Numerical Decomposition of the Solution Sets of Polynomial Systems into Irreducible Components

Andrew J. Sommese, Jan Verschelde, and Charles W. Wampler

Abstract

In engineering and applied mathematics, polynomial systems arise whose solution sets contain components of different dimensions and multiplicities. In this article we present algorithms, based on homotopy continuation, that compute much of the geometric information contained in the primary decomposition of the solution set. In particular, ignoring multiplicities, our algorithms lay out the decomposition of the set of solutions into irreducible components, by finding, at each dimension, generic points on each component. As by-products, the computation also determines the degree of each component and an upper bound on its multiplicity. The bound is sharp (i.e., equal to one) for reduced components. The algorithms make essential use of generic projection and interpolation, and can, if desired, describe each irreducible component precisely as the common zeroes of a finite number of polynomials.

AMS Subject Classification. 65H10, 68Q40.
Keywords. components of solutions, embedding, generic points, homotopy continuation, irreducible components, numerical algebraic geometry, polynomial system, primary decomposition.

Introduction

Given a system of polynomial equations in $\mathbb{C}^N$,

$$f(\mathbf{x}) := \left[\begin{array}{c} f_1(x_1,\ldots,x_N) \\ \vdots \\ f_n(x_1,\ldots,x_N) \end{array} \right], \tag{1}$$

with not all f_i equal to the zero polynomial, we seek a description, by numerical means, of its solution set. As a set with all multiplicity information ignored, this is the algebraic variety $\mathbf{V}(f) := \{ \ \mathbf{x} \in \mathbb{C}^N \ | \ f_i(\mathbf{x}) = \mathbf{0}, \text{ for } i = 1,\ldots,n \ \}$. The closures of the connected components of the set of manifold points of $\mathbf{V}(f)$ are the irreducible components of $\mathbf{V}(f)$. The irreducible components of $\mathbf{V}(f)$ can have dimensions varying from zero (isolated points) up to $N-1$ (hypersurfaces). The multiplicity of a given component, which precisely generalizes the notion of the multiplicity of a root of a polynomial in one variable, is a positive integer.

In addition to finding all of the isolated solutions, our algorithms certify the existence of each higher-dimensional irreducible component by finding one or more generic points on it. The degree of each such component and an upper bound on its multiplicity are also determined. For components having multiplicity equal to one, the computed bound is sharp. Furthermore, a post-processing algorithm describes each component precisely as the set of common zeroes of a finite set of polynomials, whose coefficients have been approximated numerically by interpolation. Accordingly, these form a numerical approximation to the primary decomposition of the radical $\sqrt{\mathcal{I}(f)}$ of the ideal $\mathcal{I}(f) \subset \mathbb{C}[x_1, \ldots, x_N]$ defined by the original system of polynomials, plus some of the multiplicity information of the primary decomposition of $\mathcal{I}(f)$.

Our principle computational tool is homotopy continuation, which has proven effective for obtaining a finite set of solutions of a given polynomial system, such that the set contains all isolated roots of the system [1, 2, 27, 29, 31]. Besides providing numerical approximations to the roots of a specific problem, which can be useful for engineering activities such as mechanism design [32], the number of isolated solutions to a single generic problem conveys information about every member of a suitably parameterized family of systems [30]. This kind of result has, for example, guided developments in theoretical kinematics, where early numerical results were subsequently confirmed by analytical means. Some of the problems solved by this approach are well beyond the current capabilities of algorithms for Gröbner bases or other symbolic reduction methods. An example is the nine-point path-synthesis problem for four-bar linkages, a problem which has total degree 7^8 and which, according to the numerical evidence obtained by homotopy continuation, has 8652 isolated, nonsingular roots in general [43]. Raghavan and Roth [36] give a survey of the situation from the viewpoint of working kinematicians.

The prior success of continuation in finding all isolated solutions encourages us to explore the extension of homotopies to find higher-dimensional components of solutions. Symbolic methods for computing primary decompositions, which are based on triangular sets [4, 5] or Gröbner bases [14], have been implemented in several computer algebra systems [18, 25]. See also [22] and [24]. According to Greuel [25], "all known algorithms for primary decompositions in $K[\mathbf{x}]$ are quite involved." In comparison, our numerical algorithms, founded soundly on results from algebraic geometry, seem more straightforward, although admittedly, careful attention to numerical issues, such as round-off and ill-conditioning, is necessary to produce a robust algorithm. For some of the examples we discuss in this article, multi-precision arithmetic was needed to arrive at meaningful results.

One product of the numerical primary decomposition is the list of isolated solutions of a system of polynomials, with no requirement that the solutions be nonsingular. Previous algorithms based on homotopy continuation produce lists of solutions that include all isolated solutions, but these algorithms cannot determine whether or not certain types of singular solutions are isolated.

This article builds on previous work in [38, 39] that gave algorithms for finding generic points on all the solution components. However, these algorithms did not determine the number of irreducible components and did not group the generic solution points by component. The algorithms of this article perform these operations to provide the numerical irreducible decomposition.

The organization of this article is as follows. We begin in §1 with a review of the basic concept of primary decomposition and then give a sketch of our algorithms in §2. To better motivate the work, in §3, we describe the application of these algorithms to an illustrative example. Then, §4 gives detailed presentations of the algorithms in terms of pseudocode. A detailed theoretical justification for the algorithms, based on results from algebraic geometry reviewed in §5, is given in §6. The efficacy of our preliminary implementation of the algorithms on several test examples is demonstrated in §7, along with a discussion of numerical requirements for fitting polynomial equations to numerically sampled points. In a final section of conclusions, §8, we sum up our results and point to areas where more research is still required.

Acknowledgments. The first author would like to thank Colorado State University, the Duncan Chair of the University of Notre Dame, the University of Notre Dame, and the Mathematical Sciences Research Institute in Berkeley, for their support. The second author acknowledges support from the NSF under Grant DMS - 9804846 at the Department of Mathematics, Michigan State University.

Contents

1 Background I: Decomposition into Irreducible Components

Before describing our algorithms for finding a numerical decomposition of the solution set of a polynomial system into irreducible components, we first review the essential concepts. For the sake of exposition, we give, in this section, a self-contained presentation of the basic ideas without getting overly technical. The reader may consult the references for precise statements of the basic results (see [16, 35] for an introductory presentation, and [19] for a full coverage of the algebraic theory with helpful discussions on the underlying geometry). The theoretical justification of our algorithms is given in detail in §6.

The solution of a system of polynomial equations can be regarded from either a geometric or an algebraic point of view. Given a system f of n polynomials on $\mathbb{C}^N$, as in (1), the geometric object of interest is the set $f^{-1}(\mathbf{0}) \subset \mathbb{C}^N$ of solutions of $f(\mathbf{x}) = \mathbf{0}$. This is often called the *variety* of f, $\mathbf{V}(f)$. On the algebraic side, the object of interest is the *ideal* $I_f =< f_1, \ldots, f_n >$, which is the set of all polynomials that are algebraic combinations of the given polynomials. For most applications, especially in engineering, the solution set $\mathbf{V}(f)$ is the main item of interest. However, a solution procedure using computer algebra works by manipulating the equations, that is, it works with the ideal I_f.

For this discussion, an *affine variety* is the solution set in $\mathbb{C}^N$ of a system of polynomial equations. An *irreducible affine variety* V is an affine variety that cannot be expressed as the union of a finite number of proper subvarieties. This is equivalent to

$$\mathcal{I}(V) := \{ \ p \in \mathbb{C}[x_1, \ldots, x_N] \mid p(V) = 0 \ \} \tag{2}$$

being a prime ideal, i.e., if $pq \in \mathcal{I}(V)$ then either $p \in \mathcal{I}(V)$ or $q \in \mathcal{I}(V)$. It is also equivalent to $V_{\rm reg}$, the subset of manifold points of V, being connected in the complex topology (this is the usual Euclidean topology, see §5.1). A well-known result is that every affine variety V has a unique decomposition into irreducible varieties as

$$V = V_1 \cup \cdots \cup V_m \tag{3}$$

where m is finite, each V_i is irreducible, and $V_i \not\subset V_j$ for $i \neq j$. Geometrically, the V_i are the closures in the complex topology of the distinct connected components of $V_{\rm reg}$, the set of manifold points of V. This is sometimes called the *minimal decomposition* or *irreducible decomposition* of the variety.

The dimension of the irreducible components can vary from zero (isolated points) up to $N-1$. An N-dimensional component would be the whole space $\mathbb{C}^N$, which happens if and only if f is identically zero, a trivial case that we ignore. It is useful in stating our algorithms to organize the irreducible components by dimension, so we write the decomposition of the entire solution set $Z = \mathbf{V}(f)$ as the nested union

$$Z := \bigcup_{i=0}^{N-1} Z_i := \bigcup_{i=0}^{N-1} \bigcup_{j \in \mathcal{I}_i} Z_{ij} \tag{4}$$

where Z_i is the union of all i-dimensional components, the Z_{ij} are the irreducible components, and the index sets $\mathcal{I}_i$ are finite and possibly empty.

Let $\sqrt{I}$ denote the radical of an ideal I, i.e.,

$$\sqrt{I} := \{ \ p \in \mathbb{C}[x_1, \ldots, x_N] \mid p^m \in I \text{ for some integer } m \geq 1 \ \}. \tag{5}$$

It is a basic fact that $\mathcal{I}(\mathbf{V}(I)) = \sqrt{I}$ for ideals $I \subset \mathbb{C}[x_1, \ldots, x_N]$. The decomposition (4) is equivalent to the unique decomposition

$$\sqrt{I_f} := \bigcap_{i=0}^{N-1} \bigcap_{j \in \mathcal{I}_i} \mathcal{I}(Z_{ij}) \tag{6}$$

of $\sqrt{I_f}$ as an intersection of the prime ideals $\mathcal{I}(Z_{ij})$.

This is a special case of the primary decomposition of an ideal. An ideal I is *primary* if $pq \in I$ implies either $p \in I$ or $q^m \in I$ for some power $m > 0$. If I is primary then $\sqrt{I}$ is a prime ideal. The *primary decomposition* of an ideal I, is an expression of the ideal as the intersection of primary ideals:

$$I = \mathcal{P}_1 \cap \cdots \cap \mathcal{P}_r, \tag{7}$$

where r is finite, $\mathcal{P}_i$ is primary, and $\mathcal{P}_i \not\supset \cap_{j \neq i} \mathcal{P}_j$. Unlike in (6), the primary decomposition is not necessarily unique. The failure of uniqueness occurs if there are two distinct primary ideals $\mathcal{P}_i, \mathcal{P}_j$ in the decomposition with $\sqrt{\mathcal{P}_i} \subset \sqrt{\mathcal{P}_j}$. In this case, the irreducible variety $\mathbf{V}(\mathcal{P}_j)$ is properly contained in some irreducible component of $\mathbf{V}(I)$, and is called an *embedded component* of I.

The primary decomposition (7), of a polynomial system f contains extra information about the system of polynomials, beyond the decomposition (4). First, we note that because of possible embedded components, $r \geq m$; that is, the number of primary ideals r in the primary decomposition can be greater that the number of irreducible varieties m. Second, the *multiplicity* of a component Z_{ij} can be determined from the primary ideal $\mathcal{P}_k$ with $\mathbf{V}(\mathcal{P}_k) = Z_{ij}$. A familiar example is the case of a multiple root in a polynomial of one variable; for example, the variety for the primary ideal $< x^2 >$ is just the point $x = 0$, but the ideal shows it to have a multiplicity of 2. We often think of it as the point $x = 0$ counted twice. This is discussed further in §5.

2 Overview of Algorithms

The terminology reviewed in §1 permits us to make clear the contribution of our algorithms. Our main goal is to numerically represent the decomposition of the solution set into its irreducible components as in (4). To this end, our algorithms can provide two levels of output. In the main processing step, which we call **IrreducibleDecomposition**, the output is a listing of the irreducible components by dimension and degree, and a set of witness points on each component. It also provides an upper bound on the multiplicity of the component. For a more complete representation of each component, a post-processing step, called **PolynomialsGenerate**, generates a set of polynomials that vanish precisely on that component.

We do not find the complete primary decomposition in the sense of (7): we do not identify the embedded components and the ideals we generate for the irreducible components are not necessarily either primary or prime. However, the multiplicity bounds give partial information about the primary ideals. In fact, **IrreducibleDecomposition** sharply identifies all components of multiplicity one.

The following paragraphs describe the main idea of each of our algorithms. A detailed description in pseudocode is given in §4 and the theory behind them is given in §6.

2.1 IrreducibleDecomposition

Given a polynomial system f as in (1) the key outputs of algorithm **IrreducibleDecomposition** are the degree of each irreducible component $\deg Z_{ij}$ and a witness point set W, defined as follows.

