SIAM J. NUMER. ANAL.
Vol. 38, No. 6, pp. 2022–2046

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.

Key words. components of solutions, embedding, generic points, homotopy continuation, irreducible components, numerical algebraic geometry, polynomial system, primary decomposition

AMS subject classifications. 65H10, 68W30

PII. S0036142900372549

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

$$f(\mathbf{x}) := \begin{bmatrix} f_1(x_1, \ldots, x_N) \\ \vdots \\ f_n(x_1, \ldots, x_N) \end{bmatrix}, \tag{0.1}$$

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 $\mathbf{V}(f) := \{ \mathbf{x} \in \mathbb{C}^N \mid f_i(\mathbf{x}) = 0, \text{ for } i = 1, \ldots, n \}$. 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 computing 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. Both the degree of each component and an upper bound on its multiplicity are also determined. For components having multiplicity equal to 1, the computed bound is sharp. Furthermore,

[*]Received by the editors May 24, 2000; accepted for publication (in revised form) October 2, 2000; published electronically February 2, 2001.
http://www.siam.org/journals/sinum/38-6/37254.html

[†]Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556 (sommese@nd.edu, http://www.nd.edu/~sommese). This author was supported by 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.

[‡]Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 851 South Morgan (M/C 249), Chicago, IL 60607-7045 (jan@math.uic.edu, jan.verschelde@na-net.ornl.gov, http://www.math.uic.edu/~jan). This author was supported by NSF grant DMS-9804846 at the Department of Mathematics, Michigan State University.

[§]General Motors Research Laboratories, Enterprise Systems Lab, Warren, MI 48090 (Charles.W.Wampler@gm.com).

a postprocessing 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 f, 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 containing all isolated roots of the system [1, 2, 28, 30, 33]. Computing numerical approximations to the roots of a specific problem is useful for many applications, such as mechanism design [34]. Beyond this, the number of isolated solutions to a single generic problem conveys information about every member of a suitably parameterized family of systems [31]. This kind of result has 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 9-point path-synthesis problem for 4-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 [44]. Raghavan and Roth [38] give a survey from the viewpoint of working kinematicians.

The prior success of continuation encourages us to explore the extension of homotopies to find higher-dimensional components of solutions. Symbolic methods for computing primary decompositions (based on triangular sets [4, 5] or Gröbner bases [15]) are available in several computer algebra systems [18, 25]. See also [22, 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 examples in this article, multiprecision arithmetic was needed to arrive at meaningful results.

Previous algorithms based on homotopy continuation produce lists of solutions that include all isolated solutions, but these algorithms cannot certify whether singular solutions are isolated. Our new algorithms construct interpolating polynomials that tell whether a given solution is isolated or not, regardless of its singularity.

This article builds on previous work in [40, 41] for finding generic points on all the solution components. However, the algorithms of [40, 41] 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 a numerical irreducible decomposition.

The organization of this article is as follows. We review in section 1 the concept of primary decomposition, and we sketch our algorithms in section 2. In section 3, we illustrate the algorithms on an example, and we give pseudocode in section 4. A theoretical justification, based on results in section 5, is in section 6. The efficacy of our preliminary implementation is demonstrated in section 7, along with a discussion of numerical aspects. Finally, in section 8 we summarize and point to future research.

1. Background I: Irreducible decomposition.

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. We refer the reader to [36, 37] for an introductory presentation and to [19] for a full discussion.

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 (0.1), the geometric object 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, one considers the *ideal* $\mathcal{I}(f) = \langle f_1, \dots, f_n \rangle$, 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, computer algebra procedures manipulate the equations working with $\mathcal{I}(f)$.

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

$$\mathcal{I}(V) := \{ \ p \in \mathbb{C}[x_1, \dots, x_N] \mid p(V) = 0 \ \} \tag{1.1}$$

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_{reg}, the subset of manifold points of V, being connected in the complex topology. (This is the usual Euclidean topology; see section 5.1.) A well-known result is that every affine variety V has a unique decomposition into irreducible varieties as

$$V = V_1 \cup \cdots \cup V_m \tag{1.2}$$

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_{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 exclude. We organize the irreducible components by dimension, writing the decomposition of the entire solution set $Z = \mathbf{V}(f)$ as

$$Z := \bigcup_{i=0}^{N-1} Z_i := \bigcup_{i=0}^{N-1} \bigcup_{j \in \mathcal{I}_i} Z_{ij}, \tag{1.3}$$

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.

The algebraic analogue of this decomposition is the primary decomposition of $\mathcal{I}(f)$ [19, 36, 37]. Our decomposition contains more information than is contained in classical primary decomposition of $\sqrt{\mathcal{I}(f)}$ but less information than the primary decomposition of $\mathcal{I}(f)$.

2. Overview of algorithms.

Our main goal is to numerically represent the decomposition of the solution set into its irreducible components as in (1.3). 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 postprocessing 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 [19]: 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 1.

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

2.1. IrreducibleDecomposition. Given a system f as in (0.1) the 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 (1.3), a *witness point set* W is a finite set of points of the form

$$W := \bigcup_{i=0}^{N-1} W_i := \bigcup_{i=0}^{N-1} \bigcup_{j \in \mathcal{I}_i} W_{ij}, \tag{2.1}$$

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 subalgorithms is outlined in the following paragraphs.

2.1.1. WitnessGenerate. This algorithm is the function provided by the prior work in [40, 41]. The fundamental idea is that a generic linear variety L_k of dimension k will cut an irreducible variety V of dimension $N-k$ into a finite set of points, equal in number to the degree of V. The qualifier "generic" is key and is discussed in detail in [40, 41]. 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 [40] is more efficient than the earlier [41], 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 endpoints of nondiverging paths not on any k-dimensional component are the start solutions of paths leading to generic points of dimension lower than k. We call those start solutions "nonsolutions." 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 $\widehat{W}$, organized as

$$\widehat{W} = \bigcup_{i=0}^{N-1} \widehat{W}_i, \quad \text{where} \quad \widehat{W}_i = W_i \cup J_i, \tag{2.2}$$

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 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 $\mathbf{v}$ is parameterized by the 1-dimensional set C, by all $\mathbf{q} \in C$. But our algorithm chooses $\mathbf{v}$ generically (i.e., randomly independent of $\mathbf{p}$) from a 2-dimensional set $\mathbb{P}^2$, so with probability 1, S does not contain $\mathbf{p}$. Hence, a test of membership in S is a probability-1 surrogate for a test of membership in C. The idea generalizes readily to varieties and spaces of any dimension, as stated rigorously in Lemma 5.2.

