Solving Polynomial Systems Equation by Equation*

Andrew J. Sommese† Jan Verschelde‡ Charles W. Wampler§

23 January 2006

Abstract

By a numerical continuation method called a diagonal homotopy, one can compute the intersection of two irreducible positive dimensional solution sets of polynomial systems. This paper proposes to use this diagonal homotopy as the key step in a procedure to intersect general solution sets that are not necessarily irreducible or even equidimensional. Of particular interest is the special case where one of the sets is defined by a single polynomial equation. This leads to an algorithm for finding a numerical representation of the solution set of a system of polynomial equations introducing the equations one by one. Preliminary computational experiments show this approach can exploit the special structure of a polynomial system, which improves the performance of the path following algorithms.

2000 Mathematics Subject Classification. Primary 65H10; Secondary 13P05, 14Q99, 68W30.

Key words and phrases. Algebraic set, component of solutions, diagonal homotopy, embedding, equation-by-equation solver, generic point, homotopy continuation, irreducible component, numerical irreducible decomposition, numerical algebraic geometry, path following, polynomial system, witness point, witness set.

*The authors acknowledge the support of Land Baden-Württemberg (RiP-program at Oberwolfach).

†Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556-4618, USA *Email:* sommese@nd.edu *URL:* http://www.nd.edu/˜sommese. This material is based upon work supported by the National Science Foundation under Grant No. 0105653 and Grant No. 0410047; and the Duncan Chair of the University of Notre Dame.

‡Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 851 South Morgan (M/C 249), Chicago, IL 60607-7045, USA *Email:* jan@math.uic.edu or jan.verschelde@na-net.ornl.gov *URL:* http://www.math.uic.edu/˜jan. This material is based upon work supported by the National Science Foundation under Grant No. 0105739, Grant No. 0134611, and Grant No. 0410036.

§General Motors Research and Development, Mail Code 480-106-359, 30500 Mound Road, Warren, MI 48090-9055, U.S.A. *Email:* Charles.W.Wampler@gm.com. This material is based upon work supported by the National Science Foundation under Grant No. 0410047 and by General Motors Research and Development.

1 Introduction

Homotopy continuation methods provide reliable and efficient numerical algorithms to compute accurate approximations to all isolated solutions of polynomial systems, see e.g. [13] for a recent survey. As proposed in [23], we can approximate a positive dimensional solution set of a polynomial system by isolated solutions, which are obtained as intersection points of the set with a generic linear space of complementary dimension.

New homotopy algorithms have been developed in a series of papers [14, 15, 18, 21, 22] to give numerical representations of positive dimensional solution sets of polynomial systems. These homotopies are the main numerical algorithms in a young field we call *numerical algebraic geometry*. See [24] for a detailed treatment of this subject.

This paper provides an algorithm to compute numerical approximations to positive dimensional solution sets of polynomial systems by introducing the equations a few at a time, i.e., "subsystem by subsystem," or even one at a time, i.e., "equation by equation." The advantage of working in this manner is that the special properties of subsets of the equations are revealed early in the process, thus reducing the computational cost of later stages. Consequently, although the new algorithm has more stages of computation than earlier approaches, the amount of work in each stage can be considerably less, producing a net savings in computing time.

This paper is organized in three parts. In §2, we explain our method to represent and to compute a numerical irreducible decomposition of the solution set of a polynomial system. In §2.3, we relate our numerical irreducible decomposition to the symbolic geometric resolution. In §3, the new diagonal homotopy algorithm is applied to solve systems subsystem by subsystem or equation by equation. Computational experiments are given in §4.

2 A Numerical Irreducible Decomposition

We start this section with a motivating illustrative example that shows the occurrence of several solution sets of different dimensions and degrees. Secondly, we define the notion of witness sets, which represent pure dimensional solution sets of polynomial systems *numerically*. Witness sets are computed by cascades of homotopies between embeddings of polynomial systems.

2.1 An Illustrative Example

Our running example (used also in [15]) is the following:

$$f(x,y,z) = \left[\begin{array}{r} (y-x^2)(x^2+y^2+z^2-1)(x-0.5) \\ (z-x^3)(x^2+y^2+z^2-1)(y-0.5) \\ (y-x^2)(z-x^3)(x^2+y^2+z^2-1)(z-0.5) \end{array} \right]. \tag{1}$$

In this factored form we can easily identify the decomposition of the solution set $Z = f^{-1}(\mathbf{0})$ into irreducible solution components, as follows:

$$Z = Z_2 \cup Z_1 \cup Z_0 = \{Z_{21}\} \cup \{Z_{11} \cup Z_{12} \cup Z_{13} \cup Z_{14}\} \cup \{Z_{01}\} \tag{2}$$

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 $(x = 0.5, y = 0.5, z = 0.5)$.

The sequence of homotopies in [14] tracks 197 paths to find a numerical representation of the solution set Z. With the new approach we have to trace only 13 paths! We show how this is done in Figure 3 in §4.1 below, but we first describe a numerical representation of Z in the next section.

2.2 Witness Sets

We define witness sets as follows. Let $f : \mathbb{C}^N \rightarrow \mathbb{C}^n$ define a system $f(\mathbf{x}) = \mathbf{0}$ of n polynomial equations $f = \{f_1, f_2, \ldots, f_n\}$ in N unknowns $\mathbf{x} = (x_1, x_2, \ldots, x_N)$. We denote the solution set of f by

$$V(f) = \{ \ \mathbf{x} \in \mathbb{C}^N \ | \ f(\mathbf{x}) = \mathbf{0} \ \}. \tag{3}$$

This is a reduced[1] algebraic set. Suppose $X \subset V(f) \subset \mathbb{C}^N$ is a pure dimensional[2] algebraic set of dimension i and degree d_X. Then, a witness set for X is a data structure consisting of the system f, a generic linear space $L \subset \mathbb{C}^N$ of codimension i, and the set of d_X points $X \cap L$.

If X is not pure dimensional, then a witness set for X breaks up into a list of witness sets, one for each dimension. In our work, we generally ignore multiplicities, so when a polynomial system has a nonreduced solution component, we compute a witness set for the reduction of the component. Just as X has a unique decomposition into irreducible components, a witness set for X has a decomposition into the corresponding irreducible witness sets, represented by a partition of the witness set representation for X. We call this a *numerical irreducible decomposition* of X.

The irreducible decomposition of the solution set Z in (2) is represented by

$$[W_2, W_1, W_0] = [[W_{21}], [W_{11}, W_{12}, W_{13}, W_{14}], [W_{01}]], \tag{4}$$

where the W_i are witness sets for pure dimensional components, of dimension i, partitioned into witness sets W_{ij}'s corresponding to the irreducible components of Z. In particular:

[1] "Reduced" means the set occurs with multiplicity one, we ignore multiplicities > 1 in this paper.
[2] "Pure dimensional" (or "equidimensional") means all components of the set have the same dimension.

1. W_{21} contains two points on the sphere, cut out by a random line,
2. W_{11} contains one point on the line $(x = 0.5, z = 0.5^3)$, cut out by a random plane,
3. W_{12} contains one point on the line $(x = \sqrt{0.5}, y = 0.5)$, cut out by a random plane,
4. W_{13} contains one point on the line $(x = -\sqrt{0.5}, y = 0.5)$, cut out by a random plane,
5. W_{14} contains three points on the twisted cubic, cut out by a random plane,
6. W_{01} is just the point $(x = 0.5, y = 0.5, z = 0.5)$.

