illustrate the behavior of our implementation comes from the following problem: Let k be a field and $S(X, Y, Z)$ be a polynomial in $k[X, Y, Z]$ such that $S(0, 0, 0) = 0$. We introduce three infinite families of new symbols $(x_i)_{i\in\mathbb{N}}$, $(y_i)_{i\in\mathbb{N}}$, $(z_i)_{i\in\mathbb{N}}$ and the following formal power series in the parameter t:

$$\hat{X} := \sum_{i\geq 0} x_i t^i, \qquad \hat{Y} := \sum_{i\geq 0} y_i t^i, \qquad \hat{Z} := \sum_{i\geq 0} z_i t^i.$$

For any nonnegative integer σ we address the problem of describing the set of power series $\hat{X}$, $\hat{Y}$ and $\hat{Z}$ satisfying $S(\hat{X}, \hat{Y}, \hat{Z}) \in \mathcal{O}(t^\sigma)$. For a fixed value of σ this problem can be turned into a polynomial system involving at most σ equations and 3σ variables.

In the rest of the paper we chose $S := (X^2 + Y^3 + Z^4)(X^2 + Y^3 + Z^5)$ and $k := \mathbb{Z}/4294967291\mathbb{Z}$, the polynomial systems we obtain for each σ are simplified a bit using the following rules:

- Each equation being a power of a variable is removed and this variable is replaced by zero in the remaining system. This rule is applied as many times as necessary to remove all these trivial equations.
- We only consider as unknowns of the system obtained after applying the previous rule the variables that do appear in the system and we discard the others.

For instance let us build the system for $\sigma = 11$: first we compute the power series expansion of $P := S(\hat{X}, \hat{Y}, \hat{Z})$ at precision $\mathcal{O}(t^{11})$, then we let f_i denote the coefficient of P of degree i, for $i = 0, \ldots, 10$. The polynomial system $F = \{f_0, \ldots, f_{10}\}$ we are to solve can be simplified this way: since $f_4 = x_1^4$ we discard f_4 and replace x_1 by 0 in F. After this substitution f_6 becomes y_1^6, hence we remove f_6 and set $y_1 = 0$. Last there are left

3 nonzero polynomials in the system, of respective degrees 6, 9 and 9. The number of effective unknowns in this system is 7. Using Kronecker we find that the solution set of this system is composed of one component of codimension 3 and degree 18 which is not multiple and one multiple component of codimension 2 and degree 1. This computation requires less than 5 seconds and about 3 MB of memory.

Dimensions and degrees of the components solutions for values of σ ranging from 11 to 14 are given in the following table. For each component we given a couple (c, d) where c is the codimension and d the degree of the component. For multiple components we display the degree in bold font. The column n contains the number of effective unknowns of the system. The column d provides the sequence of the total degrees of the equations. The last two columns contain the total time and the memory consumption of the computation.

σ	n	d	(codim,deg)	time	memory
11	7	6, 9, 9	(2,**1**), (3,18)	5 s	3 MB
12	10	6, 9, 9, 9	(2,**1**), (4,44)	28 s	5 MB
13	13	6, 9, 9, 9, 9	(3,**3**), (5,110)	285 s	14 MB
14	16	6, 9, 9, 9, 9, 9	(3,**3**), (3,200)	3382 s	100 MB

Concerning comparisons with other tools we would like to underline that no function strictly similar to our `GeometricSolve` of Algorithm 8 is available in Magma or in other computer algebra software. Comparison with straight Gröbner basis functions is not fair but for information we mention that the graded reverse lexicographical bases computed with Magma require: 1.5 s and 2 MB for $\sigma = 11$, 71 s and 9 MB for $\sigma = 12$ and for $\sigma = 13$ we have stopped the computation after 1 hour, 210 MB were in use. Primary and Prime decomposition functions available in Magma are not faster than Gröbner bases on these examples.

7 Conclusion

In this paper we have presented a practical probabilistic algorithm to compute the equidimensional decomposition of an algebraic closed set. For the first time we have been able to state a precise upper bound complexity for this problem, with an explicit reasonable exponent. Although mechanisms that handle the decomposition in itself are quite classical, the use of the deflation algorithm as a generalization of Newton's operator is very new and validates the fiber encoding approach even for multiple components. Estimating the probability of success of this algorithm is subject to further work.