Suppose we have filtering polynomials for all components of dimension greater than i. Then, we may extract W_i from $\widehat{W}_i$: $W_i = \widehat{W}_i \setminus J_i$. 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 that pick $\mathbf{w}$ out of Z_{ij}, using continuation. In this manner, we can generate an arbitrary number of new points on Z_{ij}. After picking a generic projection direction to expand Z_{ij} into a hypersurface, we find the lowest-degree polynomial that interpolates the samples, taking extra samples 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. 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 5.1, 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 subalgorithms, 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 and a specified degree, finds the best-fit polynomial of that degree.
4. *Interpolate* finds an interpolating hypersurface by sampling, projecting and fitting, starting at degree 1 and incrementing until a fit is found (subject to numerical accuracy).

2.2. PolynomialsGenerate. Once **WitnessClassify** is finished, we may construct a set of polynomials whose set of common zeroes is one of the irreducible components. By Lemma 5.3, it suffices to construct $N+1$ hypersurfaces using the Projection and Fit subalgorithms 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 **Polynomials-Generate** generates sufficiently many equations to completely identify the solution sets.

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 $\mathcal{I}(Z_{ij})$. 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:

$$f = \begin{bmatrix} (y-x^2)(x^2+y^2+z^2-1)(x-0.5) \\ (z-x^3)(x^2+y^2+z^2-1)(y-0.5) \\ (y-x^2)(z-x^3)(x^2+y^2+z^2-1)(z-0.5) \end{bmatrix}. \tag{3.1}$$

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

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

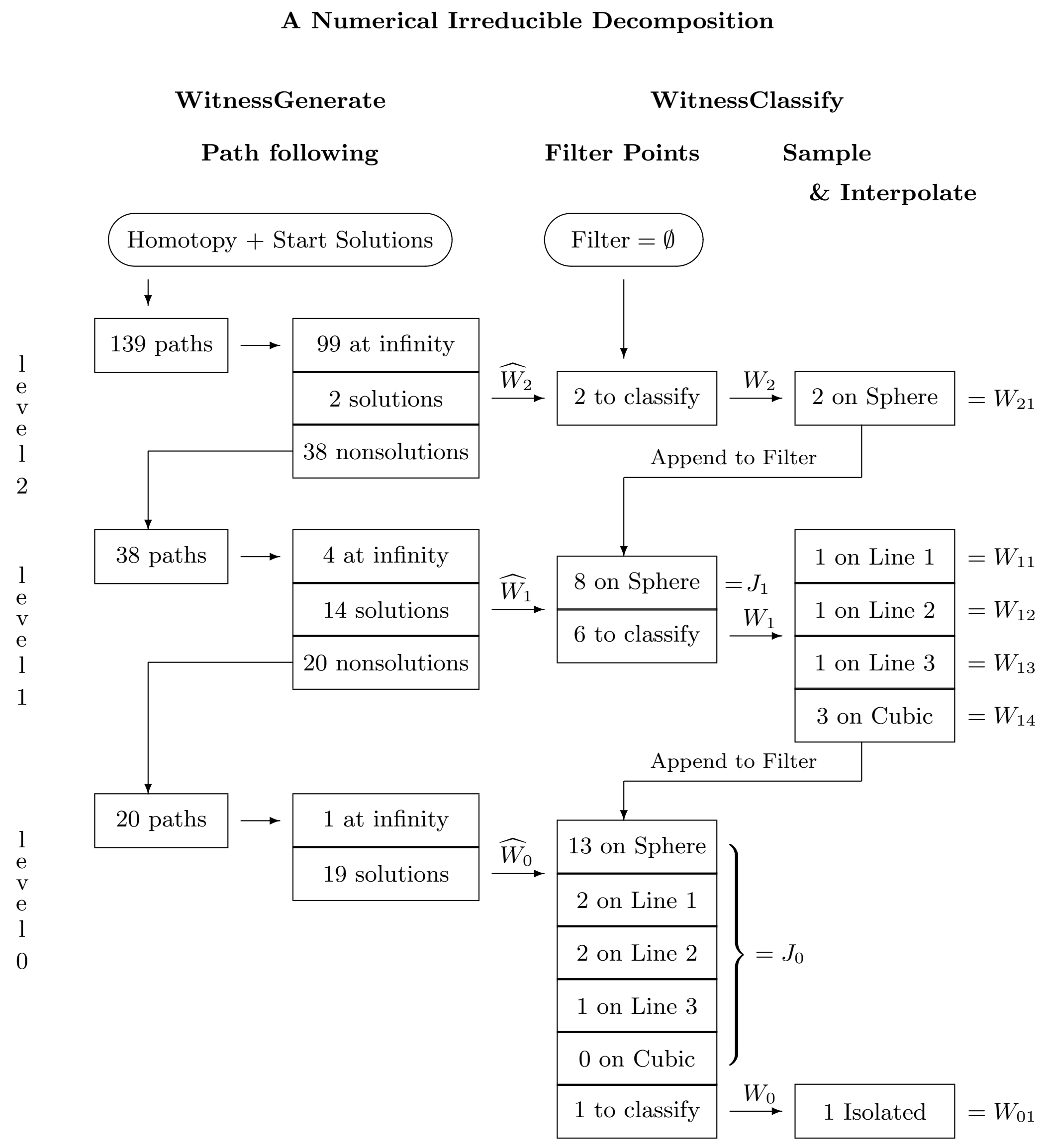

FIG. 3.1. *Flow diagram of illustrative example.*

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 $(y - x^2 = 0, z - x^3 = 0)$; and
6. Z_{01} is the point $(x = 0.5, y = 0.5, z = 0.5)$.

In Figure 3.1 we present the flow diagram for our algorithm **IrreducibleDecomposition** on this illustrative example. The left-hand 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 for classification. Samples from the first point were found to be interpolated by a quadratic surface (the sphere) and the second point also lies on the same surface. Thus, component Z_{21} is a second-degree variety of multiplicity 1. 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 1. 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, 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.

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 $\nu_{14} = 1$; and
6. W_{01} is a nonsingular solution point.