Applying the formal definition, the witness sets W_{ij} consist of witness points $\mathbf{w} = \{i, f, L, \mathbf{x}\}$, for $\mathbf{x} \in Z_{ij} \cap L$, where L is a random linear subspace of codimension i (in this case, of dimension $3 - i$). Moreover, observe $\#W_{ij} = \deg(Z_{ij}) = \#(Z_{ij} \cap L)$.

2.3 Related Work: Geometric Resolutions

Witness sets are slices by generic linear spaces of complementary dimension. Linear space sections have been studied in algebraic geometry since the nineteenth century, e.g., see [1]. There has also been considerable modern interest in commutative algebra in the properties of slices by generic linear spaces of complementary dimension, e.g., [3, 8].

Such linear slices are used in symbolic calculations under the name of *lifting fibers* which occur in a *geometric resolution* of polynomial system. The geometric resolution is a symbolic analogue to a numerical irreducible decomposition and was introduced in the symbolic community independently of the introduction in the numerical community. The origins of the numerical development go back to [23], while [4] appears to be foundational for the symbolic approach. As indicated in the title of [4], the dimension and the isolated points of a variety can be computed in polynomial time. Avoiding multivariate polynomials to describe positive dimensional solution sets and using straight-line programs, the notion of geometric resolution was introduced in [6] and lifting fibers were further developed in [5], [7], and [12], leading to a "Gröbner free algorithm" to solve polynomial systems. The numerical irreducible decomposition was introduced in [15].

Table 1 lists a dictionary relating symbolic lifting fibers and numeric witness sets for representing a pure dimensional solution set V of dimension r. Let us first look at the similarities between the two data structures: by random coordinate changes followed by a specialization of r variables or by adding r linear equations with random coefficients, we obtain as many generic points as the degree of the set V. (To emphasize the correspondences in the table, we have defined witness sets as the intersection with extrinsically-specified hyperplanes, $A\mathbf{x} = \mathbf{c}$, $\mathrm{rank}(A) = r$, but for efficiency, we may work instead with intrinsically-specified linear spaces, $\mathbf{x} = B\mathbf{y} + b$, $\mathbf{y} \in \mathbb{C}^{n-r}$, see [22, 23].) For almost all choices of random numbers, the lifting fibers and witness sets consist of $\deg(V)$ many regular points.

A major difference between the two data structures lies in the choice of number field. When working symbolically, exact arithmetic with rational numbers is used, whereas numerically, the calculations are performed in complex floating-point arithmetic. Another

Generic Points on a Pure Dimensional Solution Set V	
Symbolic: lifting fiber $\pi^{-1}(\mathbf{p})$	Numeric: witness set $W(\mathbf{c})$
computational field k: numbers in $\mathbb{Q}$ (or in a finite extension) field operations done *symbolically*	*numeric* field $\mathbb{C}$: floating point complex numbers with machine arithmetic
With a *symbolic* coordinate change we bring V to Noether position: replace $\mathbf{x}$ by $M\mathbf{y}$, $M \in k^{n \times n}$	We slice V *numerically* with some randomly chosen hyperplanes: $A\mathbf{x} = \mathbf{c}$, $A \in \mathbb{C}^{r \times n}$, $\mathbf{c} \in \mathbb{C}^r$, $\mathrm{rank}(A) = r$
choose M for coordinate change	*choose* A for slicing hyperplanes
$\dim V = r$: specialize r free variables	$\dim V = r$: cut with r hyperplanes
$\pi^{-1}(\mathbf{p}) = \{ \mathbf{y} \in \mathbb{C}^n \mid f(\mathbf{y}) = \mathbf{0}$ and $y_1 = p_1, \ldots, y_r = p_r \}$	$W(\mathbf{c}) = \{ \mathbf{x} \in \mathbb{C}^n \mid$ $f(\mathbf{x}) = \mathbf{0}$ and $A\mathbf{x} = \mathbf{c} \}$
choice of values $\mathbf{p} = (p_1, p_2, \ldots, p_r)$ for free variables $(y_1, y_2, \ldots, y_r)$ such that the fiber $\pi^{-1}(\mathbf{p})$ is finite	*choice* of r constants $\mathbf{c} = (c_1, c_2, \ldots, c_r)$ so that $\begin{cases} f(\mathbf{x}) = \mathbf{0} \\ A\mathbf{x} = \mathbf{c} \end{cases}$ has isolated solutions
for almost all $\mathbf{p} \in k^r$: $\pi^{-1}(\mathbf{p})$ consists of $\deg V$ smooth points	*for almost all* $\mathbf{c} \in \mathbb{C}^r$: $W(\mathbf{c})$ consists of $\deg V$ smooth points
where *for almost all* means except for a proper algebraic subset of bad choices	

Table 1: A dictionary between lifting fibers and witness sets.

significant difference is the form of the final answer. Symbolic solutions are given implicitly by a new set of algebraic relations having a specified form, for example, the result might be represented as a polynomial in one variable, say $p(u) \in k[u]$, such that the points in the lifting fiber are given by $\mathbf{x} = \mathbf{S}(u)$ for some straight-line program $\mathbf{S}(u)$ evaluated at the $\deg(p)$ roots of $p(u)$ (see [5]). The output of the symbolic algorithm is $p(u)$ and $\mathbf{S}(u)$. In contrast, a numerical witness set consists of a list of floating point approximations to the witness points. While mathematically, lifting fibers and witness sets are geometrically equivalent descriptions of a pure dimensional solution set, in practice, the choice of number field and the form of the representation of the solution dramatically affect the kind of algorithms that are used and the types of applications that can be solved.

2.4 Embeddings and Cascades of Homotopies

A witness superset $\widehat{W}_k$ for the pure k-dimensional part X_k of X is a set in $X \cap L$, which contains $W_k := X_k \cap L$ for a generic linear space L of codimension k. The set of "junk points" in $\widehat{W}_k$ is the set $\widehat{W}_k \setminus W_k$, which lies in $(\cup_{j>k} X_j) \cap L$.

The computation of a numerical irreducible decomposition for X runs in three stages:

1. Computation of a *witness superset* $\widehat{W}$ consisting of witness supersets $\widehat{W}_k$ for each dimension $k = 0, 1, 2, \ldots, N$.

2. Removal of junk points from $\widehat{W}$ to get a witness set W for X.

3. Decomposition of W into its irreducible components by partitioning each witness set W_k into witness sets corresponding to the irreducible components of X_k.

Up to this point, we have used the dimension of a component as the subscript for its witness set, but in the algorithms that follow, it will be more convenient to use codimension. The original algorithm for constructing witness supersets was given in [23]. A more efficient cascade algorithm for this was given in [14] by means of an embedding theorem.

In [21], we showed how to carry out the generalization of [14] to solve a system of polynomials on a pure N-dimensional algebraic set $Z \subset \mathbb{C}^m$. In the same paper, we used this capability to address the situation where we have two polynomial systems f and g on $\mathbb{C}^N$ and we wish to describe the irreducible decompositions of $A \cap B$ where $A \in \mathbb{C}^N$ is an irreducible component of $V(f)$ and $B \in \mathbb{C}^N$ is an irreducible component of $V(g)$. We call the resulting algorithm a *diagonal homotopy*, because it works by decomposing the diagonal system $\mathbf{u} - \mathbf{v} = \mathbf{0}$ on $Z = A \times B$, where $(\mathbf{u}, \mathbf{v}) \in \mathbb{C}^{2N}$. In [22], we rewrote the homotopies "intrinsically," which means that the linear slicing subspaces are not described explicitly by linear equations vanishing on them, but rather by linear parameterizations. (Note that intrinsic forms were first used in a substantial way to deal with numerical homotopies of parameterized linear spaces in [10], see also [11].) This has always been allowed, even in [23], but [22] showed how to do so consistently through the cascade down dimensions of the diagonal homotopy, thereby increasing efficiency by using fewer variables.

