Epsilon local rigidity and numerical algebraic geometry

Andrew Frohmader\textsuperscript{1} and Alexander Heaton\textsuperscript{2}

\textsuperscript{1}University of Wisconsin, Milwaukee
\textsuperscript{2}Max Planck Institute for Mathematics in the Sciences, Leipzig, and Technische Universität Berlin

February 12, 2020

Abstract

A well-known combinatorial algorithm can decide generic rigidity in the plane by determining if the graph is of Pollaczek-Geiringer-Laman type. Methods from matroid theory have been used to prove other interesting results, again under the assumption of generic configurations. However, configurations arising in applications may not be generic. We present Theorem 7 and its corresponding Algorithm 1 which decide if a configuration is \(\varepsilon\)-locally rigid, a notion we define. This provides a partial answer to a problem discussed in the 2011 paper of Hauenstein, Sommese, and Wampler. The theorem and algorithm use results from a 2012 paper of Hauenstein. We also present Algorithm 2 which uses numerical algebraic geometry to find nearby valid configurations which are not obtained by rigid motions. When successful, this method demonstrates the failure of local rigidity by explicitly constructing a sequence of configurations which are a discrete-time sample of a continuous flex.

Keywords: Numerical algebraic geometry, Real algebraic geometry, Rigidity, Kinematics, Mechanism Mobility, Homotopy Continuation

1 Introduction

Consider a graph with \(n\) nodes labelled by \(\{n\} = \{1, 2, \ldots, n\}\), and with edge set \(E \subset \binom{n}{2}\). We embed this graph in some \(\mathbb{R}^d\) by a map called the initial configuration \(p_0 : [n] \to \mathbb{R}^d\). We will make precise definitions in Section \(2\), but the basic idea is to consider the edge distances \(\ell_{ij}\) between nodes connected by an edge, but also the distances \(\widehat{\ell}_{ij}\) between nodes that are not connected by an edge. If the nodes can move so that the distances \(\ell_{ij}\) for \(\{i, j\} \in E\) remain constant, but some distance \(\widehat{\ell}_{ij}\) for \(\{i, j\} \notin E\) changes, then we say the graph is flexible. If no such continuous motion exists, we say the graph is locally rigid. We can always translate or rotate the graph inside \(\mathbb{R}^d\), called a rigid motion, which keeps all pairwise distances constant. Thus, to decide local rigidity, we need to establish if there are motions of the graph besides the rigid motions.

Deciding local rigidity for non-generic configurations \(p_0\) has long been known to be difficult. Since the simplest models are often the most useful, variations on this theme have been well-studied and therefore go by many names including configuration, truss, bar-and-joint framework, tensegrity, linkage, assembly mode, structure, mechanism, mobility, degrees of freedom, and more. In 2009, deciding local rigidity was shown to be coNP-hard for every \(d \geq 2\) [1]. The combinatorics community has studied this problem by assuming certain generic conditions on \(p_0\). This allows them to prove theorems using the graph structure alone (for overviews see [8] or Chapter 61 of [15]). In 1927 [28, 29] Hilda Pollaczek-Geiringer showed that if \(d = 2\), and \(|E| = 2n - 3\), and every subgraph on \(n'\) nodes has no more than \(2n' - 3\) edges, the graph is generically...
rigid in the plane, among many other relevant results. This result was rediscovered by Laman in 1970 [24]. Thus, for \( d = 2 \) deciding rigidity admits a combinatorial algorithm involving counts on subgraphs. Versions of this have been called the pebble game. When \( d = 3 \), things get harder. There is currently no combinatorial, polynomial-time algorithm which decides generic rigidity for an arbitrary graph in \( \mathbb{R}^3 \). However, in 1916, Dehn showed that any convex polyhedra in \( \mathbb{R}^3 \) is generically rigid [12]. This was extended by Connelly in 1980 by using second-order rigidity to show that the generic assumption can be dropped if the convex polyhedra has all faces triangles [9]. This is a fascinating area of study, but when a human writes down an embedding, or builds a structure, the configuration is often not generic. Besides, one might seek out non-generic configurations for their unusual properties, useful in applications.

