A LOCAL DIMENSION TEST FOR NUMERICALLY APPROXIMATED POINTS ON ALGEBRAIC SETS

DANIEL J. BATES*, JONATHAN D. HAUENSTEIN†, CHRIS PETERSON‡, AND ANDREW J. SOMMESE§

Abstract. Given a numerical approximation to a point $\mathbf{p}$ on the set V of common zeroes of a set of multivariate polynomials with complex coefficients, this article presents an efficient method to compute the maximum dimension of the irreducible components of V which pass through $\mathbf{p}$, i.e., a local dimension test. Such a test, used to filter out the so-called "junk points," is a crucial element in the numerical irreducible decomposition algorithms of Sommese, Verschelde, and Wampler. Computational evidence presented in this article illustrates that, with this new local-dimension test, "junk-point filtering" is no longer a bottleneck in the computation of a numerical irreducible decomposition. For moderate size examples this results in well over an order of magnitude improvement in the computation of a numerical irreducible decomposition. Also, to compute the irreducible components of a fixed dimension, it is no longer necessary to compute the numerical irreducible decomposition of all higher dimensions.
Keywords. local dimension, generic points, homotopy continuation, irreducible components, multiplicity, numerical algebraic geometry, polynomial system, primary decomposition, algebraic set, algebraic variety
AMS Subject Classification. 65H10, 68W30, 14Q99

Introduction. Given a numerical approximation of a point $\mathbf{p}$ on an algebraic set V determined by a set of polynomials $F \subset \mathbb{C}[x_1, \ldots, x_N]$, it is of great computational value to know the maximum dimension of the irreducible components containing $\mathbf{p}$, denoted here as $\dim_{\mathbf{p}}(V)$.

This article presents a rigorous numerical local dimension test. The test, which is efficient and robust, is based on the theory of Macaulay [20] and, more specifically, on the numerical approach of Dayton and Zeng [8].

This new algorithm is valuable in a number of settings, several of which are discussed in this article:

1. determining whether a given solution is isolated (and computing the multiplicity if it is);
2. computing $\dim_{\mathbf{p}}(V)$ for nonisolated points $\mathbf{p}$;
3. finding all irreducible components that contain a specified point $\mathbf{p}$;
4. computing the numerical irreducible decomposition of V more efficiently by reducing the junk point filter bottleneck; and
5. computing the irreducible components of V of a prescribed dimension.

Computational evidence indicates that the efficiency of this numerical method will have a significant impact on the structure of many of the algorithms of numerical

*Department of Mathematics, Colorado State University, Fort Collins, CO 80523 (bates@math.colostate.edu, `www.nd.edu/~dbates1`). This author was supported by Colorado State University and the Institute for Mathematics and Its Applications (IMA)

†Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556 (jhauenst@nd.edu, `www.nd.edu/~jhauenst`). This author was supported by the Duncan Chair of the University of Notre Dame, the University of Notre Dame Center for Applied Mathematics, NSF grant DMS-0410047, and NSF grant DMS-0712910

‡Department of Mathematics, Colorado State University, Fort Collins, CO 80523 (peterson@math.colostate.edu, `www.math.colostate.edu/~peterson`). This author was supported by Colorado State University, NSF grant MSPA-MCS-0434351, AFOSR-FA9550-08-1-0166, and the Institute for Mathematics and Its Applications (IMA)

§Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556 (sommese@nd.edu, `www.nd.edu/~sommese`). This author was supported by the Duncan Chair of the University of Notre Dame, NSF grant DMS-0410047, and NSF grant DMS-0712910

algebraic geometry, including the most fundamental computation: that of the numerical irreducible decomposition. For example, Section 4.3 shows that computing the numerical irreducible decomposition of the system defined by taking the 2×2 adjacent minors of a 3×9 matrix of indeterminates [13] is well over an order of magnitude cheaper.

Section 1 provides a brief overview of the basic definitions and concepts needed for the remainder of the article. More details regarding the specific algebra, geometry, and viewpoint utilized in this article can be found in [5]. An in-depth description of the general algebraic and geometric tools associated with this article can be found in [7, 9, 10, 12].

Section 2 provides very specific results which are used in the development and implementation of the algorithms of Section 3.

The algorithms themselves are presented in Section 3.1. The basic idea in calculus terms is that the dimensions T_k of the space of Taylor series expansions of degree at most k of algebraic functions on V at the point $\mathbf{p}$ eventually grow like $O(k^{\dim_{\mathbf{p}}(V)})$. When $\dim_{\mathbf{p}}(V) = 0$, the dimensions T_k steadily grow until they reach the multiplicity μ_p of the point $\mathbf{p}$ and are constant from that point on. If $\dim_{\mathbf{p}}(V) > 0$, these dimensions steadily grow without any limit. The number of paths ν_p with limit $\mathbf{p}$ for standard homotopies is an upper bound for the multiplicity of μ_p. Given such an upper bound $\nu_{\mathbf{p}}$ for the multiplicity of μ_p, we have a simple test to check whether $\mathbf{p}$ is isolated:

1. Compute the dimensions T_k until the minimum $k = \widehat{k}$, where

$$\widehat{k} := \min \left\{ k \geq 1 \mid T_k = T_{k-1} \text{ or } T_k > \nu_p \right\}.$$

2. Then $\mathbf{p}$ is isolated if and only if $T_{\widehat{k}} \leq \nu_p$.

If $V \subset \mathbb{C}^N$ is k-dimensional at a point $\mathbf{p}$, then, for $0 \leq \ell \leq k$, a general linear space L through $\mathbf{p}$ of dimension equal to $N - k + \ell$ will meet V in an algebraic subset of dimension ℓ at $\mathbf{p}$. Using this fact, we turn the above algorithm for whether a point is isolated into an algorithm for computing the maximum dimension of the components of V containing $\mathbf{p}$.

Section 3.2 describes the method of [8] and provides details on creating an efficient implementation. Examples are presented in Section 4 to illustrate the new methods.

Though the local dimension test in this article is the first rigorous numerical local dimension algorithm that applies to arbitrary polynomial systems, it is not the first numerical method proposed to compute local dimensions. Using the facts about slicing V with linear spaces L:

- If $V \subset \mathbb{C}^N$ is k-dimensional at a point $\mathbf{p}$, then a general linear space L through that point of dimension less than or equal to $N - k$ will meet V in no other points in a neighborhood of $\mathbf{p}$; and
- given a general linear space L of dimension greater than $N - k$, $L \cap V$ will contain points in a neighborhood of $\mathbf{p}$,

Sommese and Wampler [24, §3.3] showed that the local dimension $\dim_{\mathbf{p}}(V)$ could be determined by choosing an appropriate family L_t of linear spaces with $L_0 = L$ and then deciding whether the point $\mathbf{p}$ deforms in $V \cap L_t$. They did not present any numerical method to make this decision. In [14], Kuo and Li present an interesting heuristical method to make this decision. The method works well for many examples, but it does not seem possible to turn it into a rigorous local-dimension algorithm for a point on the solution set of polynomial systems. For instance, since the method is based on the presentation of the system and not on intrinsic properties of the solution set, it is not likely that any result covering general systems can be proved. Indeed, as the simple system in Section 4.2 (consisting of two cubic equations in two variables) shows: solution sets with multiple branches going through a point may well lead the method to give false answers.

1. Background material. In this section we collect basic definitions and concepts from algebraic geometry and commutative algebra.