The execution summary of our implementation on this example is described in section 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** and **WitnessClassify** applied sequentially. **PolynomialsGenerate** is a postprocessing step to generate equations for the components after they have been identified by **IrreducibleDecomposition**.

4.1. Squaring polynomial systems.

Our numerical algorithms for tracking homotopy paths require that the number of equations n and the number of unknowns N be equal. Yet, as indicated in (0.1), we wish to treat general problems where this is not necessarily the case. In the algorithm below we apply the embedding of [40] 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 $n < N$                                              [*underdetermined case*]
 then $g_i := f_i$      for $i = 1, 2, \ldots, n$;
      $g_i := 0$        for $i = n+1, \ldots, N$;        [*append zero equations*]
 elseif $n > N$                                         [*overdetermined case*]
```

```
        then $g_i := f_i + \sum_{j=N+1}^{n} a_{ij}x_j$   for $i = 1, 2, \ldots, n$;   [*the $a_{ij}$'s are random numbers*]
        else $g_i := f_i$   for $i = 1, 2, \ldots, n$;                          [*square case*]
end if.
```

For an underdetermined system ($n < N$), the system is made square by appending $N - n$ zero equations. For an overdetermined system ($n > N$), we introduce $n - N$ auxiliary variables $x_{N+1}, \ldots, x_n$. Solutions of the square system g in which all the auxiliary variables equal zero are also solutions to the original system f. All other solutions of g are spurious. Hence, in extracting the solution sets of f we discard those spurious solutions and drop the auxiliary variables from the remaining solutions. These operations are done by **UnSquare**.

ALGORITHM 4.2. $[W, S, F] =$ **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 [41].

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 $f^{-1}(\mathbf{0})$ and positive dimensional irreducible components of $F^{-1}(\mathbf{0})$. 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 $f^{-1}(\mathbf{0})$ with the irreducible component $Z_{ij} \cap L$ of $F^{-1}(\mathbf{0})$. 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 postprocess 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: IrreducibleDecomposition.

Our “probability-1” 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 postprocessing algorithm **PolynomialsGenerate**.

The two main subalgorithms are presented next. The first of these, **WitnessGenerate**, is the algorithm from [40], 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*]
for $i = k$ down to 0 do                            [*loop to process $\widehat{W}_i$*]
    $J_i := \{\mathbf{w} \in \widehat{W}_i \mid p(\mathbf{w}) \approx 0$ for some $p \in F\}$;
    $\widehat{W}_i := \widehat{W}_i \setminus J_i$;                     [*remove "junk"*]
    $j := 0$;                                       [*$j$ counts irreducible components*]
    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 $p(\mathbf{w}) \approx 0$ because in practice both the witness points $\mathbf{w}$ and the coefficients of the filtering polynomials p are subject to numerical error. The same remark applies in algorithm **Interpolate** below.

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

$$\pi(A, \mathbf{x}) = \left(a_{10} + \sum_{j=1}^{N} a_{1j} x_j, \quad \ldots \quad, a_{i0} + \sum_{j=1}^{N} a_{ij} x_j \right), \tag{4.1}$$

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 $(a_{10}, a_{20}, \ldots, a_{i0})^T$ deleted will have fewer rows than columns, and the "direction of the projection" mentioned in section 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 proper terminology appearing in section 5.2 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, the polynomial in the original coordinates is $p(\mathbf{x}) = q(\pi(A, \mathbf{x}))$. In our code we do not expand p but keep the composite form.

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

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).

```
$A := \mathrm{Rand}(i+1, N+1)$;                                   [*generate random $A \in \mathbb{C}^{(i+1)\times(N+1)}$*]
if $i = 0$                                                        [*do not sample isolated points*]
 then $S := \{\mathbf{x}\}$;
      $p(\mathbf{y}) := \pi(A, \mathbf{y}) - \pi(A, \mathbf{x})$;   [*random hyperplane through $\mathbf{x}$*]
 else $S := \emptyset$;                                           [*sample and fit*]
      $d := 1$;                                                   [*start at degree 1*]
      loop
        $m_s := \#\mathrm{monom}(d, i+1) - 1 + k_1$;                [*number of sample points needed*]
        $S := S \cup \mathbf{Sample}(f, \mathbf{x}, i, m_s - \#S)$;  [*expand the sample set*]
        $p := \mathbf{Fit}(d, i, \pi(A, S))$;                        [*fit degree d polynomial*]
        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*]
      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. **Sample** uses a random number generator for the coefficients of the new slicing hyperplanes and then calls the path trackers to find new generic points on those slices. When we have enough points, the function **Fit** constructs an interpolating polynomial, using least squares if $k_1 > 0$. The exiting condition consists of a zero test (numerically up to a certain tolerance) on the evaluation of the interpolating polynomial in $k_2 \geq 1$ extra sampled points.

Remark 4.9. We mention a modification of **WitnessClassify** that will likely yield some significant speedup for components whose dimension is higher than two. Note 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 $\mathbb{C}^{i+1}$.

4.3. Postprocessing: Generation of equations. As justified in Lemma 5.3, we need 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 4.10. $[I] =$ **PolynomialsGenerate**(D, S, F).

```
Input: Degrees $D$; sample sets $S$; filtering polynomials $F$.
Output: Defining equations $I_{ij}$ for irreducible components $Z_{ij}$.
for each irreducible component $(i, j)$ do
  $I_{ij} := F_{ij}$;                        [copy the filtering polynomial from $F$]
  if $i = N - 1$                             [determine #hypersurfaces needed]
    then $m := 1$;                           [hypersurface]
    elseif $D_{ij} = 1$                      [linear component]
        then $m := N - i$;
        else $m := N + 1$;                   [default]
  end if;
  while $\#(I_{ij}) < m$ do                  [construct hypersurfaces]
     $A := \text{Rand}(i + 1, N + 1)$;       [new random projection]
     $I_{ij} := I_{ij} \cup \textbf{Fit}(D_{ij}, i, \pi(A, S_{ij}))$;
  end while;
end for.
```

The if-block is included for efficiency. In general, $N+1$ hypersurfaces suffice, but only one is needed if the component is a hypersurface and $N-i$ suffice if the component is linear.

5. Background II: 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 *quasi-projective variety*, or *variety* for short. Though affine varieties and projective varieties are the most important types of quasi-projective varieties, there are others, e.g., $\mathbb{C}^2 \setminus \{(0,0)\}$.

