A NUMERICAL LOCAL DIMENSION TEST FOR POINTS ON THE SOLUTION SET OF A SYSTEM OF POLYNOMIAL EQUATIONS

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

Abstract. The solution set V of a polynomial system, i.e., the set of common zeroes of a set of multivariate polynomials with complex coefficients, may contain several components, e.g., points, curves, surfaces, etc. Each component has attached to it a number of quantities, one of which is its dimension. Given a numerical approximation to a point **p** on the set V, this article presents an efficient algorithm to compute the maximum dimension of the irreducible components of V which pass through **p**, i.e., a local dimension test. Such a test is a crucial element in the homotopy-based numerical irreducible decomposition algorithms of Sommese, Verschelde, and Wampler.

This article presents computational evidence to illustrate that the use of this new algorithm greatly reduces the cost of so-called "junk-point filtering," previously a significant 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. As the computation of a numerical irreducible decomposition is a fundamental *backbone* operation, gains in efficiency in the irreducible decomposition algorithm carry over to the many computations which require this decomposition as an initial step. Another feature of a local dimension test is that one can now compute the irreducible components in a prescribed dimension without first computing the numerical irreducible decomposition of all higher dimensions. For example, one may compute the isolated solutions of a polynomial system without having to carry out the full numerical irreducible decomposition.

Keywords. local dimension, generic points, homotopy continuation, irreducible components, multiplicity, numerical algebraic geometry, polynomial system

AMS Subject Classification. 65H10, 68W30, 14Q99

Introduction. The solution set V of a system of polynomial equations may contain many pieces (irreducible components), e.g., points, curves, surfaces, etc., each with its own dimension. A fundamental problem when working with such systems is to describe the set of components (the irreducible decomposition) of the solution set (algebraic set, or variety). In some areas of application, it is sufficient to compute one or more isolated (zero-dimensional) solutions. However, there are times when it is necessary to have knowledge of the entire solution set, in all dimensions. One such application is in the field of kinematics, where the dimensions of solution components indicate the number of degrees of freedom in various configurations of mechanisms [31].

Recently, numerical homotopy methods have been developed to carry out this decomposition, resulting in a numerical irreducible decomposition. Given a point **p** on at least one irreducible component, a key difficulty in these methods is the determination of the dimension(s) of the component(s) on which **p** sits. Several

algorithms to compute this information have been suggested in recent years, but all have fundamental drawbacks. This paper provides a novel method that does not suffer from these drawbacks.

More technically, given a numerical approximation of a point $\mathbf{p}$ on the algebraic set $V = \{x \in \mathbb{C}^N \mid F(x) = 0\}$ for a set of polynomials $F := \{F_1(x), \ldots, F_n(x)\} \subset \mathbb{C}[x_1, \ldots, x_N]$, it is of significant computational value to know the maximum dimension of the irreducible components containing $\mathbf{p}$, called the local dimension at $\mathbf{p}$ and 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 [21] and, more specifically, on the numerical approach of Dayton and Zeng [8] for computing multiplicities.

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 algebraic geometry, including the most fundamental computation: that of the numerical irreducible decomposition. For example, Section 3.3 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 [9, 13, 14] is well over an order of magnitude less expensive.

Section 1 provides a brief overview of the basic definitions and concepts needed for the remainder of the article. In particular, we present some background on numerical algebraic geometry [31] and we present the Dayton-Zeng approach [8, 36] to numerically computing multiplicities.

The algorithms themselves are presented in Section 2.1. 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 strictly increase 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 increase without bound. The number of paths $\nu_{\mathbf{p}}$ ending at $\mathbf{p}$ for standard homotopies used to compute $\mathbf{p}$ is an upper bound for $\mu_{\mathbf{p}}$. Given such an upper bound $\nu_{\mathbf{p}}$ for $\mu_{\mathbf{p}}$, we have a simple test to check whether $\mathbf{p}$ is isolated:

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

$$\widehat{k} := \min\{k \geq 1 \mid T_k = T_{k-1} \text{ or } T_k > \nu_{\mathbf{p}}\}.$$

2. Then $\mathbf{p}$ is isolated if and only if $T_{\widehat{k}} \leq \nu_{\mathbf{p}}$ if and only if $T_{\widehat{k}} = T_{\widehat{k}-1}$.