Definition 2.1 For a polynomial system f having a decomposition Z into irreducible components Z_{ij} as in (4), a *witness point set* W is a finite set of points of the form

$$W := \bigcup_{i=0}^{N-1} W_i := \bigcup_{i=0}^{N-1} \bigcup_{j \in \mathcal{I}_i} W_{ij}, \tag{8}$$

where

1. W_{ij} is a set of points of the irreducible component Z_{ij}, (i.e., $W_{ij} \subset Z_{ij}$),
2. W_{ij} is distinct from all the other irreducible components, (i.e., $W_{ij} \cap Z_{kl} = \emptyset$ for $(i,j) \neq (k,l)$).
3. W_{ij} contains $\deg Z_{ij}$ points, each occurring ν_{ij} times for some integer $\nu_{ij} \geq \mu_{ij}$, where μ_{ij} is the multiplicity of Z_{ij} as an irreducible component of $f^{-1}(\mathbf{0})$. Moreover, if $\mu_{ij} = 1$, then $\nu_{ij} = 1$.

IrreducibleDecomposition proceeds in two phases: **WitnessGenerate** finds an unsorted superset of witness points that **WitnessClassify** subdivides into the witness point subsets W_{ij}. The workings of these sub-algorithms is outlined in the following paragraphs.

2.1.1 WitnessGenerate

This algorithm is the function provided by the prior work in [38, 39]. The fundamental idea is that a generic linear variety L_k of dimension k will cut an irreducible variety V of dimension $N-k$ in a finite set of points, equal in number to the degree of V. The qualifier "generic" is key, and is discussed in detail in [38, 39]. Furthermore, in an appropriate homotopy, the endpoints of the solution paths will land on each irreducible component multiple times, depending on the multiplicity and degree of the component. However, some endpoints also land on the positive dimensional intersections of L_k with varieties of dimension greater than $N-k$. L_k will not intersect varieties of dimension less than $N-k$.

The algorithm given in [38] is more efficient than the earlier [39], owing to its use of an embedding strategy wherein $L_{k+1} \supset L_k$ as the algorithm ascends sequentially from $k=1$ to

$k = N$. (This implies that we find points on solution sets in descending order of dimension $i = N - k$ running from $i = N - 1$ to $i = 0$.) The linear variety L_k is specified as the intersection of $N - k$ hyperplanes, which we refer to as "slicing planes." The inclusion of L_k within L_{k+1} for each k is accomplished by removing the slicing planes one by one in a series of N homotopies. In practice, the condition of genericity is obtained by using a random number generator to choose the coefficients of the equations that define each hyperplane. The result is an algorithm that succeeds with probability 1.

We say that **WitnessGenerate** finds a witness point superset $\widehat{W}$, which is organized only by dimension. Specifically,

$$\widehat{W} = \bigcup_{i=0}^{N-1} \widehat{W}_i, \tag{9}$$

where $\widehat{W}_i = W_i \cup J_i$, in which W_i is a witness point set for components of dimension i and J_i is a set of points on components of greater than i dimensions. (J stands for "junk.") It is important to note that the points in $\widehat{W}_i$ are undifferentiated, that is, at the conclusion of **WitnessGenerate**, we do not know how the points subdivide into the subsets W_{ij} or J_i.

2.1.2 WitnessClassify

This algorithm is the heart of the new contribution of this article. Its purpose is to convert a witness point superset $\widehat{W}$ into a witness point set W. Starting at dimension $i = N - 1$ and descending sequentially to $i = 0$, **WitnessClassify** proceeds to classify the points in each of the witness supersets, $\widehat{W}_i$, by first separating out the points on higher-dimensional sets, J_i, and then subdividing the remaining points according to membership in irreducible varieties.

The key operation in the inner workings of **WitnessClassify** is the construction of *filtering polynomials*, p_{ij}, each of which vanishes on the entire irreducible component Z_{ij} but which is nonzero, with probability 1, for any point in $\widehat{W}$ that is not on Z_{ij}. The geometric concept is as follows. For an irreducible component Z_{ij}, which by definition has dimension i, pick a generic linear subspace of directions having dimension $N - i - 1$ and define a new set constructed by replacing each point of Z_{ij} with its expansion along the chosen subspace directions. The result is an $(N - 1)$-dimensional hypersurface. The filtering polynomial is the unique polynomial that vanishes on this hypersurface. It can be constructed numerically as discussed further below.

The concept of a filtering polynomial can be illustrated by considering an irreducible 1-dimensional curve C in $\mathbb{C}^3$, which can be visualized by its real part, a 1-real-dimensional curve in $\mathbb{R}^3$. Further, suppose we are given a point $\mathbf{p} \in \mathbb{C}^3$, whose membership in C is to be determined. Pick a direction $\mathbf{v} \in \mathbb{P}^2$, that is, $\mathbf{v}$ is some line through the origin in $\mathbb{C}^3$. Replace each point in C by a line through the point parallel to $\mathbf{v}$ to get a 2-dimensional surface S. S contains C, by construction. So if $\mathbf{p} \in C$, then $\mathbf{p} \in S$. On the other hand, if $\mathbf{p}$ is not on C, it will be in S if and only if there is a point $\mathbf{q} \in C$ such that $\mathbf{p} - \mathbf{q}$ is parallel to $\mathbf{v}$. So the set of all possible choices of $\mathbf{v}$ to make this happen are parameterized by $\mathbf{q} \in C$, a 1-dimensional set. But our algorithm chooses $\mathbf{v}$ generically (in practice, randomly independent of $\mathbf{p}$) from a 2-dimensional set $\mathbb{P}^2$, so with probability one, S does not contain $\mathbf{p}$. Hence, a test of membership in S is a probability-one surrogate for a test of membership in

C. The idea generalizes readily to varieties and spaces of any dimension, as stated rigorously in Lemma 5.6.

Suppose we have filtering polynomials for all components of dimension greater than i. Then, we may extract W_i from $\widehat{W}_i$ by finding the points J_i that lie on the higher-dimensional components: $W_i = \widehat{W}_i \setminus J_i$. The next task is to sort the points in W_i by their membership in the irreducible components Z_{ij}. Choosing arbitrarily some point $\mathbf{w} \in W_i$, we move the slicing planes (see previous subsection) that pick $\mathbf{w}$ out of the underlying irreducible component Z_{ij}, using continuation. In this manner, we can generate an arbitrary number of new points, preferably widely dispersed, on Z_{ij}. After picking a generic projection direction to expand Z_{ij} into a hypersurface, as discussed in the previous paragraphs, we find the lowest-degree polynomial that interpolates the samples, always taking extra samples as necessary to ensure that the true hypersurface has been correctly determined. This is the filtering polynomial p_{ij}, which can then be used to find any additional points in W_i that lie on Z_{ij}. Together with $\mathbf{w}$, these form the set W_{ij}. We then choose a new point from those in W_i that are not yet sorted, and repeat the process until all points are sorted. With all the sets W_{ij} sorted and corresponding filtering polynomials p_{ij} in hand, we proceed to dimension $i-1$ and apply the same method. The procedure starts at dimension $i = N-1$ with an empty list of filtering polynomials. This is justified because, by assumption, there are no components of dimension N, hence $J_{N-1} = \emptyset$ and $\widehat{W}_{N-1} = W_{N-1}$. After the algorithm concludes at dimension $i = 0$, the result is the witness point set W decomposed by irreducible components into the sets W_{ij}.

By Lemma 5.5 (below), the degree of the filtering polynomial p_{ij} is the same as the degree of Z_{ij}. Moreover, due to the properties of the witness point superset $\widehat{W}$, the multiplicity of Z_{ij} is bounded by $\nu_{ij} = \#(W_{ij})/\deg Z_{ij}$. We have repeated points in $\widehat{W}$, because m paths as solutions of a polynomial homotopy converge to a solution of multiplicity m.

WitnessClassify is implemented using the following list of sub-algorithms, which are necessary for constructing the filtering polynomials.

1. *Sample:* Given a point $\mathbf{w} \in Z$, produces additional points on the same irreducible component by continuation.
2. *Projection:* Projects points along a given projective subspace.
3. *Fit:* Given a set of projected sample points, finds the best-fit polynomial of a specified degree.
4. *Interpolate:* Finds an interpolating hypersurface by sampling, projecting and fitting, starting at degree 1 and incrementing until a perfect fit is found (subject to numerical accuracy).

A more complete description of **WitnessClassify** and its sub-algorithms is given in §4.

2.2 PolynomialsGenerate

Once the work of **WitnessClassify** is finished, it is straightforward to the construct a set of polynomials whose set of common zeroes is one of the irreducible components. By Lemma 5.8, it suffices to construct $N+1$ hypersurfaces using the Projection and Fit sub-algorithms already implemented within **WitnessClassify**. These can be determined using the sample points found on each component during classification: each hypersurface is determined by choosing a new generic projection. Since generic projection preserves degree, the degrees of hypersurfaces are all the same and equal to the degree already determined for Z_{ij}. The algorithm **PolynomialsGenerate** generates sufficiently many equations to completely identify the solution sets.

There are two special cases where fewer than $N+1$ hypersurfaces are sufficient. First, if the irreducible component is a hypersurface, then one polynomial is sufficient. Second, if the irreducible component of dimension i is linear (degree equal to one), then it is the intersection of $N-i$ hyperplanes.

It would be interesting to have a numerical approach to compute generators for the ideal $\mathcal{I}(Z_{ij})$ of polynomials vanishing on Z_{ij}. If Z_{ij} is smooth, the $N+1$ polynomials constructed above using $N+1$ general projections generate $\mathcal{I}(Z_{ij})$. This is a simple consequence of

1. the general fact that the differentials of $N+1$ polynomials constructed as above using $N+1$ general projections span the normal bundle of $\mathcal{I}(Z_{ij})$ at manifold points of Z_{ij}; and

2. the general fact that the higher cohomology of coherent algebraic sheaves on affine varieties vanishes.

Unfortunately, simple examples show that the minimum number of polynomials to generate $\mathcal{I}(Z_{ij})$ for Z_{ij} with singularities can grow unboundedly as the degree of Z_{ij} increases.

3 An Illustrative Example

To illustrate our procedure, we will consider the following system:

$$f = \begin{bmatrix} (y-x^2)(x^2+y^2+z^2-1)(x-0.5) \\ (z-x^3)(x^2+y^2+z^2-1)(y-0.5) \\ (y-x^2)(z-x^3)(x^2+y^2+z^2-1)(z-0.5) \end{bmatrix}. \tag{10}$$

This example has been constructed in a factored form so that it is easy to identify the decomposition of $Z = f^{-1}(\mathbf{0})$ into its irreducible solution components, as

$$Z = Z_2 \cup Z_1 \cup Z_0 = \{Z_{21}\} \cup \{Z_{11} \cup Z_{12} \cup Z_{13} \cup Z_{14}\} \cup \{Z_{01}\}$$

where

1. Z_{21} is the sphere $x^2+y^2+z^2-1=0$,

2. Z_{11} is the line $(x=0.5, z=0.5^3)$,

3. Z_{12} is the line $(x=\sqrt{0.5}, y=0.5)$,

4. Z_{13} is the line $(x=-\sqrt{0.5}, y=0.5)$,

5. Z_{14} is the twisted cubic $(y - x^2 = 0, z - x^3 = 0)$,

6. Z_{01} is the point $(x = 0.5, y = 0.5, z = 0.5)$.

In Fig. 1 we present the flow diagram for our algorithm **IrreducibleDecomposition** on this illustrative example. The left side of the diagram shows the operation of **WitnessGenerate** as it descends through successive embeddings to generate the witness point supersets $\widehat{W}_i$, $i = 2, 1, 0$, these being of size 2, 14, and 19, respectively. On the right-hand side, the operation of **WitnessClassify** proceeds as follows. At level 2, the filter is empty and both points pass on to W_2 to be classified. Samples found by continuation from the first point were found to be interpolated by a quadratic surface (the sphere) and the second point was found to also fall on the sphere. Thus, component Z_{21} is determined to be a second-degree variety of multiplicity one. The sphere equation is appended to the filter, and the algorithm proceeds to level 1. At level 1, 8 of the points in $\widehat{W}_1$ are found to lie on the sphere and are discarded as J_1. Using the sample and interpolate procedures, the remaining 6 are classified as falling on 3 lines and a cubic, each with multiplicity one. A filtering polynomial for each of these is appended to the filter and the algorithm proceeds to level 0. Here, 18 points in $\widehat{W}_0$ are found to lie on the higher-dimensional components, as shown in the diagram, leaving a single isolated root as W_{01}.

Note that the computation in **IrreducibleDecomposition** can be interleaved between **WitnessGenerate** and **WitnessClassify** to entirely complete the computation at each level before descending to the next. Except for minor issues of storage and retrieval of information, this is no different than running **WitnessGenerate** and **WitnessClassify** in sequence.

To summarize, the output of **IrreducibleDecomposition** is the witness point set W, with degrees d_{ij} and multiplicity bounds ν_{ij} as:

$$W = W_2 \cup W_1 \cup W_0 = \{W_{21}\} \cup \{W_{11} \cup W_{12} \cup W_{13} \cup W_{14}\} \cup \{W_{01}\}$$

where

1. W_{21} contains 2 points, $d_{21} = 2$, and $\nu_{21} = 1$,

2. W_{11} contains 1 point, $d_{11} = 1$, and $\nu_{11} = 1$,

3. W_{12} contains 1 point, $d_{12} = 1$, and $\nu_{12} = 1$,

4. W_{13} contains 1 point, $d_{13} = 1$, and $\nu_{13} = 1$,

5. W_{14} contains 3 points, $d_{14} = 3$, and $\nu_{14} = 1$,

6. W_{01} is a nonsingular solution point.

The execution summary of our implementation on this example is described in §7.2.