1.1. Background from numerical algebraic geometry. The common zero locus of a set of multivariate polynomials is called an algebraic set. Let $F_1, F_2, \ldots, F_n$ be multivariate polynomials (with complex coefficients) in the variables $z_1, z_2, \ldots, z_N$ and let $V = V(F_1, F_2, \ldots, F_n) \doteq \{p \in \mathbb{C}^N | F_i(p) = 0 \text{ for } 1 \leq i \leq n\}$ be the algebraic set determined by the system. The set V is said to be *reducible* if it can be decomposed as the union of two algebraic sets with neither set a subset of the other and is said to be *irreducible* if it is not reducible. An irreducible algebraic set, which is called a *variety*, has the property that its set of smooth points is connected. Every algebraic set decomposes as a union of varieties $V_1, V_2, \ldots, V_r$. If the constraint $V_i \not\subseteq V_j$ whenever $i \neq j$ is imposed then the decomposition is unique. Determining the decomposition of an algebraic set into varieties is a fundamental problem in numerical algebraic geometry and serves as crucial data for many other computations. An algorithm that accomplishes this decomposition is described in [25] and has been implemented in the numeric/symbolic systems Bertini [2] and PHCpack [28]. The algorithm is built around the well-established numerical method known as homotopy continuation (other well known homotopy based software packages for finding isolated solutions of polynomial systems include HOM4PS–2.0 [15], Hompack [29], and PHom [11]). Essentially, a set of equations determining the algebraic set, V, is cast as a member of a parameterized family of polynomial systems one of which has known isolated solutions. The known solutions are tracked via a predictor/corrector method to points numerically close to V. A collection of endgame techniques are then utilized to refine these points to lie within a prescribed tolerance of V.

The dimension of a variety is defined to be the dimension of the tangent space at a smooth point of the variety. The dimension of an algebraic set is defined to be the largest dimension among the varieties appearing in its irreducible decomposition. If a variety, V_i, has dimension d then its degree, $\deg(V_i)$, is defined to be the number of points in the intersection of V_i with a generic linear space of codimension d. For an algebraic set, V, the basic algorithms of numerical algebraic geometry produce discrete data in the form of a *witness point set* [23, 25]. For each dimension d, this consists of a set of points W_d and a generic codimension d linear space L_d with the basic property that within a user-specified tolerance, the points of W_d are the intersection of L_d with the union of the d-dimensional components of V. Since a general codimension d linear space meets each d-dimensional irreducible component, V_i of V, in exactly $\deg(V_i)$ points, each d-dimensional irreducible component has as many witness points in W_d as its degree. Let D denote the dimension of $V = V(F_1, F_2, \ldots, F_n)$. A *cascade algorithm* utilizing repeated applications of homotopy continuation to polynomial systems constructed from $F_1, F_2, \ldots, F_n$ yields the full witness set $W_0 \cup W_1 \cup \cdots \cup W_D$. Each of the polynomial systems constructed from F is obtained by adding extra linear equations (corresponding to slices by generic hyperplane sections). Thus we also obtain equations for each of the linear spaces $L_0, L_1, \ldots, L_D$ used to slice away the W_i from V and these linear spaces form a flag. In fact, a (possibly larger) witness superset $\hat{W}_i$ containing W_i is obtained. The extra *junk points* $J_i = \hat{W}_i \setminus W_i$ actually lie on irreducible components of dimension greater than i. J_i is separated from W_i using a so-called $junkpointfilter$. Using the flag together with techniques such as monodromy, it is possible to partition W_d into subsets, which are in one-to-one correspondence with the d-dimensional irreducible components of V. In particular, the points in W_d can be organized into sets such that all points of a set lie (numerically) on the same irreducible component.

Thus, given a set of polynomials $\mathcal{F}$, it is possible to produce by numerical methods a flag together with a collection of subsets of points such that the subsets are pairwise disjoint and are in one-to-one correspondence with the irreducible components of the algebraic set determined by $\mathcal{F}$. The points in a subset all lie within a prescribed tolerance of the irreducible component to which it corresponds. The number of points in the subset is the same as the degree of the irreducible component and the subset is a numerical approximation of the intersection of the irreducible component with a known linear space of complimentary dimension (coming from the flag). Classical introductions to continuation methods can be found in [1, 21]. For an overview of numerical algorithms and techniques for dealing with systems of polynomials, see [18, 25, 26]. For details on the cascade algorithm, see [22, 25].

Each system of polynomial equations generates an *ideal*. If two systems of equations generate the same ideal then the algebraic sets determined by the systems are identical. However, systems of polynomial equations that generate different ideals can still determine the same algebraic set. The set of ideals, $\mathcal{S}(V)$ defining a given algebraic set, V, form a poset (under inclusion) which has a unique maximal element. This maximal element is the unique *radical ideal* in the poset and by construction is a superset of every element in $\mathcal{S}(V)$. Thus, two systems of equations determine the same algebraic set if and only if the ideals generated by the systems have the same radical. The polynomial system can be used to label each irreducible component of its corresponding algebraic set with a positive integer known as the *multiplicity* (a radical ideal imposes a multiplicity of one on each irreducible component). Systems of polynomial equations which impose a multiplicity greater than one on a component give rise to a special type of numerical instability which can require substantial computational effort to overcome, for example with the use of deflation [16, 17] or adaptive precision techniques [3]. These instabilities, in addition to bottlenecks that arise in the decomposition and cascade algorithms, lead to a great slow-down in the computations involved in computing W_d. A significant improvement in these algorithms will follow from the ability to efficiently resolve the following problem: *Given a point* $\mathbf{p}$ *which approximates a point lying on the algebraic set* $V = V(F_1, F_2, \ldots, F_n)$*, determine the dimension of the largest irreducible component of* V *which contains* $\mathbf{p}$. We call this the local dimension of V at $\mathbf{p}$ and denote it $\dim_{\mathbf{p}}(V)$. Indeed, the standard approach to the numerical irreducible decomposition processes components starting at the top dimensions and working down to zero-dimensional components, i.e., to isolated points. For systems for which only the isolated solutions are sought, this is computationally expensive. The local dimension test allows the processing for the numerical irreducible decomposition to directly compute the decomposition of the set of components of a prescribed dimension k without first having to carry out the computation of witness sets for all dimensions greater than k.

1.2. Background commutative algebra. The following paragraphs give a brief outline of the vocabulary and tools we will be using throughout the article. All definitions, propositions, and theorems can be found in expanded detail in [7, 9, 10, 12].

Let $\mathbb{C}$ denote the field of complex numbers. Consider the ring of polynomials $R = \mathbb{C}[z_1, z_2, \ldots, z_N]$. As a set, R consists of all polynomials in the variables $z_1, z_2, \ldots, z_N$

with complex coefficients. A subset $I \subset R$ is an *ideal* if $F, G \in I \implies F + G \in I$ and if $F \in I$, $G \in R \implies FG \in I$.

