Numerical Irreducible Decomposition using Projections from Points on the Components

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

Abstract. To classify positive dimensional solution components of a polynomial system, we construct polynomials interpolating points sampled from each component. In previous work, points on an i-dimensional component were linearly projected onto a generically chosen $(i+1)$-dimensional subspace. In this paper, we present two improvements. First, we reduce the dimensionality of the ambient space by determining the linear span of the component and restricting to it. Second, if the dimension of the linear span is greater than $i+1$, we use a less generic projection that leads to interpolating polynomials of lower degree, thus reducing the number of samples needed. While this more efficient approach still guarantees — with probability one — the correct determination of the degree of each component, the mere evaluation of an interpolating polynomial no longer certifies the membership of a point to that component. We present an additional numerical test that certifies membership in this new situation. We illustrate the performance of our new approach on some well-known test systems.

1. Introduction

Polynomial systems occur in various fields of science and engineering. In many applications we encounter positive dimensional components of solutions. The problem then is to decompose the solution set into irreducible components, identifying for each component the dimension and the degree.

This manuscript is the fourth paper in a sequence that started with *Numerical Algebraic Geometry* [**15**], followed by [**13, 14**]. Our aim is to efficiently solve the above problem using numerical homotopy continuation [**1**], [**7**], [**8**]. An efficient homotopy to find generic points on each component was given in [**13**], and in [**14**] we presented a general algorithm to classify these points into the irreducible components.

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

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

The first author thanks the Duncan Chair of the University of Notre Dame for its support.

The second author acknowledges support from the NSF under Grant DMS - 9804846 at the Department of Mathematics, Michigan State University.

Since this paper departs directly from the results of [**14**], we first sketch how we decompose the solution set into irreducible components by means of an example. Suppose we are given three points as intersection points of a general line with a curve of degree three in the plane. The problem is to detect whether this curve is a genuine cubic, or whether it decomposes into a conic and a line, or even further degenerates into three lines. We proceed as follows. Start at the first given point and produce another sample point on the curve by moving the general intersecting line and tracing the intersection point as the line moves. By repeating this process, any number of generic sample points on the same component can be found. With two points, we construct the equation for the line passing through both. We then obtain a third sample and test if it satisfies this equation. If so, the component is a line, and we proceed to analyze the next component in a similar manner, starting at the second given point. Otherwise, we conduct three more samples, enough to get the equation for a conic passing through five points, and test whether the sixth sample satisfies this quadratic equation. If so, the curve breaks up in a conic and a line, otherwise the curve is a cubic. A similar procedure works for components of any degree and dimension.

It often happens that a component of the solution set lies in a linear subspace smaller than the ambient space. The smallest such subspace is called the span of the component. The first improvement of this paper over its predecessors is to take advantage of the difference between the span and the ambient space. If n general samples of the component are found to span a linear subspace of dimension $s < n - 1$, then we are assured that the entire component lies in this subspace. We can therefore restrict all further computations to this subspace. Since the dominant cost in sampling the component is the solution of linear systems in Newton iterations, we get a reduction in computational cost from $O(N^3)$ to $O(s^3)$, where N and s are the dimensions of the ambient space and the span, respectively.

In [**14**], a key step in handling an i-dimensional component is to linearly project the samples onto a generic $(i+1)$-dimensional subspace, thereby mapping the component to a hypersurface within this space. The hypersurface is described by one equation, whose coefficients are found by interpolation. The second improvement of this paper is to replace the linear projection operator of [**14**] by *central projections*, i.e., projections from points on the component. To project a component onto a general hyperplane H, first pick a general point $\mathbf{x}$ on the component as the center of the projection. Then, the projection of an arbitrary point $\mathbf{y} \neq \mathbf{x}$ is the intersection of the unique line through $\mathbf{x}$ and $\mathbf{y}$ with the hyperplane H. With this projection, the degree of the component drops by one. To project onto a linear subspace of codimension m, one may concatenate central projections from m different central points. This is equivalent to a single simultaneous projection from m distinct central points, determined by the affine linear space generated by the set of points. This is described in greater detail below.

The advantage of using *central projections* is that the interpolation requires fewer samples; for components that span many more dimensions than their inherent dimension, the efficiency gain is considerable. To uniquely determine a hypersurface of degree d in a space of dimension n, at least $(d+n)!/(d!n!)$ sample points are needed. (This includes one extra point for testing.) Since the central projection decreases the degree of the hypersurface, the amount of computation is reduced accordingly. The decrease in degree is equal to $s - (i+1)$, the difference between the dimension of the span of the component and the dimension of the final linear subspace after projection.

There is a downside of using this less generic projection: we can have positive dimensional sets of points not on the component but satisfying the interpolating polynomials. These bad sets cannot be avoided by choosing different central points for the projection. Fortunately, C. Ciliberto came to our rescue, pointing out that for the case of central projection from a single point, results of Segre [**10, 11**], ensure that these bad sets are of a lower dimension than the dimension of the component. Moreover, he and A. Calabri [**3**] showed that, in the case of central projection from a finite set of points, Segre's result still holds true. So in breaking up solution sets we can use these new projections, but there are bad sets, which are mildly enough behaved that we could develop an additional test to certify whether a point belongs to a component.

One important side effect of this new test is that we now have an effective method to check the multiplicity of a point on an irreducible component. Nevertheless, our current implementation still assumes all components are of multiplicity one. How to handle singularities will be the subject of a future paper.

The paper is organized as follows. In the next section we give a mathematical description of our methods. Pseudocode for our new algorithms is given in Section 3. In the last section we compare the performance of our new approach with the algorithms of [**14**].

Acknowledgments. We thank C. Ciliberto for pointing out B. Segre's results, and generalizing them with A. Calabri to cover our situation. We are grateful to C. Ciliberto and C. Keem for pointing out independently the useful consequences of the trisecant lemma. We thank T.-Y. Li for a number of suggestions that have improved this paper. We thank B. Sturmfels for mentioning the systems of adjacent minors to us.