The engineering and kinematics communities have also studied variations of this problem. In [14], a critical review of methods of mobility analysis is presented. The author enumerates 35 different approaches to calculating the mobility of a given configuration \( p_0 \), some of which involve an analysis of the kinematic loop equations (also called member constraints), while others are quick formulas. The author also discusses the limitations and outright failure of many of the listed methods for modern kinematic systems. It is well-known that the classical Gräbler-Kutzbach formulas (see [16, 23]) for mobility can be wrong for special configurations, and other attempts to refine the formulas still fall short. The main reason is that certain exceptional configurations cause deviations from any simple formula, for example the architecturally singular Stewart-Gough platforms in [22] or the planar manipulators in [13], whose 6 legs define lines belonging to a line complex. Due to the difficulties caused by these exceptional configurations, the author of [20] classifies mechanisms into the classes “trivial”, “exceptional”, and “paradoxical”. As yet another example, kinematotropic mechanisms are those where multiple assembly modes can meet, leading to a change in the degrees of freedom, or mobility, of the mechanism [41]. Further analysis of any of these special examples often leads to interesting results which can be useful in applied settings. The take-home message is that real algebraic geometry in \( \mathbb{R}^{nd} \) can be complicated, and any attempt to distill the results to simple formulas will miss special cases.

The main result in this paper is Theorem 7 and its associated Algorithms 1 and 2. These concern \( \varepsilon \)-local rigidity, which we depict in the illustration below. We make precise definitions in Section 7, but briefly, a configuration \( p_0 \) is \( \varepsilon \)-locally rigid if we can certify that any continuous flexes are extremely small, and do not go anywhere. Namely, we will certify that any continuous flex through \( p_0 \) stays within an \( \varepsilon \)-ball about \( p_0 \) within configuration space. For small \( \varepsilon > 0 \), knowing \( \varepsilon \)-local rigidity is practically as good as knowing local rigidity. The configuration \( p_0 \) will not noticeably move. Our methods deal directly with the real algebraic set, applying equally well to both smooth and singular configurations. In additional, the run-time and correctness are not impacted by multiplicity. Finally, \( \varepsilon \)-local rigidity may be more relevant for applications than local rigidity, since an \( \varepsilon \)-locally rigid configuration may be acceptable, even though it is not locally rigid. Finally, our results also imply Algorithm 2 which produces animations of a flex, should it exist, yielding easily-understandable information for the scientist.
In Section 2 we discuss basic notions and definitions. In Section 3 we describe the local dimension test of the 2011 paper [37], which successfully determines local rigidity whenever \( p_0 \) is smooth and the multiplicity is below a user specified bound. In Section 4 we discuss a systematic method to find configurations \( p_0 \) for which the local dimension test of [37] may fail, while Section 5 briefly introduces polynomial homotopy continuation. Section 6 describes how to change coordinates to reduce the dimension of the underlying problem by \( (d+1) \). Finally, the main Theorem 6 and Algorithm 1 are presented in Section 7, which applies a theorem of Hauenstein [17] to the case of \( \varepsilon \)-local rigidity. We extend these ideas in Section 8 to produce a discrete flex in Algorithm 2 when such a flex exists.

2 Preliminaries

We consider a connected graph \((V,E)\) with \( n \) nodes and \( m \) edges where \( V = [n] = \{1, 2, \ldots, n\} \) and an edge is written either \( \{i,j\} \), or briefly \( ij \) for \( i,j \in [n] \). The graph is embedded in \( \mathbb{R}^d \) by the map \( p_0 : [n] \rightarrow \mathbb{R}^d \), called the initial configuration. We assume that the affine span of the \( n \) nodes is \( d \) dimensional (they are not all contained in some lower-dimensional subspace). By slight abuse of notation we specify coordinates of its nodes, denoted by a tuple \((v_1, v_2, \ldots, v_n)\). The Lie algebra of the Euclidean group of rigid motions induces an \( m \times nd \) matrix of polynomials. When we evaluate this matrix at the point \( p_0 \in \mathbb{R}^{nd} \), the vectors in its right nullspace \( \text{Null}(dg) \) are sometimes called infinitesimal mechanisms, for example in [34]. These can be seen as linear approximations to flexes. The Lie algebra of the Euclidean group of rigid motions induces \((d+1)\) linearly independent infinitesimal mechanisms, whose span we denote \( RM \). Any remaining vectors outside the span of the infinitesimal rigid motions are called infinitesimal flexes, giving rise to the following orthogonal decomposition:

\[
\text{Null}(dg) = RM \oplus F,
\]

