Finding All Real Points of a Complex Curve

Ye Lu* Daniel J. Bates† Andrew J. Sommese‡
Charles W. Wampler§

August 9, 2006

Abstract

An algorithm is given to compute the real points of the irreducible one-dimensional complex components of the solution sets of systems of polynomials with real coefficients. The algorithm is based on homotopy continuation and the numerical irreducible decomposition. An extended application is made to Griffis-Duffy platforms, a class of Stewart-Gough platform robots.

2000 Mathematics Subject Classification. Primary 65H10; Secondary 65H20, 14Q99.

Key words and phrases. Homotopy continuation, numerical algebraic geometry, real polynomial systems.

In this article we give a numerical algorithm to find the real zero- and

*Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556-4618, USA *Email:* ylu4@nd.edu. This material is based upon work supported by the National Science Foundation under Grant No. 0410047.

†Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556-4618, USA *Email:* dbates1@nd.edu. This material is based upon work supported by the National Science Foundation under Grant No. 0410047.

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

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

one-dimensional parts of the solution set of a system

$$f(x) := \begin{bmatrix} f_1(x_1, \ldots, x_N) \\ \vdots \\ f_n(x_1, \ldots, x_N) \end{bmatrix} = 0 \tag{1}$$

of n polynomials on $\mathbb{R}^N$. The methods of numerical algebraic geometry are almost exclusively aimed at detecting the complex solution components of such a system: good references are [16, 17, 18, 19, 11, 22, 23]. The algorithms of this article make use of some of those known methods from the complex case.

The isolated solutions on $\mathbb{C}^N$ that lie in $\mathbb{R}^N$ may be found by first finding all the isolated solutions of $f(x) = 0$ and filtering out those with real coordinates. If the system $f(x) = 0$ depends on parameters and we need to solve the system repeatedly for different values of the parameters, it can be argued [23] that the extra work of finding all the isolated complex solutions is unavoidable.

Finding all the isolated solutions of $f(x) = 0$ on $\mathbb{R}^N$, including those on higher-dimensional solution sets, is a more difficult problem. For example, consider the simple example [5, Example 13.6] of the polynomial of degree $2d$

$$\Pi_{i=1}^{d}(x - i)^2 + \Pi_{i=1}^{d}(y - i)^2 = 0 \tag{2}$$

with d^2 isolated real solutions. These points are not isolated over the complex numbers, but are instead singular points of a complex curve.

The situation for positive-dimensional solution sets is far more complicated. Unlike the case of complex solution sets, an "irreducible component" is not necessarily connected. For example, the set of zeroes of the irreducible polynomial

$$y^2 - x(x - 1)(x - 2)$$

is irreducible over the complex numbers, i.e., the common zeroes on $\mathbb{C}^2$ forms a connected set with the property that the set of nonsingular points is connected. In contrast, the solutions on $\mathbb{R}^2$ are easily seen to consist of two one-dimensional curves.

Moreover the difficulty of one-dimensional solution sets over the real numbers that are part of higher-dimensional solution sets becomes more complicated. For example, consider Whitney's Umbrella, i.e., $V(p)$, where $p(x, y, z) = x^2 - y^2 z$. Over the complex numbers, the z-axis is contained in the irreducible two-dimensional solution set. Over the real numbers,

$$\{(0, 0, z) \mid z \in \mathbb{R}, z < 0\}$$

is a one-dimensional part of the set with the rest of the z-axis contained in a two-dimensional set.

There is the potential for confusion when describing the "dimension" of a set of real points that lies on a potentially higher-dimensional set of complex points. For example, the real solutions of real dimension zero in (2) lie on complex components of complex dimension one. In general, a real solution set of real dimension k could lie on a complex solution set of any complex dimension greater than or equal to k. For the purposes of this article, unless otherwise stated, the dimension of a set of real solutions will refer to the complex dimension of the complex solution set upon which it lies.

For applications to mechanisms, one-dimensional sets of real points have a particular importance, e.g., see the extended example of a Stewart-Gough platform robot in §4. Moreover, for one-dimensional sets, computations yielding information useful in applications can be carried out in a relatively straightforward manner.

Given a polynomial system as in (1), the algorithms of §3 will let us

1. compute isolated real points contained in the union $Z_{1,\mathbb{R}}$ of the one-dimensional complex components of $V(f)$;

2. compute the one real-dimensional connected components of $Z_{1,\mathbb{R}}$; and

3. determine whether a given real solution belongs to $Z_{1,\mathbb{R}}$, and if it does, determine which connected components and which irreducible components it belongs to.

In rough terms, we take a generic real projection and compute points where the projection is degenerate. From that set, we obtain a decomposition similar to that using a Morse function on a manifold. Symbolic algorithms along the same general lines exist, e.g., [1, 3, 9].

The numerical approach starts with the witness sets of [16, 18, 23]. Curves on which the Jacobian of the projection is singular present difficulties, which we deal with by deflation along the lines of [23, §13.3.2 and §15.2.3] generalizing the case of isolated points [4, 10, 13].

In §1, we give a general discussion of real points of solution sets of polynomial systems. Here we give the method to obtain the connected components of real points of complex curves. We also explain the data structure we use to represent real points of one-dimensional complex algebraic sets. The approach of §1 works with reduced complex curves, but curves arising in practice are in many cases of multiplicity greater than one (in §4 we

give an example of this that arose in the study of Stewart-Gough platform robots). In §2, we give a discussion of deflation as that technique provides a means of dealing with components of multiplicity greater than one. In §3, we present explicit numerical algorithms for the methods of §1. Finally, several applications of the algorithm of §3 may be found in §4.

1 Real points of complex algebraic curves

Throughout this discussion we have a system of polynomials

$$f(x) := \begin{bmatrix} f_1(x_1, \ldots, x_N) \\ \vdots \\ f_n(x_1, \ldots, x_N) \end{bmatrix} = 0 \tag{3}$$

on $\mathbb{C}^N$ with real coefficients. We are interested in describing the set of real solutions $Z_{\mathbb{R}} := V_{\mathbb{R}}(f) \subset \mathbb{R}^N$. The set of solutions $Z := V(f) \subset \mathbb{C}^N$ is taken to itself under conjugation $\kappa : \mathbb{C}^N \to \mathbb{C}^N$, and $\mathbb{R}^N$ is the set of fixed points of κ_Z, the restriction of κ to Z.

We know that the complex solution set $V(f)$ is of the form

$$Z := \bigcup_{i=0}^{\dim V(f)} Z_i = \bigcup_{i=0}^{\dim V(f)} \left(\bigcup_{j \in \mathcal{I}_i} Z_{i,j} \right) \tag{4}$$

where Z_i is the union of all i-dimensional irreducible components of $V(f)$ and $Z_{i,j}$ for $j \in \mathcal{I}_i$ are the finite number of distinct irreducible components of $V(f)$. Recently in [16, 17, 18], new techniques have been developed to numerically decompose Z into objects known as witness sets which correspond to the irreducible components $Z_{i,j}$. With the help of this decomposition, we can give a numerical description of the real solution set $Z_{\mathbb{R}}$.

Note that κ takes each component $Z_{i,j}$ to a component $Z_{i,k}$ where $Z_{i,j}$ may or may not be equal to $Z_{i,k}$. This gives a bijection of the index set $\mathcal{I}_i$ to itself, which by abuse of notation, we also denote by κ. Thus κ takes $Z_{i,j}$ to $Z_{i,\kappa(j)}$.

Given an reduced quasiprojective algebraic set X we denote the singular points of X by $\mathrm{Sing}(X)$ and the smooth points, $X \setminus \mathrm{Sing}(X)$, by $\mathrm{reg}(X)$.

Lemma 1.0.1. *Let Z_1 and κ be as above. Given a point $x \in \mathrm{reg}(Z_1)$ with $\kappa(x) = x$, there is a complex neighborhood $U \subset \mathrm{reg}(Z_1)$ of x such that the solutions of $\kappa(z) = z$ on U is a smooth connected curve.*

Proof. It is an easy fact that there is a neighborhood U of x which is κ-equivariantly diffeomorphic to a neighborhood of the origin in the real tangent space $T_{Z_1|x}$ of Z_1 at x. The induced action of κ on the two-dimensional real vector space $T_{Z_1|x}$ is the same as conjugation on $\mathbb{C}$. □

If $Z_{i,\kappa(j)} \neq Z_{i,j}$, then any real points of $Z_{\mathbb{R}}$ contained in $Z_{i,j}$ are the real points of the algebraic set $Z_{i,j} \cap Z_{i,\kappa(j)}$ with $\dim Z_{i,j} \cap Z_{i,\kappa(j)} < i$. This intersection can be computed numerically using the diagonal intersection technique of [19, 20]. The subject of this article is the curve case, $i = 1$, so any such intersections $Z_{1,j} \cap Z_{1,\kappa(j)}$ are isolated in the diagonal intersection, even though they might be contained in some other real one-dimensional component of the system.