2. Central Projection from Points

Let

$$f(\mathbf{x}) := \begin{bmatrix} f_1(x_1,\dots,x_N) \\ \vdots \\ f_n(x_1,\dots,x_N) \end{bmatrix}, \tag{2.1}$$

be a system of polynomial equations in $\mathbb{C}^N$, where for simplicity we assume that not all of the f_i are identically zero.

Let $Z = \mathbf{V}(f)$ be the union

$$Z := \bigcup_{i=0}^{N-1} Z_i := \bigcup_{i=0}^{N-1} \bigcup_{j\in\mathcal{I}_i} Z_{ij} \tag{2.2}$$

where Z_i is the union of all i-dimensional irreducible components Z_{ij} of Z, and the index sets $\mathcal{I}_i$ are finite and possibly empty. Note that Z is the reduction of the possibly non-reduced algebraic set defined by the equations $f_1,\dots,f_n$.

Let $\widehat{W} = \{\ \widehat{W}_i \mid i = 0,\dots,N-1\ \}$ denote the set of witness points produced by the routine **WitnessGenerate** of [**14**]. Each $\widehat{W}_i$ contains generic points on each i-dimensional component, plus additional junk points which will be filtered out, i.e., points lying on irreducible components of dimension $> i$. In particular, we showed in [**14**] how to do a breakup of each $\widehat{W}_i$ of $\widehat{W}$ as $\widehat{W}_i := (\cup_{j\in\mathcal{I}_i} W_{ij}) \cup J_i$, where

1. W_{ij} consists of $\deg Z_{ij}$ generic points of Z_{ij} each occurring ν_{ij} times, where ν_{ij} is a positive integer. Moreover, $\nu_{ij} \geq \mu_{ij}$, where μ_{ij} is the multiplicity of Z_{ij} in the possibly nonreduced algebraic set $f^{-1}(\mathbf{0})$, and $\nu_{ij} = 1$ if and only if $\mu_{ij} = 1$; and
2. $J_i \subset \cup_{k>i} Z_k$.

This classification is accomplished by the construction of filtering polynomials

$$\{\ p_{ij} \mid 0 \leq i \leq N-1\ ;\ j \in \mathcal{I}_i\ \}. \tag{2.3}$$

The genericity of this set of polynomials comes from the genericity of the linear projections involved in their construction. Letting P denote the product of the $N-1$ spaces of linear projections P_i from $\mathbb{C}^N \to \mathbb{C}^i$ with $1 \leq i \leq N-1$, this genericity amounts to a choice of a general point on P. Given a finite set of points S, it follows with probability one, i.e., for the polynomials p_{ij} dependent on the choice of a point in a nonempty Zariski open set of P, that $\mathbf{s} \in S$ lies on Z_{ij} if and only if $p_{ij}(\mathbf{s}) = 0$.

In this article, we construct the same type of breakup of the $\widehat{W}_i$, but with a possibly lower degree set of filtering polynomials on lower dimensional Euclidean spaces. The time savings of this new method are considerable, but we must give up some of the genericity of the filtering polynomials.

The restricted class of projections we use are described as follows. Given an i-dimensional irreducible variety $X \subset \mathbb{C}^N$, let $\mathrm{Span}(X)$ denote the smallest linear subspace of $\mathbb{C}^N$ containing X, and let $s := \dim \mathrm{Span}(X)$, $i := \dim(X)$. If $s = i$, then $X = \mathrm{Span}(X)$, and we need go no further. If $s = i+1$, then X is a hypersurface in $\mathrm{Span}(X)$, and we interpolate it without projection. In the case $s \geq i+2$, we wish to project onto a linear subspace of dimension $i+1$. To do so, choose $s-i-1$ general points $\{\mathbf{x}_1, \ldots, \mathbf{x}_{s-i-1}\}$ of X as central points of the projection, and pick an $(i+1)$-dimensional affine linear subspace L of $\mathrm{Span}(X)$ with $L \cap \langle \mathbf{x}_1, \ldots, \mathbf{x}_{s-i-1}\rangle = \emptyset$, where $\langle \mathbf{x}_1, \ldots, \mathbf{x}_{s-i-1}\rangle$ denotes the affine linear span of the points $\mathbf{x}_1, \ldots, \mathbf{x}_{s-i-1}$. We take L as the target of the central projection, defined by sending any $\mathbf{y}$ of X not in $\langle \mathbf{x}_1, \ldots, \mathbf{x}_{s-i-1}\rangle$ to the unique intersection of $\langle \mathbf{x}_1, \ldots, \mathbf{x}_{s-i-1}, \mathbf{y}\rangle$ with L.

In the original procedure of [**14**], we use two key facts about linear projections.

A) Given an i-dimensional irreducible variety $X \subset \mathbb{C}^N$ and a general linear projection $\pi : \mathbb{C}^N \to \mathbb{C}^{i+1}$, the restriction π_X of π to X is generically one-to-one on X, i.e., there exists a dense Zariski open set U of $\pi(X)$ such that π gives an isomorphism between $\pi^{-1}(U)$ and U.

B) The intersection of the algebraic sets $\pi^{-1}\left(\overline{\pi(X)}\right)$ as π runs over the set of general linear projections $\pi : \mathbb{C}^N \to \mathbb{C}^{i+1}$ equals X. Here $\overline{Y}$ denotes the closure in projective space of algebraic set Y.

For the class of central projections that we are using in this paper, assertion A is still true, but the second assertion B is false. In what follows, we first prove assertion A for central projections, i.e., that a generic central projection is generically one-to-one. Then we turn to a weaker variant of assertion B that is true for central projections, and show how to adapt the algorithm of [**14**] to give the same decomposition, while using only the this weaker assertion.

The analogue for central projections of assertion A above follows from the classical trisecant lemma. Before we state and prove the analogue, we make a few observations that are useful in passing back and forth between $\mathbb{P}^N$ and $\mathbb{C}^N \subset \mathbb{P}^N$.