(3)
where any \( f \in F \) is a (pure) infinitesimal flex. However, the mere existence of an infinitesimal flex does not answer the question of local rigidity. This brings us to the difference between a flex and an infinitesimal flex, given by the following

**Definition 2.** A **flex of** \( p_0 \) is a deformation \( p(t) : [0,1] \to \mathbb{R}^{nd} \) such that \( g(p(t)) = 0 \) for all \( t \in [0,1] \) and which is not a rigid motion.

A flex is an actual path of valid configurations in configuration space. An infinitesimal flex is a linear approximation.

**Definition 3.** The configuration \( p_0 \) is called **locally rigid** if no flex exists.

The \( m \times nd \) matrix \( dg \) comes with a sparsity pattern depending on the combinatorial graph alone. Thus for generic choices of coordinates this matrix will have a generic rank which can be studied by matroid theoretic techniques (see [30, 39, 40] for overviews). However, for non-generic configurations arising in applications, the rank may differ from this generic rank, and further analysis is required.

The Jacobian matrix \( dg \) can also be derived from Hooke’s law, treating every edge \( ij \in E \) as an elastic spring, which is accurate to first-order, and therefore well-known in engineering where it is sometimes called the **rigidity matrix**. For an exposition of the formation of the matrix \( dg \) from the perspective of Hooke’s law, see [19]. From that viewpoint, any vector in the nullspace \( \text{Null}(dg) \) can be seen as an impulse-response to zero force. Therefore, infinitesimal mechanisms allow the structure to move despite having no forces applied to it. Consider the following images of infinitesimal mechanisms, where we only draw the infinitesimal flexes.

The examples pictured above are drawn only with **infinitesimal flexes**, whereas below they are drawn only with their **infinitesimal rigid motions**. In each image, arrows of the same color come from one linearly independent null vector \((dg)u = 0\).
Any configuration which admits only the infinitesimal mechanisms coming from rigid motions, and no others, is said to be \textit{infinitesimally rigid}. The good news \cite{2} comes in the form of two well-known theorems:

**Theorem 1.** Infinitesimal rigidity implies local rigidity.

**Theorem 2.** A configuration \( p_0 \) with \( n \) nodes embedded in \( \mathbb{R}^d \) is infinitesimally rigid exactly when

\[
\text{rank}(dg) = nd - \left( \frac{d+1}{2} \right).
\]

Therefore, if we check the rank of \( dg \) and find that our configuration is \textit{infinitesimally rigid}, then it is also \textit{locally rigid}. Crucially, the converse is false - a configuration may have infinitesimal flexes but still be locally rigid. Other techniques are required to determine if infinitesimal flexes are actually realizable. One such technique is given by the local dimension test of \cite{37}, which provides an often successful approach to analyzing local rigidity.

### 2.1 Almost rigidity

After the preparation of this paper, we learned of the related 2019 paper \cite{21}, which defines the \textit{almost rigidity of frameworks}. This paper succeeds at extracting even more information from the rigidity matrix \( dg|_{p_0} \), specifically from its singular value decomposition and also existing tests for \textit{pre-stress stability} \cite{9, 10} using semidefinite programming (SDP). We will summarize the results here, noting the differences between their and our approaches. We focus on their Theorem 1, as it most directly relates to our results, but we note that they also obtain very nice results about a radius \textit{outside of which} the next nearest flex must lie, and also about the minimum edge length deformations that must occur to reach this farther configuration.