Definition 1.1. *Let $R = \mathbb{C}[z_1, z_2, \ldots, z_N]$. Let F, G be arbitrary elements in R. Let I be an ideal in R.*

- *I is* **prime** *if $FG \in I \implies F \in I$ or $G \in I$.*
- *I is* **primary** *if $FG \in I \implies F \in I$ or $G^m \in I$ for some m.*
- *The* **radical of** *I is the set $\sqrt{I} = \{F \in R | F^m \in I \text{ for some } m\}$.*
- *I is a* **radical ideal** *if $I = \sqrt{I}$.*
- *I is* $\mathfrak{p}$**-primary** *if I is a primary ideal and if $\mathfrak{p} = \sqrt{I}$.*
- *I is* **finitely generated** *if there exists a finite list $F_1, F_2, \ldots, F_r \in I$ such that every element in I can be written as an R-linear combination of $F_1, F_2, \ldots, F_r$. We call $\{F_1, F_2, \ldots, F_r\}$ a* **set of generators** *for I. We will denote this by $I = (F_1, F_2, \ldots, F_r)$.*

It should be noted that $\sqrt{I}$ is an ideal, that every prime ideal is a radical ideal, and that the radical of a primary ideal is a prime ideal. Furthermore, by the Hilbert Basis Theorem, every ideal in $\mathbb{C}[z_1, z_2, \ldots, z_N]$ is finitely generated.

The following definition introduces an algebraic idea that relates to the compactification $\mathbb{P}^N$ of $\mathbb{C}^N$. This simplifies certain proofs and algorithms.

Definition 1.2. *Let $R = \mathbb{C}[z_1, z_2, \ldots, z_N]$. Let $F \in R$.*

- *F is* **homogeneous** *if every term of F has the same degree.*
- *The* **homogenization of** *F (with respect to z_{N+1}) is the homogenous element $F^h = z_{N+1}^{\deg(F)} F\left(\frac{z_1}{z_{N+1}}, \frac{z_2}{z_{N+1}}, \ldots, \frac{z_N}{z_{N+1}}\right) \in \mathbb{C}[z_1, z_2, \ldots, z_{N+1}]$.*
- *A* **homogeneous ideal** *is an ideal that has a set of homogeneous generators.*

Our next definition introduces operations that associate geometric objects to algebraic objects and operations that associate algebraic objects to geometric objects. In particular, the operation $I(-)$ takes as input a subset of $\mathbb{C}^N$ and produces a subset of R. The operation $V(-)$ takes as input a subset of R and produces a subset of $\mathbb{C}^N$.

Definition 1.3. *Let $U \subset R = \mathbb{C}[z_1, z_2, \ldots, z_N]$ and let $T \subset \mathbb{C}^N$.*

- *Define $V(U) = \{\mathbf{z} \in \mathbb{C}^N | F(\mathbf{z}) = 0 \text{ for every } F \in U\}$.*
- *Define $I(T) = \{F \in R | F(\mathbf{z}) = 0 \text{ for every } \mathbf{z} \in T\}$.*
- *$T \subset \mathbb{C}^N$ is an (affine)* **algebraic set** *if $T = V(U)$ for some subset $U \subset \mathbb{C}[z_1, z_2, \ldots, z_N]$.*

We summarize some of the basic properties of the tools introduced thus far in the following proposition:

Proposition 1.4. *For any subset $T \subset \mathbb{C}^N$ and any subset $U \subset \mathbb{C}[z_1, z_2, \ldots, z_N]$,*

(i) $I(T)$ is a radical ideal.
(ii) $T \subseteq V(I(T))$ with equality if and only if T is an algebraic set.
(iii) $U \subseteq I(V(U))$ with equality if and only if U is a radical ideal.
(iv) If U is an ideal then $I(V(U)) = \sqrt{U}$.

An algebraic set, V, is said to be *reducible* if it is possible to write $V = V_1 \cup V_2$ with V_1, V_2 algebraic sets, with $V \neq V_1$, and with $V \neq V_2$. Algebraic sets which are not reducible are called *irreducible*. Irreducible algebraic sets are called *varieties*. Algebraic sets, varieties, ideals, and radical ideals have nice decomposition properties and relationships that are summarized as follows:

Proposition 1.5. *Decomposition properties:*

- *Every algebraic set can be written uniquely as the union of a finite number of varieties, none of which is a subset of another.*
- *Every radical ideal can be written uniquely as the intersection of a finite number of prime ideals, none of which is a subset of another.*

- *Every ideal is the intersection of a finite number of primary ideals.*
- *If V is a variety then $I(V)$ is a prime ideal.*
- *If I is a primary ideal then $V(I)$ is a variety.*
- *If $I = I_1 \cap I_2$ then $V(I) = V(I_1) \cup V(I_2)$.*

Expressing an ideal as the intersection of primary ideals is called a *primary decomposition.* Let I be an ideal and let $I = I_1 \cap I_2 \cap \cdots \cap I_k$ be a primary decomposition of I. Suppose I_i is $\mathfrak{p}_i$-primary for each i. The primary decomposition is called *reduced* if $\mathfrak{p}_i \neq \mathfrak{p}_j$ whenever $i \neq j$ and if for each i, $\cap_{j \neq i} I_j \nsubseteq I_i$. The prime ideals $\mathfrak{p}_1, \mathfrak{p}_2, \ldots, \mathfrak{p}_k$ are called *associated primes.* As defined, they depend on the choice of primary decomposition. However, the following proposition simplifies the situation and illustrates the central role of associated primes.

PROPOSITION 1.6. *(Primary decompositions)*

- *Every ideal has a reduced primary decomposition.*
- *Any reduced primary decomposition of a given ideal has the same set of associated primes.*
- *A radical ideal has a unique reduced primary decomposition.*
- *The associated primes of the radical of an ideal are a subset of the associated primes of the ideal.*

The proposition does not claim that the primary decomposition of an ideal is uniquely determined, it only claims that the associated primes are uniquely determined. However, the proposition does claim that radical ideals have a uniquely determined primary decomposition. Furthermore, the associated primes of an ideal may differ from the associated primes of its radical. The associated primes of an ideal that are not associated primes of the radical of the ideal are called *embedded primes.* It is useful to note that if $\mathfrak{p}$ is not an embedded prime of I then the $\mathfrak{p}$-primary component that appears in any reduced primary decomposition of I is the same.

To each ideal I, we can associate a degree and a dimension, denoted $deg(I)$ and $dim(I)$ respectively. Precise definitions of these terms can be found in [7, 9, 10, 12] with a particularly accessible introduction in [7]. One can use the degree function to define the *multiplicity* of a primary ideal. For instance, if I is a $\mathfrak{p}$-primary ideal then the multiplicity of I can be defined as $\mu(I) = deg(I)/deg(\mathfrak{p})$. Though multiplicity is defined as a fraction, it can be shown that the multiplicity of any $\mathfrak{p}$-primary ideal is a positive integer. If I is an ideal and if $\mathfrak{p}$ is an associated prime that is not an embedded prime then the multiplicity of I at $\mathfrak{p}$ will be taken to mean the multiplicity of the $\mathfrak{p}$-primary component of I.

2. The theory behind the algorithms. In the previous section, we outlined the general commutative algebra tools that are used. We closed the section by introducing the concept of the multiplicity of an ideal at a non-embedded associated prime. Now we will focus on several well-known theorems of multiplicity, homogenization, and local dimension that enable the development of the main algorithms.

Let $\mathcal{F} = \{F_1, \ldots, F_n\} \subseteq R = \mathbb{C}[z_1, \ldots, z_N]$ and let $V = V(\mathcal{F})$. Let $V = V_1 \cup \cdots \cup V_s$ be the reduced irreducible decomposition of V. Given a point $q \in \mathbb{C}^N$, the local dimension of V at q is defined as $\dim_q(V) = \max\{\dim(V_i) | q \in V_i\}$. If q does not lie on V then we set $\dim_q(V) = -1$. In the local dimension algorithm, we will utilize known algorithms for computing the multiplicity of an ideal at a primary component [8, 5]. The ideas presented work for both homogeneous and non-homogeneous ideals. Due to simplifications in the homogeneous setting and due to the fact that homogenization is an easy step (and is reversible with no loss in information), we often consider systems which are homogeneous. In order to make use of this simplifying assumption, it is important to understand the effect that homogenization has on multiplicity and local dimension.