Computing the prime decomposition from the equidimensional one seems to be an easy task in theory but is not at all in practice. The current distribution of Kronecker already contains experimental functions that compute prime decompositions over $\mathbb{Z}/p\mathbb{Z}$ but recombinations over $\mathbb{Q}$ are still in development.

Acknowledgments: I am very grateful to Allan Steel and all the Magma team

for their kind support and constant efforts in improving performances. I thank Olivier Piltant for having suggested me the family of examples of §6.

References

[AHU74] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The design and analysis of computer algorithms*. Addison-Wesley, 1974.

[BC95] W. Bosma and J. Cannon. Handbook of Magma functions. Sydney: School of Mathematics and Statistics, University of Sydney, 1995.

[BCM94] W. Bosma, J. Cannon, and J. Matthews. Programming with algebraic structures: design of the Magma language. In M. Giesbrecht, editor, *Proceedings of ISSAC'94*. ACM, 1994.

[BCP97] W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system I: The user language. *Journal of Symbolic Computation*, 24:235–265, 1997.

[BCS97] P. Bürgisser, M. Clausen, and M. A. Shokrolahi. *Algebraic Complexity Theory*. Springer, 1997.

[BP94] D. Bini and V. Pan. *Polynomial and matrix computations*. Progress in theoretical computer science. Birkhäuser, 1994.

[Cas01] D. Castro. *Sobre la complejidad de la representación de variedades algebraicas*. PhD thesis, Universidad de Cantabria, Departemento de Matemáticas, Estadística y Computación, Santander, Spain, July 2001.

[CG83] A. L. Chistov and D. Y. Grigoriev. Subexponential time solving systems of algebraic equations. LOMI preprint E-9-83, E-10-83, Steklov Institute, Leningrad, 1983.

[CGH$^+$] D. Castro, M. Giusti, J. Heintz, G. Matera, and L. M. Pardo. The hardness of polynomial equation solving. To appear in *Foundations of Computational Mathematics*. DOI 10.1007/s10208-002-0065-7.

[Chi96] A. L. Chistov. Polynomial-time computation of the dimension of algebraic varieties in zero-characteristic. *Journal of Symbolic Computation*, 22(1):1–25, July 1996.

[Chi97] A. L. Chistov. Polynomial-time computation of the dimensions of components of algebraic varieties in zero-characteristic. *Journal of Pure and Applied Algebra*, 117–118:145–175, 1997.

[CHMP01] D. Castro, K. Hägele, J. E. Morais, and L. M. Pardo. Kronecker's and Newton's approaches to solving: A first comparison. *Journal of Complexity*, 17(1):212–303, 2001.

[CP96] J. Cannon and C. Playoust. Magma: A new computer algebra system. *Euromath Bulletin*, 2(1):113–144, 1996.

[CW90] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251–280, March 1990.

[DST87] J. Davenport, Y. Siret, and E. Tournier. *Calcul formel. Systèmes et algorithmes de manipulations algébriques*. Masson, 1987.

[EM99] M. Elkadi and B. Mourrain. A new algorithm for the geometric decomposition of a variety. In *Proceedings of ISSAC'99*. ACM, 1999.

[Ful84] W. Fulton. *Intersection Theory*. Number 3 in Ergebnisse der Mathematik. Springer, second edition, 1984.

[GG99] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, 1999.

[GH91] M. Giusti and J. Heintz. Algorithmes - disons rapides - pour la décomposition d'une variété algébrique en composantes irréductibles et équidimensionelles. In T. Mora and C. Traverso, editors, *Proceedings of MEGA'90*, volume 94 of *Progress in Mathematics*, pages 169–194. Birkhäuser, 1991.

[GH93] M. Giusti and J. Heintz. La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial. In D. Eisenbud and L. Robbiano, editors, *Computational Algebraic Geometry and Commutative Algebra*, volume XXXIV of *Symposia Matematica*, pages 216–256. Cambridge University Press, 1993.

[GH01] M. Giusti and J. Heintz. Kronecker's smart, little black-boxes. In A. Iserles and R. DeVore, editors, *Proceedings of Foundations of Computational Mathematics, Oxford 1999*, volume 284 of *London Mathematical Society Lecture Note Series*, pages 69–104. Cambridge University Press, 2001.