Given an initial configuration \( p_0 \in \mathbb{R}^{nd} \), they find one radius \( \eta_1 > 0 \) which is computed based on the linearization at \( p_0 \) of the system \( g(x) = 0 \) of Definition \cite{1}. If certain conditions are satisfied (we describe them below) then their Theorem 1 guarantees that any continuous flex \( p(t) \) which stays within the affinely linear subspace \( p_0 + F \) will stay within a sphere of radius \( \eta_1 \) of \( p_0 \). In their setup, \( F \) can be chosen to be any complementary subspace satisfying Equation \cite{3}. In their Section 8: Conclusion, the authors note that the affine linear subspace \( p_0 + F \) (which they call \( C^p \)) only approximates the space of point configurations modulo rigid motions, and they do not yet have a “geometric or quantitative understanding of how changing \( C^p \) changes the estimates we get.” However, it seems clear that especially in the case of flexes \( p(t) \) staying near \( p_0 \), most choices of \( C^p \) will give reasonable estimates. This is also clear from their experiments. In contrast, however, our methods do not require such a linear approximation to the space of possible configurations.

We will describe their setup in more detail now. Let \( A \) be the rigidity matrix \( \frac{1}{2}dg|_{p_0} \). First, they require a choice of complementary space \( F = C^p \) such that \( \text{Null} (A) = RM \oplus C^p \). Second, choose a singular value \( \sigma \) as a cutoff to form the “almost flex space” \( V^\sigma \) which is the span of the all the right singular vectors of \( A \) with singular value below \( \sigma \), and the “almost self-stress space” \( W^\sigma \) which is the span of the left singular vectors of \( A \) with singular values below \( \sigma \). By solving an SDP, find a vector \( w \in W^\sigma \) such that the stress matrix \( \Omega(w) \) is positive semidefinite when restricted to the space of almost flexes \( V^\sigma \). Compute the minimum eigenvalue \( \lambda_0 \geq 0 \) of \( V^T \Omega V \) and \( \mu_0 \in \mathbb{R} \) of \( C^T \Omega C \), where \( V \) and \( C \) are matrices whose columns span the space \( V^\sigma \) and \( C^p \) respectively. Solve another SDP to find the minimum \( \kappa \) such that \( C^T \Omega C + 2\kappa C^T A^T AC + \lambda I \succeq 0 \), where \( \lambda \) is chosen in the interval \((0, \lambda_0)\). Compute \( \eta_1 = \frac{1}{4} \|w^T A\| \) and \( D = \frac{1}{2} (\frac{1}{\mu_0} + \frac{1}{\lambda}) \) where \( \lambda = \left( \frac{\lambda}{\sqrt{\kappa}} \right)^{1/2}, z \) is the maximum degree of any vertex in the graph, and \( \mu = 1 - \frac{\lambda}{\mu_0} \). Finally, check the condition \( D < \frac{1}{2} \).

If this is satisfied, then Theorem 1 holds, ensuring that any continuous flex \( p(t) \) (that stays within \( C^p \)) will stay within an \( \eta_1 \)-ball of \( p_0 \).

We note that our methods deal directly with the real algebraic set of valid configurations, and thus we do not require an affine linear subspace \( C^p = p_0 + F \) be used to approximate the space of possible configurations for the (curvy) real algebraic set \( g(x) = 0 \). Secondly, our methods allow the radius \( \epsilon > 0 \) be chosen arbitrarily, so it is possible to continue decreasing \( \epsilon \) until the sphere actually meets the nearest flex. Given
$p_0$, the methods of [21] output a single radius $\eta_1$, which comes with no claim of minimality. In addition, when our $\varepsilon$ is decreased so much as to meet a continuous flex, Algorithm [2] can be applied to follow that flex by a parameter homotopy in $\varepsilon$, computing more nearby points sampled from the flex. While [21] makes one $\eta_1$ estimate based on the linearization $dg$ of the system of equations $g$, our method stably computes real solutions $g(x) = 0$ to the full nonlinear system, and allows a search over all possible radii $\varepsilon > 0$. In particular, this allows the computation of radii $\varepsilon_1 < \varepsilon_2$ to replace the radii $\eta_1 < \eta_2$ of [21] with tighter bounds closer to the minimal $\varepsilon_1$ and maximal $\varepsilon_2$.