The subsequent steps of removing junk and decomposing the witness sets into irreducible pieces have been studied in [15, 16, 17, 18]. These methods presume the capability to track witness points on a component as the linear slicing space is varied continuously. This is straightforward for reduced solution components, but the case of nonreduced components, treated in [19], is more difficult. An extended discussion of the basic theory may be found in [24].

In this paper, we use multiple applications of the diagonal homotopy to numerically compute the irreducible decomposition of $A \cap B$ for general algebraic sets A and B, without the restriction that they be irreducible. At first blush, this may seem an incremental advance, basically consisting of organizing the requisite bookkeeping without introducing any significantly new theoretical constructs. However, this approach becomes particularly interesting when it is applied "equation by equation," that is, when we compute the irreducible decomposition of $V(f)$ for a system $f = \{f_1, f_2, \ldots, f_n\}$ by systematically computing $V(f_1)$, then $A_1 \cap V(f_2)$ for A_1 a component of $V(f_1)$, then $A_2 \cap V(f_3)$ for A_2 a component of $A_1 \cap V(f_2)$, etc. In this way, we incrementally build up the irreducible decomposition one equation at a time, by intersecting the associated hypersurface with all the solution components of the preceding equations. The main impact is that the elimination of junk points and degenerate solutions at early stages in the computation streamlines the subsequent stages. Even though we use only the total degree of the equations—not multihomogeneous degrees or Newton polytopes—the approach is surprisingly effective for finding isolated solutions.

3 Application of Diagonal Homotopies

In this section, we define our new algorithms by means of two flowcharts, one for solving subsystem by subsystem, and one that specializes the first one to solving equation by equation. We then briefly outline simplifications that apply in the case that only the nonsingular solutions are wanted. First, though, we summarize the notation used in the definition of the algorithms.

3.1 Symbols used in the Algorithms

A witness set W for a pure i-dimensional component X in $V(f)$ is of the form $W = \{i, f, L, \mathcal{X}\}$, where L is the linear subspace that cuts out the $\deg X$ points $\mathcal{X} = X \cap L$. In the following algorithm, when we speak of a *witness point* $\mathbf{w} \in W$, it means that $\mathbf{w} = \{i, f, L, \mathbf{x}\}$ for some $\mathbf{x} \in \mathcal{X}$. For such a $\mathbf{w}$ and for g a polynomial (system) on $\mathbb{C}^N$, we use the shorthand $g(\mathbf{w})$ to mean $g(\mathbf{x})$, for $\mathbf{x} \in \mathbf{w}$.

In analogy to $V(f)$, which acts on a polynomial system, we introduce the operator $\mathcal{V}(W)$, which means the solution component represented by the witness set W. We also use the same symbol operating on a single witness point $\mathbf{w} = \{i, f, L, \mathbf{x}\}$, in which case $\mathcal{V}(\mathbf{w})$ means the irreducible component of $V(f)$ on which point $\mathbf{x}$ lies. This is consistent in that $\mathcal{V}(W)$ is the union of $\mathcal{V}(\mathbf{w})$ for all $\mathbf{w} \in W$.

Another notational convenience is the operator $\mathcal{W}(A)$, which gives a witness set for an algebraic set A. This is not unique, as it depends on the choice of the linear subspaces that slice out the witness points. However, any two witness sets $W_1, W_2 \in \mathcal{W}(A)$ are equivalent under a homotopy that smoothly moves from one set of slicing subspaces to the other, avoiding a proper algebraic subset of the associated Grassmannian spaces, where witness points diverge or cross. That is, we have $\mathcal{V}(\mathcal{W}(A)) = A$ and $\mathcal{W}(\mathcal{V}(W)) \equiv W$, where the equivalence in the second expression is under homotopy continuation between linear subspaces.

The output of our algorithm is a collection of witness sets W_i, $i = 1, 2, \ldots, N$, where W_i is a witness set for the pure *codimension* i component of $V(f)$. (This breaks from our usual convention of subscripting by dimension, but for this algorithm, the codimension is more convenient.) Breaking W_i into irreducible pieces is a post-processing task, done by techniques described in [15, 17, 18], which will not be described here.

The algorithm allows the specification of an algebraic set $Q \in \mathbb{C}^N$ that we wish to ignore. That is, we drop from the output any components that are contained in Q, yielding witness sets for $V(f_1, f_2, \ldots, f_n) \in \mathbb{C}^N \setminus Q$. Set Q can be specified as a collection of polynomials defining it or as a witness point set.

For convenience, we list again the operators used in our notation, as follows:

$V(f)$ The solution set of $f(x) = 0$.

$\mathcal{W}(A)$ A witness set for an algebraic set A, multiplicities ignored, as always.

$\mathcal{V}(W)$ The solution component represented by witness set W.

$\mathcal{V}(\mathbf{w})$ The irreducible component of $V(f)$ on which witness point $\mathbf{w} \in \mathcal{W}(V(f))$ lies.

3.2 Solving Subsystem by Subsystem

In this section, we describe how the diagonal homotopy can be employed to generate a witness set $W = \mathcal{W}(V(f^A, f^B) \setminus Q)$, given witness sets W^A for $A = V(f^A) \setminus Q$, and W^B for $B = V(f^B) \setminus Q$. Let us denote this operation as $W = \mathbf{SysBySys}(A, B; Q)$. Moreover, suppose $\mathbf{Witness}(f; Q)$ computes a witness set $\mathcal{W}(V(f) \setminus Q)$ by any means available, such as by working on the entire system f as in our previous works, [14, 23], with junk points removed but not necessarily decomposing the sets into irreducibles. With these two operations in hand, one can approach the solution of any large system of polynomials in stages. For example, suppose $f = \{f^A, f^B, f^C\}$ is a system of polynomials composed of three subsystems, f^A, f^B, and f^C, each of which is a collection of one or more polynomials. The computation of a witness set $W = \mathcal{W}(V(f) \setminus Q)$ can be accomplished as

$$\begin{aligned} W^A &= \mathbf{Witness}(f^A; Q), \quad W^B = \mathbf{Witness}(f^B; Q) \\ W^C &= \mathbf{Witness}(f^C; Q), \quad W^{AB} = \mathbf{SysBySys}(W^A, W^B; Q), \\ W &= \mathbf{SysBySys}(W^{AB}, W^C; Q). \end{aligned}$$

This generalizes in an obvious way to any number of subsystems. Although we could compute $W = \mathbf{Witness}(f; Q)$ by directly working on the whole system f in one stage, there can be advantages to breaking the computation into smaller stages.

The diagonal homotopy as presented in [21] applies to computing $A \cap B$ only when A and B are each irreducible. To implement **SysBySys**, we need to handle sets that have more than one irreducible piece. In simplest terms, the removal of the requirement of irreducibility merely entails looping through all pairings of the irreducible pieces of A and B, followed by filtering to remove from the output any set that is contained inside another set in the output, or if two sets are equal, to eliminate the duplication. In addition to this, however, we would like to be able to proceed without first decomposing A and B into irreducibles. With a bit of attention to the details, this can be arranged.