[GHH+97] M. Giusti, K. Hägele, J. Heintz, J. E. Morais, J. L. Montaña, and L. M. Pardo. Lower bounds for Diophantine approximation. In *Proceedings of MEGA'96*, volume 117,118, pages 277–317. Journal of Pure and Applied Algebra, 1997.

[GHL+00] M. Giusti, K. Hägele, G. Lecerf, J. Marchand, and B. Salvy. Computing the dimension of a projective variety: the projective Noether Maple package. *Journal of Symbolic Computation*, 30(3):291–307, September 2000.

[GHM+98] M. Giusti, J. Heintz, J. E. Morais, J. Morgenstern, and L. M. Pardo. Straight-line programs in geometric elimination theory. *Journal of Pure and Applied Algebra*, 124:101–146, 1998.

[GHMP95] M. Giusti, J. Heintz, J. E. Morais, and L. M. Pardo. When polynomial equation systems can be solved fast? In G. Cohen, M. Giusti, and T. Mora, editors, *Applied Algebra, Algebraic Algorithms and Error Correcting Codes, Proceedings AAECC-11*, volume 948 of *Lecture Notes in Computer Science*, pages 205–231. Springer, 1995.

[GHMP97] M. Giusti, J. Heintz, J. E. Morais, and L. M. Pardo. Le rôle des structures de données dans les problèmes d'élimination. *Comptes Rendus de l'Académie des Sciences de Paris*, 325:1223–1228, 1997.

[GKZ94] I. M. Gel'fand, M. M. Kapranov, and A. V. Zelevinski. *Discriminants, Resultants and Multidimensional Determinants*. Birkhäuser, Boston, 1994.

[GLS01] M. Giusti, G. Lecerf, and B. Salvy. A Gröbner free alternative for polynomial system solving. *Journal of Complexity*, 17(1):154–211, 2001.

[Hei83] J. Heintz. Definability and fast quantifier elimination in algebraically closed fields. *Theoretical Computer Science*, 24(3):239–277, 1983.

[HMPS00] K. Hägele, J. E. Morais, L. M. Pardo, and M. Sombra. On the intrinsic complexity of the arithmetic Nullstellensatz. *Journal of Pure And Applied Algebra*, 146(2):103–183, 2000.

[HMW01] J. Heintz, G. Matera, and A. Waissbein. On the time-space complexity of geometric elimination procedures. *Applicable Algebra in Engineering, Communication and Computing*, 11(4):239–296, 2001.

[JKSS02] G. Jeronimo, T. Krick, J. Sabia, and M. Sombra. The computational complexity of the Chow form. Manuscript, March 2002.

[JPS01] G. Jeronimo, S. Puddu, and J. Sabia. Computing Chow forms and some applications. *Journal of Algorithms*, 41(1):52–68, 2001.

[JS00] G. Jeronimo and J. Sabia. Probabilistic equidimensional decomposition. *Comptes rendus de l'Académie des Sciences de Paris*, 331(6):485–490, 2000.

[JS02] G. Jeronimo and J. Sabia. Effective equidimensional decomposition of affine varieties. *Journal of Pure and Applied*, 169(2–3):229–248, 2002.

[Koi97] P. Koiran. Randomized and deterministic algorithms for the dimension of algebraic varieties. In *Proceedings of the 38th Annual IEEE Computer Society Conference on Foundations of Computer Science (FOCS)*. ACM, 1997.

[KP96] T. Krick and L. M. Pardo. A computational method for Diophantine approximation. In L. González-Vega and T. Recio, editors, *Algorithms in Algebraic Geometry and Applications. Proceedings of MEGA'94*, volume 143 of *Progress in Mathematics*, pages 193–254. Birkhäuser Verlag, 1996.

[Lec99] G. Lecerf. Kronecker, a Magma package for polynomial system solving, from 1999. http://kronecker.medicis.polytechnique.fr.

[Lec00] G. Lecerf. Computing an equidimensional decomposition of an algebraic variety by means of geometric resolutions. In *Proceedings of ISSAC'2000*, pages 209–216. ACM, 2000.

[Lec01] G. Lecerf. *Une alternative aux méthodes de réécriture pour la résolution des systèmes algébriques*. PhD thesis, École polytechnique, Palaiseau, France, 2001.