4 Algorithms

In this section we give precise statements of our algorithms using pseudocode. The main procedure is **IrreducibleDecomposition**, which consists of algorithms **WitnessGenerate**

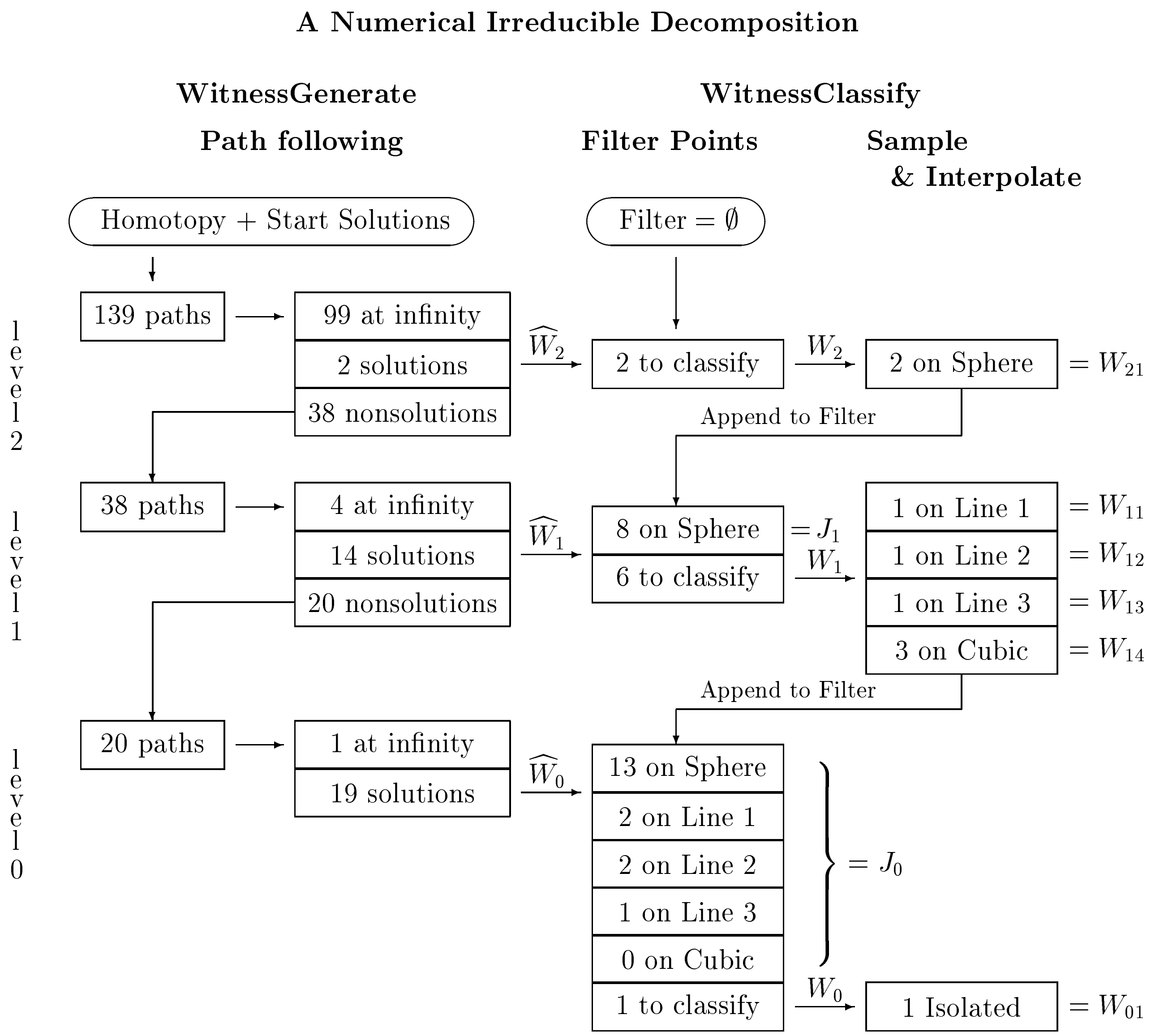

Figure 1: Flow Diagram of Illustrative Example.

and **WitnessClassify** applied sequentially. (It is also possible to interleave these two to complete the computation dimension-by-dimension, as can be seen from the example flowchart in Fig. 1.) **PolynomialsGenerate** is a post-processing step to generate equations for the components after they have been identified by **IrreducibleDecomposition**.

We lay out the algorithms in this section and then give the theoretical justification in the sequel.

4.1 Squaring Polynomial Systems

Our numerical algorithms for tracking homotopy paths currently require that the number of equations n and the number of unknowns N be equal. Yet, as indicated in (1), we wish to treat general problems in which this is not necessarily the case. In the algorithm below we apply the embedding of [38] to a "square" polynomial system, derived from the given system as follows.

Algorithm 4.1 $[g, m] = \mathbf{Square}(f, n, N)$.

Input: polynomial system $f(x_1, \ldots, x_N) = (f_1, \ldots, f_n)$, n equations in N unknowns.
Output: square polynomial system, $m = \max(n, N)$, $g(x_1, \ldots, x_m) = (g_1, \ldots, g_m)$.

if $n < N$ then [Underdetermined case.]
  $g_i := f_i$, for $i = 1, 2, \ldots, n$;
  $g_i := 0$, for $i = n + 1, \ldots, N$;
elseif $n > N$ then [Overdetermined case.]
  $g_i := f_i + \sum_{j=N+1}^{n} a_{ij} x_j$, for $i = 1, 2, \ldots, n$;
else [Square case.]
  $g_i := f_i$, for $i = 1, 2, \ldots, n$;
end if;

All constants a_{ij} in Algorithm 4.1 are randomly generated complex numbers.

For an underdetermined system, i.e., the case $n < N$, the system is made square by appending $N - n$ zero equations.

For an overdetermined system, i.e., the case $n > N$, we introduce $n - N$ auxiliary variables $x_{N+1}, \ldots, x_n$. Solutions of the square system g in which all the auxiliary variables vanish are also solutions to the original system f. All other solutions of g are spurious. Hence, the solution sets of f are extracted from those found for the system g by discarding such spurious sets and then dropping the auxiliary variables from the remaining solutions. These operations are done in procedure **UnSquare**.

Algorithm 4.2 $[W, S, F] = \mathbf{UnSquare}(W', S', F', n, N)$.

Input: Witness point set W', Sample points S', Filtering polynomials F'
from squared system, originally n equations in N unknowns.
Output: Witness point set W, Sample points S, and Filtering polynomials F
for the original system.

if $n \leq N$ then
  $[W, S, F] := [W', S', F']$;
elseif $n > N$ then
  $W := W'$; $S := S'$; $F := F'$;
  remove spurious W_{ij}, S_{ij} for $x_k \neq 0$, $k > N$;
  remove x_k, $k > N$ in W_{ij}, S_{ij}, and F;
end if;

Remark 4.3 For overdetermined systems, elimination of the auxiliary variables would give a smaller system of polynomials that are a random combination of the original ones, as proposed in [39].

Remark 4.4 For underdetermined systems, there are more efficient squaring procedures. Letting f be a system of n polynomials $f_1, \ldots, f_n$ on $\mathbb{C}^N$ with $n < N$, consider the system F obtained by appending one zero equation plus $N - n - 1$ random linear equations $L_i := b_i + \sum_{j=1}^{N} a_{ij} x_j$ for $i = n+1, \ldots, N-1$. There is a one-to-one correspondence between positive dimensional irreducible components of $f^{-1}(0)$ and positive dimensional irreducible components of $F^{-1}(0)$. Letting L be the generic linear space defined by $L_{n+1} = 0, \ldots, L_{N-1} = 0$, the correspondence is obtained by associating an irreducible component Z_{ij} of $f^{-1}(0)$ with the irreducible component $Z_{ij} \cap L$ of $F^{-1}(0)$. Under this correspondence, degrees and multiplicities are preserved, with the dimensions shifted down by $N - n - 1$. Generic points of $Z_{ij} \cap L$ are generic points of Z_{ij}.

A further improvement is to use a system G obtained from f by appending $N - n$ random linear equations, and then post-process the level zero witness points of G using the system F obtained by replacing one of the random linear equations of G with the zero equation.

4.2 Main Procedure: Irreducible Decomposition

Our "probability one" algorithm proceeds as follows.

Algorithm 4.5 $[W, D, M; S, F] =$ **IrreducibleDecomposition**(f, k)

Input: Polynomial system $f(x_1, \ldots, x_N) = (f_1, \ldots, f_n)$; top dimension k.
Output: Witness point set W; degrees D; multiplicity bounds M.
Optional Output: Sample points S; filtering polynomials F.

$[g, m] :=$ **Square**(f, n, N);
$\widehat{W} :=$ **WitnessGenerate**(g, k);
$[W', D, M; S', F'] :=$ **WitnessClassify**$(g, \widehat{W}, k)$;
$[W, S, F] :=$ **UnSquare**(W', S', F', n, N);

By default, the top dimension k is $N-1$, but for efficiency reasons, if it can be determined by some other means that there are no components of dimension greater than k, we start the algorithm with k less than $N-1$. The optional outputs are provided to avoid the need to recompute in the post-processing algorithm **PolynomialsGenerate**.

The two main subalgorithms are presented next. The first of these, **WitnessGenerate**, is the algorithm from [38], which we include only to define its specification. The algorithm finds generic slices of the solution set at each dimension i from $i = k$ down to $i = 0$. This is done using a cascade of homotopies between embedded systems.

Algorithm 4.6 $[\widehat{W}] =$ **WitnessGenerate**(f, k)

Input: Polynomial system $f(x_1, \ldots, x_N) = (f_1, \ldots, f_n)$; top dimension k.
Output: Witness point superset $\widehat{W}$.

We now come to the main contribution of this article, algorithm **WitnessClassify**, which descends through the dimensions to sort out the points in $\widehat{W}$.

Algorithm 4.7 $[W, D, M; S, F] =$ **WitnessClassify**$(f, \widehat{W}, k)$

Input: Polynomial system f; Witness point superset $\widehat{W}$; top dimension k.
Output: Witness point set W; degrees D; multiplicity bounds M;
Optional Output: Sample points S; filtering polynomials F.

$F = \emptyset$; [Initialize filter to empty.]
[Loop to process $\widehat{W}_i$, the witness point superset at dimension i.]
for $i = k$ down to 0 do
  $J_i := \{\mathbf{w} \in \widehat{W}_i \mid p(\mathbf{w}) \approx 0 \text{ for some } p \in F\}$;
  $\widehat{W}_i := \widehat{W}_i \setminus J_i$; [Remove "junk."]
  $j := 0$;
  [Each pass of the following loop finds a new irreducible component.]
  while $\widehat{W}_i \neq \emptyset$ do
    $j := j + 1$;
    choose $\mathbf{w} \in \widehat{W}_i$;
    $[p_{ij}, S_{ij}] :=$ **Interpolate**$(f, \mathbf{w}, i)$; [Get filtering polynomial & sample points.]
    $W_{ij} := \{\mathbf{w} \in \widehat{W}_i \mid p_{ij}(\mathbf{w}) \approx 0\}$; [Find witness points.]
    $D_{ij} := \deg p_{ij}$; [Degree of the component.]
    $M_{ij} := \#W_{ij}/D_{ij}$; [Multiplicity bound for the component.]
    $\widehat{W}_i := \widehat{W}_i \setminus W_{ij}$; [Remove points just classified.]
    $F := F \cup p_{ij}$; [Update the filter.]
  end while;
end for.

Two lines in this algorithm invoke an approximate test of equality $p(\mathbf{w}) \approx 0$. Ideally, we would wish to achieve exact equality, but in practice both the witness points $\mathbf{w}$ and the coefficients of the filtering polynomials p are subject to numerical error, so the tests must include some tolerance. The same remark applies in algorithm **Interpolate** below. The issue of setting numerical tolerances is discussed further in §7.1.

The next algorithm, Interpolate, makes use of a projection function $\pi(A, \mathbf{x})$:

$$\pi(A, \mathbf{x}) = \left(a_{10} + \sum_{j=1}^{N} a_{1j} x_j, \quad \ldots \quad , a_{i0} + \sum_{j=1}^{N} a_{ij} x_j \right), \tag{11}$$

where a_{ij} is the (i, j)th element of the matrix A. When applied to a set of points S, $\pi(A, S)$ is just the set of projections of the points. Matrix A with the column

$$\begin{bmatrix} a_{10} \\ \vdots \\ a_{i0} \end{bmatrix} \tag{12}$$

deleted will have fewer rows than columns, and the "direction of the projection" mentioned in §2 is just the linear subspace of projective space obtained by intersecting the closure of the null space of this truncated matrix with the hyperplane $\mathbb{P}^N \setminus \mathbb{C}^N$ of $\mathbb{P}^N$ at infinity. In the terminology used by algebraic geometers and appearing in §6, this intersection is the *center* of the projection.