5.1. The complex and Zariski topologies.

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 $\mathbb{P}^N$; see [37, Chapter 1.10].

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

Often for an irreducible variety X we want a given property to hold for a Zariski open and dense set $U \subset X$ because then, with probability 1, the property holds for a generic point of X. See [40, 41] for illustrations of this concept. Often, it is easier to work with an irreducible variety Y that is mapped algebraically onto a dense set of X; e.g., in section 5.2 below, we study a general projection as a succession of projections with 1-dimensional kernels. If we find a dense Zariski open set U of Y with the property we want, we need to produce a dense Zariski open set of X, given the Zariski open set of Y. To avoid the explicit finding of such an X we apply the theorem of Chevalley on constructible sets [36, Chapter 1.9, 35, Chapter 2C], which guarantees that there is some Zariski open dense subset of X contained in the image of U.

5.2. Projections and generic projections. "Generic" projections have been used since classical times to reduce questions about general varieties to questions about hypersurfaces; see [3, 6, 35] for a number of such applications. Here we give proofs for the facts we need but for which we know no easily accessible reference.

An algebraic map $f : Y \to X$, between algebraic varieties Y and X, is proper if given any compact subset $K \subset X$, $f^{-1}(K)$ is compact. The so-called "proper mapping theorem" states that the image of a closed subvariety under a proper map is a closed variety.

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

$$\pi(x_1, \ldots, x_N) = (L_1(\mathbf{x}), \ldots, L_m(\mathbf{x})), \quad L_i(\mathbf{x}) := a_{i0} + \sum_{j=1}^{N} a_{ij}x_j, \quad a_{ij} \in \mathbb{C}. \tag{5.1}$$

Theoretically, we work with equivalence classes of projections, considering 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 $T(\pi_1(\mathbf{x})) = \pi_2(\mathbf{x})$.

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. A projection from $\mathbb{P}^N$ to $\mathbb{P}^m$ is a surjective map $\pi_L : \mathbb{P}^N \setminus L \to \mathbb{P}^m$ with

$$\pi([x_0, \ldots, x_N]) = [L_0(\mathbf{x}), \ldots, L_m(\mathbf{x})], \quad L_i(\mathbf{x}) := \sum_{j=0}^{N} a_{ij}x_j, \quad a_{ij} \in \mathbb{C} \tag{5.2}$$

and where L is the linear projective space $\mathbb{P}^{N-m-1} \subset \mathbb{P}^N$ defined by the vanishing of the linear equations L_i. L is the *center* of the projection. Theoretically, we work with equivalence classes of projections, considering 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 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 $\langle \mathbf{x}, L \rangle$ 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 $\langle \mathbf{x}, L \rangle \cap \mathbb{P}^m$.

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 with the center of the projective extension of the projection in the affine case.

Let $X \subset \mathbb{C}^N$ be an algebraic subvariety. Let $\pi : \mathbb{C}^N \to \mathbb{C}^m$ be a linear projection. The restriction π_X, of π to 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 as image the complement of the origin and is therefore not proper. It is a part of the Noether normalization theorem [19], that if π is general, then the restriction π_X, of π to X, is in fact proper.

LEMMA 5.1. *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 follows, as noted above, from the Noether normalization theorem. By induction it suffices to show the lemma when $m = N-1 \geq k+1$. To make this induction rigorous, we need to use Chevalley's theorem, which was mentioned above, 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

$$\begin{aligned} 2k &= \dim X \times X = \dim X \times X \setminus \Delta \\ &\geq k + \dim \phi(X \times X \setminus \Delta) \geq k + \dim \big(H \setminus (\overline{X} \setminus X)\big) \\ &= k + N - 1, \end{aligned} \tag{5.3}$$

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.

Now we show that the degree of X is the same as the degree of $\pi(X)$. 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 $\pi(X)$ 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 $\overline{X}$. ∎

We need some more refined statements to separate points by using different projections.

LEMMA 5.2. *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 5.1, 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. ∎

We repeatedly use that a reduced affine subvariety $A \subset \mathbb{C}^N$, all of whose irreducible components are of dimension $N-1$, is the zero set of a polynomial on $\mathbb{C}^N$ (see [36, p. 7]). 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 5.3. *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 be 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 5.2, $\pi_{N-1}(S) \cap \pi_{N-1}(X) = \emptyset$, and thus the set of 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. ∎

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

$$\mathrm{mult}(f, \mathbf{x}^*) := \dim_{\mathbb{C}} \mathcal{O}_{\mathbb{C}^N|\mathbf{x}^*} / \langle\langle f_1, \ldots, f_n \rangle\rangle, \tag{5.4}$$

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. $\langle\langle f_1, \ldots, f_n \rangle\rangle$ denotes the ideal generated by the f_i in $\mathcal{O}_{\mathbb{C}^N|\mathbf{x}^*}$.

The following is a simple observation.

LEMMA 5.4. *Given a system f of n polynomial equations $f_1, \ldots, f_n \in \mathbb{C}[x_1, \ldots, x_N]$ and an isolated zero $\mathbf{x}^*$ of the system, and a matrix*

$$\begin{bmatrix} \lambda_{11} & \cdots & \lambda_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{N1} & \cdots & \lambda_{Nn} \end{bmatrix} \in \mathbb{C}^{N \times n} \tag{5.5}$$

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 $\mathrm{mult}(f, \mathbf{x}^*) \leq \mathrm{mult}(g, \mathbf{x}^*)$.

Proof. Simply note that $\langle\langle g_1, \ldots, g_N \rangle\rangle \subset \langle\langle f_1, \ldots, f_n \rangle\rangle$. ∎

In [40, 41], a polynomial system is replaced with a new system made out of linear combinations of the equations of the systems so 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. An example in [41] shows 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 $\mathrm{mult}(f, W)$, the multiplicity of W with respect to f, we note that it follows from [23, Chapter 10] that given a generic point $\mathbf{w} \in W$ and k linear equations $L_1, \dots, L_k$ which define a linear $\mathbb{C}^{N-k}$ passing through $\mathbf{w}$ and transverse to W at $\mathbf{w}$, then $\mathrm{mult}(f, W) = \mathrm{mult}(f, L_1, \dots, L_k, \mathbf{w})$. Thus, given generic linear equations $L_1, \dots, L_k$, $\mathrm{mult}(f, L_1, \dots, L_k, \mathbf{w})$ is the same value for all the points $\{ \mathbf{w} \in W \mid L_i(\mathbf{w}) = 0 \text{ for all } i \}$.