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 $\mathbf{p}$.

At first sight, it seems strange to hypothesize, that for the given point $\mathbf{p} \in V$, we have a positive integer, which is (in the case that $\mathbf{p}$ is isolated) an upper bound for the multiplicity of the system F at $\mathbf{p}$. In fact, such a number is typically the byproduct of the algorithms that numerically compute $\mathbf{p}$.

For example, assume that we are finding the isolated solutions of the system $F_1, \ldots, F_n$ on $\mathbb{C}^N$. If $n < N$, there are no isolated solutions, so we are in the situation $n \geq N$.

If $n = N$, then many homotopies $H(x, t) = 0$ may be used to solve the system. For example, the classical homotopies starting from general multihomogeneous systems for $t = 1$, have the property for an isolated solution $\mathbf{p}$ of F that the number of paths going to $\mathbf{p}$ as $t \to 0$ equals the multiplicity of the system F at $\mathbf{p}$. This classical result follows from [30, Lemma A.1] and the paragraph following its proof. See also [24, §5].

In the case that $n > N$, then the usual approach [31, Chapter 13.5] to finding isolated solutions of F is to first find isolated solutions of a randomized system

$$G(z) := \begin{bmatrix} G_1(z_1, \ldots, z_N) \\ \vdots \\ G_N(z) \end{bmatrix} := \begin{bmatrix} I_N & A \end{bmatrix} \cdot \begin{bmatrix} F_1(z_1, \ldots, z_N) \\ \vdots \\ F_n(z_1, \ldots, z_N) \end{bmatrix} = 0$$

where I_N is the $N \times N$ identity matrix and A is a random $((n - N) \times N)$-matrix. General theory tells us that each isolated multiplicity μ solution of $F = 0$ is an isolated solution of G of multiplicity at least μ. The cascade homotopy of [24] has the above property.

It is worth noting that the lack of such an upper bound for the multiplicity in the complex analytic case is the reason the results of this article, which are true (with pretty much verbatim arguments) for solution sets of a system of complex analytic (not necessarily algebraic) functions, are not useful in that situation.

Section 1.2 describes the method of [8] and provides details on creating an efficient implementation. Section 2.2 gives the local dimension algorithms. Examples are presented in Section 3 to illustrate the new methods.

Previous to this article the only theoretically rigorous numerical algorithm to compute the local dimension of an algebraic set V at a point was the *global* algorithm of first computing the full numerical irreducible decomposition of Sommese, Verschelde, and Wampler, and then using one of their membership tests to determine which components of V the point is on.

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 *local* 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 $\mathbf{p}$, of dimension less than or equal to $N - k$, will meet V in no other points in a neighborhood of $\mathbf{p}$; and
- a general linear space L through $\mathbf{p}$, of dimension greater than $N - k$, will meet V in points in a neighborhood of $\mathbf{p}$,

Sommese and Wampler [29, §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 [15], Kuo and Li present a useful 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 a polynomial system. 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 3.2 (consisting of two cubic equations in two variables) shows: solution sets with multiple branches going through a point may well lead that method to give false answers.

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, 10, 12].

Section A provides the specific commutative algebra results which are used in the development and implementation of the algorithms of Section 2. As is common with results about polynomial systems, we can work with either the given system of polynomials or a system consisting of the homogenenizations of the polynomials. Though we work directly with the polynomials in the article proper, it is an easy process to modify the algorithms and accompanying theory so as to work in the homogeneous setting.

1. Background material. In this section we collect basic definitions and concepts from algebraic geometry and give a description of the Dayton-Zeng multiplicity matrix sufficient for this article's needs.