This leaves only the case when $\kappa(j) = j$. Thus we have an irreducible curve C with $\kappa(C) = C$. Throughout the following discussion, if C appears with multiplicity greater than one as a solution component of $f(x) = 0$, we take C to mean the underlying reduced algebraic set. We denote the real subset of C as $C_{\mathbb{R}} = C \cap \kappa(C)$. Note that κ takes singular points to singular points. Therefore we are reduced to finding

1. the singular points of C and checking which are real, i.e., invariant under κ;

2. real points of C that are not singular, i.e., the smooth points of C invariant under κ; and

3. showing how the sets in 1) and 2) fit together.

By Lemma 1.0.1, the set in 2) is a smooth real curve.

Our approach to describing $C_{\mathbb{R}}$, the real points of C, is to take a general linear projection $\pi : \mathbb{C}^N \to \mathbb{C}$ with real coefficients. Thus the restriction of π to the reals, $\pi_{\mathbb{R}^N}$, is a linear map from $\mathbb{R}^N \to \mathbb{R}$. By the Noether normalization theorem, we know that $\pi : C \to \mathbb{C}$ is proper and thus so also is $\pi_{C_{\mathbb{R}}} : C_{\mathbb{R}} \to \mathbb{R}$. We think of $V(\pi(x) - t)$ as t ranges over $\mathbb{R}$ as a hyperplane that sweeps out the curve. For general values of $t \in \mathbb{R}$, this hyperplane intersects $C_{\mathbb{R}}$ in isolated points, and on an open interval $U \subset \mathbb{R}$ containing such a t, the corresponding segments of the curve $\pi^{-1}_{C_{\mathbb{R}}}(U)$ can be swept out numerically by continuation in t.

The numerical sweep of any particular curve segment fails at points where the tangent to the curve is either (a) not unique or (b) unique but lying wholly inside the sweep hyperplane. Together, these are the points

where $d\pi_C$ is of corank > 1. Let us denote the set of all such rank-degenerate points as $\mathcal{B}^*$. For our purposes, it is acceptable if $\mathcal{B}^*$ is expanded by including a finite number of points where $d\pi_C$ is corank 1. These may enter our computations via the processes of randomization and deflation, as described in § 1.1. By definition, $\mathrm{Sing}(C) \subset \mathcal{B}^*$, so embedded points of C are included in $\mathcal{B}^*$. Let the real points of $\mathcal{B}^*$ be denoted $\mathcal{B}^*_{\mathbb{R}}$.

It is convenient to expand this set to a finite set $\mathcal{B} \supset \mathcal{B}^*$ by also including all points of C lying in the same fiber as some point of $\mathcal{B}^*$, that is, $\mathcal{B} = C \cap \pi^{-1}(\mathcal{B}^*)$. Similarly, let $\mathcal{B}_{\mathbb{R}} = C_{\mathbb{R}} \cap \pi^{-1}(\pi(\mathcal{B}^*_{\mathbb{R}}))$. Since $\pi_{C\setminus\mathcal{B}}$ is a finite covering map, the structure of $\pi^{-1}(\mathbb{R})$ is simple. It consists of the union of the finite set $\mathcal{B}_{\mathbb{R}}$ with the set $\pi^{-1}(\mathbb{R}) \setminus \mathcal{B}_{\mathbb{R}}$, the latter of which consists of a finite number of connected sets e_k diffeomorphic to open intervals. We will refer to these connected sets as "edges."

Since open intervals of the line are simply connected, e_k is mapped diffeomorphically onto its image. Let $x_k \in e_k$ denote the unique point over a point $t_k \in \pi(e_k) \subset \mathbb{R}$. Since the real numbers have an ordering, each of the e_k has a direction of increase. By continuation from t_k in both directions, we can find, for each e_k, the points in $\mathcal{B}_{\mathbb{R}}$ which form the boundary ∂e_k of e_k. For this reason, the points of $\mathcal{B}_{\mathbb{R}}$ are referred to as "boundary points." Since we work in $\mathbb{C}^N$, it is possible that e_k maps onto an infinite interval. Therefore, we must add to our data structures a way of denoting such edges.[1]

This Morse-theoretic observation is important because by Lemma 1.0.1, each of the connected sets e_k in the union $\pi^{-1}(\mathbb{R}) \setminus \mathcal{B}_{\mathbb{R}}$ either consists of all real points or has no real points. Let us retain among these only those edges that are real, so that e_k, $k = 1, \ldots, n$, is the collection of all the real edges. Thus $C_{\mathbb{R}}$ consists of the union of the real edges with the boundary points $\mathcal{B}_{\mathbb{R}}$. Accordingly, we may define a data structure to represent $C_{\mathbb{R}}$ numerically as follows.

Definition 1.1. *A* Morse-like representation *of a real algebraic curve $C_{\mathbb{R}} \subset \mathbb{R}^N$ consists of*

1. *a generic linear projection $\pi : \mathbb{R}^N \to \mathbb{R}$,*
2. *a boundary point set $\mathcal{B}_{\mathbb{R}} = \{B_1, \ldots, B_m\}$, $B_i \in \mathbb{R}^N$ for all i; and*

[1] If we worked in $\mathbb{P}^N$, then $\overline{e_k}$ would always be compact, with the possibilities that some boundary points could be at infinity and some edges, homeomorphic to a circle, could have just one boundary point or no boundary at all.

3. *an edge set* $\mathcal{E} = \{E_1, \ldots, E_n\}$, *where for all* k, $E_k = \{x_k, \ell_k, r_k\} \in \mathbb{R}^N \times (\mathbb{Z} \cup \{-\infty\}) \times (\mathbb{Z} \cup \{+\infty\})$, *where*

   (a) $x_k \in e_k$ *over a point of* $\pi(e_k)$, *and*

   (b) B_{ℓ_k} *and* B_{r_k} *are the left and right endpoints of edge* e_k.

A value of $\ell_k = -\infty$ *means edge* e_k *extends to infinity on the left, and similarly* $r_k = +\infty$ *means* e_k *extends to infinity on the right.*

Since disjoint e_i and e_j have either identical or completely disjoint images, we can force $\pi(x_i) = \pi(x_j)$ whenever $\pi(e_i) = \pi(e_j)$. One could work instead with the set $\mathcal{B}^*_{\mathbb{R}}$ and the edges that connect them, but then the images of the edges would, in general, only partially overlap, which would complicate the bookkeeping.

To describe the real points in the full curve Z_1 instead of just a single component $C = Z_{1,j}$, one may construct a Morse-like representation for each self-conjugate component, and also find the isolated points where distinct complex conjugate components meet. If desired, one may also find the points where the real curves inside distinct components meet each other.

1.1 Extra boundary points from randomization

The most basic part of the numerical computation of a finite set $\mathcal{B}_{\mathbb{R}} \subset C_{\mathbb{R}}$ that includes all the real points where $d\pi(C)$ has corank > 1. Our methods for computing $\mathcal{B}_{\mathbb{R}}$ can introduce extra boundary points. These do not affect the validity of the data structure for representing the curve, they simply add more edges than necessary.

One way extra boundary points can arise occurs when a curve in $\mathbb{C}^N$ is given by more than $N-1$ equations. For example, consider the system $f = \{x^2, xy\}$. Then, $V(f)$ is just $V(x)$, a line, which could be described with just one edge. However, to deal with the excess of equations, we pick a random $\lambda \in \mathbb{R}$ to form a randomized system $f' = \{x^2 + \lambda xy\}$. This gives a $V(f')$ that contains both the original component $V(x)$ and an extraneous component $V(x + \lambda y)$. Any general points found on the extraneous component will not satisfy the original system, so they may be easily discovered and discarded. However, the set of boundary points $\mathcal{B}'_{\mathbb{R}}$ for the system f' consists of the point $(0, 0)$. This causes us to have two edges to describe $V(x)$, $e_1 = \{(0, y) \in \mathbb{R}^2 | y < 0\}$ and $e_2 = \{(0, y) \in \mathbb{R}^2 | y > 0\}$, instead of just one.

If we have a component of multiplicity at least two, we must reduce the multiplicity of the component to one before applying the basic method found

below. We discuss how to do so in §2. As in the case of randomization, this operation might also introduce some extra points into $\mathcal{B}_{\mathbb{R}}$, again without significant consequences.

1.2 A broader notion of conjugation

Conjugation in applications appears in different guises. For example, consider the conjugate-transpose operation that sends a matrix $A \in \mathbb{C}^{n\times n}$ to its Hermitian transpose A^*, i.e., sending that matrix A to its transpose A' and then taking the complex conjugate of every element of A'. In this case the "real points" are the Hermitian matrices. This can be put into the form of conjugation we consider in this article, by mapping points $A \in \mathbb{C}^{n\times n}$ to points

$$\left(A + A', \frac{A - A'}{\sqrt{-1}}\right) \in \mathbb{C}^{n\times 2n}.$$

Under this mapping, the Hermitian matrices are mapped isomorphically onto the set of real points of the image.