Definition 2.1. *Let I be a homogeneous ideal. The k^{th}* **homogeneous part of** *I is defined to be the set of all elements of I which are homogeneous of degree k. It is denoted I_k and is a finite dimensional vector space over $\mathbb{C}$.*

Proposition 2.2 (Multiplicity and local dimension after homogenization).

Let $\mathcal{F} = \{F_1, F_2, \ldots, F_n\} \subset R = \mathbb{C}[z_1, z_2, \ldots, z_N]$ and let $(\mathcal{F})$ denote the ideal generated by $\mathcal{F}$. Let $F^h \in R' = C[z_1, z_2, \ldots, z_N, z_{N+1}]$ denote the homogenization of F with respect to the new variable z_{N+1} and let $\mathcal{F}^h = \{F_1^h, F_2^h, \ldots, F_n^h\}$. Let $q = (q_1, q_2, \ldots, q_N) \in \mathbb{C}^N$, let $q^h = (q_1, q_2, \ldots, q_N, 1) \in \mathbb{C}^{N+1}$, and let $\langle q^h \rangle$ denote the span of q^h as a vector in $\mathbb{C}^{N+1}$. Let H be a generic hyperplane passing through q and let L be a linear polynomial vanishing on H.

(1) If $q \in V(\mathcal{F})$ then $\dim_q(V(\mathcal{F})) = \dim_{q^h}(V(\mathcal{F}^h)) - 1$.

(2) If $q \in V(\mathcal{F})$ then $\dim_q(V(\mathcal{F}, L)) = \dim_q(V(\mathcal{F})) - 1$.

(3) If $I(q)$ is a non-embedded associated prime of $(\mathcal{F})$ then $I(\langle q^h \rangle)$ is a homogenous non-embedded associated prime of $(\mathcal{F}^h)$.

(4) If q is an isolated point in $V(\mathcal{F})$ then the multiplicity of $(\mathcal{F})$ at $I(q)$ is equal to the multiplicity of $(\mathcal{F}^h)$ at $I(\langle q^h \rangle)$.

(5) If I is a homogeneous ideal with $V(I) = \langle q^h \rangle$ then the multiplicity of I at $I(\langle q^h \rangle)$ is the dimension of $(R'/I)_d$ as a $\mathbb{C}$-vector space for $d \gg 0$.

(6) If I is $I(\langle q^h \rangle)$-primary then the multiplicity of I at $I(\langle q^h \rangle)$ is the dimension of $R'/(I, M)$ as a $\mathbb{C}$-vector space where M is a general, homogeneous linear form in R'.

The following theorem is important in the development of several algorithms as it allows (locally) a reduction to the zero dimensional case.

Theorem 2.3 (Reduction to a point). *Let I be a homogeneous ideal.*

(1) Let $\mathfrak{p}$ be a non-embedded, associated prime of I; let q be a generic point on $V(\mathfrak{p})$; let $D = \dim(V(\mathfrak{p}))$; let $L_1, L_2, \ldots, L_{D-1}$ be general linear forms in $I(\langle q \rangle)$; and let $J = (I, L_1, L_2, \ldots, L_{D-1})$. Then $I(\langle q \rangle)$ is a non-embedded associated prime of J.

(2) Let q' be any point on $V(I)$; let $D = \dim_{q'}(V(I))$; let $L_1, L_2, \ldots, L_{D-1}$ be general linear forms in $I(\langle q' \rangle)$; and let $J = (I, L_1, L_2, \ldots, L_{D-1})$. Then $I(\langle q' \rangle)$ is a non-embedded associated prime of J.

The following theorem indicates a way to compute the multiplicity of a homogeneous ideal localized at a point. In particular, it provides a stopping criterion for the multiplicity algorithms which form the core of the local dimension algorithm of the next section.

Theorem 2.4 (Convergence of multiplicity at an isolated point).

Let q be a point in $\mathbb{C}^{N+1}$. Suppose $I(\langle q \rangle)$ is an associated, non-embedded prime of a homogenous ideal I. Let $J_k = (I, I(\langle q \rangle)^k)$.

(1) The multiplicity of I at $I(\langle q \rangle)$ is equal to the multiplicity of J_k at $I(\langle q \rangle)$ for $k \gg 0$.

(2) If the multiplicity of J_k at $I(\langle q \rangle)$ is equal to the multiplicity of J_{k+1} at $I(\langle q \rangle)$ then the multiplicity of I at $I(\langle q \rangle)$ is equal to the multiplicity of J_k at $I(\langle q \rangle)$

See [10, §A.8] for statements related to the previous two results. With the theorems in this section in place, it is now easy to describe the local dimension test and various related algorithms. This is carried out in the following section.

3. Algorithms and implementation details. In this section, three algorithms related to local dimension are presented along with a discussion of details of the implementation. The three algorithms are

(1) Find whether a solution to a polynomial system is isolated (and its multiplicity if it is isolated).
(2) Find the local dimension of a solution to a polynomial system.
(3) Find all irreducible components of a polynomial system passing through a given solution.

For the purposes of presentation, we are assuming that ideals are homogeneous. In this setting, we are thinking of points and algebraic sets as lying in a projective space. We will use the notation $\mathbf{p} = [p_0 : p_1 : \cdots : p_N]$ to emphasize that $\mathbf{p} \in \mathbb{P}^N$, i.e., that for any $\lambda \neq 0$, $\lambda[p_0 : p_1 : \cdots : p_N]$ represents the same point $\mathbf{p}$. We will let $I_{\mathbf{p}}$ denote the homogeneous ideal corresponding to the point $\mathbf{p}$.

The fourth item listed at the beginning of the Introduction, i.e., the use of the local dimension test as a junk point filter during the computation of a numerical irreducible decomposition, is straightforward. The standard way of determining whether a candidate witness point is junk is to check whether that point lies on any of the previously discovered components of higher dimension by following paths originating from the witness points in all higher dimensions. Instead, algorithm (2) above very simply indicates whether a candidate witness point lies on a component of the current dimension of interest or some component of a higher dimension.

The fifth item from that list, i.e., the ability to study a single dimension of a given algebraic set, relies on algorithm (2) above. Indeed, a witness superset for the codimension d components of a given algebraic set may be computed by appending d generic hyperplane equations to the system and using standard zero-dimensional techniques. The junk points lying on components of codimension less than d may be removed from this set using (2) above, after which standard pure-dimensional decomposition methods, e.g., monodromy, may be utilized to complete the codimension d irreducible decomposition.

3.1. The algorithms. The local dimension algorithm and the related algorithms presented in the following pages rely on a method for finding the multiplicity of a homogeneous ideal supported at a point $\mathbf{p}$ (called *multiplicity* below) and an upper bound for the multiplicity if the point $\mathbf{p}$ is isolated (called *find_bound* below). The operation *multiplicity*$(\{F_1, \ldots, F_n\}, \mathbf{p}, k)$ returns the level k approximation of the multiplicity of the ideal at $I_{\mathbf{p}}$; see Section 3.2.2 for details. Two different options for computing the multiplicity can be found in [8, 5]. For *find_bound*, an upper bound can be determined by using homotopy continuation then simply counting the number of solution paths converging to the point $\mathbf{p}$. This upper bound, along with Theorem 2.4, provides the stopping criterion if the point $\mathbf{p}$ is not isolated.