### 3 Local dimension test

In this section, we sketch the local dimension test of [37]. We begin by realizing $p_0$ is a point on the real algebraic set

$$V_\mathbb{R}(g) = \{ x \in \mathbb{R}^n : g(x) = 0 \}.$$  

The configuration $p_0$ is locally rigid exactly when the local real dimension of $V_\mathbb{R}$ is $(d+1)$, the dimension of the Euclidean group of rigid motions. The local dimension test described in [37] determines the local complex dimension successfully, even in the case when $p_0$ is on a non-reduced component, provided that the multiplicity is below some user-specified bound. However, with polynomial systems an a priori upper bound on the multiplicity can often be obtained and used directly. To be clear, this treats $p_0$ as a point on the algebraic set $V(g) \subseteq \mathbb{C}^n$ given by

$$V(g) = V_\mathbb{C}(g) = \{ x \in \mathbb{C}^n : g(x) = 0 \}.$$  

Since the polynomials in the matrix $dg$ have real coefficients, the real and complex local dimensions will agree whenever $p_0$ is a smooth point on the variety. To see this, choose a basis for the nullspace given by vectors with real components. This basis will span the real and complex tangent spaces. Therefore, this method succeeds whenever $p_0$ is smooth. We describe the local dimension test now. First, we note that for our polynomial system $g : \mathbb{C}^n \rightarrow \mathbb{C}^m$, we have the following inequality:

$$nd - m \leq \text{Null}(dg|_z) \leq \dim_{p_0} V_\mathbb{C} \leq \text{Null}(dg|_{p_0}),$$

where $z \in \mathbb{C}^n$ is a generic point, and we refer to the rank of the Jacobian at such a generic point as the generic rank of $dg$. Again following [37], if $p_0$ is on a reduced component, then

$$\dim_{p_0} V_\mathbb{C} = \text{Null}(dg|_{p_0}).$$

If $p_0$ is on a non-reduced component, then

$$\dim_{p_0} V_\mathbb{C} < \text{Null}(dg|_{p_0}).$$

To determine which of these two cases holds, the local dimension test forms certain Macaulay matrices [11, 26], which generalize the Jacobian $dq$ to higher order partial derivatives. There is one Macaulay matrix for each $q \in \mathbb{N}$, formed from all partial derivatives up to and including order $q$. Consider the sequence $c_q$ which counts the dimension of the nullspace of the $q$th Macaulay matrix. Just like the dimension of an ideal is reflected in the growth rate of the number of standard monomials in each degree [27], so the growth rate of the $c_q$ reflects the local complex dimension at $p_0$. By slicing the system of equations with generic linear spaces, the authors of [37] reduce the problem of determining the growth rate to the problem of determining if the sequence $c_q$ stabilizes. The local dimension test discovers $\dim_{p_0} V_\mathbb{C}$ by examining how many linear equations were needed before the sequence $c_q$ became eventually constant. In this way they successfully determine local rigidity whenever $p_0$ is smooth.

However, for singular $p_0$, the real dimension may differ from the complex dimension, which is the only dimension the local dimension test of [37] can compute. The authors acknowledge this point, and leave it open for future work. The local dimension test is useful in many situations unrelated to local rigidity. For
example, this test is used during the numerical irreducible decomposition, a backbone algorithm of numerical algebraic geometry. Therefore, the perspective of papers studying ideas related to the local dimension test is quite general, since this test has many uses (see [6, 11, 37, 38, 42]). The purpose of the present paper is to fully specialize to the problem of local rigidity, making no attempt at generality.

4 Singular points and primary decomposition

Due to the discussion in Section 3, we know that the local dimension test can be used to decide local rigidity in the case that \( p_0 \) is a smooth point on \( V(g) \). In this section, we point out a systematic method for finding non-smooth points.

**Definition 4.** A point \( p_0 \in V(g) \) is *singular* if the rank of \( dg|_{p_0} \) is less than generic rank of \( dg \).