1. An r dimensional affine linear space in $\mathbb{C}^N$ closes up to be a linear $\mathbb{P}^r \subset \mathbb{P}^N$, and given a linear $L := \mathbb{P}^r \subset \mathbb{P}^N$, not contained in $\mathbb{P}^N \setminus \mathbb{C}^N$, the set $L \cap \mathbb{C}^N$ is affine linear. Moreover, a general linear $\mathbb{P}^r \subset \mathbb{P}^N$ is not contained in $\mathbb{P}^N \setminus \mathbb{C}^N$.
2. Given an i dimensional variety $X \subset \mathbb{C}^N$, and a general affine linear space L of dimension $N - i$, then it follows that $L \cap X = \overline{L} \cap \overline{X}$, since the $(i-1)$-dimensional set $\overline{X} \setminus X$ is missed by any general $\mathbb{P}^r \subset \mathbb{P}^N$ with $r \leq N - i$.

LEMMA 2.1. *Let $X \subset \mathbb{C}^N$ be an irreducible i-dimensional variety. Assume that X belongs to no affine linear hyperplane $\mathbb{C}^{N-1} \subset \mathbb{C}^N$. For a set K of $k \leq N - i - 1$ general points of X, it follows that the central projection $\pi_K : \overline{X} \to \mathbb{P}^{i+1}$ is generically one-to-one, and in particular, that there is a non-empty Zariski open set $U \subset \pi_K(X)$ such that $\pi_K : \pi_K^{-1}(U) \to U$ is an isomorphism.*

PROOF. This would follow if we knew that the smallest linear space spanned by K and another general point $\mathbf{x}$ of X, has dimension k and meets $\overline{X}$ in $K \cup \{\mathbf{x}\}$. This would follow if we knew that a general linear $L := \mathbb{P}^{N-i}$ meets X (or equivalently $\overline{X}$ by the remarks before the lemma), in a set of cardinality $\deg X$ with the property that

1. no subset of $L \cap X$ of cardinality $k + 2 \leq N - i + 1$ lies on a linear $\mathbb{P}^k$; or equivalently
2. given any $k + 2 \leq N - i + 1$ points of $L \cap X$, the linear span of the points is a $\mathbb{P}^{k+1}$.

Noting [**12**, Theorem 3.44] that, if $\dim X \geq 2$, the intersection of X with a general linear $\mathbb{P}^{N-i+1}$ is an irreducible algebraic curve, we are reduced to the classical trisecant lemma [**2**, General Position Theorem, page 109]. $\square$

One consequence of Lemma 2.1, that follows from [**9**, Theorem 5.11, page 76], is that there is a Zariski open dense set $U \subset X^{s-i-1}$ such that for $(\mathbf{x}_1, \dots, \mathbf{x}_{s-i-1}) \in U$ and any $(i+1)$-dimensional affine linear space L of $\mathbb{C}^s$ disjoint from $\langle \mathbf{x}_1, \dots, \mathbf{x}_{s-i-1} \rangle$, it follows that under the central projection $\pi : \mathbb{C}^s \setminus \langle \mathbf{x}_1, \dots, \mathbf{x}_{s-i-1} \rangle \to L$, associated to $\langle \mathbf{x}_1, \dots, \mathbf{x}_{s-i-1} \rangle$, $\deg(\pi(X)) = \deg X - (s-i-1)$. Note that π is a meromorphic mapping on X, and by $\pi(X)$ is meant the closure of $\pi(X \setminus \langle \mathbf{x}_1, \dots, \mathbf{x}_{s-i-1} \rangle \cap X)$.

As noted above, assertion B, one of two key facts about generic linear projections used in [**14**], is false for the class of central projections, but fortunately the generalization [**3**] of classical results of B. Segre [**10, 11**] show that a slightly weakened variant of assertion B is true for central projections.

To be precise, given an irreducible algebraic set $X \subset \mathbb{C}^N$, let $\mathrm{Cone}(X, \mathbf{x}_1, \dots, \mathbf{x}_{s-i-1})$ be the closure of the set of all affine linear spaces generated by the distinct points $\mathbf{x}_1, \dots, \mathbf{x}_{s-i-1}$ of X, where $s = \dim \mathrm{Span}(X)$, and a point of $\mathbf{y}$ varying over X. Let

$$B(X) := \bigcap_{(\mathbf{x}_1, \dots, \mathbf{x}_{s-i-1}) \in X^{s-i-1} - \Delta} \mathrm{Cone}(X, \mathbf{x}_1, \dots, \mathbf{x}_{s-i-1}), \tag{2.4}$$

where Δ is the set of $(s-i-1)$-tuples of points whose affine linear span is of dimension less than $s-i-2$. It is a consequence of the trisecant lemma [**2**, page 109] that Δ is a proper algebraic subset of X^{s-i-1}.

For the case of cones from a single point $\mathbf{x}_1$, it is a classical result of Segre [**10, 11**] that $B(X) = X \cup X'$ where $\dim X' < \dim X$. We refer to [**3**] for the generalization of Segre's result needed in our algorithm, i.e., that in general, $B(X) = X \cup X'$ where $\dim X' < \dim X$.

Fix an irreducible i-dimensional component Z_{ij} of $V(f)$ and set $B_{ij} := B(Z_{ij}) = Z_{ij} \cup Z'_{ij}$ with $\dim Z'_{ij} < i$. The new filtering polynomials q_{ij}, which are only constructed when $s := \dim \mathrm{Span}(Z_{ij}) \geq i+2$, have the property that, given a finite set of points S, making a generic choice of points $\mathbf{x}_1, \ldots, \mathbf{x}_{s-i-1}$ on Z_{ij}, it follows with probability one that $\mathbf{s} \in S$ lies on $Z_{ij} \cup Z'_{ij}$ if and only if $q_{ij}(\mathbf{s}) = 0$. To make up for the loss of generality, i.e., to decide whether or not $\mathbf{s} \in Z_{ij}$ and not just in $Z_{ij} \cup Z'_{ij}$, we need a simple numerical test. It is based on the following lemma, which summarizes some facts about linear spaces in general position. We will apply it with A being one of the Z_{ij} and $\mathbb{C}^a$ being $\mathrm{Span}(Z_{ij})$.