In [40, 41], to study a dimension k component W of the zero set of a given system f, f is replaced with a new system g of polynomials $g_1, \dots, g_{N-k}$ of random linear combinations of the f_i. Then the system g is augmented with k general linear equations $L_1, \dots, L_k$. The combination of Lemma 5.4 and the last paragraph guarantees that

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

If homotopy continuation is used to solve the system g augmented with $L_1, \dots, L_k$, then the quantity $\mathrm{mult}(g, L_1, \dots, L_k, \mathbf{w})$ is the number of paths ending at $\mathbf{w}$. Thus, we find a bound for $\mathrm{mult}(f, W)$. Since the bound is the same for all the generic points of W we compute, we use it as a check on our breakup of the witness points.

6. Theoretical justification.

Let f be a system as in (0.1) with not all the f_i equal to the zero polynomial. We have the decomposition (1.3) 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** to describe it explicitly) with

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

Moreover, as shown in [40, 41], for each $\mathbf{w} \in W_{ij}$ **WitnessGenerate** has a number of paths $\nu_{ij} \ge \mathrm{mult}(f, Z_{ij})$ 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.

For the top dimension k, J_k is empty. If we move the linear space defining a generic point $\mathbf{x}$ of some Z_{kj}, we can trace out by continuation as large a finite set S_{kj} as we want of points of the same component Z_{kj}, 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 section 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 $\deg Z_{k1}$. 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**.

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.

7.1. Numerics of fitting.

We denote the interpolating polynomial $p(\mathbf{x})$ of degree d in n variables as

$$p(\mathbf{x}) = \sum_{|\mathbf{a}| \leq d} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}} \quad \text{with} \quad \begin{array}{l} \mathbf{x}^{\mathbf{a}} = x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n} \\ |\mathbf{a}| = a_1 + a_2 + \cdots + a_n \end{array} \quad \text{and} \quad m = \frac{(n+d)!}{n!d!}, \tag{7.1}$$

where m is the number of monomials of degree $\leq d$ in n variables in this dense representation. 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 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 the accuracy of the sample points $\mathbf{x}_i$ and the conditioning of the matrix $\mathbf{Y}$. To improve the conditioning of $\mathbf{Y}$, we disperse the sample points widely. Points close together will lead to rows of $\mathbf{Y}$ that are nearly equal. For example, consider 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 higher degrees, the numerical issues are thornier because 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 took (7.1) with $n = 2$ and $c_{\mathbf{a}} = 1$ for various degrees. In the numerical experiment we reconstructed this polynomial from sampled points (x_k, y_k) satisfying $f(x_k, y_k) = 0$, for $k = 1, 2, \ldots, m-1$. 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 $c_{\mathbf{0}} = 1$, formed the matrix $\mathbf{Y}$, and solved $\mathbf{Y}\mathbf{c} = 0$ by classical LU decomposition, with 32-digit multiprecision arithmetic. Since the correct values of the coefficients are known from the outset (all $c_{\mathbf{a}} = 1$), we can evaluate the error. The experiment was repeated for degrees from 2 to 12. The results in Table 7.1 indicate a steady worsening of the conditioning as the degree increases, with a corresponding increase in the error of the computed coefficients. For degrees greater than or equal to 4, the conditioning is such that double precision calculations—instead of 32 decimal places—will give coefficients only accurate to 10^{-8} or worse. Obviously multiple precision is needed for higher degrees.

7.2. Test problems.

The algorithms have been implemented in an extra module of PHCpack [42]. Reported timings concern a Pentium III 800 Mhz processor running Linux. The first two examples were done with standard machine arithmetic, but two systems required multiprecision facilities to refine the witness points and to construct the interpolating polynomials.

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.

Table 7.1
Under "rcond" we list the estimate for the inverse of the condition number of the linear system. Column "error" is the maximal error in the coefficients $c_{\mathbf{a}}$.

deg	#terms	rcond	error
2	6	2.935E-06	1.409E-27
3	10	2.331E-05	1.437E-28
4	15	1.062E-08	1.761E-25
5	21	1.226E-07	9.069E-27
6	28	3.661E-08	6.636E-26
7	36	1.639E-10	4.200E-24
8	45	1.050E-12	7.589E-22
9	55	2.486E-14	7.840E-21
10	66	6.722E-14	2.859E-21
11	78	1.359E-13	2.086E-21
12	91	2.686E-16	3.540E-19

Table 7.2
Execution summary for the illustrative example. We find one 2-dimensional component of degree 2, four curves (three linear and one cubic), and one isolated point.

	WitnessGenerate					**WitnessClassify**						
Level	$\#\mathbf{x}(t)$	#ns	$\#\widehat{W}$	$\#\infty$	CPU time	2	1	1	1	1	0	CPU time
2	139	38	2	99	1m 31s 50ms	2	0	0	0	0	0	6s 460ms
1	38	20	16	2	4s 540ms	10	3	1	1	1	0	9s 560ms
0	20	—	14	6	2s 720ms	8	0	1	2	2	1	0ms
Total	197	58	32	107	1m 35s 300ms	20	3	2	3	3	1	16s 20ms

7.2.1. The illustrative example. Table 7.2 collects all the numbers of the flow diagram in Figure 3.1. 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 computational work for this stage.

The second half of Table 7.2 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.

7.2.2. Butcher's problem. This problem arose in the construction of Runge–Kutta formulas; see [16, 13]. There are two versions of this problem, one with seven and another with eight equations. The computations are summarized in Tables 7.3 and 7.4.

In both examples, standard machine arithmetic sufficed in **WitnessClassify**. We see that with low degree components, this classification is swift compared to **WitnessGenerate**.

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 [8, 9, 10, 11, 12, 17, 21, 29]. By a trick of Canny, described in [20], we can reduce the dimension of the cyclic n-roots problem by 1.

We are interested in the dimensions $n = 4, 8$, and 9, because those systems have

Table 7.3

Execution summary for the 7-variable version of Butcher's problem. There are three linear 3-dimensional components, two linear 2-dimensional components, and five isolated points.