Before formally stating the algorithms, let us first indicate informally how they work. If $\mathbf{p}$ is not isolated then a certain sequence of approximations to the multiplicity at $\mathbf{p}$ (called $\mu(k)$ below) will eventually grow beyond the upper bound determined using *find_bound*. If $\mathbf{p}$ is isolated then the sequence of approximations will stabilize to the correct multiplicity (bounded by *find_bound*). Algorithm 2 simply pairs this idea with the use of a flag of linear spaces through $\mathbf{p}$ to determine the local dimension. Algorithm 3 indicates how this method may be used to find all irreducible components which contain $\mathbf{p}$.

Algorithm 1. *is_isolated*$(\{F_1, F_2, \ldots, F_n\}, \mathbf{p}, B; \textit{is\_isolated}_{\mathbf{p}}, \mu_{\mathbf{p}})$

Input:

• $\{F_1, F_2, \ldots, F_n\}$: a set of homogeneous polynomials in $\mathbb{C}[z_0, z_1, \ldots, z_N]$.
• $\mathbf{p} = [p_0 : p_1 : \cdots : p_N]$: a point on $V(F_1, F_2, \ldots, F_n) \subset \mathbb{P}^N$.
• B: an upper bound on the multiplicity of $(F_1, F_2, \ldots, F_n)$ at $\mathbf{I_p}$ if $\mathbf{p}$ is isolated.

Output:
• $is\_isolated_{\mathbf{p}}$: True, if $\mathbf{p}$ is isolated, otherwise False.
• $\mu_{\mathbf{p}}$: the multiplicity of $(F_1, F_2, \ldots, F_n)$ at $\mathbf{I_p}$ if $\mathbf{p}$ is isolated.

Algorithm:
Set $k := 0$, $\mu(0) := 1$.
do
  $k := k + 1$.
  $\mu(k) = multiplicity(\{F_1, F_2, \ldots, F_n\}, \mathbf{p}, k)$.
while $\mu(k) \neq \mu(k-1)$ and $\mu(k) \leq B$.
If $\mu(k) \leq B$, then $is\_isolated_{\mathbf{p}} :=$ True and $\mu_{\mathbf{p}} := \mu(k)$, else $is\_isolated_{\mathbf{p}} :=$ False.

Using *is_isolated* with linear slicing, the local dimension test is straightforward and can be performed with the following algorithm.

Algorithm 2. $local\_dimension(\{F_1, F_2, \ldots, F_n\}, \mathbf{p}; \dim_{\mathbf{p}}, \mu_{\mathbf{p}})$

Input:
• $\{F_1, F_2, \ldots, F_n\}$: a set of homogeneous polynomials in $\mathbb{C}[z_0, z_1, \ldots, z_N]$.
• $\mathbf{p} = [p_0 : p_1 : \cdots : p_N]$: a point on $V(F_1, F_2, \ldots, F_n) \subset \mathbb{P}^N$.

Output:
• $\dim_{\mathbf{p}}$: the local dimension of $\mathbf{p}$.
• $\mu_{\mathbf{p}}$: the multiplicity of $(F_1, F_2, \ldots, F_n)$ at $\mathbf{I_p}$, if $\mathbf{p}$ is isolated.

Algorithm:
Choose $L_1, \ldots, L_N$, random linear forms through $\mathbf{p}$ on $\mathbb{P}^N$, and set $k := -1$.
do
  $k := k + 1$.
  $B := find\_bound(\{F_1, F_2, \ldots, F_n, L_1, \ldots, L_k\}, \mathbf{p})$.
  $(is\_isolated_k, \mu_k) := is\_isolated(\{F_1, F_2, \ldots, F_n, L_1, \ldots, L_k\}, \mathbf{p}, B)$.
while $is\_isolated_k = False$.
Set $\dim_{\mathbf{p}} := k$ and $\mu_{\mathbf{p}} := \mu_k$.

From an irreducible decomposition of $V(F_1, F_2, \ldots, F_n)$, all irreducible components passing through $\mathbf{p}$ can be obtained using a membership test, called *membership* below, which decides whether a given solution lies on a given irreducible component. The canonical membership test is described in [23].

Algorithm 3. $irreducible\_components(\{F_1, F_2, \ldots, F_n\}, W, \mathbf{p}; J)$

Input:
• $\{F_1, F_2, \ldots, F_n\}$: a set of homogeneous polynomials in $\mathbb{C}[z_0, z_1, \ldots, z_N]$.
• $\mathbf{p} = [p_0 : p_1 : \cdots : p_N]$: a point on $V(F_1, F_2, \ldots, F_n) \subset \mathbb{P}^N$.
• W: an irreducible decomposition of $V(F_1, F_2, \ldots, F_n)$.

Output:
• J: a set of pairs of numbers, each representing the dimension and component number of an irreducible component of which $\mathbf{p}$ is a member.

Algorithm:
Set $J := \{\}$.
Compute $(\dim_{\mathbf{p}}, \mu_{\mathbf{p}}) := \mathit{local\_dimension}(\{F_1, F_2, \ldots, F_n\}, \mathbf{p})$.
for $j = 0, \ldots, \dim_{\mathbf{p}}$
  $m :=$ number of irreducible components of dimension j in W_j.
  for $k = 1, \ldots, m$
    Compute $\mathit{is\_member} := \mathit{membership}(\{F_1, F_2, \ldots, F_n\}, W_{jk}, \mathbf{p})$.
    if $\mathit{is\_member} = True$
      $J := J \cup \{(j, k)\}$.

3.2. Details of the implementation. The algorithms presented above have been implemented as a module of the Bertini software package [2, 4] which is under development by the first, second, and fourth authors and Charles Wampler of General Motors Research and Development. This implementation uses the multiplicity algorithm described in [8] rather than that of [5] as the structure of the former is more convenient from an implementation standpoint than that of the latter. The implementation makes use of the upper bound from homotopy continuation described above, and the witness set membership test described in [23] was used for the algorithm *membership*. As the implementation of the algorithm *multiplicity* is at the core of the algorithms presented in Section 3.1, the remainder of this section is dedicated to describing its implementation.

3.2.1. Construction of multiplicity matrix M_k. The Dayton-Zeng method for computing multiplicity structures is described in detail in [8]. At the heart of the method is the construction of a sequence of so-called *multiplicity matrices*, M_k. Given a (possibly approximate) solution, $\mathbf{p}$, of a polynomial system $\{F_1, F_2, \ldots, F_n\}$, M_k is a matrix of partial derivatives evaluated at the point $\mathbf{p}$. More precisely, the entries of the matrix consist of all evaluated partial derivatives up to (and including) those of order k of the functions of the form $(x-p)^\alpha F_i$, where

- $\alpha = (\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_n) \in (Z_{\geq 0})^{n+1}$ so that $(x-p)^\alpha = (x_0 - p_0)^{\alpha_0}(x_1 - p_1)^{\alpha_1}(x_2 - p_2)^{\alpha_2} \ldots (x_n - p_n)^{\alpha_n}$;
- α ranges over all elements with coordinate sum less than k; and
- all possible combinations of partial derivatives and monomials are taken.