Figure 1 gives a flowchart for algorithm **SysBySys**. For this to be valid as shown, we require that the linear subspaces for slicing out witness sets are chosen once and for all and used in all the runs of **Witness** and **SysBySys**. In other words, the slicing subspaces for W^A and W^B at the top of the algorithm must be the same as each other and as the output W^{AB}. This ensures that witness sets from one stage can, under certain

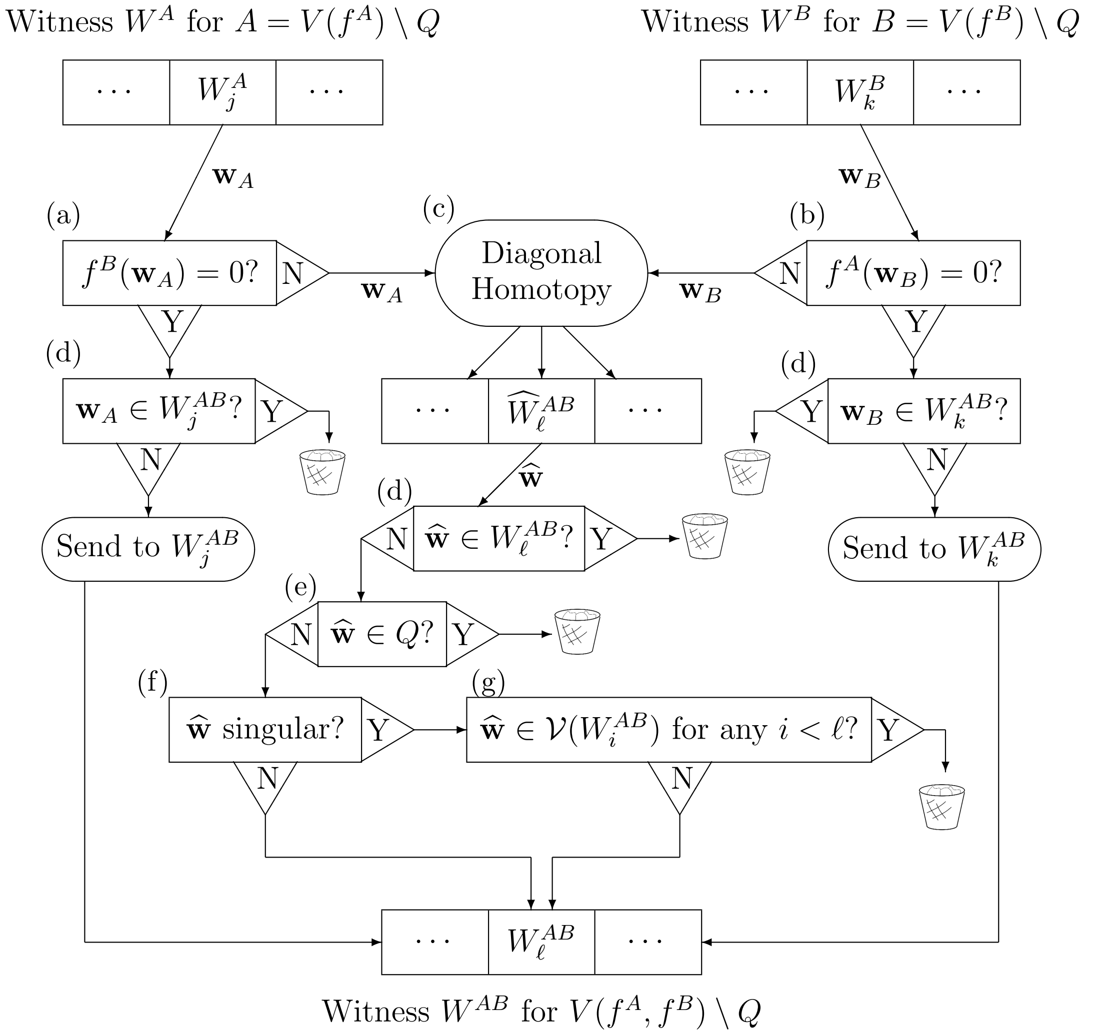

Figure 1: Subsystem-by-subsystem generation of witness sets for $V(f^A, f^B) \setminus Q$.

circumstances, pass directly through to the next stage. Otherwise, a continuation step would need to be inserted to move from one slicing subspace to another.

The setup of a diagonal homotopy to intersect two irreducibles $A \subset \mathbb{C}^N$ and $B \subset \mathbb{C}^N$ involves the selection of certain random elements. We refer to [21, 22] for the full details. All we need to know at present is that in choosing these random elements the only dependence on A and B is their dimensions, $\dim A$ and $\dim B$. If we were to intersect another pair of irreducibles, say $A' \subset \mathbb{C}^N$ and $B' \subset \mathbb{C}^N$, having the same dimensions as the first pair, i.e., $\dim A' = \dim A$ and $\dim B' = \dim B$, then we may use the same random elements for both. In fact, the random choices will be generic for any finite number of intersection pairs. Furthermore, if A and A' are irreducible components of the solution set of the same system of polynomials, f^A, and B and B' are similarly associated to system f^B, then we may use exactly the same diagonal homotopy to compute $A \cap B$ and $A' \cap B'$. The only difference is that in the former case, the start points of the homotopy are pairs of points $(\alpha, \beta) \in \mathcal{W}(A) \times \mathcal{W}(B) \subset \mathbb{C}^{2N}$, while in the latter, the start points come from $\mathcal{W}(A') \times \mathcal{W}(B')$.

To explain this more explicitly, consider that the diagonal homotopy for intersecting A with B works by decomposing $\mathbf{u} - \mathbf{v}$ on $A \times B$. To set up the homotopy, we form the randomized system

$$\mathcal{F}(\mathbf{u}, \mathbf{v}) = \begin{bmatrix} R_A f^A(\mathbf{u}) \\ R_B f^B(\mathbf{v}) \end{bmatrix}, \tag{5}$$

where R_A is a random matrix of size $(N - \dim A) \times \#(f^A)$ and R_B is random of size $(N - \dim B) \times \#(f^B)$. [By $\#(f^A)$ we mean the number of polynomials in system f^A and similarly for $\#(f^B)$.] The key property is that $A \times B$ is an irreducible component of $V(\mathcal{F}(\mathbf{u}, \mathbf{v}))$ for all (R_A, R_B) in a nonzero Zariski open subset of $\mathbb{C}^{(N - \dim A) \times \#(f^A)} \times \mathbb{C}^{(N - \dim B) \times \#(f^B)}$, say R_{AB}. But this property holds for $A' \times B'$ as well, on a possibly different Zariski open subset, say $R_{A'B'}$. But $R_{AB} \cap R_{A'B'}$ is still a nonzero Zariski open subset, that is, almost any choice of (R_A, R_B) is satisfactory for computing both $A \cap B$ and $A' \cap B'$, and by the same logic, for any finite number of such intersecting pairs.