LEMMA 2.2. *Let A' and A be reduced and irreducible algebraic subsets of $\mathbb{C}^a$ with $\dim A' < \dim A$, e.g., let $\mathrm{Span}(A) = \mathbb{C}^a$, $a \geq \dim A + 2$, and let A' be the algebraic set as above such that $B(A) = A \cup A'$. For $\mathbf{s} \in A \cup A'$, choose a general linear subspace $L \subset \mathbb{C}^a$ of dimension $a - \dim A$ containing $\mathbf{s}$. Then $L \cap A'$ contains at most the point $\mathbf{s}$, and therefore $(A \cup A') \cap L$ consists of either*

1. $\deg A$ *distinct points of A not including $\mathbf{s}$ if $\mathbf{s} \notin A$; or*
2. $\mathbf{s}$ *plus $m < \deg A$ distinct points of A if $\mathbf{s} \in A$. In this case the multiplicity of $\mathbf{s}$ as a point of A is $\deg A - m$.*

To make this into a membership test for $\mathbf{s}$ in the set Z_{ij}, first check membership of $\mathbf{s}$ in $\mathrm{Span}(Z_{ij})$. Next take the general linear space L_{N-i} of [**13**] which meets Z_{ij} in W_{ij}, which consists of a set, $W_{ij,\mathrm{red}}$, of $\deg Z_{ij}$ distinct generic points of Z_{ij} each occurring ν_{ij} times. Note that by induction we have already determined the junk J_i and the breakup $W_i = \cup_{j \in \mathcal{I}_i} W_{ij}$. With probability one, L_{N-i} misses the set $\cup_{k \in \mathcal{I}_i} Z'_{ik}$. Apply homotopy continuation to move the set $W_{ij,\mathrm{red}}$ (equal as a set to $L_{N-i} \cap \mathrm{Span}(Z_{ij})$) within $\mathrm{Span}(Z_{ij})$ to a general linear space $K \subset \mathrm{Span}(Z_{ij})$ of dimension $\dim \mathrm{Span}(Z_{ij}) - i$ containing $\mathbf{s}$. (This step is only implemented numerically in this paper for the components of multiplicity one.) Among the set of points obtained with this continuation, let m denote the number of points in the set, excluding $\mathbf{s}$, on which q_{ij} is zero. If $m < \deg Z_{ij}$, then $\mathbf{s} \in Z_{ij}$, and if $m = \deg Z_{ij}$, then $\mathbf{s} \notin Z_{ij}$. For components of multiplicity one, we will make this construction precise in the second part of the following section.

3. The algorithms

In this section we detail the computation of central projections. Then we give the revised algorithm to compute the filtering polynomials. We end this section with a description of the new membership test.

The computation of a central projection $\pi(\mathbf{y})$ of a point $\mathbf{y} \in X$ ($i = \dim(X) > 0$) may proceed as follows. First, restrict the problem to $\mathrm{Span}(X)$ by a linear change of coordinates. Next, homogenize the problem and work on projective space, hence points are represented by $s+1$ coordinates, where $s = \dim \mathrm{Span}(X)$.

Let the linear subspace L be given by $s-i-1$ linear equations on $\mathbb{P}^s$; that is, $L = \{\mathbf{q} \in \mathbb{P}^s | A\mathbf{q} = \mathbf{0}\}$ for some full-rank matrix $A \in \mathbb{C}^{(s-i-1)\times(s+1)}$. Further, let P be a $(s+1)\times(s-i)$ matrix given as $P = (\mathbf{x}_1 \ \cdots \ \mathbf{x}_{s-i-1} \ \mathbf{y})$, so that the span of the central points $\{\mathbf{x}_1, \dots, \mathbf{x}_{s-i-1}\}$ and $\mathbf{y} \in \mathbb{P}^s$ may be represented as $\langle \mathbf{x}_1, \dots, \mathbf{x}_{s-i-1}, \mathbf{y} \rangle = \{\mathbf{q} \in \mathbb{P}^s | \text{for some } \mathbf{b} \in \mathbb{P}^{s-i-1}, \mathbf{q} = P\mathbf{b}\}$. The intersection of the span and L is given by the solution of $AP\mathbf{b} = \mathbf{0}$, which we denote by $\mathbf{b}_0$. Then the projected point is $\pi(\mathbf{y}) = P\mathbf{b}_0$. It is sufficient to keep only the first $i+1$ coordinates of $\pi(\mathbf{y})$ to determine the irreducible decomposition of X. When A is chosen generically, all of the operations in this procedure are nonsingular with probability one.

As in (2.3), for each Z_{ij}, we construct one filtering polynomial p_{ij} defined as $p_{ij}(\mathbf{x}) = q_{ij}(\pi_{ij}(\mathbf{x}))$, where π_{ij} is the central projection operator for that component. Here, q_{ij} is a polynomial that interpolates projected points of the component. There are three cases, depending on the difference between $s = \dim \mathrm{Span}(Z_{ij})$ and $i = \dim Z_{ij}$:

1. Case $s = i$. Then $Z_{ij} = \mathrm{Span}(Z_{ij})$, and we have a complete description of Z_{ij} as the linear combination of $s+1$ points in Z_{ij}.
2. Case $s = i+1$. Then Z_{ij} is a hypersurface in $\mathrm{Span}(Z_{ij})$, so π_{ij} is just the identity and the projected point $\pi_{ij}(\mathbf{x})$ can be represented by s coordinates.
3. Case $s \geq i+2$. In this case, we use the central projection as described in the previous paragraph to project $\mathbf{x}$ into an $(i+1)$-dimensional linear subspace.