	WitnessGenerate					**WitnessClassify**						
Level	$\#\mathbf{x}(t)$	#ns	$\#\widehat{W}$	$\#\infty$	CPU time	3	3	3	2	2	0	CPU time
3	247	195	3	49	8m 32s 460ms	1	1	1	0	0	0	1s 600ms
2	195	165	11	19	21s 90ms	3	1	5	1	1	0	1s 50ms
1	165	70	11	79	28s 820ms	0	0	8	1	2	0	0ms
0	79	—	7	68	11s 810ms	0	0	2	0	0	5	0ms
Total	682	435	32	215	9m 34s 180ms	5	4	16	2	3	5	2s 650ms

Table 7.4

Execution summary for the 8-variable version of Butcher's problem. Three linear components of dimension 3 were found, followed by a linear component of dimension 2, and then by 2 lines, and finally 16 isolated points.

	WitnessGenerate					**WitnessClassify**							
Level	$\#\mathbf{x}(t)$	#ns	$\#\widehat{W}$	$\#\infty$	CPU time	3	3	3	2	1	1	0	CPU time
3	228	181	3	44	31m 3s 910ms	1	1	1	0	0	0	0	1s 370ms
2	181	157	9	15	25s 600ms	4	2	2	1	0	0	0	150ms
1	157	86	14	57	24s 570ms	10	0	0	2	1	1	0	760ms
0	86	—	21	65	14s 430ms	5	0	0	0	0	0	16	0ms
Total	652	434	47	181	32m 8s 510ms	20	3	3	3	1	1	16	2s 280ms

Table 7.5

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

	WitnessGenerate					**WitnessClassify**					
Level	$\#\mathbf{x}(t)$	#ns	$\#\widehat{W}$	$\#\infty$	CPU time	1	1	1	1	0	CPU time
1	775	750	18	7	9m 40s 190ms	8	1	8	1	0	7m 29s 610ms
0	750	—	235	515	3m 27s 30ms	43	0	48	0	144	52s 800ms
Total	1525	750	253	522	13m 7s 220ms	51	1	56	1	144	8m 22s 410ms

components of solutions. Backelin [7] proved that if n has a quadratic divisor, then there are infinitely many solutions. For prime dimensions, Fröberg conjectured and 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 in Table 7.5. Although the lines can be distinguished without multiprecision arithmetic, the breakup of the remainder of the whole 1-dimensional component into two curves of degree 8 is impossible to do with standard floating-point double-precision arithmetic.

No multiprecision arithmetic was needed for the reduced cyclic 9-roots problem whose 2-dimensional component breaks up in two linear components; see Table 7.6.

For $n = 10$, all solutions are isolated: there are 34940 of them in total. Since there are no positive dimensional components, this problem is much easier for homotopy continuation than the cyclic 8-roots or cyclic 9-roots problems.

TABLE 7.6
Execution summary for reduced cyclic 9*-roots. The* 2*-dimensional component of degree* 2 *breaks up into* 2 *linear pieces. Standard floating-point arithmetic sufficed. Solutions with* 0 *components are spurious and counted as on "toric" infinity, listed under* $\#\infty$*. The list of* 730 *isolated solutions contains* 650 *regular and* 20 *quadruple solutions.*

	WitnessGenerate					**WitnessClassify**			
Level	$\#\mathbf{x}(t)$	#ns	$\#\widehat{W}$	$\#\infty$	CPU time	2	2	0	CPU time
2	4044	4018	2	24	2h 33m 19s 160ms	1	1	0	2s 40ms
1	4018	3009	8	1001	16m 39s 20ms	6	2	0	136ms
0	3009	—	730	2279	36m 53s 870ms	0	0	730	0ms
Total	11071	7027	740	3104	3h 26m 52s 50ms	7	3	730	2s 176ms

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 top of Figure 7.1.

We wish 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 Figure 7.1. 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.

Using the formulation in [43] for problems of this type, the problem can be formulated as a system of polynomial equations:

$$\theta_j \hat{\theta}_j = 1, \qquad j = 1, \dots, 6, \tag{7.2}$$

$$\begin{aligned} -a_0 + a_1\theta_1 + a_2\theta_2 - a_3\theta_3 &= 0, \\ -b_0 + b_2\theta_2 + a_3\theta_3 - a_4\theta_4 + a_5\theta_5 &= 0, \\ -c_0 + a_4\theta_4 + b_5\theta_5 - a_6\theta_6 &= 0, \end{aligned} \tag{7.3}$$

$$\begin{aligned} -\bar{a}_0 + \bar{a}_1\hat{\theta}_1 + \bar{a}_2\hat{\theta}_2 - \bar{a}_3\hat{\theta}_3 &= 0, \\ -\bar{b}_0 + \bar{b}_2\hat{\theta}_2 + \bar{a}_3\hat{\theta}_3 - \bar{a}_4\hat{\theta}_4 + \bar{a}_5\hat{\theta}_5 &= 0, \\ -\bar{c}_0 + \bar{a}_4\hat{\theta}_4 + \bar{b}_5\hat{\theta}_5 - \bar{a}_6\hat{\theta}_6 &= 0. \end{aligned} \tag{7.4}$$

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 (7.4), $\bar{a}_i$, $\bar{b}_i$, and $\bar{c}_i$ denote the complex conjugate of a_i, b_i, and c_i. One may notice that the coefficients in (7.4) are the conjugates of those in (7.3). 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; see [27] for a demonstration using a different formulation. For the dimensions shown at the top of Figure 7.1, 8 of these are "real" solutions having $|\theta_i| = 1$.

For certain special linkages, higher-dimensional solution sets can occur. One such example can be constructed by making the two 4-bar linkages $ABFEG$ and $CDIHG$ to be Roberts cognates [39] (see also [14, p. 340]), so that the solution set must include a 4-bar coupler curve, having degree 6. A particular example is as follows.