Algorithm **Interpolate** finds a filtering polynomial by fitting a polynomial to the projection of a set of sample points. Letting q denote the defining equation in projected coordinates, which is unique up to nonconstant multiple, then the polynomial in the original coordinates is $p(\mathbf{x}) = q(\pi(A, \mathbf{x}))$. In our computer code, we do not expand this to express p directly as a polynomial in $\mathbf{x}$, but rather we keep it in the composite form.

Algorithm 4.8 $[p, S] =$ **Interpolate**$(f, \mathbf{x}, i)$

Input: Polynomial system f; Solution point $\mathbf{x}$; Working dimension i.
Output: Interpolating polynomial p; Sample points S.

Parameter: Oversampling numbers $k_1 \geq 0$, $k_2 \geq 1$ (integers).

```
A := Rand(i + 1, N + 1); [Generate random matrix A ∈ C^((i+1)×(N+1)).]
if i = 0 then    [Isolated points do not need sampling.]
    S := x;
    p(y) := π(A, y) − π(A, x);  [Random hyperplane through x.]
else             [For positive dimensional sets (i > 0), sample and fit.]
    S := ∅;
    d := 1;      [Start at degree 1.]
    loop
        m_s := #monom(d, i + 1) − 1 + k_1;        [Number of sample points needed.]
        S := S ∪ Sample(f, x, i, m_s − #S);       [Expand the sample set.]
```

$p := \mathbf{Fit}(d, i, \pi(A, S));$ [Fit degree d polynomial to projection.]
exit when $p(\pi(A, \mathbf{Sample}(f, \mathbf{x}, i, k_2))) \approx 0;$ [Test k_2 extra points.]
$d := d + 1;$ [If fit not good, increment d and loop.]
end loop;
end if.

The function **Sample** generates new generic points from the current one $\mathbf{x}$ by homotopy continuation that moves slicing hyperplanes, initially passing through $\mathbf{x}$. The hyperplanes that pick out the new sample points are chosen randomly to produce widely-dispersed, generic points on the irreducible component on which $\mathbf{x}$ sits. A polynomial p of degree d in $i+1$ variables is uniquely defined, up to scaling, by $m-1$ general points, where $m = \#\mathrm{monom}(d, i+1) = \binom{d+i+1}{i+1}$ is the maximum number of monomials in p. When we have enough points, the function **Fit** constructs an interpolating polynomial, using least squares if $k_1 > 0$. The exiting condition consists in a zero test (numerically up to a certain tolerance) on the evaluation of the interpolating polynomial in $k_2 \geq 1$ extra sampled points.

There is one important missing link in our description of the **Sample** algorithm. If the component to be sampled has multiplicity one, then the sample points will, after slicing, be nonsingular, and sampling comes down to standard path tracking. However, if the multiplicity is higher, the slicing still cuts out a unique point, but the point will be multiple, and hence, singular. Consequently, a more sophisticated path tracker will be necessary. One possibility is to use the secant method combined with a modified Newton corrector, see e.g. [41].

4.3 Post-processing: Generation of Equations

Once the work of **IrreducibleDecomposition** is finished, it is a simple matter to construct equations for each irreducible component. As justified in Lemma 5.8, we need only to interpolate the component with $N+1$ generically projected hypersurfaces. A by-product of **IrreducibleDecomposition** is set S_{ij} of sample points on each irreducible component Z_{ij}, which is large enough to uniquely determine a hypersurface once a projection has been fixed. Also, **IrreducibleDecomposition** has already found one such polynomial for each component for use in the filtering process. Thus, procedure **PolynomialsGenerate** is easily built from existing functions: the projection function π and the fitting routine **Fit**, as follows.

Algorithm 4.9 $[I] = \mathbf{PolynomialsGenerate}(D, S, F)$

Input: Degrees D; Sample sets S; Filtering polynomials F.
Output: Radical ideal I.

for each irreducible component (i, j) do
$I_{ij} := F_{ij}$; [Copy the known filtering polynomial from F].
[Determine the number of hypersurfaces needed.]
if $i = N - 1$ then [Hypersurface]
$m := 1;$
elseif $D_{ij} = 1$ then [Linear component]
$m := N - i;$

else [Default]
$m := N + 1$;
end if
[Loop to construct the Hypersurfaces]
while $\#(I_{ij}) < m$ do
$A := \text{Rand}(i + 1, N + 1)$; [New random projection]
$I_{ij} := I_{ij} \cup \mathbf{Fit}(D_{ij}, i, \pi(A, S_{ij}))$;
end while;
end for.

The if-block is included as a matter of efficiency. In general, $N + 1$ hypersurfaces are sufficient, but only one is needed if the irreducible component is itself a hypersurface and $N - i$ suffices if the component is linear.

5 Background II: Generic Projections and Related Material

Using homogeneous polynomials and projective space in place of polynomials and $\mathbb{C}^N$, we arrive at the concept of a *projective variety* as the common zeroes of a set of homogeneous polynomials. A projective variety minus a proper projective subvariety is called a *quasiprojective variety*, or *variety* for short. Though affine varieties and projective varieties are the most important types of quasiprojective varieties, there are others, e.g., $\mathbb{C}^2 \setminus \{(0, 0)\}$.

5.1 The Complex Topology, the Zariski Topology, and Constructible Sets

Unless we say otherwise, closures will be in the complex topology, i.e., in the topology induced on a subset of complex Euclidean space (or projective space) by the underlying manifold topology of complex Euclidean space (or projective space). The key fact connecting the complex topology and the Zariski topology, is that, given a subvariety X of a variety Y, the closure $\overline{X}$ of X in Y is a subvariety of Y, and X is Zariski open and dense in $\overline{X}$, i.e., $\overline{X} \setminus X$ is a proper subvariety of $\overline{X}$ not containing any nonempty Zariski open set of $\overline{X}$. Usually, we use this fact when X is a variety in $\mathbb{C}^N \subset \mathbb{P}^N$ and we close up X in $\mathbb{P}^N$. Mumford [35, Chap. 1.10] is a good place to learn about these results. For historical reasons, connected to the many different approaches to algebraic geometry, and the progressively more general developments of the subject, the names and definitions of the most basic words like varieties and quasiprojective varieties differ from place to place. In some places what we call varieties would be called reduced quasiprojective varieties. Also, though we use variety in a standard way, it is used, in a few references, to refer to what we call an irreducible variety. So the reader needs always to check the basic vocabulary being used.

The *dimension* of a variety is the complex dimension of the regular points, i.e., manifold points, of the variety.

Often we have an irreducible variety X and we need to know that a given property holds for a Zariski open and dense set $U \subset X$. Knowing this will let us know that, with probability one, the property holds for a generic point of X: for a discussion of generic points and related matters see [38, 39]. Often, it is easier to work with an irreducible variety Y that maps onto a dense set of X, e.g., in the discussion of projections in the subsection §5.2, it is convenient to study a general projection as a succession of projections with one dimensional kernels. In such a situation we find a dense Zariski open set of Y with the property we want, and then need to produce a dense Zariski open set of X. Though, in most given situations, it is straightforward to find such a Zariski open set of X given the Zariski open set of Y, it can be tedious and it is usually unnecessary. This is because of the simple, but very useful theorem of Chevalley on constructible sets. The constructible sets on a variety X are the subsets of X constructed from subvarieties by an arbitrary, but finite number, of complementations and unions. Thus $\{(0,0)\} \cup \{\mathbb{C}^2 - \{(0,0)\} \times \mathbb{C}\}$ is a constructible subset of $\mathbb{C}^2$. The key facts about constructible sets are given in the following Theorem, see [35, Chap. 1.9] and [34, Chap. 2C] for a proof and detailed discussion of the results in this subsection.

Theorem 5.1 (Chevalley's Theorem) *Let $p : Y \to X$ be an algebraic map from an variety Y to an variety X. The set $p(Y)$ is constructible on X. Further, if Y is irreducible and if the Zariski closure of $p(Y)$ equals X, then there is a Zariski open dense set U of X contained in $p(Y)$.*

Remark 5.2 Note that in this result the hypothesis "if the Zariski closure of $p(Y)$ equals X" is the same as the hypothesis "if the closure of $p(Y)$ in the complex topology equals X." Indeed, since in Theorem 5.1, $U \subset p(Y) \subset X$, we conclude from §5.1 that the closure of $p(Y)$ in the complex topology contains $\overline{U} = X$, and thus equals X. On the other hand if we assumed that the $\overline{p(Y)} = X$ in the complex topology, then since the complex topology has at least as many open sets as the Zariski topology, it follows that the closure of $p(Y)$ in the Zariski topology is X.

5.2 Projections and Generic Projections

"Generic" projections have many very useful properties. They have been used since classical times to reduce questions about general varieties to questions about hypersurfaces. They have also been used to produce equations vanishing on a given variety, e.g., [33] for details on this; and also to discover and prove results about the projective invariants of varieties, e.g., [6]. Many of these results hold in some form even for differentiable, not necessarily algebraic manifolds, and can be used to build fast probability one algorithms in such situations, e.g., see, [3] for such an algorithm. In this subsection, we give proofs for the facts we need in this article, but for which, we know no easily accessible reference.

An algebraic map $f : Y \to X$, between algebraic varieties Y and X, is said to be proper if given any compact subset $K \subset X$, $f^{-1}(K)$ is compact. It is an easy check that a proper map is closed. A key fact about proper mappings is the so called "proper mapping theorem", which states that the image of a closed subvariety under a proper map is a closed variety.

By a *linear projection* (or *projection* for short) $\pi : \mathbb{C}^N \to \mathbb{C}^m$ we mean a surjective affine map

$$\pi(x_1, \ldots, x_N) = (L_1(\mathbf{x}), \ldots, L_m(\mathbf{x})), \tag{13}$$

where for each i we have $L_i(\mathbf{x}) := a_{i0} + \sum_{j=1}^{N} a_{ij}x_j$, with the a_{ij} constants. From the theoretical point of view of view, it is enough to work with equivalence classes of projections, where we consider two projections π_1, π_2 from $\mathbb{C}^N$ onto $\mathbb{C}^m$ equivalent if there is an affine linear isomorphism $T : \mathbb{C}^m \to \mathbb{C}^m$ with $T(\pi_1(\mathbf{x})) = \pi_2(\mathbf{x})$. Of course, numerically, when it comes to implementation, the choice of which projection to use is important in scaling considerations.

We need to consider the extension of projections to projective space. Let $[x_0, \ldots, x_N]$ denote linear coordinates on $\mathbb{P}^N$. Abusing notation, we regard $\mathbb{C}^N \subset \mathbb{P}^N$ using the inclusion $(x_1, \ldots, x_N) \to (1, x_1, \ldots, x_N)$. Thus $\mathbb{C}^N = \mathbb{P}^N \setminus H$, where $H := \{x_0 = 0\}$ is the hyperplane at infinity. By a *linear projection* (or *projection* for short) from $\mathbb{P}^N$ to $\mathbb{P}^m$ we mean a surjective map $\pi_L : \mathbb{P}^N \setminus L \to \mathbb{P}^m$

$$\pi([x_0, \ldots, x_N]) = [L_0(\mathbf{x}), \ldots, L_m(\mathbf{x})], \tag{14}$$

where for each i we have $L_i(\mathbf{x}) := \sum_{j=0}^{N} a_{ij}x_j$, with the a_{ij} constants; and where L is the linear $\mathbb{P}^{N-m-1} \subset \mathbb{P}^N$ defined by the vanishing of the linear equations L_i. L is said to be the *center* of the projection. From the theoretical point of view of view, it is enough to work with equivalence classes of projections, where we consider two projections π_1, π_2 from $\mathbb{P}^N$ onto $\mathbb{P}^m$ equivalent if they have a common center L, and there is a projective linear isomorphism $T : \mathbb{P}^m \to \mathbb{P}^m$ with $T(\pi_1(\mathbf{x})) = \pi_2(\mathbf{x})$ on $\mathbb{P}^N - L$. Note that, in fact, two projections from $\mathbb{P}^N$ to $\mathbb{P}^m$ are equivalent if and only if they have the same center. Geometrically, the projection has a simple description. Let L be the center of the projection. Choose any $\mathbb{P}^m \subset \mathbb{P}^N$ with the property that $L \cap \mathbb{P}^m = \emptyset$. Given a point $\mathbf{x} \in \mathbb{P}^N \setminus L$ and letting $< \mathbf{x}, L >$ denote the linear subspace $\mathbb{P}^{N-m} \subset \mathbb{P}^N$ generated by $\mathbf{x}$ and L, the projection from $\mathbb{P}^N$ to $\mathbb{P}^m$ with center L sends $\mathbf{x}$ to $< \mathbf{x}, L > \cap \mathbb{P}^m$.