The pseudocode to find the interpolating polynomials is given next. The difference with the **Interpolate** of [**14**] is the restriction to the spanning subspace of the component and the central projection once this restriction has been done. The subroutine $\#\mathrm{monom}(d,n) = \binom{d+n}{n}$ gives the number of monomials of a dense polynomial of degree d in n variables; **Sample** generates new generic points on the component; and **Fit** constructs an interpolating polynomial through the samples.

The procedure **CentralSample** generates new samples on the component which will be used as central points in the projection. Therefore the argument modified by this routine is the projection operator π. For a list S, we use $\pi(S)$ to denote the list of projected points; $S(a, \dots, b)$ collects those points of S on positions ranging from a till b.

ALGORITHM 3.1. $[p, S] = \textbf{Interpolate}(f, \mathbf{x}, i)$

Input: Polynomial system f; Solution point $\mathbf{x}$; Working dimension $i > 0$.
Output: Interpolating polynomial p; Sample points S; Equations for Span(S).

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

restricted:= false;	[*true if restricted to subspace*]
$S := \emptyset$; $d := 1$;	[*start at degree 1*]
$n_\pi := 0$;	[*#central projection points*]
loop	
$m_s := \#\mathrm{monom}(d, i+1) - 1 + k_1 + k_2$;	[*#sample points*]
if restricted	
then $S := S \cup \textbf{Sample}(f_{\mid L}, \mathbf{x}, i, m_s - \#S)$;	[*expand S using $f_{\mid L}$*]
else $S := S \cup \textbf{Sample}(f, \mathbf{x}, i, m_s - \#S)$;	[*expand S using f*]

```
        L := Span(S);                                   [subspace spanned by S]
        if dim L = N or dim L < #S − 1
         then restricted := true;                       [working space becomes L]
              S := S_{|L};                              [restrict S to L]
              n_π := dim L − i − 1;                     [#central projection points]
    end if;
    exit when dim L = i;                                [linear component found]
    if n_π > 0
     then CentralSample(f_L, x, i, n_π, π);             [sample n_π central points]
          n_π := 0;                                     [do this only once]
    end if;
    p := Fit(d, i, π(S(1, ..., #S − k_2)));             [fit degree d polynomial]
    exit when (p(π(S(#S − k_2 + 1, ..., #S))) ≈ 0);     [test k_2 extra points]
    d := d + 1;                                         [no good fit, continue loop]
  end loop.
```

The ≈ 0 in the algorithm above indicates the test succeeds if the absolute value of the evaluation result does not exceed a given tolerance.

Once we know for an i-dimensional component that $\dim \mathrm{Span}(S) = i + k$, then its degree is at least $k + 1$. Therefore, we sample the central points all at once. To get the actual degree of the component, we have to add to the degree of the interpolating polynomial the number of central points.

To illustrate the workings of **Interpolate**, we consider the 7-bar mechanism described in [**14**], which has a one-dimensional solution component of degree six that spans a four-dimensional space. In terms of the algorithm, we have $i = 1$, $s = 4$, and $d = 6$. Also, the ambient space is size $N = 12$. The dimension of the target space for the projection is $i + 1 = 2$, so we use $s - (i + 1) = 2$ central points. So the two improvements of the new algorithm are: (1) after finding that $s = 4$, we restrict all further computation to four dimensions instead of 12, and (2) the degree of the filtering polynomial is reduced from $d = 6$ to $d - (s - (i + 1)) = 4$. To make the algorithm clear, we follow through the steps of **Interpolate**, as follows below. For the purpose of counting samples, we assume the minimum requirements of $k_1 = 0$ (no oversampling for **Fit**) and $k_2 = 1$ (only test one extra point). The inputs to the routine are the system f, a witness point $\mathbf{x}$ that has been confirmed to lie on a component of dimension $i = 1$. (That is, $\mathbf{x} \in W_1$, where W_1 has been extracted from $\widehat{W}_1$ by filtering out the junk J_1.)

init: No restriction, S is empty, $d = 1$, $n_\pi = 0$.

loop: First pass through loop

1. $m_s = 3$, so sample 3 points, making $\#S = 3$.
2. $\dim \mathrm{Span}(S) = 2$, which is equal to $\#S - 1$, so no restriction.
3. Fitting and testing finds that points do not lie on a $d = 1$ curve.
4. Increment to $d = 2$ and loop back.

loop: Second pass through loop

1. $m_s = 6$, so sample 3 more points, making $\#S = 6$.
2. $s = \dim \mathrm{Span}(S) = 4$, which is less than $\#S - 1$, so restrict the problem.
   -: Restrict all further computation to the span.

3. Sample $n_\pi = 2$ central points. Use central projection for all future fits.
4. Fitting and testing finds that points do not lie on a $d = 2$ curve.
5. Increment to $d = 3$ and loop back.

loop: Third pass through loop
1. $m_s = 10$, so sample 4 more points, making $\#S = 10$.
2. Fitting and testing finds that points do not lie on a $d = 3$ curve.
3. Increment to $d = 4$ and loop back.

loop: Fourth pass through loop
1. $m_s = 15$, so sample 5 more points, making $\#S = 15$.
2. Fitting and testing finds that points lie on a $d = 4$ curve.
3. Exit with $d = 4$ and return the fitted polynomial.

The total number of samples, including those for the central projection, equals 17. Without central projections we need 28 samples to construct an interpolating polynomial of degree six on a two-dimensional space. Computation times for this example are reported in the next section.

The algorithm **Interpolate** yields the spanning subspace and — if the component is nonlinear — a filtering polynomial for one component Z_{ij}. To obtain the complete irreducible decomposition of all i-dimensional components Z_i, we apply this algorithm repeatedly to all points $\mathbf{x} \in \widehat{W}_i$ that are not members of a known component of dimension i or higher.

The numerical irreducible decomposition outlined in [**14**] is performed by **WitnessClassify** which processes the output of **WitnessGenerate**. Completion of an algorithm for the irreducible decomposition requires a membership test, which we describe in the remainder of this section.