From the definition, M_k is embedded into M_{k+1} in the form

$$M_{k+1} = \begin{bmatrix} M_k & A_k \\ 0 & B_k \end{bmatrix}. \tag{3.1}$$

The structure of this matrix can be exploited to create an efficient implementation of this method, described in Section 3.2.3.

As an example, consider the system $F_1 = x_1 - x_2 + x_1^2$ and $F_2 = x_1 - x_2 + x_2^2$ at the point $p = (0,0)$, from §4 of [8]. In this case, M_1 is simply

	$\{0,0\}$	$\{1,0\}$	$\{0,1\}$
F_1	0	1	-1
F_2	0	1	-1

where the columns correspond to multi-indices $\{i, j\}$ which in turn correspond to the partial derivative $\frac{\partial}{\partial x_1^i \partial x_2^j}$. For this example, M_3 has the following form:

	$\{0,0\}$	$\{1,0\}$	$\{0,1\}$	$\{2,0\}$	$\{1,1\}$	$\{0,2\}$	$\{3,0\}$	$\{2,1\}$	$\{1,2\}$	$\{0,3\}$
F_1	0	1	-1	1	0	0	0	0	0	0
F_2	0	1	-1	0	0	1	0	0	0	0
x_1F_1	0	0	0	1	-1	0	1	0	0	0
x_1F_2	0	0	0	1	-1	0	0	0	1	0
x_2F_1	0	0	0	0	1	-1	0	1	0	0
x_2F_2	0	0	0	0	1	-1	0	0	0	1
$x_1^2F_1$	0	0	0	0	0	0	1	-1	0	0
$x_1^2F_2$	0	0	0	0	0	0	1	-1	0	0
$x_1x_2F_1$	0	0	0	0	0	0	0	1	-1	0
$x_1x_2F_2$	0	0	0	0	0	0	0	1	-1	0
$x_2^2F_1$	0	0	0	0	0	0	0	0	1	-1
$x_2^2F_2$	0	0	0	0	0	0	0	0	1	-1

3.2.2. The Dayton-Zeng multiplicity method. Both the method of [8] and that of [5] compute a sequence of nonnegative integers $\mu(k)$ that converge to the multiplicity of $I = (F_1, F_2, \ldots, F_n)$ at $I_{\mathbf{p}}$ when $\mathbf{p}$ is isolated. In fact, the sequence $\{\mu(k)\}$ stabilizes as soon as $\mu(k) = \mu(k+1)$. If $\mathbf{p}$ is not isolated, $\mu(k)$ will grow indefinitely. In this setting, $\mu(k)$ will eventually grow like a polynomial of degree $\dim_{\mathbf{p}}(V)$, the local dimension of $\mathbf{p}$.

The values of $\mu(k)$ are computed from the dimension of the null space of M_k. When the system $F_1, \ldots, F_n$ is considered as defining an object in affine space, $\mu(k)$ is the dimension of the null space of M_k. When the system is considered as defining an object in projective space, $\mu(k) = \text{nullity}(M_k) - \text{nullity}(M_{k-1})$. Using this trivial modification, the algorithm can be used in either setting.

Algorithm 1 in §3.1 considers the input system as defining an object in projective space. Since this algorithm loops over k, we provide the central step of this method as Algorithm 4.

Algorithm 4. *multiplicity*$(\{F_1, F_2, \ldots, F_n\}, \mathbf{p}, k; \mu(k))$

<u>**Input**</u>:
- $\{F_1, F_2, \ldots, F_n\}$: a set of homogeneous polynomials in $\mathbb{C}[z_0, z_1, \ldots, z_N]$.
- $\mathbf{p} = [p_0 : p_1 : \cdots : p_N]$: a point on $V(F_1, F_2, \ldots, F_n) \subset \mathbb{P}^N$.
- k: the level at which $\mu(k)$ is to be computed.

<u>**Output**</u>:
- $\mu(k)$: The level k approximation to the multiplicity of the ideal at $\mathbf{p}$.

<u>**Algorithm**</u>:
Form M_{k-1} and M_k.
Evaluate M_{k-1} and M_k at $\mathbf{p}$.
Compute $\mu(k) = \text{nullity}(M_k) - \text{nullity}(M_{k-1})$.

The formation and evaluation of M_k may be carried out rapidly due to the multivariate version of the Leibnitz rule. The efficient computation of the dimension of the null space of M_k is made simpler by a few key observations. The implementation of these steps within Bertini is described in the next section.

3.2.3. Implementation details and efficiency. The bulk of the computation time involved with the multiplicity computation is in the computation of the dimension of the null space of M_k (corresponding to step 3 of Algorithm 4). The manner in which this has been implemented in Bertini will be described later in this section.

First, it is worthwhile to note that there is a fast way to form M_k and evaluate it at $\mathbf{p}$, these operations form the first two steps of Algorithm 4. For $\alpha, \beta \in (Z_{\geq 0})^{n+1}$, define

- $\alpha! = \alpha_0!\alpha_1!\alpha_2! \ldots \alpha_n!$,
- if $\alpha \geq \beta$, $C(\alpha, \beta) = \frac{\alpha!}{\beta!(\alpha-\beta)!}$,
- $\partial^\alpha = \frac{\partial^{|\alpha|}}{\partial x^\alpha}$, and
- $\partial_\alpha = \frac{1}{\alpha!}\partial^\alpha$.

In this notation, entries of M_k have the form $\partial_\alpha((x - \mathbf{p})^\beta F_j)(\mathbf{p})$.

Consider the multivariate form of the Leibnitz Rule:

$$\partial^\alpha(fg) = \sum_{0 \leq \beta \leq \alpha} C(\alpha, \beta)(\partial^\beta f)(\partial^{\alpha-\beta} g).$$

Notice that

$$\partial^\alpha((x - \mathbf{p})^\beta)(\mathbf{p}) = \begin{cases} \alpha!, \text{ if } \alpha = \beta, \\ 0, \text{ otherwise.} \end{cases} \tag{3.2}$$

Applying the Leibnitz Rule, we have that

$$\partial_\alpha((x - \mathbf{p})^\beta F_j)(\mathbf{p}) = \begin{cases} \partial_{\alpha-\beta} F_j(\mathbf{p}), \text{ if } \beta \leq \alpha, \\ 0, \text{ otherwise.} \end{cases} \tag{3.3}$$

As a result, the multiplicity matrices may be rapidly populated with zeroes and the appropriate evaluated partial derivatives, simply by observing the α and β corresponding to the given entry.

As for the computation of the dimension of the null space of a matrix, there are several key observations to make. First, columns of zeroes may be discarded. This condition is trivially checked for each column. The first column (as defined above) will always be zero while other columns may be zero as well, depending upon the monomial structure of the polynomial system.

Also, exploiting the block upper triangular structure and the embedding of M_k in M_{k+1} shown in Eq. 3.1 greatly speeds evaluation. Since a decomposition of M_k is already computed and its (co)rank is known from the previous step, the complexity of computing a decomposition of M_{k+1} and finding its (co)rank is reduced. See [8, §5] for a discussion on efficiently computing a QR decomposition of M_{k+1} using a QR decomposition of M_k.