1.1. Background from numerical algebraic geometry. A general reference for the material presented here is [31]. 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 subset V° of V consisting of manifold points is an open subset of V that is dense in V in the usual topology on $\mathbb{C}^N$. The algebraic set V is said to be *irreducible* if V° is connected. An irreducible algebraic set is called a *variety*. For an arbitrary algebraic set V, there are finitely many connected components $V_1^\circ, \ldots, V_r^\circ$ of V°. Setting V_i equal to the closure $\overline{V_i^\circ}$, the decomposition

$$V = V_1 \cup \ldots \cup V_r$$

is called the irreducible decomposition of the algebraic set V. For this decomposition $V_i \not\subset \bigcup_{j \neq i} V_j$ for all i. The dimension of V_i is defined to be the dimension of V_i°, which in turn is defined as the dimension of the complex tangent space of V_i° at any point. The dimension of the algebraic set V is defined to be the largest dimension of the varieties appearing in its irreducible decomposition. The dimension $\dim_{\mathbf{p}}(V)$ of V at a point $\mathbf{p} \in V$ is the maximum of the dimensions of the components containing $\mathbf{p}$.

Numerically determining the decomposition of an algebraic set into varieties is a fundamental problem in numerical algebraic geometry and serves as crucial data for other computations. Algorithms of Sommese, Verschelde, and Wampler that accomplish this decomposition are presented in [25, 27, 28] and described with full background details in [31]. The computation of the numerical irreducible decomposition has been implemented in the numeric/symbolic systems Bertini [2] and PHCpack [33]. 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 [16], Hompack [34], and PHoM [11]).

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. A good reference for this is [23]. For an algebraic set, V, the basic algorithms of numerical algebraic geometry produce discrete data in the form of a *witness point set* [25, 31]. 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 junk-point filter. 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 complementary dimension (coming from the flag). Classical introductions to continuation methods can be found in [1, 22]. For an overview of numerical algorithms and techniques for dealing with systems of polynomials, see [19, 31, 32]. For details on the cascade algorithm, see [24, 31].

The polynomial system can be used to attach a positive integer, known as the *multiplicity*, to each irreducible component of its corresponding algebraic set. 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 [17, 18] and 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* $\mathbf{p}$, i.e., determine the local dimension $\dim_{\mathbf{p}}(V)$ of V at $\mathbf{p}$. 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. 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 *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 $(x - \mathbf{p})^\alpha F_i$, where

- $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n) \in (Z_{\geq 0})^n$ so that $(x - \mathbf{p})^\alpha = (x_1 - \mathbf{p}_1)^{\alpha_1} \ldots (x_n - \mathbf{p}_n)^{\alpha_n}$;
- 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

$$M_{k+1} = \begin{bmatrix} M_k & A_k \\ 0 & B_k \end{bmatrix}. \tag{1.1}$$

The structure of this matrix can be exploited to create an efficient implementation of this method, described in Section 1.2.2.

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 $\mathbf{p} = (0,0)$, 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 $\frac{\partial^{i+j}}{\partial x_1^i \partial x_2^j}$. 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
$x_1 F_1$	0	0	0	1	-1	0	1	0	0	0
$x_1 F_2$	0	0	0	1	-1	0	0	0	1	0
$x_2 F_1$	0	0	0	0	1	-1	0	1	0	0
$x_2 F_2$	0	0	0	0	1	-1	0	0	0	1
$x_1^2 F_1$	0	0	0	0	0	0	1	-1	0	0
$x_1^2 F_2$	0	0	0	0	0	0	1	-1	0	0
$x_1 x_2 F_1$	0	0	0	0	0	0	0	1	-1	0
$x_1 x_2 F_2$	0	0	0	0	0	0	0	1	-1	0
$x_2^2 F_1$	0	0	0	0	0	0	0	0	1	-1
$x_2^2 F_2$	0	0	0	0	0	0	0	0	1	-1

1.2.1. The Dayton-Zeng multiplicity method. The methods of [8] and [5] both compute a sequence of nonnegative integers $\mu(k)$ that converge to the multiplicity of $I = (F_1, F_2, \ldots, F_n)$ at $\mathbf{p}$ 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 §2.1 will consider the input system as defining an object in affine space. This algorithm will loop over a dimension k, with the main numerical computation for each k given by Algorithm 0.