Membership is determined by three tests. Failure of any of the three indicates nonmembership. The tests for whether point $\mathbf{w} \in \widehat{W}_k$ is in component Z_{ij}, $i \geq k$ are organized in three steps:

Step 1: Point $\mathbf{w}$ must lie in $\mathrm{Span}(Z_{ij})$.
Step 2: It must satisfy the filtering polynomial: $p_{ij}(\mathbf{w}) = 0$.
Step 3: If $i > k$, we must check by homotopy continuation whether $\mathbf{w} \in Z_{ij}$ or $\mathbf{w} \in Z'_{ij}$, where Z'_{ij} is the union of the lower-dimensional (less than i) irreducible components of the bad set $B(Z_{ij})$, as described in section 2.

These steps are executed in the order listed, with termination at the first indication of nonmembership. Of course, step 1 is sufficient by itself if Z_{ij} is linear. Step 3 is the most expensive one, therefore we only invoke it after the tests in step 1 and 2 fail. We note, however, that step 2 often must be done in higher precision than step 3, because of the numerical conditioning of high-degree polynomials. Since extra precision can be quite expensive, it is conceivable that in some cases it may be preferable to skip step 2 and proceed directly with step 3.

Step 3 in the membership test requires a bit more explanation. The generalized theorem of Segre guarantees that, although the set of all possible p_{ij} does not necessarily cut out Z_{ij} exactly, they cut out a set of the form $Z_{ij} \cup Z'_{ij}$, with $\dim Z'_{ij} < i$. Since the points in the set $W_i \subset \widehat{W}_i$ are all general in their respective components, and these components are of dimension $i > \dim Z'_{ij}$, we can assume that none of them lie on any of the Z'_{ij}. This is why step 2 of the membership test is sufficient for $k = i$.

To motivate step 3 of the membership test we give a situation where it is unavoidable, when $Z_{i-1,l} \subset Z'_{ij}$. Such examples are straightforward to construct:

$$\left\{ \begin{array}{r} -x^2 + y^2 + z^2 = 0 \\ y(x^3 + y^3 + z^3 - 1) = 0 \\ z(x^3 + y^3 + z^3 - 1) = 0 \end{array} \right. \tag{3.1}$$

The solution set of the system consists of the origin as a double isolated root and the irreducible curve Z_{11} of degree six defined by $-x^2 + y^2 + z^2 = 0 = x^3 + y^3 + z^3 - 1$. This curve does not contain the origin. Note that given any solution (x, y, z) on the curve, the point (bx, by, bz) with b a cube root of unity is also on the curve. The line through these points contain the origin and therefore evaluation of the interpolation polynomial with central projections at the origin will yield zero. Thus in this case taking Z_{11} we have Z'_{11} equal to the set $\{(0, 0, 0)\}$.

Notice, if we choose a general $(N - i + 1)$-dimensional linear space through a point $\mathbf{x}$, it will contain exactly $\deg Z_{ij}$ points of Z_{ij} exactly when $\mathbf{x} \notin Z_{ij}$, and if $\mathbf{x} \in Z_{ij}$, then it will contain $\deg Z_{ij} - \mu_{Z_{ij},\mathbf{x}}$ points of Z_{ij} other than $\mathbf{x}$ (here $\mu_{Z_{ij},\mathbf{x}}$ is the multiplicity of the point $\mathbf{x}$ as a point on the reduced irreducible variety Z_{ij}). Since for a point $\mathbf{x}$ on Z_{ij}, $\mu_{Z_{ij},\mathbf{x}} = 1$ if and only if $\mathbf{x}$ is a smooth point of Z_{ij}, this also gives us a test for smoothness of any given point of Z_{ij}.

So, for a nonlinear component, in case $\mathbf{x}$ lies in $\mathrm{Span}(Z_{ij})$ and satisfies p_{ij}, then we choose a general hyperplane L in $\mathrm{Span}(Z_{ij})$ through $\mathbf{x}$. If the number of solutions $\mathbf{y} \neq \mathbf{x}$ of f_L that satisfy p_{ij} equals $\deg Z_{ij}$, then $\mathbf{x} \notin Z_{ij}$. Note that we generate more samples in this way. If we have to test sufficiently many points, we may use additional samples we found to construct filtering polynomials of higher degree.

The membership routine is given in pseudocode below, for testing whether a point $\mathbf{w}$, known to lie on some component of dimension at least k, actually lies on component Z_{ij}, $i \geq k$. In the normal usage of the algorithm, $\mathbf{w}$ is a member of the witness point superset $\widehat{W}_k$. However, one can also test an arbitrary point, not necessarily belonging to a witness point set, by calling this routine with $k = 0$.

ALGORITHM 3.2. [ismember] = **MembershipTest**$(f, p_{ij}, \mathrm{Span}(Z_{ij}), L, W_{ij}, \mathbf{w}, k)$

Input: Polynomial system f; filtering polynomial p_{ij} for component Z_{ij};
linear equations defining $\mathrm{Span}(Z_{ij})$; slices L: $\dim(L) = N - i$;
generic points W_{ij}: $W_{ij} \cap L \subset Z_{ij}$, $\#W_{ij} = \deg(Z_{ij})$;
test point $\mathbf{w}$; current dimension k.
Output: ismember $\in \{\text{true}, \text{false}\}$.

```
if w ∉ Span(Z_ij) or p_ij(w) ≠ 0                          [step 1 and 2]
 then return false;
 else if w ∈ Span(Z_ij) and deg(Z_ij) = 1                  [if Z_ij is linear]
        then return true;                                  [then test is conclusive]
       end if;
       if i ≤ k                                            [does Segre's result apply?]
        then return true;                                  [w ∈ Span(Z_ij), p_ij(w) = 0]
        else take general slices K: w ∈ K, dim K = dim L;  [slices K pass through w]
```

$H(\mathbf{x},t) := \left(\begin{array}{c} f \\ L \end{array} \right) t + \left(\begin{array}{c} f \\ K \end{array} \right) (1-t);$ [*homotopy between L and K*]
trace paths $\mathbf{x}(t) : H(\mathbf{x}(t),t) = \mathbf{0}$, [*invoke path tracker*]
start at $t = 1$ with all $\mathbf{x}(1) \in W_{ij}$ to $t = 0$; [*paths originate at W_{ij}*]
$W := \{ \mathbf{x} \mid H(\mathbf{x},0) = \mathbf{0} \}$; [*collect end points of paths*]
return $\mathbf{w} \in W$;
end if;
end if.