A special case of this occurs in planar mechanism theory, in which isotropic coordinates can be used to represent vectors in the plane [2, 26]. These are pairs $(z, \widehat{z}) \in \mathbb{C}^2$ that are "real" when z and $\widehat{z}$ are complex conjugates of each other, that is, real points are the stationary points of the generalized conjugation operation $\kappa : (z, \widehat{z}) \mapsto (\mathrm{conj}(\widehat{z}), \mathrm{conj}(z))$, where conj() means complex conjugation.

With minor adjustments, the techniques of this paper can be applied to determine real curves under any of these generalized forms of conjugation.

2 Deflation

Let C be an irreducible component of Z_1 as in the last section. To numerically deal with C in an efficient way, we need to use homotopy continuation to compute the points of $L_t \cap C$ where L_t is a one-parameter family of generic hyperplanes. This is straightforward when the multiplicity of C as a component of the solution set is one. Unfortunately, this assumption is not always true in practice, e.g., the motion path of the Stewart-Gough platform robot described in §4.

Fortunately, we can use the deflation technique to effectively replace such a component with a one-dimensional multiplicity one component of a system of polynomials on a higher-dimensional space. This is discussed for

isolated points in [10] and for components in [23, §13.3.2 and §15.2.3]. We do not give the details here. Let it suffice to say that given a system $f(x)$ and a generic point x^* on some irreducible component $Z \subset V(f)$ having multiplicity greater than one, deflation produces a new system, say $F(x, \xi)$ with a multiplicity one solution component Y, which projects under the map $(x, \xi) \mapsto (x)$ generically one-to-one onto Z with an isomorphism in a Zariski neighborhood of x^*. Deflation proceeds in stages, each stage guaranteed to reduce the multiplicity by at least one.

The main difficulties with deflation are that:

1. each stage of deflation roughly doubles the number of variables; and
2. to form the deflated system, one must determine the rank of the Jacobian matrix at a solution point.

For low-multiplicity singularities the increase in the number of variables is not severe. For components with high multiplicity, there can be difficulties, but this will likely be true with any approach to computing such components. More research in this direction is warranted.

The determination of rank can be difficult as the numerical approximation of a singular point is less accurate than at well-conditioned points. Research in "rank-revealing" algorithms is relevant to this difficulty [12]. Our current approach is to use adaptive multiprecision with a singular endgame to accurately compute the points of $C \cap L_t$ for a given value of t and so that the rank of the Jacobian matrix at such a point can be found accurately. If more than one stage of deflation is required, we would continue to use multiprecision on the intermediate deflated systems until the final deflation is obtained. (So far, we have only treated systems where one stage of deflation suffices.) The computational cost for the deflations pays off since the ranks stay constant over Zariski open dense sets of $t \in \mathbb{C}$.

In general, a multiplicity μ component may require up to $\mu - 1$ stages of deflation to fully reduce it, although a sharp bound was developed in [4]. Also, it should be noted that at any stage of deflation, say f_i, where the rank of the Jacobian matrix, $J_i = \partial f_i / \partial x$ is r, we restrict λ_i to a generic r-dimensional linear subspace to form the new system $f_{i+1} := \{f_i, J_i \lambda_i\}$. Only when $r = 0$ is λ_i constant; otherwise, new variables are introduced to parameterize the linear subspace. In this manner, the number of variables usually increases with each stage of deflation.

2.1 Extra boundary points from deflation

Let us see an example of how deflation reduces a multiplicity two curve in $\mathbb{C}^2$ to multiplicity one and in the process introduces extra boundary points into the description of the curve. Let $g = x_1^2 + x_2^2 - 1$, $f = (x_1^2 + x_2^2 - 1)^2$. The Jacobian matrix for f is

$$J := \partial f/\partial x = \left[\begin{array}{cc} 2x_1(x_1^2 + x_2^2 - 1) & 2x_2(x_1^2 + x_2^2 - 1) \end{array} \right],$$

which has rank zero on $V(f)$. In other words, the kernel of J is all of $\mathbb{C}^2$. Consequently, on $V(f)$, we may set $J\lambda = 0$ for

$$\lambda := \left[\begin{array}{c} \lambda_1 \\ \lambda_2 \end{array} \right],$$

a generic point in $\mathbb{C}^2$. This gives the deflated system

$$f_1(x) := \left[\begin{array}{c} f \\ J\lambda \end{array} \right] = \left[\begin{array}{c} (x_1^2 + x_2^2 - 1)^2 \\ (x_1^2 + x_2^2 - 1)(\lambda_1 2x_1 + \lambda_2 2x_2) \end{array} \right] = 0.$$

$V(x_1^2 + x_2^2 - 1)$ is a component of $V(f_1)$. Moreover, $\operatorname{rank}(\partial f_1/\partial x) = 1$ on a Zariski open subset of $V(g)$, specifically, on the complement of $V(\lambda_1 2x_1 + \lambda_2 2x_2)$. The exceptional points are where the tangent to $V(g)$ is orthogonal to the randomly selected direction λ. Since $\operatorname{rank}_{V(f_1)}(\partial f_1/\partial x) = \operatorname{codim} V(f_1)$, the component is multiplicity one and no further deflation is needed. The deflated system $f_1(x)$ is suitable for describing the real curve in $V(f)$, but extra boundary points $V(x_1^2 + x_2^2 - 1) \cap V(\lambda_1 2x_1 + \lambda_2 2x_2)$ will be included in $\mathcal{B}_{\mathbb{R}}$.

A Zariski open subset of the regular points of a deflated curve are guaranteed to project onto a dense Zariski-open subset of the original nonreduced curve. Singular points of the deflated curve still project to the original curve, but some may project to points that are not singular for the original curve. Including such boundary points in the Morse-like representation of the curve has the mild disadvantage of creating extra edges but also has the crucial advantage of allowing us to compute with the numerically stable deflated system instead of the original numerically unstable nonreduced system.

3 Algorithms

In this section, we give a method for finding all real solutions lying on the one-dimensional complex irreducible components in the solution set of a

polynomial system with real coefficients as in Eq. 1. We also give a membership algorithm that decides whether a given point lies on the real solution set of complex dimension one.

Algorithm 3.1. *(An algorithm for finding real solutions contained in the one-dimensional complex component of $V(f)$.)*

Input: *A polynomial system $f(x)$ as in (1).*

Output: *An enumeration of the one-dimensional irreducible components $Z_{1,j} \subset V(f)$, $j \in \mathcal{I}_1$, as in (4), and a numerical description of the real points contained therein. This is a set of isolated points P arising as the intersection of two distinct complex conjugate components, along with a Morse-like representation of the real curves in each self-conjugate complex component. For the ith such component, this is given as a boundary set $\mathcal{B}_{\mathbb{R}i}$ and an edge set $\mathcal{E}_i$, as in Definition 1.1. Optionally, we can also find all points where the real curves meet, thus completing a connection diagram for all real parts of the complex one-dimensional components in $V(f)$.*

Step 1 *Compute a numerical irreducible decomposition of $V(f)$. Retain only the one-dimensional components $Z_{1,j}$, $j \in \mathcal{I}_1$. Each component $Z_{1,j}$ is represented by a witness set $W_{1,j} = \{Y_j, L_j, g_j, \pi_j\}$, where L_j is a generic hyperplane in $\mathbb{C}^N$, g_j is a deflated polynomial system such that $V(g_j)$ has a reduced and irreducible component, say $\mathcal{Y}$, whose closure $\overline{\mathcal{Y}}$ projects under the projection π_j to the closure of $Z_{1,j}$, i.e., $\pi_j(\overline{\mathcal{Y}}) = \overline{Z_{1,j}}$, and Y_j is a set of witness points such that $\pi_j(Y) = Z_{1,j} \cap L_j$. In the case that $Z_{1,j}$ is reduced, we have $g_j = f$ and $\pi_j = \mathrm{id}$, the identity map. The number of witness points, $\#(Y_j)$, is equal to the degree of the reduction of $Z_{1,j}$. We choose L_j as a linear equation with real coefficients.*

Step 2 *For each $Z_{1,i}$ determine which component $Z_{1,j}$ is its complex conjugate, i.e., $\kappa(i) = j$. Since L_j is real, these can be recognized by the property $\kappa(\pi_i(Y_i)) = \pi_j(Y_j)$. A component and its conjugate will have the same degree and the same sequence of ranks in the deflation process, both of which can be used to limit the number of comparisons.*

Step 3 *For each pair of distinct complex conjugate components, say $Z_{1,i}$ and $Z_{1,j}$, $\kappa(i) = j$, $i \neq j$, find the isolated points in $Z_{1,i} \cap Z_{1,j}$. This can be done using a diagonal homotopy [19, 20, 23] to solve the system*

$$\begin{bmatrix} g_i(x) \\ g_j(y) \\ \pi_i(x) - \pi_j(y) \end{bmatrix} = 0.$$

The witness sets for $Z_{1,i}$ and $Z_{1,j}$ enable the diagonal homotopy to find just the intersection of these two components, even though the above system may have other solution components. Since $Z_{1,i}$ and $Z_{1,j}$ are distinct irreducible components of dimension one, their intersection is at most dimension zero. However, in the case of nonreduced components that have been deflated, it could happen that the fiber product defined by the above system has higher-dimensional components whose images in $\mathbb{C}^N$ are isolated. That is, the extra dimensions lie entirely within the subspace of the extra variables introduced in the deflation. The collection of the images in $\mathbb{C}^N$ is a set of points that is output as set P.