Algorithm 0. $multiplicity(\{F_1, F_2, \ldots, F_n\}, \mathbf{p}, k; \mu(k))$

<u>**Input**</u>:
- $\{F_1, F_2, \ldots, F_n\}$: a set of polynomials in the variables $z_1, \ldots, z_N$.
- $\mathbf{p} = (p_1, \ldots, p_N)$: a point on $V(F_1, F_2, \ldots, F_n) \subset \mathbb{C}^N$.
- 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 and evaluate M_k at $\mathbf{p}$.
Compute $\mu(k) = \text{nullity}(M_k)$.

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.

1.2.2. 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 the second step of Algorithm 0). 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 step of Algorithm 0. For $\alpha, \beta \in (Z_{\geq 0})^n$, define
- $\alpha! = \alpha_1! \alpha_2! \ldots \alpha_n!$,
- 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

$$\partial^\alpha((x - \mathbf{p})^\beta)(\mathbf{p}) = \begin{cases} \alpha!, & \text{if } \alpha = \beta, \\ 0, & \text{otherwise.} \end{cases} \tag{1.2}$$

Applying the Leibnitz Rule, we have that

$$\partial_\alpha((x - \mathbf{p})^\beta F_j)(\mathbf{p}) = \begin{cases} \partial_{\alpha-\beta} F_j(\mathbf{p}), & \text{if } \beta \leq \alpha, \\ 0, & \text{otherwise.} \end{cases} \tag{1.3}$$

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. Additionally, exploiting the embedding of M_k in M_{k+1} shown in Eq. 1.1 greatly speeds evaluation.

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.

Additionally, an orthonormal basis for the null space of M_k can be used to expedite the computation of an orthonormal basis for the null space of M_{k+1}. See [8, §5] for a detailed discussion of a numerical method that performs this computation using QR factorizations and the very efficient rank-revealing method presented in [20].

Finally, increasing the accuracy of the approximated point $\mathbf{p}$ and rerunning the computation using higher precision increases the security of numerical methods. In Bertini, $\mathbf{p}$ is an endpoint of a homotopy path so that rerunning an endgame [31] with higher precision yields a more accurate approximation. More specifically, Bertini computes $\mathbf{p}$ using two different levels of accuracy and precision, and performs the rank/null space computations simultaneously on the two corresponding versions of each M_k.

2. 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.
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.

2.1. The algorithms. The local dimension algorithm and the related algorithms presented in the following pages rely on the method for finding the multiplicity of a polynomial system at a point $\mathbf{p}$ (see §1.2) and rely on a positive number $\nu_{\mathbf{P}}$, which is an upper bound for the multiplicity if the point $\mathbf{p}$ is isolated. Computing such a bound was described in detail in the Introduction. The operation

multiplicity($\{F_1, \ldots, F_n\}, \mathbf{p}, k$) returns the level k approximation of the multiplicity of the polynomial system $\{F_1, \ldots, F_n\}$ at $\mathbf{p}$; see §1.2.1 for details. Two different options for computing the multiplicity can be found in [8, 5]. An upper bound on the multiplicity and Theorem A.1 provide 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 on the multiplicity. If $\mathbf{p}$ is isolated then the sequence of approximations will stabilize to the correct multiplicity (bounded by the upper bound on the multiplicity). 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*($\{F_1, F_2, \ldots, F_n\}, \mathbf{p}, B$; *is_isolated*$_\mathbf{p}, \mu_\mathbf{p}$)

Input:
• $\{F_1, F_2, \ldots, F_n\}$: a set of polynomials in the variables $z_1, \ldots, z_N$.
• $\mathbf{p} = (p_1, \ldots, p_N)$: a point on $V(F_1, F_2, \ldots, F_n) \subset \mathbb{C}^N$.
• B: an upper bound on the multiplicity of $(F_1, F_2, \ldots, F_n)$ at $\mathbf{p}$ 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 $\mathbf{p}$ 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 polynomials in the variables $z_1, \ldots, z_N$.
• $\mathbf{p} = (p_1, \ldots, p_N)$: a point on $V(F_1, F_2, \ldots, F_n) \subset \mathbb{C}^N$.