The projections π_L from $\mathbb{P}^N$ to $\mathbb{P}^m$ that are extensions of projections from $\mathbb{C}^N$ to $\mathbb{C}^m$ are precisely the projections with center $L \subset H$. For example, the projection $(x_1, \ldots, x_N) \to (x_1, \ldots, x_{N-1})$ extends to the projection $[x_0, \ldots, x_N] \to [x_0, x_1, \ldots, x_{N-1}]$ with center $L := \{[0, \ldots, 0, 1]\}$.

Note that an equivalence class of projections is naturally identified with the center of the projection in the projective case, and the center of the projection of the projective extension in the affine case. Thus the projections from $\mathbb{C}^N \to \mathbb{C}^m$ are parameterized by the Grassmannian $\mathrm{Grass}(N - m, N + 1)$ of linear $\mathbb{P}^{N-m-1}$s in $\mathbb{P}^N$.

Let $X \subset \mathbb{C}^N$ be an algebraic subvariety. Let $\pi : \mathbb{C}^N \to \mathbb{C}^m$ be a linear projection. The map p_X does not have to be proper. For example, the map from the hyperbola $\{x_1x_2 - 1 = 0\}$ to the x_1 axis has image the complement of the origin, and is therefore not proper. It is a part of the Noether Normalization Lemma [19], that if π is general, then π_X is in fact proper. We give an easy direct argument for this, because it is useful to have an explicit understanding of what genericity is needed to obtain this property.

Lemma 5.3 *Let X be a closed k-dimensional subvariety of $\mathbb{C}^N$. The linear projection $\pi : \mathbb{C}^N \to \mathbb{C}^m$ with $m \geq k$, the map π_X is proper if the center L of the associated projection from*

$\mathbb{P}^N$ to $\mathbb{P}^m$ is disjoint from $\overline{X} - X$. In particular, for a general, linear projection $\pi : \mathbb{C}^N \to \mathbb{C}^m$ with $m \geq k$, the map π_X is proper.

Proof. For simplicity of exposition, we show this with $m = N - 1 \geq k$. We can assume without loss of generality that X is irreducible. Let $\overline{X}$ denote the closure of X in $\mathbb{P}^N$ in the complex topology. Note that $\overline{X}$ is a k-dimensional subvariety of $\mathbb{P}^N$, and that X is a Zariski open set of $\overline{X}$. As noted above projections of $\mathbb{C}^N$ to $\mathbb{C}^{N-1}$ are the restrictions to $\mathbb{C}^N$ of the projections $\mathbb{P}^N \to \mathbb{P}^{N-1}$ of the projections $\pi_{\mathbf{x}}$ from points $\mathbf{x}$ on the hyperplane at infinity, $H := \mathbb{P}^N \setminus \mathbb{C}^N$.

Note that $\pi_{\mathbf{x}|X}$ is proper if $\mathbf{x} \notin \overline{X} - X$. Indeed, given a compact set $K \subset \pi_{\mathbf{x}}(X)$, $\pi_{\mathbf{x}|X}^{-1}(K) = \overline{\pi_{\mathbf{x}}^{-1}(K)} \cap X$. Noting that, $\overline{\pi_{\mathbf{x}}^{-1}(K)} \cap \overline{X} \subset \mathbb{P}^N$ is compact, and that if $\mathbf{x} \notin \overline{X} - X$, then $\overline{\pi_{\mathbf{x}}^{-1}(K)} \cap H = \{\mathbf{x}\}$, we see that if $\mathbf{x} \notin \overline{X} - X$, then $\overline{\pi_{\mathbf{x}}^{-1}(K)} \cap \overline{X} = \overline{\pi_{\mathbf{x}}^{-1}(K)} \cap X$ is compact. □

Remark 5.4 Note that it can happen that the center of the projection meets $\overline{X} - X$, but the restriction of the projection to X is still proper. A simple example is obtained by taking the zero set C on $\mathbb{P}^2$ of $x_1^2 - x_0 x_2$. As usual, let the hyperplane at infinity be defined by $H := \{x_0 = 0\}$, and let $\mathbb{C}^2 := \mathbb{P}^2 \setminus H$. Let $X := C \setminus C \cap H$. The restriction to X of the projection from $[0, 0, 1]$ is an isomorphism onto $\mathbb{C}$. Note, that in this special case, the restriction of every projection of $\mathbb{C}^2$ to $\mathbb{C}$ is proper.

Lemma 5.5 *Let X be a closed subvariety of $\mathbb{C}^N$, all of whose irreducible components are of dimension k. For a general linear projection $\pi : \mathbb{C}^N \to \mathbb{C}^m$ with $m \geq k+1$, the map π_X is proper and generically one-to-one. In particular $\pi(X)$ is a closed subvariety of $\mathbb{C}^m$ of degree equal to the degree of X in $\mathbb{C}^N$.*

Proof. It can be assumed without loss of generality that X is irreducible. The properness has already been shown above by Lemma 5.3. By induction it suffices to show the lemma when $m = N - 1 \geq k + 1$. To make this induction rigorous we need to use Lemma 5.1 to conclude that the generic choices at each stage give a generic choice at the end.

Let $\overline{X}$ denote the closure of X in $\mathbb{P}^N$ in the complex topology. $\overline{X}$ is a k-dimensional subvariety of $\mathbb{P}^N$, and X is a Zariski open set of $\overline{X}$. As noted above projections of $\mathbb{C}^N$ to $\mathbb{C}^{N-1}$ are the restrictions to $\mathbb{C}^N$ of the projections $\mathbb{P}^N \to \mathbb{P}^{N-1}$ of the projections $\pi_{\mathbf{x}}$ from points $\mathbf{x}$ on the hyperplane at infinity, $H := \mathbb{P}^N \setminus \mathbb{C}^N$. Let Δ denote the diagonal of $X \times X$. Note that we have an algebraic map $\phi : X \times X \setminus \Delta \to H$, obtained by sending $(\mathbf{x}, \mathbf{y})$ to the intersection of H with the unique projective line through $\mathbf{x}$ and $\mathbf{y}$.

If none of the projections $\pi_{\mathbf{x}}$ with $\mathbf{x} \notin H \cap \overline{X}$ are generically one-to-one, then we conclude that the fiber of ϕ over each point of $H \setminus (\overline{X} \setminus X)$, is at least k-dimensional. Thus we conclude that

$$\begin{aligned} 2k &= \dim X \times X = \dim X \times X \setminus \Delta \\ &\geq k + \dim \phi(X \times X \setminus \Delta) \geq k + \dim \big(H \setminus (\overline{X} \setminus X)\big) \\ &= k + N - 1, \end{aligned} \tag{15}$$

i.e., that $k \geq N-1$. Since we are assuming that $k < N-1$, we see that the generic fiber dimension of the map $\phi : X \times X \to H$ is less than k, in which case there is a Zariski open subset of $H \setminus (\overline{X} \setminus X)$ with projections that are generically one-to-one.

The fact that the degree of X is the same as the degree of $\pi(X)$ is seen as follows. Note that since $\deg X = \deg \overline{X}$, $\deg \pi(X) = \deg \overline{\pi(X)}$, and $\pi(\overline{X}) = \overline{\pi(X)}$, it suffices to prove the result for a generically one-to-one projection in projective space. Let V denote the proper subvariety of $\pi(X)$ equal to the union of the singular points of $\overline{\pi(X)}$ and the image under π of the ramification locus of π. A generic linear $L := \mathbb{P}^{m-k}$ meets $\overline{\pi(X)}$ transversely in $\deg \overline{\pi(X)}$ points contained in $\pi(\overline{X}) \setminus V$. By construction, the $(N-k)$-dimensional linear space $\pi^{-1}(L)$ meets $\overline{X}$ transversely in $\deg \overline{X}$ points contained in the regular points of $\overline{X}$. $\square$

We need some more refined statements that guarantee we can separate points by using different projections.

Lemma 5.6 *Let X be a closed subvariety of $\mathbb{C}^N$, all of whose irreducible components are of dimension k. Fix a finite set $S \subset \mathbb{C}^N$. For a general linear projection $\pi : \mathbb{C}^N \to \mathbb{C}^m$ with $m \geq k+1$, $\pi(\mathbf{x}) = \pi(\mathbf{y})$ for $\mathbf{x} \in S$ and $\mathbf{y} \in X \cup S$ implies that $\mathbf{x} = \mathbf{y}$.*

Proof. As in Lemma 5.5, we can assume by induction that $m = N-1$. Let $H := \mathbb{P}^{N-1}$ denote the hyperplane at infinity in $\mathbb{P}^N$. Let $\mathbf{y}$ be a point of S. If $\mathbf{y} \notin X$ consider the map $\phi_{\mathbf{y}} : X \to H$ given by sending $\mathbf{x} \in X$ to the point $\phi_{\mathbf{y}}(\mathbf{x})$ equal to the intersection of H with the line spanned by $\mathbf{x}$ and $\mathbf{y}$. If $\mathbf{y} \in X$, let $\phi_{\mathbf{y}} : X \setminus \{\mathbf{y}\} \to H$ be the analogous map. The union $\mathcal{T}$ of the closures of the images of these maps as $\mathbf{y}$ runs over the set S is at most $\dim X$. Since $\dim X = k \leq N-2 < \dim H$, we conclude that the projection corresponding to a general point of $H \setminus \mathcal{T}$ has the desired properties. $\square$

The following simple Lemma is useful, see [34, page 7].

Lemma 5.7 *Let $A \subset \mathbb{C}^N$ be a reduced affine subvariety all of whose irreducible components are of dimension $N-1$. Then, there is a polynomial q in $\mathbb{C}^N$ whose zero set is exactly A.*

Given a reduced variety X, the following classical lemma will let us construct polynomials whose set of common zeroes is exactly the underlying set of X.

Lemma 5.8 *Let X be a subvariety of $\mathbb{C}^N$, all of whose irreducible components are of dimension $k < N$. Given $N+1$ generic projections $\pi_i : \mathbb{C}^N \to \mathbb{C}^{k+1}$ with q_i the defining $\deg X$ polynomial of $\pi_i(X)$ for $i = 0, \ldots, N$; the set of common zeroes of the polynomials $q_0(\pi_0(\mathbf{x})), \ldots, q_N(\pi_N(\mathbf{x}))$ is X.*

Proof. Choose a generic projection $\pi_N : \mathbb{C}^N \to \mathbb{C}^{k+1}$ and let q_N the defining $\deg X$ polynomial of $\pi_N(X)$. Then $q_N(\pi_N(\mathbf{x}))$ vanishes on an $(N-1)$-dimensional set X_N containing X. Let S be a finite set consisting of one point from each irreducible component of $X_N \setminus X$. Choose a generic projection $\pi_{N-1} : \mathbb{C}^N \to \mathbb{C}^{k+1}$. By Lemma 5.6, $\pi_{N-1}(S) \cap \pi_{N-1}(X) = \emptyset$, and thus the common zeroes X_{N-1} of $q_N(\pi_N(\mathbf{x})), q_{N-1}(\pi_{N-1}(\mathbf{x}))$ minus X is of dimension at most $N-2$. This step can be repeated, in an obvious induction, to give the conclusion of the lemma. $\square$

We leave the proof of the next statement to the reader. Details can be found in [33].

Lemma 5.9 *Let X be a closed subvariety of $\mathbb{C}^N$, all of whose irreducible components are of dimension k. Fix a point $\mathbf{x} \in X_{\text{reg}}$. For a general linear projection $\pi : \mathbb{C}^N \to \mathbb{C}^m$ with $m \geq k+1$, π gives a isomorphism of a Zariski open set $U \subset X$ containing $\mathbf{x}$ with $\pi(U)$.*

5.3 Multiplicities

We take a pedestrian view of multiplicities in this section, and refer the reader to Fulton [23] for a thorough description.

If we have a system f of n polynomial equations $f_1, \ldots, f_n \in \mathbb{C}[x_1, \ldots, x_N]$ and $\mathbf{x}^*$ is an isolated zero of the system, then the multiplicity of $\mathbf{x}^*$ is

$$\operatorname{mult}(f, \mathbf{x}^*) := \dim_{\mathbb{C}} \mathcal{O}_{\mathbb{C}^N|\mathbf{x}^*} / << f_1, \ldots, f_n >>, \tag{16}$$

where

1. $\mathcal{O}_{\mathbb{C}^N|\mathbf{x}^*}$ denotes the ring of convergent power series at $\mathbf{x}^* \in \mathbb{C}^N$, or equivalently, the ring of germs of holomorphic functions at $\mathbf{x}^* \in \mathbb{C}^N$; and
2. $<< f_1, \ldots, f_n >>$ denotes the ideal generated by the f_i in $\mathcal{O}_{\mathbb{C}^N|\mathbf{x}^*}$.

The following is a simple observation.

Lemma 5.10 *Given a system f of n polynomial equation $f_1, \ldots, f_n \in \mathbb{C}[x_1, \ldots, x_N]$ and an isolated zero $\mathbf{x}^*$ of the system, and a matrix*

$$\begin{bmatrix} \lambda_{11} & \cdots & \lambda_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{N1} & \cdots & \lambda_{Nn} \end{bmatrix} \in \mathbb{C}^{N \times n} \tag{17}$$

such that the system g of polynomial equations $g_1 := \lambda_{11} f_1 + \cdots + \lambda_{1n} f_n, \ldots, g_N := \lambda_{N1} f_1 + \cdots + \lambda_{Nn} f_n$ also has $\mathbf{x}^*$ as an isolated zero, it follows that $\operatorname{mult}(f, \mathbf{x}^*) \leq \operatorname{mult}(g, \mathbf{x}^*)$.