Step 4 *Choose at random a real projection $\pi : \mathbb{C}^N \to \mathbb{C}$ defined by $\pi(x) = a \cdot x = a_1 x_1 + \cdots + a_N x_N$, $a \in \mathbb{R}^N$.*

Step 5 *For each self-conjugate component, i.e., $Z_{1,i}$ with $\kappa(i) = i$, compute a Morse-like representation $\mathcal{B}_{\mathbb{R}i}$, $\mathcal{E}_i$, using Algorithm 3.2, below, and the projection π.*

Step 6 *(Optional) Find all points where real components meet. Any point where a reduced self-conjugate curve meets any other curve must be in the set of boundary points $\mathcal{B}_{\mathbb{R}}$ for that curve. Thus, since the same projection π is used for all curves in Step 5, points where two such curves meet can be found by simple comparison. Points where such a curve meets a nonreduced curve can be found by testing the boundary points of the reduced one for membership in the nonreduced one. Points where two nonreduced curves meet can be computed by a diagonal intersection of the same form as appears in Step 3, above.*

Done

Algorithm 3.1 calls Algorithm 3.2, below, to find the real part of each self-conjugate one-dimensional component of $V(f)$. Before stating the algorithm, we justify here our approach to formulating the main step of finding the boundary points $\mathcal{B}^*$. Recall that these are the points of the curve where either (a) the tangent to the curve is not well-defined, i.e., it is not a unique point in the tangent space $\mathbb{P}^N$, or (b) the tangent is orthogonal to the sweep direction a that defines the sweep projection π. Both of these conditions can be captured in a single condition: if the intersection of the tangent space at x with the hyperplane orthogonal to a is not empty, then x is a boundary point. Steps 1–3 of the algorithm formulate equations for this tangent subspace and Step 4 checks every possible dimension for solutions.

Algorithm 3.2. *(An algorithm for finding real solutions contained in a self-conjugate one-dimensional irreducible complex algebraic set Z.)*

Input: *A deflated witness set $W = \{Y, L, g, \pi'\}$ for Z, and a real sweep projection $\pi(x) = a^T x$ given by a column vector $a \in \mathbb{R}^N$.*

Output: *A Morse-like representation of the real points in Z, i.e., $Z \cap \mathbb{R}^N$, in terms of a set of boundary points $\mathcal{B}_\mathbb{R}$ and an edge set $\mathcal{E}$, as given in Definition 1.1.*

Notation: *Denote the dimension of the ambient space for the deflated system as $N' \geq N \geq 2$, that is, $N' - N$ is the number of extra variables appended by the deflation process. Let $\mathcal{Y} \in \mathbb{C}^{N'}$ denote the component of $V(g)$ that projects to Z, i.e., $\pi'(\overline{\mathcal{Y}}) = \overline{Z}$. $Y \subset \mathcal{Y}$ is the set of witness points $\mathcal{Y} \cap \pi'^{-1}(\pi'(\mathcal{Y}) \cap L)$. Since Z is one-dimensional, g is a polynomial system $g : \mathbb{C}^{N'} \to \mathbb{C}^{N'-1}$.*

Step 1 *Pad the sweep vector a with $N'-N$ zeros to form $a' = \begin{bmatrix} a \\ 0 \end{bmatrix} \in \mathbb{R}^{N'}$.*

Step 2 *Find a basis B for the linear subspace in $\mathbb{R}^{N'}$ orthogonal to a'. That is, find a matrix $B \in \mathbb{R}^{N' \times (N'-1)}$ such that $\operatorname{rank}[\ a' \quad B\ ] = N'$ and $(a')^T B = 0$. This can be done, for example, by choosing $B' \in \mathbb{R}^{N' \times (N'-1)}$ at random and applying Gram-Schmidt orthogonalization.*

Step 3 *Let $J(y) = \partial g / \partial y$ be the $(N'-1) \times N'$ Jacobian matrix of $g(y)$. Define $H : \mathbb{C}^{N'} \times \mathbb{P}^{N'-2}$ as $h(y, z) = J(y)Bz$. The variables z parameterize the tangent space orthogonal to a'.*

Step 4 *Define the projections $\pi_y : (y, z) \mapsto (y)$ and $\pi_z : (y, z) \mapsto (z)$. We wish to find the set $A = \left(\mathcal{Y} \times \mathbb{P}^{N'-2}\right) \cap V(h)$. Since $\mathcal{Y}$ is one-dimensional and since the fiber in A over a generic point in $\mathcal{Y}$ is empty, we know that the projection of A onto Z, i.e., $\pi'(\pi_y(A))$ is a finite set of points. These are the boundary points $\mathcal{B}^*$. The components of A can be any dimension from 0 to $N' - 2$.*

We find a witness superset $\mathbb{W}$ containing witness points on the components of dimension M by intersecting $\pi_z(A)$ with a generic linear subspace of $\mathbb{P}^{N'-2}$ of dimension $N' - 2 - M$. For computational purposes, we may treat $\mathbb{P}^{N'-2}$ as the lines through the origin in $\mathbb{C}^{N'-1}$, where we find components of dimension $M' = M + 1$ by intersecting with an affine linear subspace of $\mathbb{C}^{N'-1}$ of dimension $N' - 1 - M'$. These intersections can be computed independently one at a time in the

style of [23, ch.13] or we may use a cascade as outlined in Remark 3.4 below.

The projection of the witness superset $\mathbb{W}$ onto the original coordinates are the boundary points $\mathcal{B}^* = \pi'(\pi_y(\mathbb{W}))$. $\mathcal{B}^*_{\mathbb{R}}$ is the subset of $\mathcal{B}^*$ having all real coordinates.

Step 5 *Initialize $\mathcal{B}_{\mathbb{R}} = \mathcal{B}^*_{\mathbb{R}}$ and initialize $\mathcal{E} = \emptyset$.*

Step 6 *Let $t_1 < t_2 < \cdots < t_m$ be the distinct values $\pi(\mathcal{B}^*_{\mathbb{R}})$, sorted in ascending order. Pick values $t'_0 < \cdots < t'_m$, one in each interval $(-\infty, t_1), (t_1, t_2), \ldots, (t_{m-1}, t_m), (t_m, +\infty)$. For $i = 1, \ldots, m-1$, the midpoint $t'_i = (t_i + t_{i+1})/2$ suffices. For the infinite intervals, $t'_0 = t_1 - 1$ and $t'_m = t_m + 1$ suffice.*

Step 7 *For each $i = 0, \ldots, m$, follow the solution paths of the homotopy*

$$\begin{bmatrix} g(y) \\ sL(y) + (1-s)(\pi(\pi'(y)) - t_i) \end{bmatrix} = 0 \tag{5}$$

as s goes from 1 to 0, starting at the witness points Y. Each real endpoint, say y^*, of these solution paths is a point on an edge of the Morse-like representation. Append the partially complete edge information $(y^*, \cdot, \cdot)$ as a new edge in $\mathcal{E}$. There may be multiple endpoints over the same t'_i and there may be none at all over others.

Step 8 *For every incomplete edge $(y^*, \cdot, \cdot)$ in $\mathcal{E}$, complete it by starting at $y = y^*$ and tracking t left from t'_k to t_k and right from t'_k to t_{k+1} using the homotopy*

$$\begin{bmatrix} g(y) \\ (\pi(\pi'(y)) - t) \end{bmatrix} = 0 \tag{6}$$

Do not track left from t'_0 or right from t'_m, as these must go to infinity. The endpoint $\pi'(y)$ obtained by tracking left is compared to $\mathcal{B}_{\mathbb{R}}$ and the index of the matching point, if any, is the left index, ℓ, for that point. If there is no match, the endpoint is appended to $\mathcal{B}_{\mathbb{R}}$ and its index becomes ℓ. Similarly, tracking to the right determines the right index r, completing the edge descriptor $(\pi'(y^*), \ell, r)$. It is important to use a singular endgame in these homotopies, as some edges are expected to end at the singular boundary points $\mathcal{B}^*_{\mathbb{R}}$.

Done

Remark 3.3. The algorithm assumes the problem is formulated on $\mathbb{C}^N$. If instead it is formulated on $\mathbb{P}^N$, i.e., if the system consists of homogeneous polynomials on $\mathbb{C}^{N+1}$, then we may work on an arbitrary patch $b \cdot x = 1$, $b \in \mathbb{R}^{N+1}$, and use a projection $\pi(x) = a \cdot x$, $a \in \mathbb{R}^{N+1}$. In the case of a nonreduced component, $N' > N$, pad both a and b with $N' - N$ zeros in Step 1 to get a' and b'. Then in Step 2, B is a $(N'+1) \times (N'-1)$ matrix whose columns span the subspace orthogonal to both a' and b'. The rest of the algorithm is unchanged.