The upshot of this is that if we wish to intersect a pure dimensional set $A = \{A_1, A_2\} \subset V(f^A)$ with a pure dimensional set $B = \{B_1, B_2\} \subset V(f^B)$, where A_1, A_2, B_1, and B_2 are all irreducible, we may form one diagonal homotopy to compute all four intersections $A_i \cap B_j$, $i, j \in \{1, 2\}$, feeding in start point pairs from all four pairings. In short, the algorithm is completely indifferent as to whether A and B are irreducible or not. Of course, it can happen that the same irreducible component of $A \cap B$ can arise from more than one pairing $A_i \cap B_j$, so we will need to take steps to eliminate such duplications.

We are now ready to examine the details of the flowchart in Figure 1 for computing $W^{AB} = \mathcal{W}(V(f^A, f^B) \setminus Q)$ from $W^A = \mathcal{W}(V(f^A) \setminus Q)$ and $W^B = \mathcal{W}(V(f^B) \setminus Q)$. It is assumed that the linear slicing subspaces are the same for W^A, W^B, and W^{AB}. The following items (a)–(g) refer to labels in that chart.

(a) Witness point $\mathbf{w}_A$ is a generic point of the component of $V(f^A)$ on which it lies,

$\mathcal{V}(\mathbf{w}_A)$. Consequently, $f^B(\mathbf{w}_A) = 0$ implies, with probability one, that $\mathcal{V}(\mathbf{w}_A)$ is contained in some component of $V(f^B)$. Moreover, we already know that $\mathbf{w}_A$ is not in any higher dimensional set of A, and therefore it cannot be in any higher dimensional set of C. Accordingly, any point $\mathbf{w}_A$ that passes test (a) is an isolated point in witness superset $\widehat{W}^{AB}$. The containment of $\mathcal{V}(\mathbf{w}_A)$ in B means that the dimension of the set is unchanged by intersection, so if $\mathbf{w}_A$ is drawn from W_j^A, its correct destination is W_j^{AB}.

On the other hand, if $f^B(\mathbf{w}_A) \neq 0$, then $\mathbf{w}_A$ proceeds to the diagonal homotopy as part of the computation of $\mathcal{V}(\mathbf{w}_A) \cap B$.

(b) This is the symmetric operation to (a).

(c) Witness points for components not completely contained in the opposing system are fed to the diagonal homotopy in order to find the intersection of those components. For each combination (a, b), where $a = \dim \mathcal{V}(\mathbf{w}_A)$ and $b = \dim \mathcal{V}(\mathbf{w}_B)$, there is a diagonal homotopy whose random constants are chosen once and for all at the start of the computation.

(d) This test, which appears in three places, makes sure that multiple copies of a witness point do not make it into W^{AB}. Such duplications can arise when A and B have components in common, when different pairs of irreducible components from A and B share a common intersection component, or when some component is nonreduced.

(e) Since a witness point $\widehat{\mathbf{w}}$ is sliced out generically from the irreducible component, $\mathcal{V}(\widehat{\mathbf{w}})$, on which it lies, if $\widehat{\mathbf{w}} \in Q$, then $\mathcal{V}(\widehat{\mathbf{w}}) \subset Q$. We have specified at the start that we wish to ignore such sets, so we throw them out here.

(f) In this test, "singular" means that the Jacobian matrix of partial derivatives for the sliced system that cuts out the witness point is rank deficient. We test this by a singular value decomposition of the matrix. If the point is nonsingular, it must be isolated and so it is clearly a witness point. On the other hand, if it is singular, it might be either a singular isolated point or it might be a junk point that lies on a higher dimensional solution set, so it must be subjected to further testing.

(g) Our current test for whether a singular test point is isolated or not is to check it against all the higher dimensional sets. If it is not in any of these, then it must be an isolated point, and we put it in the appropriate output bin.

In the current state of the art, the test in box (g) is done using homotopy membership tests. This consists of following the paths of the witness points of the higher dimensional set as its linear slicing subspace is moved continuously to a generically disposed one passing through the test point. The test point is in the higher dimensional set if, and only if, at the end of this continuation one of these paths terminates at the test point, see [16]. In the future, it may be possible that a reliable local test, based just on the local behavior of the polynomial system, can be devised that determines if a point is isolated or not. This might substantially reduce the computation required for the test. As it stands, one must test the point against all higher dimensional solution components, and so points reaching box (g) may have to wait there in limbo until all higher dimensional components have been found.

The test (e) for membership in Q would entail a homotopy membership test if Q is given by a witness set. If Q is given as $V(f^Q)$ for some polynomial system f^Q, then the test is merely "$f^Q(\widehat{\mathbf{w}}) = 0$?" We have cast the whole algorithm on $\mathbb{C}^N$, but it would be equivalent to cast it on complex projective space $\mathbb{P}^N$ and use Q as the hyperplane at infinity.

As a cautionary remark, note that the algorithm depends on A and B being complete solution sets of the given polynomial subsystems, excepting the same set Q. Specifically, test (a) is not valid if B is a partial list of components in $V(f^B) \setminus Q$. Indeed, suppose $B' \subset B$ is a partial set of components and we wish to find $A \cap B'$. Then, a point $\mathbf{w}_A$ for which $f^B(\mathbf{w}_a) = 0$ is necessarily in B but not necessarily in B', so it would be unjustified to pass it directly into the output witness set. A similar objection holds for test (b) if A were not the complete solution set.

3.3 Solving Equation by Equation

The equation-by-equation approach to solving a polynomial system is a limiting case of the subsystem-by-subsystem approach, wherein one subsystem is just a single polynomial equation. Accordingly, we begin by computing a witness set X^i for the solution set $V(f_i)$, $i = 1, 2, \ldots, n$ of each individual polynomial. If any polynomial is identically zero, we drop it and decrement n. If any polynomial is constant, we terminate immediately, returning a null result. Otherwise, we find $X^i = (V(f_i) \cap L) \setminus Q$, where L is a 1-dimensional generic affine linear subspace. A linear parameterization of L involves just one variable, so X^i can be found with any method for solving a polynomial in one variable, discarding any points that fall in Q.

Next, we randomly choose the affine linear subspaces that will cut out the witness sets for any lower dimensional components that appear in succeeding intersections.

The algorithm proceeds by setting $W^1 = X^1$ and then computing $W^{k+1} =$ **SysBySys** $(W^k, X^{k+1}; Q)$ for $k = 1, 2, \ldots, n-1$. The output of stage k is a collection of witness sets W_i^{k+1} for i in the range from 1 to $\min(N, k+1)$. (Recall, we are using the codimension for the subscript.) Of course, some of these may be empty, in fact, in the case of a total intersection, only the lowest dimensional one, W_{k+1}^{k+1}, is nontrivial.

In applying the subsystem-by-subsystem method to this special case, we can streamline the flowchart a bit, due to the fact that $V(f_{k+1})$ is a hypersurface. The difference comes in the shortcuts that allow some witness points to avoid the diagonal homotopy.