[Lec02] G. Lecerf. Quadratic Newton iteration for systems with multiplicity. *Foundations of Computational Mathematics*, 2(3):247–293, 2002.

[LS01] G. Lecerf and É. Schost. Fast multivariate power series multiplication in characteristic zero. To appear in *SADIO Electronic Journal on Informatics and Operations Research*, preliminary version of 2001. Manuscript available at http://www.math.uvsq.fr/~lecerf.

[Mag] Magma. http://magma.maths.usyd.edu.au.

[Mat86] H. Matsumura. *Commutative Ring Theory*. Cambridge University Press, 1986.

[Mat99] G. Matera. Probabilistic algorithms for geometric elimination. *Applicable Algebra in Engineering, Communication and Computing*, 9(6):463–520, 1999.

[Mor97] J. E. Morais. *Resolución eficaz de sistemas de ecuaciones polinomiales*. PhD thesis, Universidad de Cantabria, Santander, Spain, 1997.

[Mum95] David Mumford. *Algebraic Geometry I - Complex Projective Varieties*. Springer-Verlag, 1995.

[Par95] L. M. Pardo. How lower and upper complexity bounds meet in elimination theory. In G. Cohen, M. Giusti, and T. Mora, editors, *Applied Algebra, Algebraic Algorithms and Error Correcting Codes, Proceedings of AAECC-5*, volume 948 of *Lecture Notes in Computer Science*, pages 33–69. Springer, Berlin, 1995.

[PS78] F. P. Preparata and D. V. Sarwate. An improved parallel processor bound in fast matrix inversion. *Information Processing Letters*, 7(3):148–150, April 1978.

[Roj00] M. Rojas. Computing complex dimension faster and deterministically. Manuscript of City University of Hong Kong, 2000.

[Sam67] P. Samuel. *Méthodes d'algèbre abstraite en géométrie algébrique*. Springer-Verlag, 2nd edition, 1967.

[Sch00] É. Schost. *Sur la résolution des systèmes polynomiaux à paramètres*. PhD thesis, École polytechnique, Palaiseau, France, 2000.

[Sch02] É. Schost. Degree bounds and lifting techniques for triangular sets. In Teo Mora, editor, *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, pages 238–235. ACM, 2002.

[Sch03] É. Schost. Computing parametric geometric resolutions. *Applicable Algebra in Engineering, Communication and Computing*, 13(5):349–393, 2003.

[Stu94] B. Sturmfels. On the Newton polytope of the resultant. *Journal of Algebraic Combinatorics*, 3:207–236, 1994.

[SV00] A. J. Sommese and J. Verschelde. Numerical homotopies to compute generic points on positive dimensional algebraic sets. *Journal of Complexity*, 16(3):572–602, 2000.

[SVW01a] A. J. Sommese, J. Verschelde, and C. W. Wampler. Numerical decomposition of the solution sets of polynomial systems into irreducible components. *SIAM Journal of Numerical Analysis*, 38(6):2022–2046, 2001.

[SVW01b] A. J. Sommese, J. Verschelde, and C. W. Wampler. Numerical irreducible decomposition using projections from points on the components. In E. L. Green, S. Hosten, R. C. Laubenbacher, and V. A. Powers, editors, *Symbolic Computation: Solving Equations in Algebra, Geometry, and Engineering*, volume 286 of *Contemporary Mathematics*. AMS, 2001.

[SVW01c] A. J. Sommese, J. Verschelde, and C. W. Wampler. Using monodromy to decompose solution sets of polynomial systems into irreducible components. In C. Ciliberto, F. Hirzebruch, R. Miranda, and M. Teicher, editors, *Application of Algebraic Geometry to Coding Theory, Physics, and Computation, Proceedings of NATO Conference 2001*. Kluwer, 2001. http://www.math.uic.edu/~jan/.

[SVW02] A. J. Sommese, J. Verschelde, and C. W. Wampler. Symmetric functions applied to decomposing solution sets of polynomial systems. *SIAM Journal of Numerical Analysis*, 40(6):2026–2046, 2002.

[Vor99] N. Vorobjov. Complexity of computing the local dimension of semialgebraic set. *Journal of Symbolic Computation*, 27(6):565–579, 1999.