The actual implementation of a rank-finding algorithm in Bertini uses the following well known fact. If A is an $m \times n$ matrix of rank r (with $r \leq n$) and if U is a random $n \times r$ matrix satisfying $U^*U = I_r$, then AU is an $m \times r$ matrix of rank r. By letting U_k be a sequence of random matrices of the appropriate size and satisfying the appropriate constraint on $U_k^* U_k$, $\text{rank}(M_k U_k) = \text{rank}(M_k)$, and $\text{rank}(M_{k+1}) = \text{rank}(\widehat{M}_{k+1})$ where

$$\widehat{M}_{k+1} = M_{k+1} \begin{bmatrix} U_k & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} M_k & A_{k+1} \\ 0 & B_{k+1} \end{bmatrix} \begin{bmatrix} U_k & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} M_k U_k & A_{k+1} \\ 0 & B_{k+1} \end{bmatrix},$$

where I is the identity matrix of the appropriate size. Letting

$$M_k U_k = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}$$

be a QR decomposition of M_kU_k, we have

$$\widehat{M}_{k+1} = \begin{bmatrix} Q_1 & Q_2 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} R & C_1 \\ 0 & C_2 \\ 0 & B_{k+1} \end{bmatrix},$$

where

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} Q_1^* \\ Q_2^* \end{bmatrix} A_{k+1}.$$

By construction,

$$\operatorname{rank}(M_{k+1}) = \operatorname{rank}(\widehat{M}_{k+1}) = \operatorname{rank}(M_k) + \operatorname{rank}\left(\begin{bmatrix} C_2 \\ B_{k+1} \end{bmatrix}\right). \tag{3.4}$$

Computing the rank on the right of Eq. 3.4 can be carried out using rank-revealing methods found in [27] or the very efficient technique found in [19].

4. Examples. The local dimension computations were run using the implementation described in Section 3.2 in the Bertini software package [2, 4]. The computational examples we will discuss were run using an Opteron 250 processor with 64-bit Linux. The parallel examples of Section 4.3 were run on a cluster consisting of a manager that uses one core of a Xeon 5410 processor and 8 computing nodes each containing two Xeon 5410 processors running 64-bit Linux.

In Section 4.1, Bertini is used to compute the multiplicity structure of an isolated solution, which is a significant part of the local-dimension computation. ApaTools [30] is the first implementation of the methods described in [8]. Both Bertini and ApaTools can perform the multiplicity structure computations numerically for (possibly) inexact solutions, but ApaTools can also perform symbolic computations for exact solutions. The current version of ApaTools, which is a Maple toolbox, took longer than Bertini on many examples, e.g.,for $S_7 \cap \widehat{S}_5$, ApaTools took 2.60 seconds and Bertini took 0.02 seconds. In Section 4.2, an example is presented to illustrate potential difficulties arising in the heuristic local-dimension approach of [14]. Finally, Section 4.3 presents examples to demonstrate the application of the local dimension test to junk point filtering.

4.1. A class with isolated solutions. Rhodonea curves, such as those displayed in Figure 4.1, are defined by polar equations of the form $r = \sin(k\theta)$. Denote the curve defined by the polar equation $r = \sin(k\theta)$ by S_k and let $\widehat{S}_k$ denote the curve obtained by applying a random rotation about the origin to S_k. For k even, S_k has $2k$ "petals" and for k odd, S_k has k petals. We restricted our attention to pairs of odd integers m and n. In this setup, the origin will be an isolated solution of the pair of equations defining the curves S_m and $\widehat{S}_n$. Due to the random rotation about the origin, the petals of S_m and $\widehat{S}_n$ do not share any common tangent directions at the origin. As a result, the multiplicity of the isolated solution at the origin is mn.

Tables 4.2 and 4.3 display the sequence $\mu(k)$ and the timings for the implementation in Bertini for various (odd) values of m and n.

4.2. A positive-dimensional example. The Rhodonea curve S_1, as described in Section 4.1, is the solution set of the polynomial $g(x, y) = x^2 + y^2 - y$. In particular, S_1 is the circle of radius $\frac{1}{2}$ centered at $\left(0, \frac{1}{2}\right)$. For a 2×2 random real orthogonal matrix A, define $(\widehat{x}, \widehat{y})$ by

$$\begin{bmatrix} \widehat{x} \\ \widehat{y} \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix}.$$

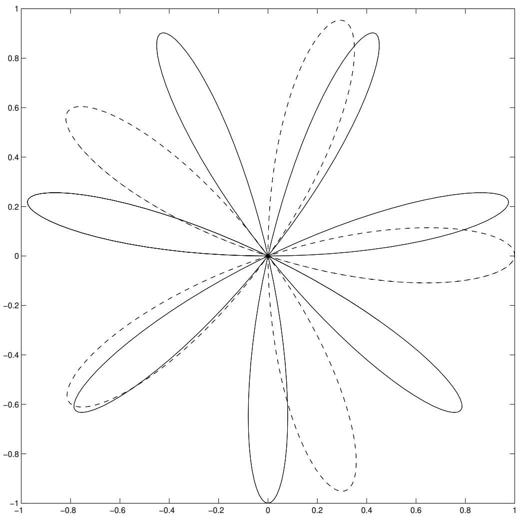

FIG. 4.1. *Rhodonea curves* S_7 *and* $\widehat{S}_5$.

n	5	7
$\mu(k)$	$1, 3, 6, 10, 15, 20, 25, 29, 32, 34, 35, 35$	$1, 3, 6, 10, 15, 21, 28, 34, 39, 43, 46, 48, 49, 49$
Bertini	0.020s	0.066s

FIG. 4.2. *Computations of the multiplicity structure at the origin for* $S_7 \cap \widehat{S}_n$

$\mu(k)$	$1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 76, 85, 93, 100, 106, 111, 115, 118, 120, 121, 121$
Bertini	3.478s

FIG. 4.3. *Computations of the multiplicity structure at the origin for* $S_{11} \cap \widehat{S}_{11}$

The solution set of $\widehat{g}(x, y) = g(\widehat{x}, \widehat{y})$ is a rotation of S_1 about the origin. The solution set of the system $g(x, y) = \widehat{g}(x, y) = 0$ is the 2 intersection points of the corresponding two circles. One of the intersection points is the origin while the other is a point $(\bar{x}, \bar{y})$ away from the origin.

Let $F_1(x, y) = xg(x, y)$, $F_2(x, y) = x\widehat{g}(x, y)$ and $I = (F_1, F_2)$. The algebraic set corresponding to I has 2 irreducible components: the point $p = \{(\bar{x}, \bar{y})\}$ and the line $L = \{x = 0\}$. These irreducible components correspond to the two associated primes of $\sqrt{I}$ given by $\mathfrak{p}_1 = (x - \bar{x}, y - \bar{y})$ and $\mathfrak{p}_2 = (x)$. However, I has three associated primes consisting of $\mathfrak{p}_1, \mathfrak{p}_2$ and $\mathfrak{p}_3 = (x, y)$. In other words, the origin is an embedded prime. Bertini used a total degree homotopy and found 2 paths that lead to the origin. Since the Jacobian of the system is the zero matrix at the origin, the multiplicity must be at least 3 in order for it to be an isolated solution. The algorithm *is_isolated* identified that the origin is not isolated in 0.005 seconds. The full numerical irreducible decomposition was completed by Bertini in 0.11 seconds.

The Matlab module described in [14] used a polyhedral homotopy which also had 2 paths leading to the origin. However, it was unable to identify that the origin lies on a one-dimensional component. The module did correctly identify $(\bar{x}, \bar{y})$ as isolated and that another point of the form $(0, y')$ (with $y' \neq 0$) was a point lying on a one-dimensional component.