Proof. Simply note that $<< g_1, \ldots, g_N >> \subset << f_1, \ldots, f_n >>$. □

In the procedures of [38, 39], a polynomial system is replaced with a new system made out of linear combinations of the equations of the systems in such a way that an isolated solution $\mathbf{x}^*$ of the original system is still isolated in the new system. The above lemma shows that the multiplicity of the isolated solution of the new system is at least as great as the isolated solutions of the original system. This is discussed briefly in [39] with an example showing that the multiplicity can increase.

Let W be a k-dimensional irreducible component of the affine variety $\mathbf{V}(f)$ of a system f of polynomial equations $f_1, \ldots, f_n \in \mathbb{C}[x_1, \ldots, x_N]$. Rather than give the definition of $\operatorname{mult}(f, W)$, the multiplicity of W with respect to the system f, we note that it follows from [23, Chap. 10] that given a generic point $\mathbf{w} \in W$, and k linear equations,

$L_1, \ldots, L_k$ which define a linear $\mathbb{C}^{N-k}$ passing through $\mathbf{w}$ and transverse to W at $\mathbf{w}$, then $\mathrm{mult}(f, W) = \mathrm{mult}(f, L_1, \ldots, L_k, \mathbf{w})$. Thus, in particular, given generic linear equations $L_1, \ldots, L_k$, $\mathrm{mult}(f, L_1, \ldots, L_k, \mathbf{w})$ is the same value for all the points $\{ \mathbf{w} \in W \mid L_i(\mathbf{w}) = 0 \text{ for all } i \}$.

In the procedures of [38, 39], to study a dimension k component W of the zero set of a given system f consisting of equations $f_1, \ldots, f_n$ on $\mathbb{C}^N$, f is replaced with a new system g of polynomials $g_1, \ldots, g_{N-k}$ of random linear combinations of the f_i. Then the system g is augmented with k general linear equations $L_1, \ldots, L_k$. The combination of Lemma 5.10 and the last paragraph, guarantee that

1. $\mathrm{mult}(f, W) \leq \mathrm{mult}(g, W)$; and
2. $\mathrm{mult}(g, W) = \mathrm{mult}(g, L_1, \ldots, L_k, \mathbf{w})$ for any solution $\mathbf{w} \in W$ of all the L_i.

The importance of this is that, in the homotopy continuation process to solve the system $g, L_1, \ldots, L_k$, the quantity $\mathrm{mult}(g, L_1, \ldots, L_k, \mathbf{w})$ is the number of paths that end up at $\mathbf{w}$. Thus, a bound for $\mathrm{mult}(f, W)$ is computed as a byproduct of our computations. Since the bound is the same for all the generic points of W that we compute, we use it as a check on our breakup of the witness points into the subsets corresponding to given components.

6 Theoretical Justification

Let f be a system of polynomials $f_1, \ldots, f_n$ on $\mathbb{C}^N$, with not all the f_i equal to the zero polynomial. Let

$$Z := \bigcup_{i=0}^{N-1} Z_i := \bigcup_{i=0}^{N-1} \bigcup_{j \in \mathcal{I}_i} Z_{ij} \tag{18}$$

denote the decomposition of $\mathbf{V}(f)$ into irreducible components with $\dim Z_{ij} = i$. As the output of **WitnessGenerate** we get a finite set of points $\widehat{W} := \sum_{i=0}^{N-1} \widehat{W}_i$, where each $\widehat{W}_i$ has a disjoint decomposition $\bigcup_{j \in \mathcal{I}_i} W_{ij} + J_i$ (we know this breakup exists, and it is the purpose of **WitnessClassify** is to describe it explicitly) with

1. W_{ij} consists of $\deg Z_{ij}$ generic points of Z_{ij}; and
2. $J_i \subset \bigcup_{j>i} Z_j$.

Moreover, as shown in [38, 39], for each $\mathbf{w} \in W_{ij}$, **WitnessGenerate** has a number of paths $\nu_{ij} \geq \mathrm{mult}(f, Z_{ij})$ ending at $\mathbf{w}$. The algorithms do an explicit breakup $\widehat{W}_i$ into the disjoint subsets W_{ij} and J_i. In this section we show why the algorithms work.

If $\dim Z = k$, then J_k is empty. If we move the linear space defining a generic point $\mathbf{x}$ of some Z_{ij}, we can trace out by continuation as large a finite set S_{ij} as we want of points of the same component Z_{ij}, which can also be assumed generic. Take a generic projection $\pi : \mathbb{C}^N \to \mathbb{C}^{k+1}$. By the generic projection theorems of §5, we can assume that π is

1. one-to-one on the set $\widehat{W} \cup \bigcup_{j \in \mathcal{I}_k} S_{kj}$;
2. proper on each Z_{kj};
3. gives an isomorphism of a dense Zariski open set of Z_{kj} with its image; and
4. given $\mathbf{x} \in \widehat{W} \cup \bigcup_{j \in \mathcal{I}_k} S_{kj}$, $\pi(\mathbf{x}) \in Z_{kj}$ if and only if $\mathbf{x} \in Z_{kj}$.

Now fix a component Z_{kj}, which we can assume after renaming to be Z_{k1}. Given a large enough finite set of points S_{k1}, we can find the lowest degree polynomial p_{k1} on $\mathbb{C}^{k+1}$ vanishing on S_{k1}, and know that it will vanish on Z_{k1} and be of degree $\deg Z_{k1}$. The exact details of carrying this out efficiently is a numerical interpolation problem. Using the composition $p_{k1}(\pi(\mathbf{x}))$ of the projection π with the polynomial p_{k1} we can pick out the points W_{k1} as the zeroes of $p_{k1}(\pi(\mathbf{x}))$ on W_k, and thus split W_{k1} off from W_k. We can proceed until we have completely broken up W_k as desired. Now a point $\mathbf{x} \in \widehat{W}_{k-1}$ is in J_{k-1} if and only if it belongs to one of the Z_{kj}. Since $\pi(\mathbf{x}) \in Z_{kj}$ for $\mathbf{x} \in \widehat{W} \cup \bigcup_{j \in \mathcal{I}_k} S_{kj}$ if and only if $\mathbf{x} \in Z_{kj}$, we see that J_{k-1} is precisely the set of points of $\widehat{W}_{k-1}$ for which $p_{kj}(\pi(\mathbf{x})) = 0$ for some j. Thus we have computed J_{k-1}, and therefore also W_{k-1}. Repeating the above argument successively for i from $k-1$ to 0 gives the desired decomposition, i.e., the output of **WitnessClassify**.

Remark 6.1 We would like to mention a modification of **WitnessClassify**, that will likely yield some significant speedup. We have used many times the fact that if $X \subset \mathbb{C}^N$ is an irreducible affine variety of dimension $k > 1$, then $X \cap L \subset L$ is irreducible of the same degree for any generic linear subspace $L \subset \mathbb{C}^N$ of dimension $\geq N - k + 1$. Thus for $i > 1$, if we slice Z_{ij} with a generic linear space L of dimension $N - i + 1$ and project $Z_{ij} \cap L$ to a generic $\mathbb{C}^2$ before the construction of a given polynomial p_{ij}, we can ascertain the degree of the polynomial p_{ij} by interpolation in $\mathbb{C}^2$ instead of $\mathbb{C}^{i+1}$.

We have not implemented this improvement in this article, because it leads quickly to many numerical questions connected with least squares and the inductive construction of the interpolation polynomials p_{ij}. These important questions would obscure the main lines of this article, and so are being dealt with in a sequel on efficient numerical implementation.

7 Computational Experiments

In this section we first comment on the numerical aspects of constructing the interpolating polynomials and then discuss several numerical experiments using the new algorithm. Some problems require the use of multi-precision arithmetic, especially when dealing with curves of high degree, as is the case of the last two test examples.

7.1 Numerics of Fitting

We denote the interpolating polynomial $p(\mathbf{x})$ of degree d in n variables using a multi-index notation as

$$p(\mathbf{x}) = \sum_{|\mathbf{a}|\leq d} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}}, \quad \text{with} \quad \mathbf{x}^{\mathbf{a}} = x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n} \quad \text{and} \quad |\mathbf{a}| = a_1 + a_2 + \cdots + a_n. \tag{19}$$

In this dense representation, the number of monomials m of degree $\leq d$ in n variables is:

$$m = \frac{(n+d)!}{n!d!}. \tag{20}$$

To determine the coefficients $c_{\mathbf{a}}$ we need at least $m-1$ generic points $\mathbf{x}_i$, $i = 1, 2, \ldots, s$, $s \geq m-1$. The interpolation conditions $p(\mathbf{x}_i) = 0$ constitute a linear system, say $\mathbf{Y}\mathbf{c} = 0$, where $\mathbf{c}$ is the $m \times 1$ column of coefficients and $\mathbf{Y}$ is an $s \times m$ matrix composed of the monomials $\mathbf{x}_i^{\mathbf{a}}$ evaluated at the generic points $\mathbf{x}_i$. Since the scale of $\mathbf{c}$ is undetermined, we may add one extra condition, such as a generic inhomogeneous linear equation, to make the solution unique.

In our implementation we solve the linear system directly from the interpolation conditions. This can be done by least squares using QR decomposition of the matrix $\mathbf{Y}$.

The accuracy of the fit depends on two factors: the accuracy of the sample points $\mathbf{x}_i$ and the conditioning of the matrix $\mathbf{Y}$. To improve the conditioning of $\mathbf{Y}$, it is generally helpful to disperse the sample points widely. Points close together will lead to rows of $\mathbf{Y}$ that are nearly equal. The phenomenon is also clear from considering the problem of fitting a line through two points in the plane. For a given level of absolute accuracy in the points, the error in the slope of the line diminishes proportionately to the distance between the points. For the nonlinear case, the numerical issues are thornier, because for higher degrees, the entries in $\mathbf{Y}$ tend to blow up (or vanish) when the magnitude of the sample points is large (or small).

To get a sense of the problem we conducted the following numerical experiment. We wish to reconstruct the polynomial

$$f(x,y) = 1 + \sum_{1\leq i+j\leq d} x^i y^j \tag{21}$$

from sampled points (x_k, y_k) satisfying $f(x_k, y_k) = 0$, for $k = 1, 2, \ldots$, #terms-1. The samples were generated by choosing x_k randomly in a uniform distribution from the unit circle in the complex plane, and then solving for y_k by Newton's method. We fixed $c_0 = 1$, formed the matrix $\mathbf{Y}$, and solved $\mathbf{Y}\mathbf{c} = 0$ by classical LU decomposition, with 32-digit multi-precision arithmetic. Since the correct values of the coefficients are known from the outset (all $c_{\mathbf{a}} = 1$), we can evaluate the error in the computed values. The experiment was repeated for degrees from 2 to 12. The results, shown in Table 1, indicate a steady worsening of the conditioning as the degree increases, with a corresponding increase in the error of the computed coefficients. One sees that for degrees greater than 4, the conditioning is such that double precision calculations will give coefficients only accurate to 10^{-5} or worse. It is clear that multiple precision would be needed for higher degrees. Our other examples below show similar behavior.

deg	#terms	rcond	error
2	6	8.749E-06	3.727E-28
3	10	4.859E-08	7.183E-26
4	15	1.041E-06	8.514E-28
5	21	4.724E-10	1.042E-24
6	28	4.678E-11	9.006E-24
7	36	1.557E-10	3.401E-24
8	45	6.680E-14	4.376E-21
9	55	1.627E-14	1.808E-20
10	66	7.423E-13	3.299E-22
11	78	9.158E-18	2.486E-17
12	91	5.228E-18	2.834E-17

Table 1: The precision for these experiments is 10^{-32}. Column "rcond" is the estimate for the inverse of the condition number of the linear system. Column "error" is the largest imaginary part in the coefficients $c_{\mathbf{a}}$.

7.2 Test Problems

The algorithms have been implemented as an extra driver to PHC [40]. Reported timings concern a Pentium II 400 Mhz running Linux. No specialized methods were implemented to speed up the computations in **WitnessGenerate**. In particular, the multi-precision arithmetic, that was needed in **WitnessClassify** to refine the witness points and construct the interpolating polynomials, is done at a very elementary level. The first two examples were done with standard machine arithmetic, but the final two systems required multi-precision facilities.

Except for the illustrative example, we started **WitnessGenerate** at the level that corresponds with the known top dimension of the solution set of the systems, because doing otherwise would be computationally too expensive.

7.2.1 The illustrative example

Table 2 collects all the numbers of the flow diagram in Fig. 1. The computations started at level 2 and went down to level 0. The number of paths ($\#\mathbf{x}(t)$) breaks up into three categories: nonsolutions (counted by #ns), solutions that are candidate witness points ($\#\widehat{W}$), and spurious solutions at infinity ($\#\infty$). Note that we did not record the number of paths traced by the polyhedral homotopies to construct the start systems. The timings however include also the computation work for this stage.