Remark 3.4. The points in $\mathbb{W}$ in Step 4 may be computed in the style of a cascade [23, ch.14]. In the cascade approach, we use the embedding

$$H(y,z,s) = \begin{bmatrix} g(y) \\ J(y)Bz + \Lambda \begin{bmatrix} s_1 c_1 z \\ \vdots \\ s_{N'-2} c_{N'-2} z \end{bmatrix} \\ dz - 1 \end{bmatrix} \tag{7}$$

where Λ is a random complex $(N'-1) \times (N'-2)$ matrix, each c_i and d is a random complex $1 \times (N'-1)$ row vector. The final equation, $dz - 1 = 0$, picks a projective patch for representing $\mathbb{P}^{N'-2}$ in $\mathbb{C}^{N'-1}$, while the conditions $c_i z = 0$ slice out linear subspaces of $\mathbb{P}^{N'-2}$. We begin by setting all $s_i = 1$ and solving this by diagonal homotopy. Call this stage $M' = N' - 1$. Then, for stage $M' = N' - 2, \ldots, 1$, we follow homotopy paths as $s_{M'}$ goes from 1 to 0. At the end of stage M', the path endpoints that satisfy $c_i z = 0$, all $i < M'$, are the witness superset for boundary points having an orthogonal tangent space of dimension $M = M' - 1$ in $\mathbb{P}^{N'-2}$. We append these to the witness superset $\mathbb{W}$. The other endpoints are the start points for the next stage of the computation.

Remark 3.5. Instead of the cascade technique of Remark 3.4, one could avoid introducing the variables z and replace the equations $J(y)Bz = 0$ with the single condition $\det(J(z)B) = 0$. While valid, the determinantal equation can have a high degree, making it less stable numerically than the cascade.

Remark 3.6. For the purpose of numerically sampling edges, we should store the deflated system $g(y)$ along with each edge. For simplicity, this was dropped from the notation. Also, one may wonder if in Step 8 it is sufficient to store $\pi'(y^*) \in \mathbb{C}^N$ instead of $y^* \in \mathbb{C}^{N'}$. It is sufficient, because the auxiliary variables to complete y^* in the deflated representation can be reconstructed by linear algebra given just $\pi'(y^*)$.

Among the boundary points $\mathcal{B}_{\mathbb{R}}$ returned by Algorithm 3.1, there may be some points not connected to any edge. Such points are isolated solutions. Given x^*, a solution of the polynomial system, one may use the following algorithm to identify the real component to which it belongs.

Algorithm 3.7. *(An algorithm for membership test)*

Input: *A polynomial system f as in (1), a solution x^* of the system, a projection π, the set of isolated solutions P, and for all self-conjugate irreducible components of the one-dimensional solutions of $V(f)$, a Morse-like representation $\mathcal{B}_{\mathbb{R}i}$, $\mathcal{E}_i$ for the real points in the ith such component.*

Output: *The component upon which x^* lies, if any.*

Step 1 *Check if x^* is in the set of isolated points P. If so, return with this information.*

Step 2 *Test for membership in each self-conjugate component, as follows:*

Step 2a *Set $t^* = \pi(x^*)$ and let $t_1 < \cdots < t_m$ be the sorted values of $\pi(\mathcal{B}_{\mathbb{R}})$. Clearly, $t^* \in [t_i, t_{i+1}]$ for some i (taking into account the infinite left and right intervals as well). If t^* is equal to one of the endpoints of these intervals, then x^* must be in the set of boundary points $\mathcal{B}_{\mathbb{R}}$. Find the matching point and return that information.*

Step 2b *If we reach this step, x^* must either be on one of the edges that maps to the interval containing t^* or it is not in the component. Suppose edge $E_k = \{y_k, \ell_k, r_k\}$ has a value $\pi(\pi'(y_k))$ that lies in the same interval as t^*. We test for membership in the edge by starting at $y = y_k$ and tracking the solution path of*

$$\begin{bmatrix} g(y) \\ \pi(\pi'(y)) - t \end{bmatrix} = 0, \tag{8}$$

as t goes from t_k to t^*. If the endpoint y^* of this path matches x^*, that is, if $\pi'(y^*) = x^*$, then x^* is on this edge. If so, return this information; otherwise, try the next edge.

Step 3 *If no component contains x^*, then even though x^* is a solution of $f(x)$, it does not lie on the real subset of the one-dimensional component of $V(f)$.*

Done

Remark 3.8. As is common, suppose several edges share the same t value, say t_k. It would be more efficient to track in step 2b from x^* as t goes from t^* to t_k and compare the endpoint against all these edges. This is fine for edges that come from reduced complex components, but it is less straightforward for edges coming from nonreduced components, because these would require reconstructing the y^* that lies over x^* in the deflated representation. We have sacrificed efficiency for simplicity in the above statement of the algorithm.

4 Examples

This section starts with three very simple examples that can also be derived by hand. These serve to illustrate how our method finds and represents solutions. A more challenging fourth example from the theory of mechanisms completes the section. For brevity, we report solutions to only a few decimal places, although all computations were done in double precision using HomLab[23], adding new routines for the algorithms of this paper. Also for brevity in the presentation, we choose "random" constants of only two decimal places, although our standard procedure is to use a random number generator in double precision.

4.1 Isolated real point

Consider the polynomial $f = x^2 + y^2$, which clearly has $(0, 0)$ as its only real root. The results of each step of Algorithm 3.1 are as follows.

Step 1 The irreducible decomposition of $V(f)$ reveals two one-dimensional components. Each component has a single witness point, hence the components are linear. This reflects the factorization

$$x^2 + y^2 = (x + iy)(x - iy),$$

where $i = \sqrt{-1}$. To be explicit, let

$$L(x, y) = 0.38x + 0.54y - 0.5.$$

Then the witness sets are $W_j = \{w_j, L, f, \mathrm{id}\}$, $j = 1, 2$, where

$$\begin{aligned} w_1 &= [0.43578 + 0.61927i, 0.61927 - 0.43578i], \\ w_2 &= [0.43578 - 0.61927i, 0.61927 + 0.43578i]. \end{aligned}$$

Call the two components Z_1 and Z_2.

Step 2 Since the witness points are complex conjugates, $\kappa(w_1) = w_2$, so are the components they represent, $\kappa(Z_1) = Z_2$.

Step 3 Find the intersection $Z_1 \cap Z_2$ using the diagonal homotopy:

$$\begin{bmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 \\ \gamma s L(x_1, y_1) + (1-s)(x_1 - x_2) \\ \gamma s L(x_2, y_2) + (1-s)(y_1 - y_2) \end{bmatrix} = 0, \tag{9}$$

where γ is a random complex number. Track one path from the start point $[(x_1, y_1), (x_2, y_2)] = [w_1, w_2]$ as s goes from 1 to 0. The endpoint at $s = 0$ is the point $[(x_1, y_1), (x_2, y_2)] = [(0,0), (0,0)]$. (Although the point is singular, it is easily computed to full precision using a singular endgame.) Accordingly, the set P of isolated real points is $P = \{(0,0)\}$.

Steps 4-6 There are no self-conjugate components, so the algorithm is complete.

Return $P = \{(0,0)\}$, $\mathcal{B}_{\mathbb{R}} = \mathcal{E} = \emptyset$.

4.2 Line with an embedded point

Consider the polynomial system

$$f(x, y) = \begin{bmatrix} x(y-1) \\ (y-1)^2 \end{bmatrix} = 0. \tag{10}$$

The real solution set of this system consists of the line $y = 1$. This system has an embedded point of higher multiplicity at $(x, y) = (0, 1)$. Algorithm 3.1 proceeds as follows.

Step 1 Using the slice

$$L(x, y) = 0.82x - 0.27y + 0.66,$$

we get a single witness point for the one-dimensional component:

$$(x, y) = (-.47561, 1).$$

Since the component has codimension 1, we represent it with a single polynomial via randomization, say $g(x, y) = [1 \quad -0.92] f(x, y)$.

Step 2 The component is its own conjugate.

Step 3 Their are no conjugate pairs, so $P = \emptyset$.

Step 4 We choose a sweep projection $\pi(x, y) = 0.44x - 0.83y$.

Step 5 Algorithm 3.2 returns a single boundary point $\mathcal{B}_{\mathbb{R}} = \{(0, 1)\}$ and two edges in $\mathcal{E}$ (details below).

Step 6 There is just one component, so do nothing more.

Return $P = \emptyset$, $\mathcal{B}_{\mathbb{R}} = \{(0, 1)\}$, and $\mathcal{E}$ from Step 5.

The actions of Algorithm 3.2, called in Step 5 above, are as follows.

Step 1 The component is reduced; $N' = N = 2$. So $a' = a$.

Step 2 The orthogonal complement is $B = [0.83\ \ 0.44]^T$.