4.3. A collection of high-dimensional examples. When computing a witness superset, the cascade algorithm [22, 25] is widely believed to result in fewer junk points than a witness superset created by slicing at each dimension. For problems with many components at different dimensions, junk point filtering using a membership test can result in a bottleneck. With *is_isolated*, the filtering of junk points is very efficient.

For example, let G_m denote the homogeneous system defined by taking the 2×2 adjacent minors of a $3 \times m$ matrix of indeterminates [13]. It is well-known that, for $m \geq 3$, the solution set of G_m consists of components of different dimensions. Table 4.4 compares the timings of witness superset methods, i.e., the cascade algorithm and slicing at each dimension, along with junk point filtering methods, i.e., the membership test of [23] and *is_isolated*, for computing a numerical irreducible decomposition for G_m, $3 \leq m \leq 9$. It should be noted that computing a witness set is the majority of the computational cost for computing a numerical irreducible decomposition for G_m. In particular, for $m = 7$, it took 272.79 seconds to compute a witness set using the cascade algorithm with *is_isolated* and 15.91 seconds to decompose this witness set into its irreducible components.

	is_isolated		membership test	
m	slicing	cascade	slicing	cascade
3	0.12	0.15	0.12	0.17
4	0.71	1.12	1.15	1.32
5	4.96	7.30	11.86	10.68
6	29.26	71.51	149.59	92.28
7	183.14	288.70	2036.73	854.33
8	1157.74	1714.35	17362.71	8720.14
9	7296.78	9533.50	219509.84	83060.43

FIG. 4.4. *Comparison for computing a numerical irreducible decomposition for G_m, in seconds*

In Bertini, a parallel junk point filtering is achieved using a dynamic distribution of the points since each point can be handled independently. Table 4.5 compares the timings for computing a numerical irreducible decomposition in parallel for G_m, $7 \leq m \leq 9$.

	is_isolated		membership test	
m	slicing	cascade	slicing	cascade
7	15.83	16.36	82.59	30.03
8	35.87	49.88	350.96	168.46
9	138.91	213.23	3320.04	1399.43

FIG. 4.5. *Comparison for computing a numerical irreducible decomposition in parallel for G_m, in seconds*

5. Conclusions. This article provides an effective numerical local dimension test, an algorithm that should prove useful as a subroutine in many other algorithms within numerical algebraic geometry. This algorithm relies heavily on the Dayton-Zeng [8] and Bates-Peterson-Sommese [5] methods for the computation of multiplicity information at a solution of a system of multivariate equations. The utility of this method has been described in a few settings and several numerical examples were presented to illustrate the various related algorithms of the article.

REFERENCES

[1] E. Allgower and K. Georg, *Introduction to numerical continuation methods*, Classics in Applied Mathematics 45, SIAM Press, Philadelphia, 2003.

[2] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, Bertini: software for numerical algebraic geometry, Available at `www.nd.edu/∼sommese/bertini`.

[3] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, Adaptive multiprecision path tracking, *SIAM J. Numer. Anal.*, 46 (2008), pp. 722–746.

[4] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, Software for numerical algebraic geometry: a paradigm and progress towards its implementation, in *IMA Volume 148: Software for Algebraic Geometry*, M. Stillman, N. Takayama, and J. Verschelde, eds., Springer, New York, 2008, pp. 1–14.

[5] D.J. Bates, C. Peterson, and A.J. Sommese, A numerical-symbolic algorithm for computing the multiplicity of a component of an algebraic set, *J. Complexity*, 22 (2006), pp. 475-489.

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

[7] D. Cox, J. Little and D. O'Shea, *Ideals, varieties, and algorithms*, Second Edition, Undergraduate Texts in Mathematics, Springer, New York, 1996.

[8] B. Dayton and Z. Zeng, Computing the multiplicity structure in solving polynomial systems, in Proceedings of ISSAC 2005, ACM, New York, 2005, pp. 116–123.

[9] W. Fulton, *Algebraic curves*, W.A. Benjamin, New York, 1969.

[10] G-M. Greuel and G. Pfister, *A* **Singular** *introduction to commutative algebra*, Springer, Berlin, 2002.

[11] T. Gunji, S. Kim, M. Kojima, A. Takeda, K. Fujisawa, and T. Mizutani, PHoM – Polyhedral homotopy continuation software for polynomial systems, Available at `www.is.titech.ac.jp/∼kojima`.

[12] R. Hartshorne, *Algebraic geometry*, Graduate Texts in Mathematics 52, Springer, New York, 1977.

[13] S. Hoşten and S. Sullivant, Ideals of adjacent minors, *J. Algebra*, 277 (2004), pp. 615–642.

[14] Y.C. Kuo and T.Y. Li, Determining dimension of the solution component that contains a computed zero of a polynomial system, *J. Math. Anal. Appl.*, 338 (2008), pp. 840–851.

[15] T.L. Lee, T.Y. Li, and C.H. Tsai, HOM4PS–2.0, A software package for solving polynomial systems by the polyhedral homotopy continuation method, preprint.

[16] A. Leykin, J. Verschelde, and A. Zhao, Newton's method with deflation for isolated singularities of polynomial systems, *Theoret. Comput. Sci.*, 359 (2006), pp. 111–122.

[17] A. Leykin, J. Verschelde, and A. Zhao, Higher-order deflation for polynomial systems with isolated singular solutions, in *IMA Volume 146: Algorithms in Algebraic Geometry*, A. Dickenstein, F.-O. Schreyer, and A.J. Sommese, eds., Springer, New York, 2008, pp. 79–97.

[18] 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*, F. Cucker, ed., North-Holland, 2003, pp. 209–304.

[19] T.Y. Li and Z. Zeng, A rank-revealing method with updating, downdating, and applications, *SIAM J. Matrix Anal. Appl.*, 26 (2005), pp. 918–946.

[20] F.S. Macaulay, *The algebraic theory of modular systems*, Cambridge University Press, 1916.

[21] A. Morgan, *Solving polynomial systems using continuation for engineering and scientific problems*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[22] A.J. Sommese and J. Verschelde, Numerical Homotopies to compute generic points on positive dimensional Algebraic Sets, *J. Complexity*, 16 (2000), pp. 572–602.

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

[24] A.J. Sommese and C.W. Wampler, Numerical algebraic geometry, in *The Mathematics of Numerical Analysis*, J. Renegar, M. Shub, and S. Smale, eds., volume 32 of *Lectures in*

Applied Mathematics, 1996, pp. 749–763. Proceedings of the AMS-SIAM Summer Seminar in Applied Mathematics, Park City, Utah, July 17-August 11, 1995, Park City, Utah.

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

[26] H. Stetter, *Numerical polynomial algebra*, SIAM, Philadelphia, 2004.

[27] G.W. Stewart, The QLP approximation to the singular value decomposition, *SIAM J. Sci. Comput.*, 20 (1999), pp. 1336–1348.

[28] J. Verschelde, PHCPACK: A general-purpose solver for polynomial systems by homotopy continuation, Paper and software available at `www.math.uic.edu/~jan`.

[29] L. Watson, A suite of FORTRAN 77 subroutines for solving nonlinear systems of equations by homotopy methods, Download available at `www.netlib.org/hompack`.

[30] Z. Zeng, ApaTools: A software toolbox for approximate polynomial algebra, Article and software available at `www.neiu.edu/~zzeng`.