In this case, the local real dimension may differ from the local complex dimension, and the local dimension test of [37] may not give the correct answer for the local real dimension. The set of singular configurations for a given system of member constraints \( g(x) = 0 \) are exactly the configurations which cause the rank of the Jacobian matrix \( dg \) to drop below its generic rank. These are the configurations which cause the vanishing of all (possibly non-maximal) minors of a fixed size. As described in [27], an open problem in commutative algebra is to describe the ideal of polynomial relations among non-maximal minors of a rectangular matrix. For maximal minors, these are the Plücker relations of the Grassmannian. The minors of our matrix \( dg \) should be simpler, however, because \( dg \) has a sparsity pattern determined by the edge structure of the graph.

**Example 1.** Consider the example of a configuration with \( n = 5 \), \( m = 7 \) and \( p_0 \) displayed below. We will briefly discuss the computation required for a primary decomposition of the ideal corresponding to the singular locus.

![Diagram](image)

The Jacobian matrix \( dg \) is

\[
\begin{bmatrix}
  x_{11} - x_{21} & x_{12} - x_{22} & -x_{11} + x_{21} & -x_{12} + x_{22} & 0 & 0 & 0 & 0 & 0 & 0 \\
  x_{11} - x_{31} & x_{12} - x_{32} & 0 & 0 & -x_{11} + x_{31} & -x_{12} + x_{32} & 0 & 0 & 0 & 0 \\
  x_{11} - x_{41} & x_{12} - x_{42} & 0 & 0 & 0 & 0 & -x_{11} + x_{41} & -x_{12} + x_{42} & 0 & 0 \\
  0 & 0 & x_{21} - x_{31} & x_{22} - x_{32} & -x_{21} + x_{31} & -x_{22} + x_{32} & 0 & 0 & 0 & 0 \\
  0 & 0 & x_{21} - x_{41} & x_{22} - x_{42} & 0 & 0 & -x_{21} + x_{41} & -x_{22} + x_{42} & 0 & 0 \\
  0 & 0 & 0 & 0 & x_{31} - x_{51} & x_{32} - x_{52} & 0 & 0 & -x_{31} + x_{51} & -x_{32} + x_{52} \\
  0 & 0 & 0 & 0 & 0 & x_{41} - x_{51} & x_{42} - x_{52} & -x_{41} + x_{51} & -x_{42} + x_{52} & 0
\end{bmatrix}
\]

Let \( I \subset \mathbb{C}[x_{ik}] \) be the ideal generated by the \( m \) polynomials

\[
I = \langle g_1, \ldots, g_m \rangle.
\]

Let \( r \) be the rank of \( dg|_z \) evaluated at a generic point \( z \in \mathbb{C}^{nd} \), and let

\[
S = \langle \text{all } r \times r \text{ minors of } dg \rangle
\]
Finally let $q_1, q_2, \ldots, q_a$ be the associated primes of a primary decomposition of the ideal sum $I + S$. This provides a systematic method for finding the singular points of $V(g)$. In the case of this example we have four associated prime ideals. We list the output of SAGE [35] below, followed by illustrations of the meaning of each of the four prime ideals. For any valid configuration, we can always translate node 1 to the origin, and then rotate the coordinate frame until node 2 is along the $x$-axis. This corresponds to setting $(x_{11}, x_{12}) = (0, 0)$ and $(x_{21}, x_{22}) = (1, 0)$, a simplification we have made in our computations.

```
Ideal (x52, x51 - 2, x42 - 1, x41 - 2, x32 + 1, x31 - 2)
of Multivariate Polynomial Ring in x31, x32, x41, x42, x51, x52 over Rational Field
Ideal (x42 + 1, x41 - 2, x32 + 1, x31 - 2, x51^2 + x52^2 - 4*x51 + 2*x52 + 4)
of Multivariate Polynomial Ring in x31, x32, x41, x42, x51, x52 over Rational Field
Ideal (x52, x51 - 2, x42 + 1, x41 - 2, x32 - 1, x31 - 2)
of Multivariate Polynomial Ring in x31, x32, x41, x42, x51, x52 over Rational Field
Ideal (x42 - 1, x41 - 2, x32 - 1, x31 - 2, x51^2 + x52^2 - 4*x51 - 2*x52 + 4)
of Multivariate Polynomial Ring in x31, x32, x41, x42, x51, x52 over Rational Field
```

Our configuration $p_0$ satisfies the equations in just one of these prime ideals. Finding solutions to the equations in the other prime ideals is to find new singular configurations. We will analyze this example further in Section 7.