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

Algorithm:
Choose $L_1, \ldots, L_N$, random linear polynomials through $\mathbf{p}$ and set $k := -1$.
do

$k := k + 1$.
Compute an upper bound B on the multiplicity of $\{F_1, F_2, \ldots, F_n, L_1, \ldots, L_k\}$
      at $\mathbf{p}$. See the Introduction for details.
$(is\_isolated_k, \mu_k) := is\_isolated(\{F_1, F_2, \ldots, F_n, L_1, \ldots, L_k\}, \mathbf{p}, B)$.
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 homotopy membership test was first described in [26].

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 polynomials in the variables $z_1, \ldots, z_N$.
• $\mathbf{p} = (p_1, \ldots, p_N)$: a point on $V(F_1, F_2, \ldots, F_n) \subset \mathbb{C}^N$.
• 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 $(\dim_{\mathbf{p}}, \mu_{\mathbf{p}}) := local\_dimension(\{F_1, F_2, \ldots, F_n\}, \mathbf{p})$.
for $j = 0, \ldots, \dim_{\mathbf{p}}$
      $m :=$ number of irreducible components of dimension j in W_j.
      for $k = 1, \ldots, m$
            Compute $is\_member := membership(\{F_1, F_2, \ldots, F_n\}, W_{jk}, \mathbf{p})$.
            if $is\_member = True$
                  $J := J \cup \{(j, k)\}$.

2.2. 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 (see § 1.2). The implementation makes use of the upper bound from homotopy continuation described in the Introduction, and the witness set membership test described in [26, 31] was used for the algorithm *membership*.

3. Examples. The local dimension computations were run using the implementation described in § 2.2 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 3.3 were run on a cluster consisting of a manager that uses one core of a Xeon 5410 (quad-core 2.33 GH) processor and 8 computing nodes each containing two Xeon 5410 processors running 64-bit Linux.

In Section 3.1, Bertini is used to compute the multiplicity structure of an isolated solution, which is a significant part of the local-dimension computation. ApaTools [35] is a Maple toolbox that provided the first implementation of the methods described in [8]. ApaTools is designed for educational purposes: it is a very good tool for understanding the algorithm. 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. In Section 3.2, an example is presented to illustrate potential difficulties arising in the heuristic local-dimension approach of [15]. Finally, Section 3.3 presents examples to demonstrate the application of the local dimension test to junk-point filtering.
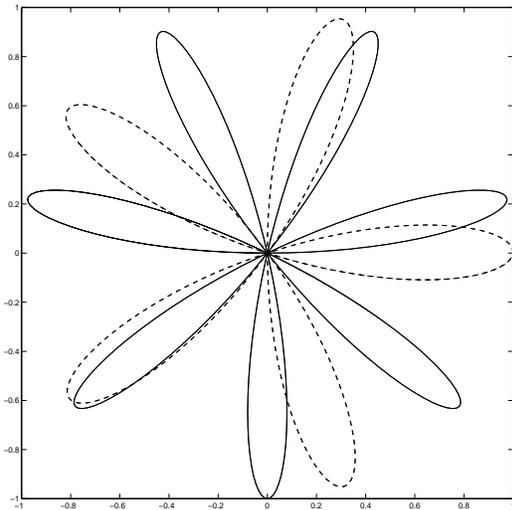

FIG. 3.1. *Rhodonea curves S_7 and $\widehat{S}_5$.*

3.1. A class with isolated solutions. Rhodonea curves, such as those displayed in Figure 3.1, 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 3.2 and 3.3 display the sequence $\mu(k)$ and the timings for the implementation in Bertini for various (odd) values of m and n.

3.2. A positive-dimensional example. The Rhodonea curve S_1, as described in § 3.1, 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

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

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

TABLE 3.2. *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

TABLE 3.3. *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 $L = \{x = 0\}$. 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 [15] 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.