Step 3 In the formula $J(x, y)Bz = 0$, z lives in $\mathbb{P}^0$, that is, it is a nonzero constant. The choice $z = 1$ suffices.

Step 4 The cascade (7) has just one stage:

$$H(x, y) = \begin{bmatrix} g(x, y) \\ J(x, y)B \\ z - 1 \end{bmatrix} = 0,$$

We can drop the last equation, since z does not appear in the first two. We have a single witness point for the first polynomial and the second polynomial is linear, so we may solve the system with a single continuation path in a diagonal homotopy. The result is the point $\mathcal{B}^*_{\mathbb{R}} = (0, 1)$.

Step 5 Set $\mathcal{B}_{\mathbb{R}} = \{(0, 1)\}$ and $\mathcal{E} = \emptyset$.

Step 6 We have $t_1 = \pi(0, 1) = -0.83$. Choose $t'_0 = -1.83$ and $t'_1 = 0.17$.

Step 7 We have one continuation path for each of t'_0 and t'_1, giving an edge point on each side of t_1.

Step 8 We track from t'_0 right to t_1 and also from t'_1 left to t_1. Both edges end at the point $(0, 1)$. The edge description is

$$\mathcal{E} = \left\{ \begin{array}{[cccc]} (-2.2727, 1), & -\infty, & 1 & \\ (+2.2727, 1), & 1, & +\infty & \end{array} \right\}.$$

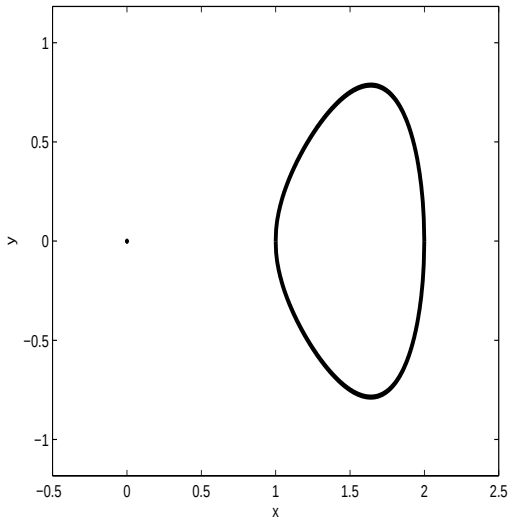


Figure 1: The real solutions of $y^2 + x^2(x-1)(x-2)$.

Remark 4.1. We should note that an exception could be made for lines: two sample points on a line suffice to describe it as a geometric object. For illustrative purposes, we have followed every step of the algorithm, revealing the embedded point in the process.

4.3 Isolated boundary point plus curve

Consider the following simple polynomial system:

$$y^2 + x^2(x-1)(x-2) = 0. \tag{11}$$

As depicted in Figure 1, the real solution set of this system consists of a bounded curve and an isolated point at the origin. On the other hand, the complex solution set consists of a single irreducible curve.

Let us describe the numerical treatment of this problem more briefly than in the previous two. In Step 1 of the main algorithm, we find a single irreducible complex component of degree four. Being just one component, it must be self-conjugate, and indeed, the four witness points appear as either two complex conjugate pairs, or one such pair and two real points, depending on the linear slice used. We choose a projection, say $\pi(x, y) = 0.8x + 1.2y$, and proceed to Algorithm 3.2 to find the Morse-like representation.

As in the prior example, there is only one stage in the cascade of Step 4. The four witness points for the component give rise to four homotopy paths. Two of these end at the point $b_1 = (0.0, 0.0)$, while the others go to $b_2 = (1.4413970, -0.71573118)$, and $b_3 = (1.7632215, 0.74955463)$. Letting $t_i = \pi(b_i)$, we have that $t_1 = 0.0$, $t_2 = 0.29424019$, and $t_3 = 2.3100428$. It is therefore reasonable to choose the values $t'_0 = -1.0$, $t'_1 = 0.14712009$, $t'_2 = 1.3021415$, and $t'_3 = 3.0$.

In Step 7, for each t'_i, $i = 0, \ldots, 3$, we track four homotopy paths. Only t'_2 has any real solutions, and these are $(1.1090746, 0.34573486)$ and $(1.9853600, -0.23845543)$. Tracking these points from $t = t'_2$ to $t = t_2$, both paths end at b_2. Similarly, tracking from $t = t'_2$ to $t = t_3$, both paths end at b_3. Therefore, we may conclude that b_1 is an isolated solution while the other two boundary points are connected by two edges. In other words, the real solution set consists of an isolated solution and a bounded curve, as expected.

To summarize, the final output $\mathcal{B}_{\mathbb{R}}$, $\mathcal{E}$, and their projection values are as follows.

$$\begin{aligned}
\mathcal{B}_{\mathbb{R}} &= \{(0,0),\ (1.4413970, -0.71573118),\ (1.7632215, 0.74955463)\} \\
\pi(\mathcal{B}_{\mathbb{R}}) &= \{0.0,\ 0.29424019,\ 2.3100428\} \\
\mathcal{E} &= \left\{ \begin{array}{llllll} [& (1.1090746, & 0.34573486), & 1, & 3 &] \\ [& (1.9853600, & -0.23845543), & 1, & 3 &] \end{array} \right\} \\
\pi(\mathcal{E}) &= \{1.3021415,\ 1.3021415\}.
\end{aligned}$$

4.4 The Stewart-Gough platform problem

A generic Stewart-Gough platform consists of two rigid bodies, called the base and endplate, joined by six legs. The legs are connected to the base and endplate by spherical joints. When the six ball joints in the base and the six ball joints in the end plate are in general position, a Stewart-Gough platform has forty isolated solutions. This was first established by continuation [14] and later proven analytically [7, 25]. One formulation of the kinematic equations is as follows, where $(u, v) \in \mathbb{P}^7$ are Study coordinates for rigid body motion [25] and we treat each of u and v as a 4×1 column vector. We note that u is a quaternion that describes the orientation of the endplate, and v is a quaternion that encodes the position of the endplate such that $p = vu'/(u'u)$ (where u' is the quaternion conjugate of u and quaternion multiplication is implied) is the position vector from the origin of the base to the origin of the endplate. In the following, we use 4×1 column vectors

to represent the quaternions, such that $u = a\mathbf{i} + b\mathbf{j} + c\mathbf{k} + d$ becomes the column vector $u = [a, b, c, d]^T$, and similar for v.

$$u^T v = 0 \tag{12}$$

$$u^T A_i u + 2v^T B_i u + v^T v = 0, \quad i = 1, \ldots 6. \tag{13}$$

Formulas for the 4×4 constant matrices A_i and B_i in terms of the ball joint positions and the leg length may be found in [25].

A special kind of Stewart-Gough platform is called a Griffis-Duffy platform [6, 8]. In this platform, the base and endplate are triangles with ball joints at each vertex and along each side. The leg connects a side point of one body to a vertex of the other. A special Griffis-Duffy platform first identified in [8] is to make the base and endplate triangles equilateral and the ball joints on the sides are at the midpoints. This platform is called Griffis-Duffy I platform in [21]. A further specialization of Griffis-Duffy I platform is to make the base and endplates congruent and to make all six leg lengths equal, which is called Griffis-Duffy II platform. It is well known that Griffis-Duffy I and Griffis-Duffy II platforms have one degree of freedom of motion if one specifies a general position of endplate with respect to the base and sets the leg lengths to match. In [8, 21], the motion of Griffis-Duffy I and Griffis-Duffy II platforms are analyzed. Especially in [21], the movement curve is broken into twelve lines and five one-dimensional irreducible components of degree $\{6, 6, 6, 6, 4\}$.

The nature of the real components varies with the leg length. For example, for leg length less than one-third the altitude of the base triangle, the mechanism cannot be assembled: there are no real solutions. For our present study, we choose leg length equal to the altitude of the base. This allows an assembly in which the base and endplate exactly coincide and the legs all coincide with one of the altitudes. By physical intuition, we expect that the existence of such a special assembly configuration signals a mechanism with a particularly interesting motion.

For the numerical study, we scale the mechanism so that the edge length of the base and endplates are 2, making the altitudes equal to $\sqrt{3}$. To best reveal the three way symmetry of the problem, we place the origin of coordinates at the center of the triangles, and let the initial reference configuration

$$x_{\text{ref}} = (u, v) = ([0, 0, 0, 1]^T, [0, 0, 0, 0]^T) \tag{14}$$

be the configuration where the base and endplate coincide, as described in the previous paragraph. In other words, the coordinates of the joint locations

in the base and endplate are as follows.

Base =

	1	2	3	4	5	6
x	1	$1/2$	0	$-1/2$	-1	0
y	$-1/\sqrt{3}$	$1/(2\sqrt{3})$	$2/\sqrt{3}$	$1/(2\sqrt{3})$	$-1/\sqrt{3}$	$-1/\sqrt{3}$
z	0	0	0	0	0	0

Endplate =

	1	2	3	4	5	6
x	$-1/2$	-1	0	1	$1/2$	0
y	$1/(2\sqrt{3})$	$-1/\sqrt{3}$	$-1/\sqrt{3}$	$-1/\sqrt{3}$	$1/(2\sqrt{3})$	$2/\sqrt{3}$
z	0	0	0	0	0	0

The ith point of the base is connected to the ith point of the endplate by a leg of length $L = \sqrt{3}$.