The second half of Table 2 lists the output of **WitnessClassify**. The numerical header in every column indicates the dimension of the components found. The first nonzero entry in each column is the degree of the component and the subsequent entries are the number of points classified as on that component.

level	WitnessGenerate					WitnessClassify						
	$\#\mathbf{x}(t)$	#ns	$\#\widehat{W}$	$\#\infty$	cpu time	2	1	1	1	1	0	cpu time
2	139	38	2	99	2m 1s 950ms	2	0	0	0	0	0	8s 800ms
1	38	20	14	4	4s 540ms	8	3	1	1	1	0	2s 20ms
0	20	—	19	1	2s 720ms	13	0	1	2	2	1	0ms
total	197	58	35	104	2m 9s 210ms	23	3	2	3	3	1	10s 820ms

Table 2: Execution summary for the illustrative example. The components are: one 2-dimensional component of degree two, four curves (three linear and one cubic), and one isolated point.

7.2.2 Butcher's problem

This problem arose in the construction of Runge Kutta formulas, see [15] and [12]. There are two versions of this problem, one with seven and another with eight equations. The computations are summarized in the Table 3 and Table 4.

level	WitnessGenerate					WitnessClassify						
	$\#\mathbf{x}(t)$	#ns	$\#\widehat{W}$	$\#\infty$	cpu time	3	3	3	2	2	0	cpu time
3	247	188	3	56	29m 43s 700ms	1	1	1	0	0	0	11s 390ms
2	188	156	14	18	1m 53s 580ms	4	3	5	1	1	0	1s 810ms
1	156	70	10	76	3m 29s 570ms	0	0	8	1	1	0	0ms
0	70	—	6	64	1m 51s 900ms	0	0	2	0	0	4	0ms
total	661	414	33	214	36m 58s 750ms	5	4	16	2	2	4	13s 200ms

Table 3: Execution summary for the seven variable version of Butcher's problem. There are three linear 3-dimensional components, two linear 2-dimensional components, and four isolated points.

In both examples, standard floating-point arithmetic sufficed to classify the components. We see that with low degree components, this classification is swift compared to **Witness-Generate**.

7.2.3 The cyclic n-roots problems

One of the most notorious benchmark problems in polynomial system solving is the so-called cyclic n-roots problem, see [7], [8], [9], [10], [11], [17], and [21]. By a trick of J. Canny, described in [20], we can reduce the dimension of the cyclic n-roots problem by one.

We are interested in the dimensions $n = 4, 8$, and 9, because those systems have components of solutions. R. Fröberg conjectured [28] that if n has a quadratic divisor, then there are infinitely many solutions. For prime dimensions, U. Haagerup proved in [26] that the number of roots is always finite, and equals $\binom{2n-2}{n-1}$.

The summary of the computation for the reduced version of the cyclic 8-roots problem is

level	$\#\mathbf{x}(t)$	#ns	$\#\widehat{W}$	$\#\infty$	cpu time	3	3	3	2	1	1	0	cpu time
	WitnessGenerate					**WitnessClassify**							
3	228	178	3	47	1h 25m 15s 750ms	1	1	1	0	0	0	0	2s 950ms
2	178	154	9	15	2m 18s 250ms	2	2	4	1	0	0	0	600ms
1	154	83	14	57	3m 57s 300ms	0	0	10	2	1	1	0	550ms
0	83	—	20	63	2m 45s 290ms	0	0	4	0	0	0	16	10ms
total	643	415	46	182	1h 34m 16s 590ms	3	3	19	3	1	1	16	4s 110ms

Table 4: Execution summary for the eight variable version of Butcher's problem. Three linear components of dimension three were found, followed by two lines and 16 isolated points.

in Table 5. Although the lines can be distinguished without multi-precision arithmetic, the breakup of the remainder of the whole 1-dimensional component into two curves of degree eight is impossible to do with standard floating-point double-precision arithmetic.

level	$\#\mathbf{x}(t)$	#ns	$\#\widehat{W}$	$\#\infty$	cpu time	1	1	1	1	0	cpu time
	WitnessGenerate					**WitnessClassify**					
1	775	722	18	35	24m 7s 820ms	8	1	8	1	0	1h 11m 36s 470ms
0	722	—	222	500	12m 5s 60ms	44	2	32	0	144	4m 53s 790ms
total	1497	722	240	535	37m 38s 750ms	52	3	40	1	144	1h 16m 30s 260ms

Table 5: Execution summary for the reduced cyclic 8-roots problem. The interpolation was done with 32 decimal places. Two lines and two curves of degree eight were found. Solutions with zero components are spurious for this application, and counted as on "toric" infinity, listed under $\#\infty$.

No multi-precision arithmetic was needed to treat the next problem. The 2-dimensional component of the reduced cyclic 9-roots problem breaks up in two linear components. The execution summary is in Table 6.

For $n = 10$, there are no components of solutions. Cyclic 10-roots has 34940 isolated solutions, so for homotopy continuation this problem is much easier to do than cyclic 8-roots or cyclic 9-roots.

7.2.4 A 7-bar mechanism

This problem tests a known result from the kinematics of planar linkages. Suppose we are given a collection of seven rigid planar pieces: one quadrilateral, two triangles, and four line segments with vertices labeled as shown at the left of Fig. 2.

We are told to assemble the pieces so as to align A with A', B with B', etc. It is not permitted to flip the pieces over, but they can be translated and rotated in any fashion within the plane. One such assembly is shown at the right of Fig. 2. The problem is to find all possible assemblies. It is simplest to hold one of the links, say the quadrilateral, in a fixed location and determine the locations of the remaining links.

	WitnessGenerate					WitnessClassify			
level	$\#\mathbf{x}(t)$	#ns	$\#\widehat{W}$	$\#\infty$	cpu time	2	2	0	cpu time
2	4044	3762	2	280	6h 37m 1s 540ms	1	1	0	1s 990ms
1	3762	2789	6	967	39m 52s 190ms	3	3	0	136ms
0	2789	—	730	2059	1h 4m 20s 890ms	0	0	730	0ms
total	10595	6551	738	3306	8h 21m 14s 620ms	4	4	730	2s 126ms

Table 6: Execution summary for the reduced cyclic 9-roots problem. The 2-dimensional component of degree two breaks up into two linear pieces. Standard floating-point arithmetic sufficed. Solutions with zero components are spurious for this application, and counted as on "toric" infinity, listed under $\#\infty$. The list of 730 isolated solutions contains 650 regular and 20 quadruple solutions.

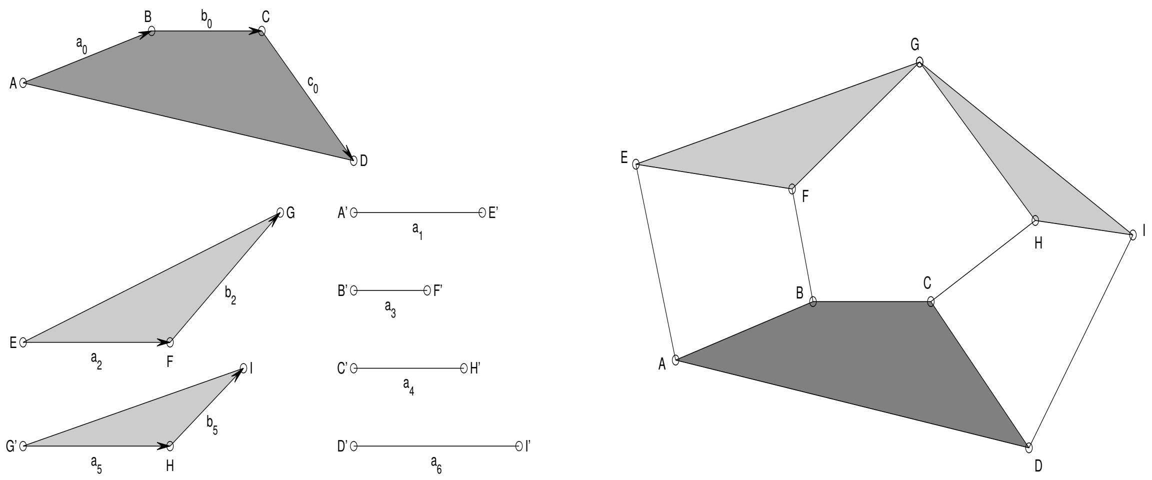

Figure 2: Find all possible assemblies of pieces at the left into a 7-bar mechanism.

It is a well-known result from kinematics that if ℓ is the number of links and v is the number of vertex pairs to be aligned, then for links of general shape, the dimensionality of the solution set is $M = 3(\ell - 1) - 2v$. (Kinematicians call M the *mobility*.) In the case at hand, we have $\ell = 7$ and $v = 9$, so $M = 0$; that is, we expect only isolated solutions. Using the formulation in [42] for problems of this type, the problem can be formulated as a system of polynomial equations:

$$\theta_j \hat{\theta}_j = 1, \qquad j = 1, \ldots, 6 \tag{22}$$

$$\begin{array}{rcl} -a_0 + a_1\theta_1 + a_2\theta_2 - a_3\theta_3 & = & 0 \\ -b_0 + b_2\theta_2 + a_3\theta_3 - a_4\theta_4 + a_5\theta_5 & = & 0 \\ -c_0 + a_4\theta_4 + b_5\theta_5 - a_6\theta_6 & = & 0 \end{array} \tag{23}$$

$$-\bar{a}_0 + \bar{a}_1\hat{\theta}_1 + \bar{a}_2\hat{\theta}_2 - \bar{a}_3\hat{\theta}_3 = 0$$

$$\begin{aligned} -\bar{b}_0 + \bar{b}_2\hat{\theta}_2 + \bar{a}_3\hat{\theta}_3 - \bar{a}_4\hat{\theta}_4 + \bar{a}_5\hat{\theta}_5 &= 0 \\ -\bar{c}_0 + \bar{a}_4\hat{\theta}_4 + \bar{b}_5\hat{\theta}_5 - \bar{a}_6\hat{\theta}_6 &= 0 \end{aligned} \tag{24}$$

The parameters $a_0, b_0, c_0, a_1, a_2, b_2, a_3, a_4, a_5, b_5, a_6$ are complex numbers that describe the shape of the links. In Eqs.(24), $\bar{a}_i$, $\bar{b}_i$, and $\bar{c}_i$ denote the complex conjugate of a_i, b_i, and c_i. One may notice that the coefficients in Eqs.(24) are the conjugates of those in Eqs.(23). The variable θ_i represents the rotation of link i as a complex number. Solutions having $|\theta_i| = 1$ (all i) correspond to actual solutions of the geometric problem.

For generic parameters, this problem has 18 distinct solutions in complex space. For the dimensions shown at the left of Fig. 2, eight of these are "real" solutions having $|\theta_i| = 1$.

For certain special linkages, the generic mobility M is exceeded, and this can happen for the seven-bar problem at hand. Note that if vertices G and G' are not constrained to be aligned, then quadrilaterals $ABFE$ and $CDIH$ can deform independently. The paths traced out by G and G' are called *four-bar coupler curves*. A classic result due to Roberts [37] states that every four-bar curve is triply generated; that is, for every four-bar linkage, there are two other four-bars which generate the same coupler curve. Such four-bar triples are said to be "cognates." If the two four-bars hidden inside our seven-bar linkage are cognates, then the assembly problem will have a 1-dimensional component.

Roberts' proof of the existence of cognates is nonconstructive, but the exact conditions have subsequently been determined [13]. We have constructed a particular test example as follows. First, choose

$$b_0 = 0,\ b_2 = -0.11 + 0.49i,\ a_2 = 0.46,\ a_5 = 0.41,\ c_0 = 1.2,\ \alpha = 0.6 + 0.8i,\ \beta = e^{1.8i}. \tag{25}$$

Then, derive the remaining parameters as

$$\begin{aligned} &a_3 = a_5,\ \gamma = b_2/a_2,\ b_5 = a_5\gamma,\ a_0 = c_0/\gamma,\ a_4 = |b_2|, \\ &a_1 = |a_0 + a_3\alpha - a_4\beta/\gamma|,\ a_6 = |a_4\beta - b_5\alpha - c_0|. \end{aligned} \tag{26}$$

The result is the linkage as shown at the left of Fig. 3, having a solution curve of sixth degree. The linkage also has isolated solutions, such as the one shown at the right of Fig. 3.

There are two such isolated solutions associated with each double point of the four-bar coupler curve, of which there are three, for a total of six isolated roots.

The summary for using the algorithms of this article to compute a numerical irreducible decomposition for this problem is in Table 7. Here multi-precision was used to compute the interpolating polynomial correct up to 23 decimal places accurately. Standard floating-point arithmetic is sufficient here to verify there is only one component of degree six, but we need the additional accuracy of the interpolating polynomial when we want to use the polynomial as a filter to classify the end points of the solution paths.

For generic choices of the parameters, there are 18 isolated solutions. For a generic test problem, PHC finds the 18 solutions in only 13s 240ms.