3.3. A collection of high-dimensional examples. When computing a witness superset, the cascade algorithm [24, 31] 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 [9, 13, 14]. It is well-known that, for $m \geq 3$, the solution set of G_m consists of components of different dimensions. Table 3.4 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 [26] 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.

In Bertini, a parallel junk-point filtering is achieved using a dynamic distribution of the points since each point can be handled independently. Table 3.5 compares the timings for computing a numerical irreducible decomposition in paral-

lel for G_m, $7 \leq m \leq 9$.

4. 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.

Appendix A. Theoretical justification of the algorithms. We assume the reader is familiar with basic notions of commutative algebra such as polynomial ring, ideal, localization at a point, and Nakayama's Lemma. Good references are [7, 10, 12]. The reference [10, §A.8] has a particularly good discussion of multiplicity, and, in particular (see [10, Theorem A.8.13]), of the behavior of the multiplicity of a point q on an algebraic set $X \subset \mathbb{C}^N$ under slicing by a generic linear space $L \subset \mathbb{C}^N$ containing q and of codimension $\dim X$.

Let $R = \mathbb{C}[z_1, z_2, \ldots, z_N]$ denote the ring of all polynomials in the variables $z_1, z_2, \ldots, z_N$ with complex coefficients. Let $F(z) := \{F_1(z), F_2(z), \ldots, F_r(z)\}$ be a set of elements of R, i.e., a system of r polynomials with complex coefficients on $\mathbb{C}^N$.

Let $F(q) = 0$, i.e., let q denote a common root of the F_i. Let A denote the localization of R at q, i.e., A is the ring consisting of all rational functions $f(z)/g(z)$ such that f and g have no non-constant polynomial factors and $g(q) \neq 0$. The functions F define an ideal $I = (F_1, F_2, \ldots, F_r)$ in A with the quotient ring A/I finite dimensional as a complex vector space if and only if q is an isolated solution of $F(z) = 0$, in which case the dimension

$$\mu := \dim_{\mathbb{C}} A/I$$

is called the *multiplicity* of q as a solution of the system $F(z) = 0$.

	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

TABLE 3.4. *Comparison for computing a numerical irreducible decomposition for G_m, in seconds*

	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

TABLE 3.5. *Comparison for computing a numerical irreducible decomposition in parallel for G_m, in seconds*

Though the local ring A is used, any of its completions, e.g., either the convergent or the formal complex power series at q, would yield the same results and the same multiplicities. Since we always work with finite dimensional quotients of local rings, the results in the dimension test of this article work for systems of holomorphic functions. The difficulty is that only for systems of algebraic functions do we have algorithms that give a solution q of the system with a positive integer ν, which bounds the multiplicity if q turns out to be an isolated solution of the system of functions.

Working with convergent power series would be closer to the numerical analysis, but the use of the localization of polynomial rings requires less background.

Let $\mathcal{A} := A/I$ be the local ring of the algebraic set defined by I at q. Denote the maximal ideal of q in A by M_q and the maximal ideal of $\mathcal{A}$ by $\mathfrak{m}$. Note that M_q is the ideal generated by the linear functions $L_1, \ldots, L_N$ vanishing at q. Let $J_k = (I, M_q^k)$. Since q is an isolated solution of $F_1, \ldots, F_r, L_1, \ldots, L_N$, the ring A/J_k is finite dimensional as a complex vector space. Note that there is a natural isomorphism

$$A/J_k \approx \mathcal{A}/\mathfrak{m}^k$$

induced by the surjection $A \to \mathcal{A}$. By defining $\mu_k := \dim_{\mathbb{C}} A/J_k$ and noting that $J_{k+1} \subset J_k$ for all $k \geq 1$, we have

$$\mu_k \leq \mu_{k+1} \tag{A.1}$$

for all $k \geq 1$.