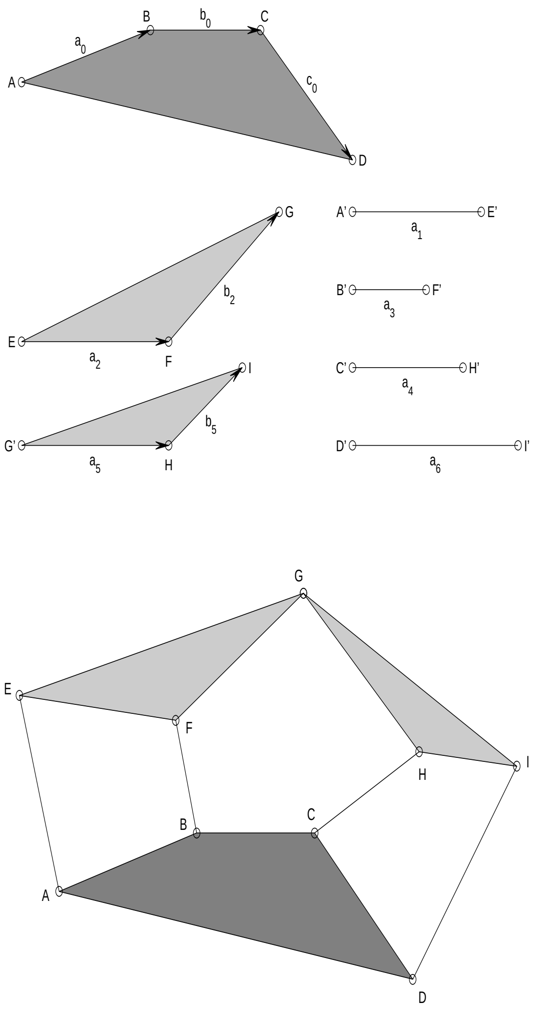

FIG. 7.1. *Find all possible assemblies of pieces at the top into a* 7*-bar mechanism.*

First, choose

$$b_0 = 0, \quad b_2 = -0.11 + 0.49i, \quad a_2 = 0.46, \quad a_5 = 0.41, \\ c_0 = 1.2, \quad \alpha = 0.6 + 0.8i, \quad \beta = e^{1.8i}. \tag{7.5}$$

Then, derive the remaining parameters as

$$a_3 = a_5, \quad \gamma = b_2/a_2, \quad b_5 = a_5\gamma, \quad a_0 = c_0/\gamma, \quad a_4 = |b_2|, \\ a_1 = |a_0 + a_3\alpha - a_4\beta/\gamma|, \quad a_6 = |a_4\beta - b_5\alpha - c_0|. \tag{7.6}$$

The result is the linkage as shown in Figure 7.2. In addition to a solution curve of sixth degree, shown on the top, the linkage also has six isolated solutions, such as the one shown at the bottom of Figure 7.2. There are two isolated solutions associated

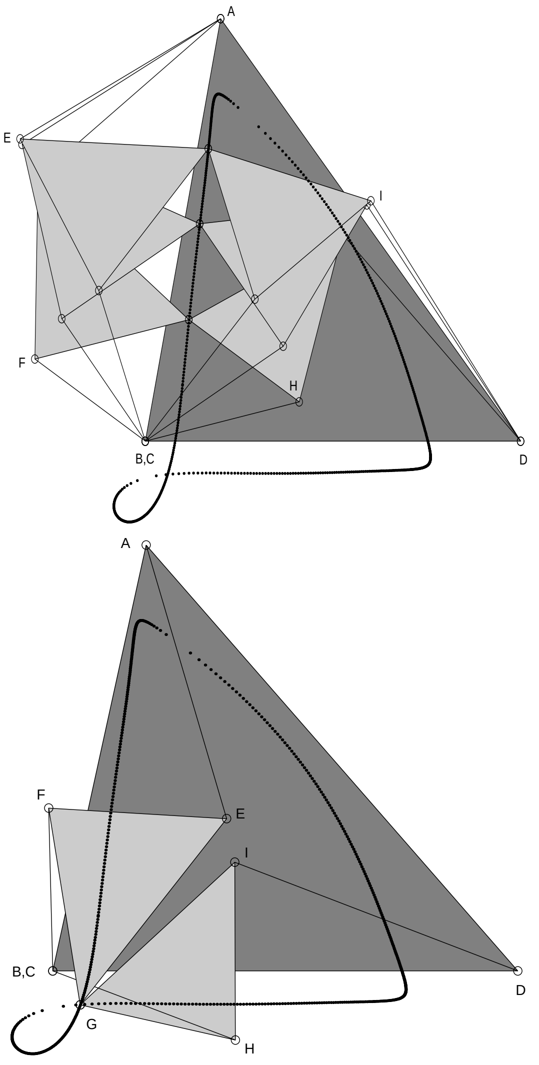

FIG. 7.2. *7-bar linkage with a solution curve of degree* 6*, shown on the top, and six isolated solutions, one of which is displayed at the bottom.*

with each of three double points of the four-bar coupler curve. In this example, only one of the three double points is real.

The execution summary for this problem is in Table 7.7. Standard floating-point arithmetic is sufficient to detect that there is only one component of degree 6, but we need an accurate interpolating polynomial to use 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 4s 700ms.

8. Conclusions.

We presented algorithms for finding a numerical decomposition of the solution set of a polynomial system into irreducible components. The methods are general and thoroughly justified by algebraic geometry. Experiments on the test

Table 7.7
Execution summary for 7*-bar mechanism. The interpolation was done with* 40 *decimal places. A curve of degree* 6 *and six isolated points are found.*

	WitnessGenerate					**WitnessClassify**		
Level	$\#\mathbf{x}(t)$	#ns	$\#\widehat{W}$	$\#\infty$	CPU time	1	0	CPU time
1	48	42	6	0	8s 670ms	6	0	42s 180ms
0	42	—	6	36	3s 700ms	0	6	290ms
Total	90	42	12	36	12s 370ms	6	6	42s 470ms

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 multiprecision 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. Another missing piece is a method for tracking the singular paths occurring when sampling a higher-dimensional solution set that has multiplicity greater than 1. Finally, the current slicing method used in **WitnessGenerate** creates spurious paths leading to infinity. A formulation that avoids or at least mitigates this phenomenon will be needed for treating large problems.

Acknowledgment. We thank the referee for helpful comments.

REFERENCES

[1] E.L. Allgower and K. Georg, *Numerical Continuation Methods. An Introduction*, Springer Ser. Comput. Math. 13, Springer-Verlag, Berlin, 1990.

[2] E.L. Allgower and K. Georg, *Numerical Path Following*, in Handb. Numer. Anal. V, P.G. Ciarlet and J.L. Lions, eds., North-Holland, Amsterdam, 1997, pp. 3–207.

[3] E.L. Allgower and A.J. Sommese, *Piecewise Linear Approximations of Smooth Fibers*, preprint, available at http://www.nd.edu/~sommese.