The first difference is at the output of test (a), which now sends $\mathbf{w}$ directly to the final output without any testing for duplicates. This is valid because we assume that on

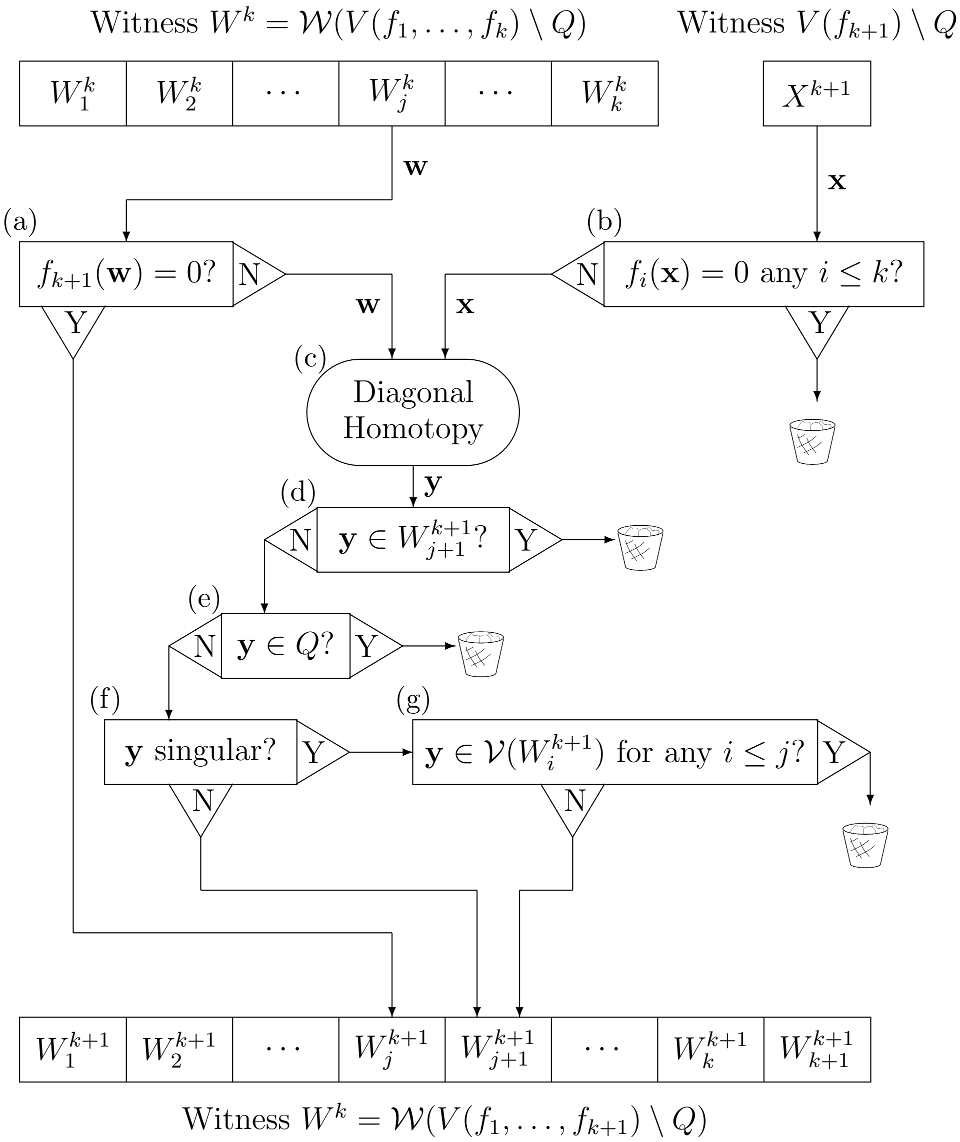

Figure 2: Stage k of equation-by-equation generation of witness sets for $V(f_1,\ldots,f_n) \in \mathbb{C}^N \setminus Q$

input $\mathcal{V}(\mathbf{w})$ is not contained within any higher dimensional component of $\mathcal{V}(W^k)$, and in the intersection with hypersurface $V(f_{k+1})$ that is the only place a duplication could have come from.

On the opposing side, test (b) is now stronger than before. The witness point $\mathbf{x}$ only has to satisfy one polynomial among $f_1, f_2, \ldots, f_k$ in order to receive special treatment. This is because we already threw out any polynomials that are identically zero, so if $f_j(\mathbf{x}) = 0$ it implies that $\mathcal{V}(\mathbf{x})$ is a factor of $V(f_j)$. But the intersection of that factor with all the other $V(f_i)$, $i \neq j$, is already in $V(f_1, f_2, \ldots, f_k)$, so nothing new can come out of intersecting $\mathcal{V}(\mathbf{x})$ with $V(f_1, f_2, \ldots, f_k)$. Accordingly, we may discard $\mathbf{x}$ immediately.

Another small difference from the more general algorithm is that the test for junk at box (g) never has to wait for higher dimensional computations to complete. Points reaching this box cannot be placed in the output witness set until they are checked against all higher dimensional components. When working subsystem by subsystem, the output of the diagonal homotopy in Figure 1 can generate witness points at several different dimensions, so a witness point for a lower dimensional component might be generated before all higher dimensional ones have been found. In the equation by equation case, a witness point either passes test (a) and goes to the output at its original codimension or else its codimension promotes by just one when it goes through the diagonal homotopy. When carrying out the algorithm, we can make sure to process points in increasing order of codimension by obeying just two rules: (1) process all points in W_j^k before proceeding to W_{j+1}^k, and (2) check all points in W_j^k against test (a), depositing those that pass into the output witness set and queueing up those that fail, before feeding any of them through the diagonal homotopy. This ensures that all higher dimensional sets are in place before we begin computations on W_{j+1}^{k+1}. This is not a matter of much importance, but it can simplify coding of the algorithm.

In the test at box (d), we discard duplications of components, including points that appear with multiplicity due to the presence of nonreduced components. However, for the purpose of subsequently breaking the witness set into irreducible components, it can be useful to record the number of times each root appears. By the abstract embedding theorem of [21], points on the same irreducible component must appear the same number times, even though we cannot conclude from this anything about the actual multiplicity of the point as a solution of the system $\{f_1, f_2, \ldots, f_n\}$. Having the points partially partitioned into subsets known to represent distinct components will speed up the decomposition phase.

A final minor point of efficiency is that if $n > N$, we may arrive at stage $k \geq N$ with some zero dimensional components, W_N^k. These do not proceed to the diagonal homotopy: if such a point fails test (a), it is not a solution to system $\{f_1, f_2, \ldots, f_{k+1}\} = 0$, and it is discarded.

3.4 Seeking only Nonsingular Solutions

In the special case that $n \leq N$, we may seek only the multiplicity-one components of codimension n. For $n = N$, this means we seek only the nonsingular solutions of the system. In this case, we discard points that pass test (a), since they give higher dimensional components. Indeed, as discussed in [24, Chapter 12.2.1], for any $j \leq n$, any component Z of $V(f_1, \ldots, f_j)$ has codimension at most j, and if it is less than j, any point on Z will lie on a component of $V(f_1, \ldots, f_n)$ of codimension less than n.

Furthermore, we keep only the points that test (f) finds to be nonsingular and discard the singular ones. To see this, note that if in the end we want points z where the Jacobian matrix of f at z has rank n, then it follows that the rank at z of the Jacobian of $f_1, \ldots, f_j$ for $j \leq n$, must be j. This can greatly reduce the computation for some systems.

In this way, we may use the diagonal homotopy to compute nonsingular roots equation by equation. This performs differently than traditional approaches based on continuation, which solve the entire system all at once. In order to eliminate solution paths leading to infinity, these traditional approaches use multihomogeneous formulations or toric varieties to compactify $\mathbb{C}^N$. But this does not capture other kinds of structure that give rise to positive dimensional components. The equation-by-equation approach has the potential to expose some of these components early on, while the number of intrinsic variables is still small, and achieves efficiency by discarding them at an early stage. However, it does have the disadvantage of proceeding in multiple stages. For example, in the case that all solutions are finite and nonsingular, there is nothing to discard, and the equation-by-equation approach will be less efficient than a one-shot approach. However, many polynomial systems of practical interest have special structures, so the equation-by-equation approach may be commendable. It is too early to tell yet, as our experience applying this new algorithm on practical problems is very limited. Experiences with some simple examples are reported in the next section.