Note that there exists a $K > 0$ such that $\mathfrak{m}^k = 0$ in $\mathcal{A}$ for $k \geq K$ if and only if $\mathcal{A}$ is finite dimensional as a complex vector space, which, as noted above, is equivalent to q being an isolated solution of $F(z)$. This is also equivalent to $M_q^k \subset I$ for $k \geq K$, which in turn is equivalent to $J_k := (I, M_q^k) = I$ for $k \geq K$.

Thus if q is an isolated solution of F, we have that $\mu = \mu_k = \dim_{\mathbb{C}} A/J_k$ for all sufficiently large k.

Since $\mu_k = \mu_{k+1}$ if and only if $\mathfrak{m}^k = \mathfrak{m} \cdot \mathfrak{m}^k$, we conclude by Nakayama's Lemma that this happens only if

$$\mathfrak{m}^k = 0 \ \text{ or equivalently if } \ J_k = J_{k+1}.$$

The converse is immediate.

Putting the above statements together we have the following theorem.

THEOREM A.1. *Let $F(z) := \{F_1(z), F_2(z), \ldots, F_r(z)\}$ be a system of polynomials with complex coefficients on $\mathbb{C}^N$, i.e., the F_i are elements of the polynomial ring $R := \mathbb{C}[z_1, \ldots, z_N]$. Let q be a solution of this system. Let I denote the ideal $(F_1, \ldots, F_r) \subset R$, let $M_q \subset R$ denote the ideal generated by linear functions vanishing at q, and let $J_k := (I, M_q^k)$ for $k \geq 1$. Let $\mu_k := \dim_{\mathbb{C}} R/J_k$. Then*

  1. *$\mu_k \leq \mu_{k+1}$ for all $k \geq 1$;*
  2. *there is a $k \geq 1$ such that $\mu_k = \mu_{k+1}$ if and only if q is an isolated solution of $F(z) = 0$. In this case $\mu_k = \mu$, where μ is the multiplicity of q as a solution of $F(z) = 0$.*

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] P. Diaconis, D. Eisenbud, and B. Sturmfels, Lattice walks and primary decomposition, In *Mathematical essays in honor of Gian-Carlo Rota (Cambridge, MA, 1996)*, volume 161 of *Progr. Math.*, pages 173–193, Birkhäuser Boston, Boston, MA, 1998.

[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 J. Shapiro, Primary decomposition of lattice basis ideals, *J. Symbolic Comput.*, 29(4-5):625–639, 2000. Symbolic computation in algebra, analysis, and geometry (Berkeley, CA, 1998).

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

[15] 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.

[16] 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, *Computing*, 83 (2008), pp. 109–133.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

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

[22] A. Morgan, *Solving polynomial systems using continuation for engineering and scientific problems*, Classics in Applied Mathematics 57, SIAM Press, Philadelphia, 2009.

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

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

[25] 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.

[26] A. J. Sommese, J. Verschelde, and C. W. Wampler, Numerical Irreducible decomposition using projections from points on components, in *Symbolic Computation: Solving Equations in Algebra, Geometry, and Engineering*, ed. by Green, Hosten, Laubenbacher, and Power, Contemporary Mathematics, 206 (2001), pp. 37–51.

[27] A.J. Sommese, J. Verschelde and C.W. Wampler, Using Monodromy to Decompose Solution

Sets of Polynomial Systems into Irreducible Components, in Proceedings of the 2001 NATO Advance Research Conference, Eilat, Israel, on Applications of Algebraic Geometry to Coding Theory, Physics, and Computation, edited by C. Ciliberto, F. Hirzebruch, R. Miranda, and M. Teicher, (2001), pp 297–315.

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

[29] 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.

[30] A.J. Sommese, J. Verschelde, and C.W. Wampler, Homotopies for intersecting solution components of polynomial systems, SIAM Journal on Numerical Analysis, 42 (2004), 1552–1571.

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

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

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

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

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

[36] Z. Zeng, The closedness subspace method for computing the multiplicity structure of a polynomial system, to appear in Interactions of Classical and Numerical Algebraic Geometry, ed. by D. Bates, G. Besana, S. Di Rocco, and C. Wampler, Contemporary Mathematics, 2009.