8 Conclusions

We have given algorithms for finding a numerical primary decomposition of the solution set of a polynomial system into irreducible components. The algorithm is general and is thoroughly

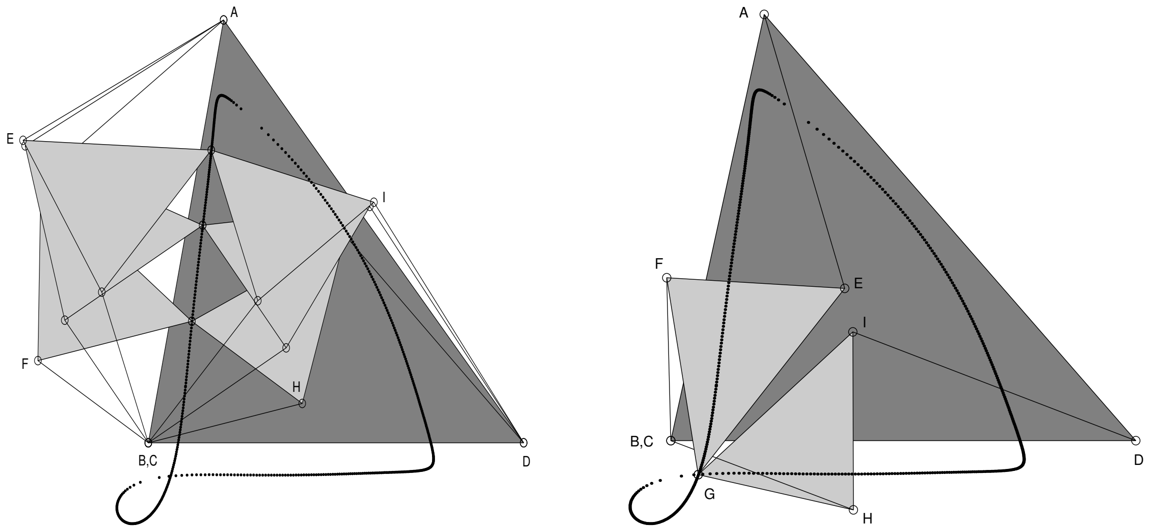

Figure 3: At the left we see a linkage with a solution curve of degree six that has an isolated solution, displayed at the right.

	WitnessGenerate					WitnessClassify		
level	$\#\mathbf{x}(t)$	#ns	$\#\widehat{W}$	$\#\infty$	cpu time	1	0	cpu time
1	48	42	6	0	46s 810ms	6	0	9m 23s 770ms
0	42	—	6	36	1m 7s 350ms	0	6	1s 390ms
total	90	42	12	36	1m 54s 160ms	6	6	9m 25s 160ms

Table 7: Execution summary for 7-bar mechanism. The interpolation was done with 40 decimal places. A curve of degree six and six isolated points are found.

justified on results from algebraic geometry. The results obtained on all of the test problems are very encouraging and agree with known results.

The field of numerical algebraic geometry, in which this algorithm falls, is in its infancy, and there is much work yet to be done. To this point, we have concentrated mainly on the geometric aspects of the algorithms, but it is clear that the numerical analysis of the methods deserves further attention. In particular, as the degree of a solution component increases, the numerical conditioning is seen to worsen. Hence, methods to surmount this problem require multi-precision to adapt the accuracy to the problem. In a related vein, the algorithm must at several points decide when a polynomial function evaluates to zero, so a good method is needed to set the tolerances for such tests, preferably one based on a sound model of the numerical processes used. Another missing piece is a method for tracking the singular paths that will occur when the algorithm must sample a higher-dimensional solution set that has multiplicity greater than one. Finally, the current slicing method used in **WitnessGenerate** creates spurious homotopy paths leading to infinity. A formulation, that avoids or at least mitigates this phenomenon, would be very helpful for treating large problems.

References

[1] E.L. Allgower and K. Georg. *Numerical Continuation Methods, an Introduction*, volume 13 of *Springer Ser. in Comput. Math.* Springer–Verlag, Berlin Heidelberg New York, 1990.

[2] E.L. Allgower and K. Georg. Numerical Path Following. In P.G. Ciarlet and J.L. Lions, editors, *Techniques of Scientific Computing (Part 2)*, volume 5 of *Handbook of Numerical Analysis*, 3–203. North-Holland, 1997.

[3] E.L. Allgower and A.J. Sommese. Piecewise linear approximations of smooth fibers, preprint.

[4] P. Aubry, D. Lazard, and M.M. Maza. On the theories of triangular sets. *J. Symbolic Computation*, 28(1&2):105–124, 1999. Special Issue on Polynomial Elimination – Algorithms and Applications. Edited by M. Kalkbrener and D. Wang.

[5] P. Aubry and M.M. Maza. Triangular sets for solving polynomial systems: a comparative implementation of four methods. *J. of Symbolic Computation*, 28(1&2):125–154, 1999. Special Issue on Polynomial Elimination – Algorithms and Applications. Edited by M. Kalkbrener and D. Wang.

[6] M. Beltrametti, A. Howard, M. Schneider, and A.J. Sommese. Projections from subvarieties. In *Complex Analysis and Algebraic Geometry, Bayreuth, June 1998*, edited by T. Peternell and F.O. Schreyer, (2000), De Gruyter.

[7] J. Backelin and R. Fröberg. How we proved that there are exactly 924 cyclic 7-roots. In *Proceedings of ISSAC-91*, 101–111. ACM, 1991.

[8] G. Björck. Functions of modulus one on Z_p whose Fourier transforms have constant modulus. In *Proceedings of the Alfred Haar Memorial Conference, Budapest*, volume 49 of *Colloquia Mathematica Societatis János Bolyai*, 193–197. 1985.

[9] G. Björck. Functions of modulus one on Z_n whose Fourier transforms have constant modulus, and "cyclic n-roots". In J.S. Byrnes and J.F. Byrnes, editors, *Recent Advances in Fourier Analysis and its Applications*, volume 315 of *NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci.*, 131–140. Kluwer, 1989.

[10] G. Björck and R. Fröberg. A faster way to count the solutions of inhomogeneous systems of algebraic equations, with applications to cyclic n-roots. *J. Symbolic Computation*, 12(3):329–336, 1991.

[11] G. Björck and R. Fröberg. Methods to "divide out" certain solutions from systems of algebraic equations, applied to find all cyclic 8-roots. In M. Gyllenberg and L.E. Persson, editors, *Analysis, Algebra and Computers in Math. research*, volume 564 of *Lecture Notes in Applied Mathematics*, 57–70. Marcel Dekker, 1994.

[12] W. Boege, R. Gebauer, and H. Kredel. Some examples for solving systems of algebraic equations by calculating Groebner bases. *J. Symbolic Computation*, 2:83–98, 1986.

[13] O. Bottema and B. Roth. Theoretical Kinematics. North-Holland, Amsterdam, 1979.

[14] B. Buchberger and F. Winkler, editors, *Gröbner Bases and Applications*. Volume 251 of *London Mathematical Lecture Note Series*, Cambridge University Press, 1998.

[15] C. Butcher. An application of the Runge-Kutta space. *BIT*, 24:425–440, 1984.

[16] D. Cox, J. Little and D. O'Shea. Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer–Verlag, New York, 1992.

[17] J. Davenport. Looking at a set of equations. Technical report 87-06, Bath Computer Science, 1987.

[18] W. Decker, G.M. Greuel, and G. Pfister. Primary Decomposition: Algorithms and Comparisons. In B.H. Matzat, G.M. Greuel, G. Hib (editors): *Algorithmic Algebra and Number Theory. Selected Papers of a Conference held at the University of Heidelberg in October 1997.* Pages 187-220, Springer–Verlag, New York, 1998. Available via `http://loge.math.uni-sb.de/~agdecker/`

[19] D. Eisenbud. *Commutative algebra. With a view toward algebraic geometry.* volume 150 of *Graduate Texts in Math.* Springer–Verlag, New York, 1995.

[20] I.Z. Emiris. *Sparse Elimination and Applications in Kinematics.* PhD thesis, Computer Science Division, Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, 1994.

[21] I.Z. Emiris and J.F. Canny. Efficient incremental algorithms for the sparse resultant and the mixed volume. *J. Symbolic Computation*, 20(2):117–149, 1995. Software available at `http://www.inria.fr/saga/emiris`.

[22] J.C. Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, 1999. Proceedings of MEGA'98, 22–27 June 1998, Saint-Malo, France.

[23] W. Fulton. *Intersection Theory*, volume (3) 2 of *Ergeb. Math. Grenzgeb.* Springer–Verlag, Berlin, 1984.

[24] M. Giusti, K. Hägele, G. Lecerf, J. Marchand, and B. Salvy. The projective Noether Maple package: computing the dimension of a projective variety. Manuscript available at `ftp://medicis.polytechnique.fr/pub/publications/lecerf/`

[25] G.-M. Greuel. Computer algebra and algebraic geometry — Achievements and perspectives. To appear in *J. Symbolic Comp.*

[26] U. Haagerup. Orthogonal MASA's in the n x n - matrices, complex Hadamard matrices and cyclic n-roots. Talk at the conference on "Operator Algebras and Quantum Field Theory", Rome, July 1-6, 1996.

[27] T.Y. Li. Numerical solution of multivariate polynomial systems by homotopy continuation methods. *Acta Numerica*, 6, 399–436, 1997.

[28] H.M. Möller. Gröbner bases and numerical analysis. In B. Buchberger and F. Winkler, editors, *Gröbner Bases and Applications*, volume 251 of *London Mathematical Lecture Note Series*, 159–178. Cambridge University Press, 1998.

[29] A.P. Morgan. *Solving polynomial systems using continuation for engineering and scientific problems.* Prentice-Hall, Englewood Cliffs, N.J., 1987.

[30] A.P. Morgan and A.J. Sommese. Coefficient-parameter polynomial continuation. *Appl. Math. Comput.*, 29:123–160, 1989. Errata: *Appl. Math. Comput.* 51 (1992), p. 207.

[31] A.P. Morgan and A.J. Sommese. Generically nonsingular polynomial continuation. *Proc. AMS-SIAM Summer Seminars on Computational Solution of Nonlinear Systems of Equations*, Lectures in Applied Math. 26 (1990), 467–493.

[32] A.P. Morgan and C.W. Wampler. Solving a planar four-bar design problem using continuation. *ASME J. of Mechanical Design*, 112:544–550, 1990.

[33] D. Mumford. Varieties defined by quadratic equations. in E. Marchionna, editor, *Questions on algebraic varieties*, CIME course 1969, 30–100, Edizioni Cremonese, Rome, 1970.

[34] D. Mumford. *Algebraic Geometry I*, volume 221 of *Grundlehren Math. Wiss.* Springer-Verlag, New York, 1976.

[35] D. Mumford. *The Red Book of Varieties and Schemes*, volume 1358 of *Lecture Notes in Math.* Springer-Verlag, New York, 1988.

[36] M. Raghavan and B. Roth. Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and robot manipulators. *ASME J. of Mechanical Design*, 117:71–79, 1995.

[37] S. Roberts. On three-bar motion in plane space. *Proc. London Math. Soc.* VII:14–23, 1875.

[38] A.J. Sommese and J. Verschelde. Numerical Homotopies to compute generic Points on positive dimensional Algebraic Sets. To appear *Journal of Complexity.* Available at `http://www.nd.edu/~sommese` and `http://www.math.msu.edu/~jan`.

[39] A.J. Sommese and C.W. Wampler. Numerical algebraic geometry. In J. Renegar, M. Shub, and S. Smale, editors, *The Mathematics of Numerical Analysis*, volume 32 of *Lectures in Applied Mathematics*, 749–763, 1996. Proceedings of the AMS-SIAM Summer Seminar in Applied Mathematics, Park City, Utah, July 17-August 11, 1995, Park City, Utah.

[40] J. Verschelde. PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software* 25(2): 251-276, 1999. Paper and software available at `http://www.math.msu.edu/~jan`.

[41] H.F. Walker. Newton-like methods for underdetermined systems. In E.L. Allgower and K. Georg, editors, *Computational Solution of Nonlinear Systems of Equations. Proceedings of the Summer Seminar, Fort Collins, Colorado, 1988*, volume 26 of *Lectures in Applied Mathematics*, 679–699. AMS, 1990.

[42] C.W. Wampler. Solving the kinematics of planar mechanisms. *ASME J. Mechanical Design*, 121:387–391, 1999.

[43] C.W. Wampler, A.P. Morgan and A.J. Sommese. Complete solution of the nine-point path synthesis problem for four-bar linkages. *ASME J. Mechanical Design*, 114:153–159, 1992.

Andrew J. Sommese
Department of Mathematics
University of Notre Dame
Notre Dame, IN 46556, U.S.A.
sommese@nd.edu
http://www.nd.edu/∼sommese

Jan Verschelde
Michigan State University
Department of Mathematics
East Lansing, MI 48824-1027, U.S.A.
jan@math.msu.edu
jan.verschelde@na-net.ornl.gov
http://www.math.msu.edu/∼jan

Charles W. Wampler
General Motors Research Laboratories
Enterprise Systems Lab
Warren, MI 48090, U.S.A.
Charles.W.Wampler@gm.com