4 Computational Experiments

The diagonal homotopies are implemented in the software package PHCpack [25]. See [20] for a description of a recent upgrade of this package to deal with positive dimensional solution components.

4.1 An illustrative example

The illustrative example (see Eq. 1 for the system) illustrates the gains made by our new solver. While our previous sequence of homotopies needed 197 paths to find all candidate witness points, the new approach shown in Figure 3 tracks just 13 paths. Many of the paths take shortcuts around the diagonal homotopies, and five paths that diverge to infinity in the first diagonal homotopy need no further consideration. It happens that none of the

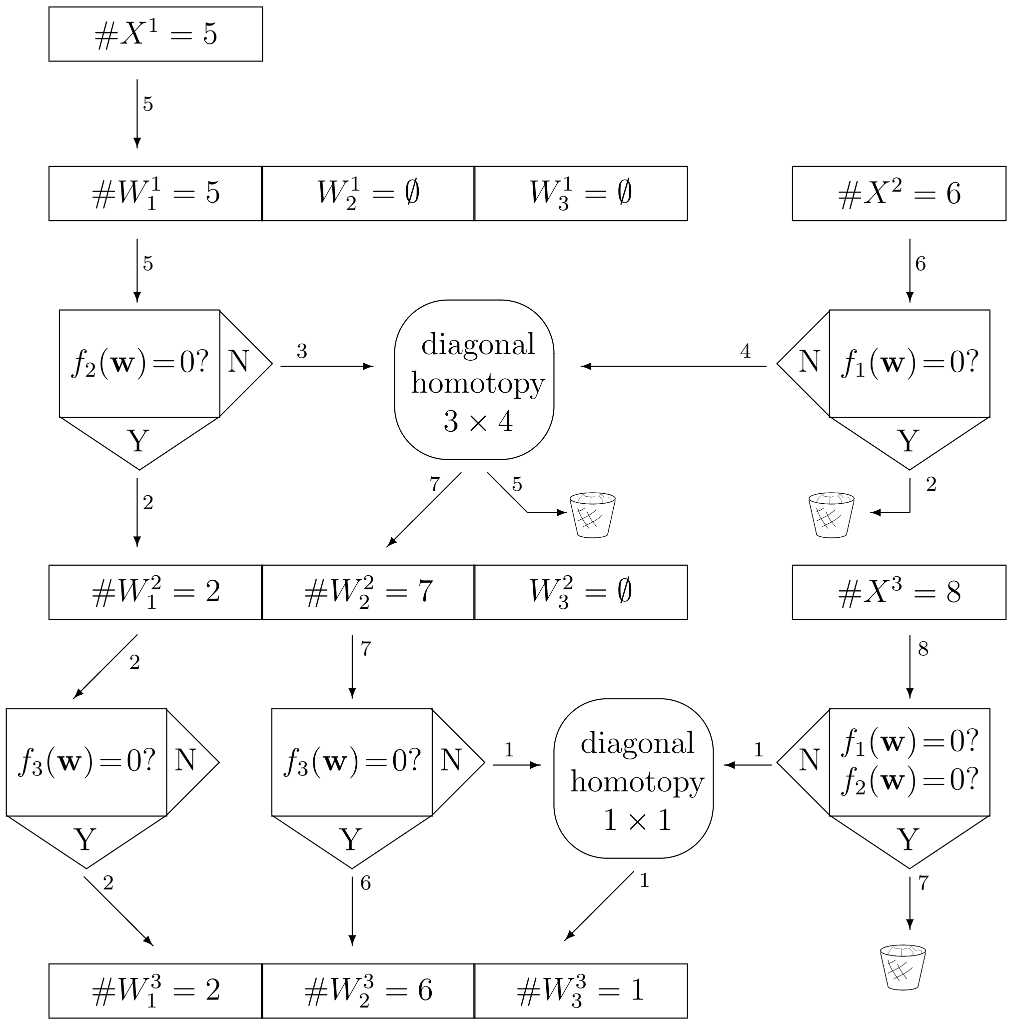

Figure 3: Flowchart for the illustrative example.

witness points generated by the diagonal homotopies is singular, so there is no need for membership testing.

On a 2.4Ghz Linux workstation, our previous approach [14] requires a total of 43.3 cpu seconds (39.9 cpu seconds for solving the top dimensional embedding and 3.4 cpu seconds to run the cascade of homotopies to find all candidate witness points). Our new approach takes slightly less than a second of cpu time. So for this example our new solver is 40 times faster.

4.2 Adjacent Minors of a General 2-by-9 Matrix

In an application from algebraic statistics [2] (see also [9] for methods dedicated for these type of ideals) one considers all adjacent minors of a general matrix. For instance, consider this general 2-by-9 matrix:

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} & x_{17} & x_{18} & x_{19} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} & x_{27} & x_{28} & x_{29} \end{bmatrix}$$

Two minors are adjacent if they share one neighboring column. Taking all adjacent minors from this general 2-by-9 matrix gives 8 quadrics in 18 unknowns. This defines a 10-dimensional surface, of degree 256.

We include this example to illustrate that the flow of timings is typical as in Table 2. Although we execute many homotopies, most of the work occurs in the last stage, because both the number of paths and the number of variables increases at each stage. We are using the intrinsic method of [22] to reduce the number of variables. With the older extrinsic method of [21], the total cpu time increases five-fold from 104s to 502s.

stage	#paths			time/path	time
1	4	=	2×2	0.03s	0.11s
2	8	=	4×2	0.05s	0.41s
3	16	=	8×2	0.10s	1.61s
4	32	=	16×2	0.12s	3.75s
5	64	=	32×2	0.19s	12.41s
6	128	=	64×2	0.27s	34.89s
7	256	=	128×2	0.41s	104.22s
				total user cpu time	157.56s

Table 2: Timings on Apple PowerBook G4 1GHz for the 2×9 adjacent minors, a system of 8 quadrics in 18 unknowns.

4.3 A General 6-by-6 Eigenvalue Problem

Consider $f(\mathbf{x}, \lambda) = \lambda\mathbf{x} - A\mathbf{x} = \mathbf{0}$, where $A \in \mathbb{C}^{6\times 6}$, A is a random matrix. These 6 equations in 7 unknowns define a curve of degree 7, far less than what may be expected from the application of Bézout's theorem: $2^6 = 64$. Regarded as a polynomial system on $\mathbb{C}^7$, the solution set consists of seven lines, six of which are eigenvalue-eigenvector pairs while the seventh is the trivial line $\mathbf{x} = \mathbf{0}$.

Clearly, as a matter of practical computation, one would employ an off-the-shelf eigenvalue routine to solve this problem efficiently. Even with continuation, we could cast the problem on $\mathbb{P}^1 \times \mathbb{P}^6$ and solve it with a seven-path two-homogeneous formulation. However, for the sake of illustration, let us consider how the equation-by-equation approach performs, keeping in mind that the only information we use about the structure of the system is the degree of each equation. That is, we treat it just like any other system of 6 quadratics in 7 variables and let the equation-by-equation procedure numerically discover its special structure.