On this problem, Algorithm 3.1 proceeds as follows.

Step 1 The 7 polynomials of (12) have only rank 6: they are linearly dependent. We replace them with 6 linear combinations of the original equations. The first stage of computing an irreducible decomposition is the generation of a witness superset using the cascade algorithm [15], [23, ch.14]. The system of 6 quadratics has $2^6 = 64$ paths in the cascade. Four of these end in a pair of double roots on the two-dimensional set $S = \{(u, v) \in \mathbb{P}^7 | u = 0, v^T v = 0\}$. This set is not physically meaningful, because the position vector $p = vu'/(u'u)$ is not defined. The remaining 60 paths carry over to the final cascade stage for dimension one. Here 26 end as singular points, while 34 are nonsingular. Of the 26 singular ones, 20 are junk points on the two-dimensional set S, while 6 appear as three double points.

The three double points require deflation for further numerical work. Since the multiplicity is two, one stage of deflation suffices. A linear trace test shows that each of the three points represents an irreducible component, that is, these are three double lines.

Next, monodromy and trace tests are applied to decompose the 34 nonsingular roots into irreducible components. This gives 12 linear factors, 3 quadrics, and 4 quartics. Since all 12 lines satisfy $u'u = 0$, they are nonphysical, and we cast them aside.

In short, we have 3 double lines, 3 quadrics, and 4 quartics requiring further study.

Step 2 Each of the 10 components is self-conjugate.

x_0	T	x_0	T	x_0	T
0.2601	-0.4119	-2.5334	0.0423	-0.3198	1.3401
-0.4506	0.7134	-4.3880	0.0733	0.0000	0.0000
-0.0000	-0.0000	0.0000	0.0000	-0.0000	-0.0000
1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
-0.0000	0.0000	0.0000	-0.0000	-0.0000	-0.0000
-0.0000	0.0000	-0.0000	0.0000	-0.0000	0.0000
-0.6008	0.9512	5.8506	-0.0977	-0.3693	1.5474
0.0000	0.0000	0.0000	-0.0000	0.0000	0.0000

Table 1: Data for the double lines of the Griffis-Duffy example

Step 3 There are no complex conjugate pairs of components.

Step 4 We choose a random real projection, $\pi(x) = a \cdot x$, with

$$a = [\ 0.90 \quad -0.54 \quad 0.21 \quad -0.03 \quad 0.78 \quad 0.52 \quad -0.09 \quad -0.96\ ]. \tag{15}$$

Since the Griffis-Duffy system (12) is formulated on $\mathbb{P}^7$, we recast it onto $\mathbb{C}^7$ by choosing a projective patch, $b \cdot x = 1$, where we choose b arbitrarily as

$$b = [\ 0.59 \quad 0.90 \quad 0.56 \quad 0.07 \quad 0.67 \quad 0.93 \quad 0.23 \quad 0.53\ ]. \tag{16}$$

Step 5 We may describe the three double lines as $s_1 x_0 + s_2 T$, where x_0 is the witness point found in Step 1, T is the tangent vector computed from the Jacobian matrix for the deflated system, and $(s_1, s_2) \in \mathbb{P}^1$ are the free parameters of the line. The results are given to four digits in Table 1.

This leaves the 3 quadrics and 4 quartics for treatment by Algorithm 3.2. For brevity, let's label the quadratics as Q_1, Q_2, and Q_3 and the quartics as R_1, R_2, R_3, and R_4. In Step 4 of Algorithm 3.2, we use the cascade of Remark 3.4, which has 12 cascade paths for each quadratic and 24 cascade paths for each quartic. For all three quadrics, 2 paths end at stage $M' = 2$ and the other 10 end at stage $M' = 1$. For quartic R_1, 2 paths end at $M' = 3$, 8 paths end at $M' = 2$ and 14 paths end at $M' = 1$. Quartics R_2, R_3, and R_4 have the same pattern: 6 paths end at stage $M' = 2$ and 18 end at stage $M' = 1$. Some of the endpoints at lower dimensions are "junk" points on the sets found at higher dimension. These are easily identified as they lie over the same point $\mathcal{B}^*$.

	B_1	B_2	B_3	x_{ref}	B_1	B_2	B_3	x_{ref}
u_i	-0.4330	-0.4330	0.8660	0.0000	$-\sqrt{3}/4$	$-\sqrt{3}/4$	$\sqrt{3}/2$	0
u_j	-0.7500	0.7500	0.0000	0.0000	$-3/4$	$3/4$	0	0
u_k	0.0000	0.0000	0.0000	0.0000	0	0	0	0
u_1	0.0000	0.0000	0.0000	1.0000	0	0	0	1
v_i	0.0000	0.0000	0.0000	0.0000	0	0	0	0
v_j	0.0000	0.0000	0.0000	0.0000	0	0	0	0
v_k	1.0000	1.0000	1.0000	0.0000	1	1	1	0
v_1	0.0000	0.0000	0.0000	0.0000	0	0	0	0
M'	2	2	2	3	2	2	2	3

Table 2: Boundary points $\mathcal{B}^*_{\mathbb{R}}$ having higher-dimensional tangencies (numerical and exact).

The counts of points $\mathcal{B}^*$ over the complexes will remain the same for different choices of projection and projective patch. However, the number of real endpoints $\mathcal{B}^*_{\mathbb{R}}$ will vary, because as the sweep direction varies, real tangencies in the orthogonal direction come in and out of existence. In contrast to these, the higher-dimensional points in the cascade remain fixed, as for example, a self crossing of the curve remains singular independent of the sweep direction. There are four such fixed singularities, listed in Table 2. Each of the quadrics passes through two of $\{B_1, B_2, B_3\}$ at stage $M' = 2$ of the cascade, and one of the quartics has x_{ref} as a double root at stage $M' = 3$. The other three quartics have no real boundary points at all.

The next step is to fill in the edges between the boundary points. Rather than list all the results numerically, we present them in graphical form in Figure 2. Here, the arcs of the curve are traced out in the projective patch defined by (16) and then projected onto a 2-plane, using the projection (15) as the horizontal axis and using a second projection $\hat{a} \cdot x$ as the vertical axis with

$$\hat{a} = [\ -0.21 \quad -0.37 \quad 0.00 \quad 0.03 \quad -0.13 \quad 0.02 \quad -0.07 \quad 0.89\ ]. \quad (17)$$

In Figure 2, the quadrics are shown in solid line (each appears as an hyperbola in the projection chosen), the singular double lines are shown in dotted line, and one quartic is shown in dashed line. The remaining three quartics contain no real points. On the curves, each open circle marks a point in the middle of each edge found, a box filled with an asterisk marks one of the singular boundary points $\mathcal{B}^*_{\mathbb{R}}$, and

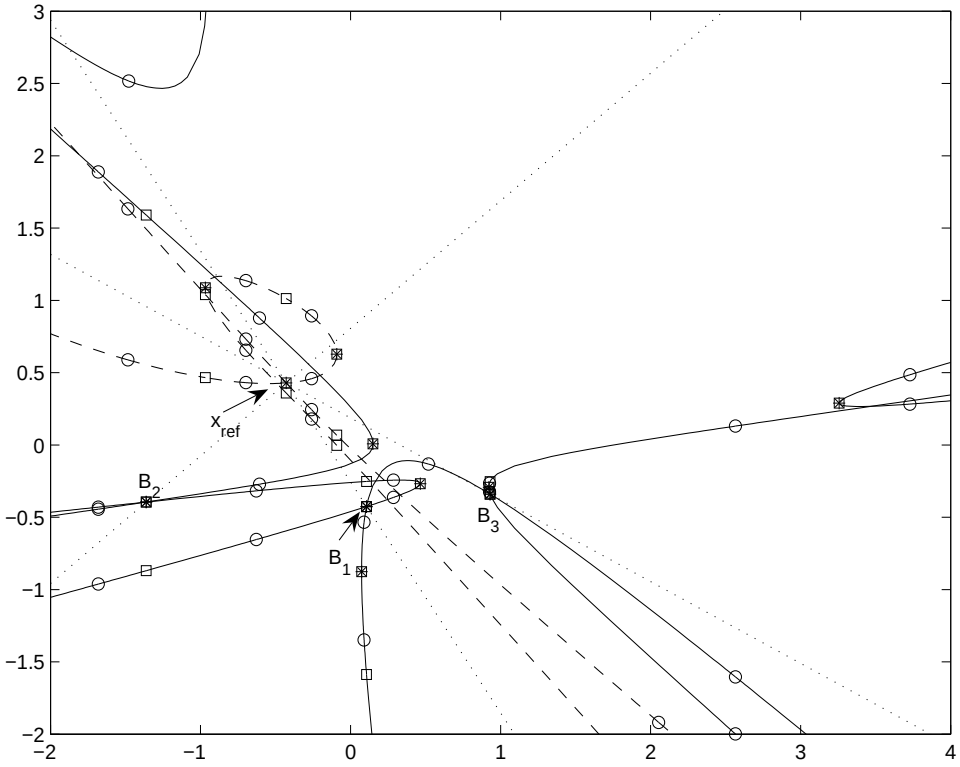
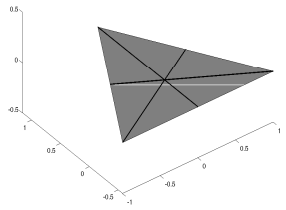


Figure 2: A projection of the Griffis-Duffy real curve for $L = \sqrt{3}$.