[4] P. Aubry, D. Lazard, and M.M. Maza, *On the theories of triangular sets*, J. Symbolic Comput., 28 (1999), pp. 105–124.

[5] P. Aubry and M.M. Maza, *Triangular sets for solving polynomial systems: A comparative implementation of four methods*, J. Symbolic Comput., 28 (1999), pp. 125–154.

[6] M. Beltrametti, A. Howard, M. Schneider, and A.J. Sommese, *Projections from subvarieties*, in Complex Analysis and Algebraic Geometry, T. Peternell and F.O. Schreyer, eds., de Gruyter, Berlin, 2000, pp. 71–107.

[7] J. Backelin, *Square multiples n give infinitely many cyclic n-roots*, Reports, no. 8, Matematiska Institutionen, Stockholms Universitet, Stockholm, Sweden, 1989.

[8] J. Backelin and R. Fröberg, *How we proved that there are exactly* 924 *cyclic* 7*-roots*, in Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC-91), ACM, New York, 1991, pp. 101–111.

[9] G. Björck, *Functions of modulus one on Z_p whose Fourier transforms have constant modulus*, in Proceedings of the Alfred Haar Memorial Conference, Budapest, Hungary, Colloq. Math. Soc. János Bolyai 49, 1985, pp. 193–197.

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

[11] 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 Comput., 12 (1991), pp. 329–336.

[12] G. Björck and R. Fröberg, *Methods to "divide out" certain solutions from systems of algebraic equations, applied to find all cyclic* 8*-roots*, in Analysis, Algebra and Computers in Mathematical Research, M. Gyllenberg and L.E. Persson, eds., Lecture Notes in Pure and Appl. Math. 156, Dekker, New York, 1994, pp. 57–70.

[13] W. Boege, R. Gebauer, and H. Kredel, *Some examples for solving systems of algebraic equations by calculating Groebner bases*, J. Symbolic Comput., 2 (1986), pp. 83–98.

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

[15] B. Buchberger and F. Winkler, eds., *Gröbner Bases and Applications*, London Math. Soc. Lecture Note Ser. 251, Cambridge University Press, Cambridge, UK, 1998.

[16] C. Butcher, *An application of the Runge-Kutta space*, BIT, 24 (1984), pp. 425–440.

[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 Algorithmic Algebra and Number Theory, B.H. Matzat, G.M. Greuel, G. Hiss, eds., Springer-Verlag, Berlin, 1999, pp. 187-220.

[19] D. Eisenbud, *Commutative Algebra. With a View Toward Algebraic Geometry*, Grad. Texts in Math. 150, Springer-Verlag, New York, 1995.

[20] I.Z. Emiris, *Sparse Elimination and Applications in Kinematics*, Ph.D. thesis, University of California, Berkeley, CA, 1994.

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

[22] J.C. Faugère, *A new efficient algorithm for computing Gröbner bases (F_4)*, in Effective Methods in Algebraic Geometry, H. Lombardi and M.-F. Roy, eds., J. Pure Appl. Math., 139 (1999), pp. 61–88.

[23] W. Fulton, *Intersection Theory*, Ergeb. Math. Grenzgeb(3), 2, 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*, J. Symbolic Comput., 30 (2000), pp. 291–307.

[25] G.-M. Greuel, *Computer algebra and algebraic geometry—achievements and perspectives*, J. Symbolic Comput., 30 (2000), pp. 253–289.

[26] U. Haagerup, *Orthogonal maximal abelian $*$-subalgebras of the $n \times n$ matrices and cyclic n-roots*, in Operator Algebras and Quantum Field Theory, International Press, Cambridge, MA, 1996, pp. 296–322.

[27] C. Innocenti, *Polynomial solution to the position analysis of the* 7*-link Assur kinematic chain with one quaternary link*, Mech. Mach. Theory, 30 (1995), pp. 1295–1303.

[28] T.Y. Li, *Numerical solution of multivariate polynomial systems by homotopy continuation methods*, Acta Numer., 6 (1997), pp. 399–436.

[29] H.M. Möller, *Gröbner bases and numerical analysis*, in Gröbner Bases and Applications, B. Buchberger and F. Winkler, eds., London Math. Soc. Lecture Note Ser. 251, Cambridge University Press, Cambridge, UK, 1998, pp. 159–178.

[30] A.P. Morgan, *Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[31] A.P. Morgan and A.J. Sommese, *Coefficient-parameter polynomial continuation*, Appl. Math. Comput., 29 (1989), pp. 123–160.

[32] A.P. Morgan and A.J. Sommese, *Errata: "Coefficient-parameter polynomial continuation,"* Appl. Math. Comput., 51 (1992), p. 207.

[33] A.P. Morgan and A.J. Sommese, *Generically nonsingular polynomial continuation*, in Computational Solution of Nonlinear Systems of Equations, Lectures in Appl. Math. 26, AMS, Providence, RI, 1990, pp. 467–493.

[34] A.P. Morgan and C.W. Wampler, *Solving a planar four-bar design problem using continuation*, ASME J. Mech. Design, 112 (1990), pp. 544–550.

[35] D. Mumford, *Varieties defined by quadratic equations*, in Questions On Algebraic Varieties, E. Marchionna, ed., Edizioni Cremonese, Rome, 1970, pp. 29–100.

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

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

[38] M. Raghavan and B. Roth, *Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and robot manipulators*, ASME J. Mech. Design, 117 (1995), pp. 71–79.

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

[40] A.J. Sommese and J. Verschelde, *Numerical homotopies to compute generic points on positive dimensional algebraic sets*, J. Complexity, 16 (2000), pp. 572-602.

[41] A.J. Sommese and C.W. Wampler, *Numerical algebraic geometry*, in The Mathematics of Numerical Analysis, J. Renegar, M. Shub, and S. Smale, eds., Lectures in Appl. Math. 32, AMS, Providence, RI, 1996, pp. 749–763,

[42] J. Verschelde, *Algorithm* 795: *PHCpack: A general-purpose solver for polynomial systems by homotopy continuation*, ACM Trans. Math. Software, 25 (1999), pp. 251-276; paper and software available at http://www.math.uic.edu/~jan.

[43] C.W. Wampler, *Solving the kinematics of planar mechanisms*, ASME J. Mech. Design, 121 (1999), pp. 387–391.

[44] 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. Mech. Design, 114 (1992), pp. 153–159.