Just as in the description of the **Interpolate** algorithm, we make abstraction of the numerical tolerances needed in tests to decide whether computed results evaluate to zero.

We conclude this section with the observation that if we are concerned only with the breakup $\widehat{W_i} = W_i \cup J_i$ for each i, and not with the breakup $W_i = \cup_{j \in \mathcal{I}_i} W_{ij}$, then using the implementation of Step 3 in **MembershipTest** the filtering polynomials p_{ij} become superfluous. If we ignore the numerical difficulties of dealing with high degree components, then this observation has little practical value. However, while the cost of evaluating a polynomial might seem negligible compared to the cost of path tracking, we emphasize that interpolating (and thus also evaluating) high degree polynomials always requires multi-precision arithmetic, whereas standard machine precision suffices to perform the path tracking. Efficiency considerations left aside, this observation has significant implications if the coefficients of the input polynomial system are contaminated by errors or only known with limited precision.

4. Computational Experiments

Our new algorithms are implemented under a separate module of PHC [**16**]. Timings concern a Pentium III 800Mhz running Linux with 512Mb internal memory. The tables of timings have four columns:

1. "Generic points" is the time to compute generic witness points using the algorithm **WitnessGenerate** from [**13**], also described in [**14**].
2. "Previous projection" is the time to break up the witness points into components using the algorithm **WitnessClassify** from [**14**].
3. "Subspace restriction" is the time required when **WitnessClassify** is modified to find the linear span of components and using that result to work in the possibly lower dimensional space of the span.
4. "New approach" is **WitnessClassify** modified to use both the subspace restriction and central projections.

In all cases, the correct top dimension was given on input. This means that, if the top dimension equals k, the embedded polynomial systems that have the generic points as their solutions are obtained directly by adding k random hyperplanes to the original given systems.

4.1. The system Schwartz. We found the following system in the collection [**4**], related to constructing idempotents in group theory:

$$\begin{cases} -ab - b^2 - 2de - 2ch = 0 \\ -ac - 2bc - e^2 - 2dh = 0 \\ -c^2 - ad - 2bd - 2eh = 0 \\ -2cd - ae - 2be - h^2 = 0 \\ -d^2 - 2ce - ah - 2bh = 0 \end{cases}$$

This system yields a nice illustration of the calculation of the space spanned by the components. In this case, there are 32 lines as solutions of this system. Our method finds representations for those lines as the equations of the spanning subspaces. Finding 32 generic witness points takes 1s 410ms and in 980ms we get all spanning subspaces representing the lines.

4.2. A 7-bar system and reduced cyclic 8-roots. Here we reconsider two examples of [**14**]. The working precision was in both cases 40 decimal places. In Table 1 we summarize the results for decomposing the solution set into irreducible components.

	generic points	previous projection	subspace restriction	new approach
7barsys	18s 790ms	34s 370ms	22s 490ms	14s 250ms
redcyc8	9 m 8s 400ms	3m 36s 980ms	2m 15s 380ms	1m 37s 230ms

TABLE 1. CPU user times for the 7-bar system and the reduced cyclic 8-roots.

The sextic of the 7-bar system lies in a 4-dimensional space, so we can project twice from a point. This brings the number of samples down from 28 to 17 (15 for the interpolating filter and 2 central points). Column four of the five column Table 1 illustrates the efficiency gain — when we compare with the time in the third column — from taking samples in this 4-dimensional space, instead of in the ambient space of dimension 12. In particular, 6 samples are used to determine the span, which takes 5s 860ms. It takes then an additional 6s 540ms to compute the remaining 22 samples in the restricted space needed to construct the interpolating filter of degree six. With the previous projection, the program did all 28 samples in the ambient space, which took 5s 860ms + 21s 690ms = 27s 55ms.

For the 7-bar system we have to certify whether six solutions found as part of $\widehat{W}_0$ are isolated or belong to J_0 (which means they lie on the sextic). Step 3 in **MembershipTest** requires on this system timings between 110ms and 170ms to execute for one point, leading to a total of 920ms to test six points. If we have an interpolating filter of degree six, Step 1 and 2 in **MembershipTest** suffices and requires only 290ms to evaluate six points. We conclude the discussion of this example by mentioning that standard floating point arithmetic would now already suffice — instead of working with 40 decimal places — because the use of two central points leads to an interpolation filter of degree four instead of degree six, producing less roundoff. Constructing this quartic with standard arithmetic only takes 480ms as opposed to 14s 250ms when working with 40 decimal places.

The reduced cyclic 8-roots problems has two lines — for which we find the explicit equations — and two curves of degree 8. Those curves each live in a 3-dimensional space, so we can only project once of a point. Still this projection brings the number of samples from 45 down to 39 (38 for the interpolating filter and 1 central point) for each curve. As before, we can draw similar conclusions concerning the efficiency gain from sampling in the lower dimensional spaces.

4.3. Adjacent minors of a general matrix. This example was brought to our attention by B. Sturmfels. As polynomial equations we take all adjacent minors of a $2 \times (n+1)$-matrix. For example, for $n = 3$ we have as matrix and polynomial system:

$$\left[\begin{array}{cccc} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \end{array} \right] \qquad \left\{ \begin{array}{l} x_{11}x_{22} - x_{21}x_{12} = 0 \\ x_{12}x_{23} - x_{22}x_{13} = 0 \\ x_{13}x_{24} - x_{23}x_{14} = 0 \end{array} \right. \tag{4.1}$$

A theorem proven in [**5**] states that the ideal of adjacent 2×2-minors of a generic $2 \times (n+1)$-matrix is the intersection of F_n prime ideals, F_n being the nth Fibonacci number, and that the ideal is radical. Moreover all irreducible components of the solution set to the system in (4.1) is pure dimensional, have as dimension $\#\text{variables} - \#\text{equations} = 2(n+1) - n$; and the sum of the degrees of the irreducible components equals 2^n. In (4.2) we list the degrees of the irreducible components of dimension, for $n = 3, 4, \ldots, 8$.