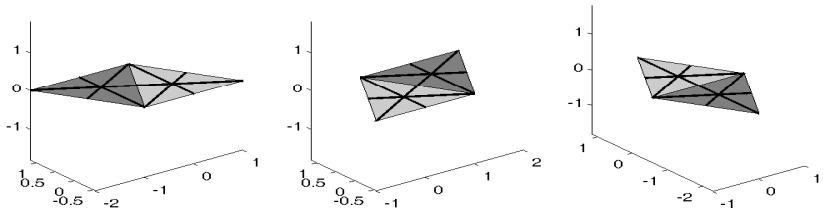
an empty box marks the end of an edge in the fiber over a singular boundary point, i.e., a point in $\mathcal{B}_{\mathbb{R}} \setminus \mathcal{B}^*_{\mathbb{R}}$.

Step 6 Computing intersections between the singular lines reveals that they all pass through the reference configuration, x_{ref}, see (14). This configuration is also a singular double boundary point on the quartic: close inspection of Figure 2 confirms that the quartic passes through this point twice. All intersections with the reduced components must appear as a point $\mathcal{B}^*_{\mathbb{R}}$ already found. Checking these, we find that the points B_1, B_2, B_3 in Table 2 are triple intersections: two of the quadrics and one of the lines meet in each.

The components of the curve can be better understood by drawing the mechanism at selected configurations. First, in Figure 3, we display the four configurations where multiple components meet. We see that in the reference position, the top and bottom triangles coincide. In the other three



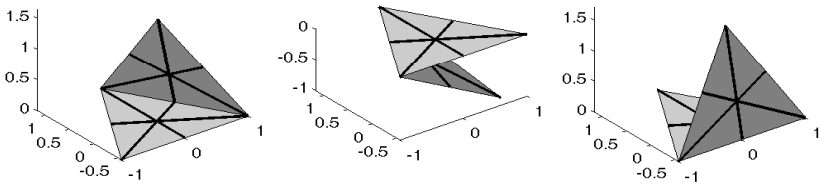
The reference pose x_{ref}



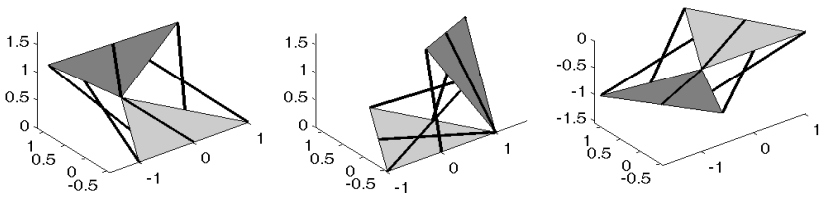
The folded out poses B_1, B_2, B_3

Figure 3: Singular poses of the Griffis-Duffy mechanism for $L = \sqrt{3}$.

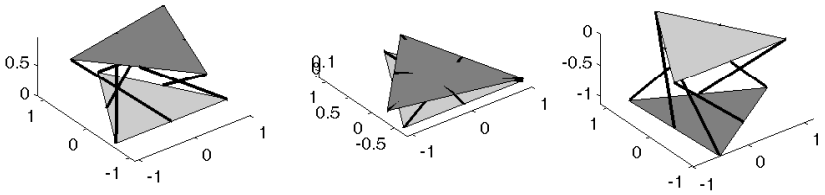
meeting places, the two triangles are coplanar, sharing one edge: call these the "folded out" configurations. In Figure 4, we display general configurations along the various components of the curve. For each of the lines, we see that the triangles rotate along a coincident edge, thus taking the mechanism from a folded out configuration to the reference configuration. At a general configuration of the quadrics, the two triangles have one vertex that coincides. These take the mechanism from one folded out position to another. Finally, on the quartic, the two triangles are in parallel planes, one centered above the other. As the moving triangle rotates, it descends from above to pass through the reference position, then loops below until it rises again through the reference position, eventually coming back to its initial configuration. From these observations, the complete motion of the mechanism is now well understood.



Poses on the singular lines



Poses on the quadrics



Poses on the quartic: above, near x_{ref}, below

Figure 4: Selected regular poses of the Griffis-Duffy mechanism for $L = \sqrt{3}$.

5 Conclusions

This paper shows that the real points contained in a one complex-dimensional algebraic curve can be described as a finite list of isolated real points along with a Morse-like representation of the real one-dimensional part of the curve. The latter consists of a finite set of boundary points along with a set of edges connecting them, each edge represented by a point in its interior and indices for its left and right boundary points. We have described how to numerically compute this representation using the operations of numerical algebraic geometry, with polynomial continuation as the main computational technique. The method is illustrated on four examples, including an instance of the Griffis-Duffy type II platform robot. For leg lengths equal to the altitude of the triangles, this mechanism has seven real curve components: 3 lines, 3 quadrics, and a quartic. The algorithm also discovers the singular poses where two or more components meet, thus determining that all the components connect.

References

[1] S. Basu, R. Pollack and M.F.Roy. *Algorithms in real algebraic geometry*. Springer-Verlag, Berlin, 2003.

[2] O. Bottema and B. Roth. *Theoretical Kinematics*. Dover, New York, 1990.

[3] C.W. Brown. Improved projection for cylindrical algebraic decomposition. *J. Symbolic Comput.*, 32:447–465, 2001.

[4] B.H. Dayton and Z. Zeng. Computing the multiplicity structure in solving polynomial systems. In *Proceedings of the 2005 international symposium on symbolic and algebraic computation*, pages 116–123. ACM Press, New York, 2005.

[5] W. Fulton. *Intersection theory*. Springer-Verlag, Berlin, 1998.

[6] M. Griffis and J. Duffy. Method and apparatus for controlling geomtrically simple parallel mechanisms with distinctive connections. US Patent 5,179,525, 1993.

[7] M.L. Husty. An algorithm for solving the direct kinematics of general Stewart-Gough Platforms. *Mech. Mach. Theory*, 31(4):365–380, 1996.

[8] M.L. Husty and A. Karger. Self-motions of Griffis-Duffy type parallel manipulators. Proc. 2000 IEEE Int. Conf. Robotics and Automation, CDROM, San Francisco, CA, April 24-28, 2000.

[9] D. Lazard and F. Rouillier. Solving parametric polynomial systems. *Preprint.*

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

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

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

[13] T. Ojika. Modified deflation algorithm for the solution of singular problems. I. A system of nonlinear algebraic equations. *J. Math. Anal. Appl.*, 123:199–221, 1987.

[14] R. Raghavan. The Stewart platform of general geomrtry has 40 configurations. *ASME J. Mech. Design*, 115:277–282, June 1993.

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

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

[17] A.J. Sommese, J. Verschelde, and C.W. Wampler. Using monodromy to decompose solution sets of polynomial systems into irreducible components. In *Application of Algebraic Geometry to Coding Theory, Physics and Computation*, edited by C. Ciliberto, F. Hirzebruch, R. Miranda, and M. Teicher. Proceedings of a NATO Conference, February 25 - March 1, 2001, Eilat, Israel, pages 297–315. Kluwer Academic Publishers, Dordrecht, 2001.

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

[19] A.J. Sommese, J. Verschelde and C.W. Wampler. Homotopies for intersecting solution components of polynomial systems. *SIAM J. Numer. Anal.* 42:1552–1571, 2004.

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

[21] A.J. Sommese, J. Verschelde and C.W. Wampler. Advances in polynomial continuation for solving problems in kinematics. *ASME J. Mech. Des.*, 126:262–268, 2004.

[22] A.J. Sommese, J. Verschelde, and C.W. Wampler. Introduction to numerical algebraic geometry. In *Solving Polynomial Equations: Foundations, algorithms, and applications*, ed. by A. Dickenstein and I. Emiris, pages 301–337. Springer Verlag, Berlin, 2005.

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

[24] A. Tarski. *A decision method for elementary algebra and geometry.* RAND Corporation, Santa Monica, California, 1948.

[25] C.W. Wampler. Forward displacement analysis of general six-in-parallel SPS(Stewart) platform manipulators using soma coordinates. *Mech. Mach. Thoery*, 31(3):331–337, 1996.

[26] C.W. Wampler. Solving the kinematics of planar mechanisms by Dixon determinant and a complex-plane formulation. *ASME J. of Mech. Des.*, 123(3):382–387, 2001.