In a direct approach of solving the system in one total-degree homotopy, adding one generic linear equation to slice out an isolated point on each solution line, we would have 64 paths of which 57 diverge. This does not even consider the work that would be needed if we wanted to rigorously check for higher dimensional solution sets.

Table 3 shows the evolution of the number of solution paths tracked in each stage of the equation-by-equation approach. The size of each initial witness set is $\#(X^i) = 2$, so each new stage tracks two paths for every convergent path in the previous stage. If the quadratics were general, this would build up exponentially to 64 paths to track in the final stage, but the special structure of the eigenvalue equations causes there to be only $i + 2$ solutions at the end of stage i. Accordingly, there are only 12 paths to track in the final, most expensive stage, and only 40 paths tracked altogether. The seven convergent paths in the final stage give one witness point on each of the seven solution lines.

stage in solver	1	2	3	4	5	total
#paths tracked	4	6	8	10	12	40
#divergent paths	1	2	3	4	5	15
#convergent paths	3	4	5	6	7	25

Table 3: Number of convergent and divergent paths on a general 6-by-6 eigenvalue problem.

5 Conclusions

The recent invention of the diagonal homotopy allows one to compute intersections between algebraic sets represented numerically by witness sets. This opens up many new possibilities for ways to manipulate algebraic sets numerically. In particular, one may solve a system of polynomial equations by first solving subsets of the equations and then intersecting the results. We have presented a subsystem-by-subsystem algorithm based on this idea, which when carried to extreme gives an equation-by-equation algorithm. The approach can generate witness sets for all the solution components of a system, or it can be specialized to only seek the nonsingular solutions at the lowest dimension. Applying this latter form to a system of N equations in N variables, we come full circle in the sense that we are using methods developed to deal with higher dimensional solution sets as a means of finding just the isolated solutions.

Experiments with a few simple systems indicates that the method can be very effective. Using only the total degrees of the equations, the method numerically discovers some of their inherent structure in the early stages of the computation. These early stages are relatively cheap and they can sometimes eliminate much of the computation that would otherwise be incurred in the final stages.

In future work, we plan to exercise the approach on more challenging problems, especially ones where the equations have interrelationships that are not easily revealed just by examining the monomials that appear. Multihomogeneous homotopies and polyhedral homotopies are only able to take advantage of that sort of structure, while the equation-by-equation approach can reveal structure encoded in the coefficients of the polynomials. One avenue of further research could be to seek a formulation that uses multihomogeneous homotopies or polyhedral homotopies in an equation-by-equation style to get the best of both worlds.

Acknowledgement

We acknowledge the assistance of Anton Leykin in constructing the dictionary shown in Table 1. We also thank the referees for their corrections and helpful comments.

References

[1] Beltrametti, M. and Sommese, A.J. *The adjunction theory of complex projective varieties.* Expositions in Mathematics, 16, Walter De Gruyter, Berlin, 1995.

[2] Diaconis, P., Eisenbud, D., and Sturmfels, B. Lattice Walks and Primary Decomposition. In *Mathematical Essays in Honor of Gian-Carlo Rota*, edited by B.E. Sagan, R.P. Stanley, volume 161 of *Progress in Mathematics*, pages 173–193. Birkhäuser, 1998.

[3] Geramita, A.V. *The Curves Seminar at Queen's.* Vol. IX. Proceedings of the seminar held at Queen's University, Kingston, Ontario, 1992–1993. Queen's University Press, Ontario, Canada, 1993.

[4] M. Giusti and J. Heintz. La détermination de la dimension et des points isolées d'une variété algébrique peuvent s'effectuer en temps polynomial. In *Computational Algebraic Geometry and Commutative Algebra, Cortona 1991*, edited by D. Eisenbud and L. Robbiano, Symposia Mathematica XXXIV, pages 216–256. Cambridge UP, 1993.

[5] M. Giusti and J. Heintz. Kronecker's smart, little black boxes. In *Foundations of Computational Mathematics* edited by DeVore, R.A. and Iserles, A. and Süli, E., volume 284 of *London Mathematical Society Lecture Note Series*, pages 69–104. Cambridge University Press, 2001.

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

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

[8] J. Harris, *Curves in projective space*, with the collaboration of D. Eisenbud, Université de Montreal Press, Montreal (Québec), Canada, 1982.

[9] S. Hosten and J. Shapiro. Primary Decomposition of Lattice Basis Ideals. *J. of Symbolic Computation* 29(4&5): 625–639, 2000.

[10] B. Huber, F. Sottile, and B. Sturmfels. Numerical Schubert calculus. *J. of Symbolic Computation* 26(6):767–788, 1998.

[11] B. Huber and J. Verschelde. Pieri homotopies for problems in enumerative geometry applied to pole placement in linear systems control. *SIAM J. Control Optim.* 38(4):1265–1287, 2000.

[12] G. Lecerf. Computing the equidimensional decomposition of an algebraic closed set by means of lifting fibers. *Journal of Complexity* 19(4):564–596, 2003.

[13] T.Y. Li. Numerical solution of polynomial systems by homotopy continuation methods. In *Handbook of Numerical Analysis. Volume XI. Special Volume: Foundations of Computational Mathematics*, edited by F. Cucker, pages 209–304. North-Holland, 2003.

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

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

[16] A.J. Sommese, J. Verschelde, and C.W. Wampler. Numerical irreducible decomposition using projections from points on the components. In *Symbolic Computation: Solving Equations in Algebra, Geometry, and Engineering*, volume 286 of *Contemporary Mathematics*, edited by E.L. Green, S. Hoşten, R.C. Laubenbacher, and V. Powers, pages 37–51. AMS 2001.

[17] A.J. Sommese, J. Verschelde, and C.W. Wampler. Using monodromy to decompose solution sets of polynomial systems into irreducible components. In *Application of Algebraic Geometry to Coding Theory, Physics and Computation*, edited by C.

Ciliberto, F. Hirzebruch, R. Miranda, and M. Teicher. Proceedings of a NATO Conference, February 25 - March 1, 2001, Eilat, Israel, pages 297–315, Kluwer Academic Publishers.

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

[19] A.J. Sommese, J. Verschelde, and C.W. Wampler. A method for tracking singular paths with application to the numerical irreducible decomposition. In *Algebraic Geometry, a Volume in Memory of Paolo Francia*, edited by M.C. Beltrametti, F. Catanese, C. Ciliberto, A. Lanteri, C. Pedrini, pages 329–345, W. de Gruyter, 2002.

[20] A.J. Sommese, J. Verschelde, and C.W. Wampler. Numerical irreducible decomposition using PHCpack. In *Algebra, Geometry, and Software Systems*, edited by M. Joswig and N. Takayama, pages 109–130, Springer-Verlag 2003.

[21] A.J. Sommese, J. Verschelde, and C.W. Wampler. Homotopies for Intersecting Solution Components of Polynomial Systems. *SIAM J. Numer. Anal.* 42(4):1552–1571, 2004.

[22] A.J. Sommese, J. Verschelde, and C.W. Wampler. An intrinsic homotopy for intersecting algebraic varieties. *Journal of Complexity*, 21:593–608, 2005.

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

[24] A.J. Sommese and C.W. Wampler. *The Numerical solution of systems of polynomials arising in engineering and science.* World Scientific Press, Singapore, 2005.

[25] J. Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software* 25(2): 251–276, 1999. Software available at `http://www.math.uic.edu/~jan`.