(4.2)

$$\begin{array}{rcrcl} n = 3 & : & 8 & = & 4 + 2 + 2 \\ n = 4 & : & 16 & = & 5 + 4 + 3 + 3 + 1 \\ n = 5 & : & 32 & = & 6 + 6 + 6 + 4 + 4 + 2 + 2 + 2 \\ n = 6 & : & 64 & = & 9 + 8 + 8 + 7 + 5 + 5 + 4 + 4 + 4 + 3 + 3 + 3 + 1 \\ n = 7 & : & 128 & = & 12 + 12 + 10 + 10 + 8 + 8 + 6 + 6 + 6 + 6 + 6 \\ & & & & +6 + 6 + 6 + 4 + 4 + 4 + 2 + 2 + 2 + 2 \\ n = 8 & : & 256 & = & 16 + 15 + 15 + 12 + 12 + 12 + 12 + 12 + 9 \\ & & & & +9 + 9 + 9 + 8 + 8 + 8 + 8 + 8 + 8 + 7 + 7 + 5 + 5 + 5 \\ & & & & +4 + 4 + 4 + 4 + 4 + 4 + 3 + 3 + 3 + 3 + 1 \end{array}$$

In Table 2 we summarize our computational experiences. We observe that computing the witness point superset $\widehat{W}$, which is size 2^n, does not take long compared to the time needed for decomposing the solution set. This is mainly due to the use of multi-precision arithmetic, with 40 decimal places as working precision. For the $n = 8$ case, 56 decimal places were used. Comparing the third with the fourth column of the five column table, we notice the advantage of sampling in the appropriate spanning subspace. The difference between the fifth and the third column illustrates the efficiency gain between the projection used in [**14**] and our new approach.

As a class of polynomial systems we found it an interesting test case for the software and we present this here with the sole intent of comparing our approaches. Since this system has so much additional structure, we expect our approach to be inferior against methods like the ones discussed in [**6**](and the references cited therein) which are dedicated to binomial systems. However, this class of examples was previously unsolvable for numerical homotopy continuation methods.

n	generic points	previous projection	subspace restriction	new approach
3	570ms	20s 710ms	19s 950ms	17s 120ms
4	1s 740ms	1m 11s 550ms	59s 410ms	50s 160ms
5	8s 40ms	3m 58s 150ms	2m 54s 140ms	2m 11s 970ms
6	21s 530ms	14m 30s 660ms	9m 33s 230ms	6m 16s 690ms
7	1m 19s 40ms	1h 15m 30s 40ms	34m 30s 340ms	16m 30s 840ms
8	3m 56s 880ms	6h 40m 30s 630ms	4h 41m 35s 960ms	1h 26m 47s 400ms

TABLE 2. CPU user times for the system of adjacent minors.

5. Conclusions

This paper presents two improvements over the algorithms in [**14**] for the numerical irreducible decomposition of the solution sets of systems of polynomial equations. The first improvement is to check the dimension of the linear span of a solution component as the component is being sampled. If it is found that the span is smaller than the ambient space, computation is restricted to the spanning space. The second improvement is the use of central projections to reduce the degree of projections of the solution components, hence reducing the number of samples needed to construct an interpolating polynomial. The overall reduction in computation time for witness point classification ranged from 17% to 78% on our test problems, with the greatest percentage applying to the most difficult problem. The two improvements are complementary in the sense that if the span is narrow, then the subspace restriction pays dividends, and if the span is large compared to the dimension of the component, then the central projection produces larger gains. The improvement to be gained depends on the nature of the polynomial system and its solution set, in particular, the size of the ambient space, the size of the spanning spaces, and the degrees of the solution components.

References

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

[2] E. Arbarello, M. Cornalba, P.A. Griffiths, and J. Harris. *Geometry of Algebraic Curves*, Volume I. Grundlehren Math. Wiss. 267, Springer-Verlag, New York, (1985).

[3] A. Calabri and C. Ciliberto. On special projections of varieties: epitome to a theorem of Beniamino Segre. *Advances in Geometry* 1, 97-106, 2001.

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

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

[6] S. Hosten and J. Shapiro. Primary Decomposition of Lattice Basis Ideals. *Journal of Symbolic Computation* 29(4&5): 625–639, 2000. Special Issue on Symbolic Computation in Algebra, Analysis and Geometry. Edited by E. Cattani and R.C. Laubenbacher.

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

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

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

[10] B. Segre. On the locus of points from which an algebraic variety is projected multiply. Proceedings of the Phys.-Math. Soc. of Japan, ser. III, 18, 425–426, 1936.

[11] B. Segre. Sul luogo dei punti da cui una data varietà algebrica iperspaziale è proiettata multiplicamente. Atti I Congresso Unione Matematica Ital., 264–269, 1937.

[12] B. Shiffman and A.J. Sommese. *Vanishing Theorems on Complex Manifolds*, volume 56 of *Progress in Mathematics*. Birkhäuser, Boston, 1985.

[13] 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.

[14] 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.

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

[16] 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`.

Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556, U.S.A.

E-mail address: `sommese@nd.edu` *URL:* `http://www.nd.edu/~sommese`

Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 851 South Morgan (M/C 249), Chicago, IL 60607-7045, U.S.A.

E-mail address: `jan@math.uic.edu jan.verschelde@na-net.ornl.gov`
URL: `http://www.math.uic.edu/~jan`

General Motors Research Laboratories, Enterprise Systems Lab, Warren, MI 48090, U.S.A.

E-mail address: `Charles.W.Wampler@gm.com`