5 Polynomial homotopy continuation

In this section we review the basics of homotopy continuation for polynomial systems, which plays a fundamental role in the subject of numerical algebraic geometry and is central to our approach for determining $\varepsilon$-local rigidity. For general references see any of [5, 18, 33]. The main idea is simple. If we have a system of equations $f : \mathbb{C}^N \to \mathbb{C}^N$, whose solutions $f(x) = 0$ we want to find, then we first try to construct another system $g : \mathbb{C}^N \to \mathbb{C}^N$, whose solutions we know. By perturbing $g$ into $f$, we can follow the known solutions, arriving at the previously unknown solutions of $f$. For example, if $f$ is a system of polynomials of degrees $d_i$, then the total degree homotopy begins with a very simple system $g$ whose $D = \prod d_i$ solutions are tuples...
of roots of unity. We then consider the one-parameter homotopy which connects the two systems

\[ H_\gamma(z, t) = (1 - t) \begin{bmatrix} f_1(z) \\ f_2(z) \\ \vdots \\ f_N(z) \end{bmatrix} + \gamma t \begin{bmatrix} z_1^{d_1} - 1 \\ z_2^{d_2} - 1 \\ \vdots \\ z_N^{d_N} - 1 \end{bmatrix}, \]

where \( z = (z_1, \ldots, z_N) \in \mathbb{C}^N \), \( t \in (0, 1] \subset \mathbb{R} \), and \( \gamma \in \mathbb{C} \) is a random complex number which allows us to follow a path through parameter space that avoids problems with probability 1. We can start at one of the known solutions \( z^* \in \mathbb{C}^N \) at parameter value \( t = 1 \), and then we can solve a system of ordinary differential equations written (very briefly) as

\[ \frac{\partial H_\gamma}{\partial z} \frac{dz}{dt} + \frac{\partial H_\gamma}{\partial t} = 0. \]

Here \( \frac{\partial H_\gamma}{\partial z} \) is the Jacobian of \( H_\gamma \) with respect to the variables \( z_1, \ldots, z_N \). The solution \( z(t) \) is a path through \( \mathbb{C}^N \), with initial conditions \( z(1) = z^* \). We can repeat this for each of the \( D \) solutions \( z^* \). The final position \( z(0) \), or its limit \( \lim_{t \to 0} z(t) \), will be a solution to our original system of equations \( f \). We call such a solution path trackable if there is a smooth map \( z : (0, 1] \to \mathbb{C}^N \) with \( z(1) = z^* \) and \( z(t) \) is a nonsingular solution of \( H(z, t) \) for all \( t \in (0, 1] \). In particular, this allows for singular solutions to occur at the endpoints, where more sophisticated endgame path tracking methods are employed. This is the basic idea of polynomial homotopy continuation. The main computation is called path tracking because predictor-corrector methods of ODE solvers are used to follow the solution paths from the known system to the unknown system.

We now briefly consider the situation where \( f : \mathbb{C}^N \to \mathbb{C}^n \), which allows possibly different numbers of equations and unknowns. The algebraic set \( V(f) \) can have irreducible components \( X \) of any dimension among \( N - n, N - n + 1, \ldots, N - 1 \). An irreducible component \( X \) of dimension \( k \) is one such that the set \( X_{\text{reg}} \subset X \) is open, dense, and connected, where \( X_{\text{reg}} \) is the set of all points which are locally biholomorphic to \( \mathbb{C}^k \). For positive-dimensional solution sets the numerical representation of an irreducible component is called a witness set. A witness set is a tuple \((f, L, W)\) where \( f \) is the system of equations, \( L \) is a linear space represented in components by a rectangular matrix whose nullspace is the linear space, and finally \( W \) is a set of finitely many isolated solutions to the set of equations obtained by appending \( Lx = 0 \) to the equations \( f = 0 \). A witness set is obtained by first computing a witness superset \( \tilde{W} \supset W \) and then eliminating extraneous points. In these cases we utilize randomization and Bertini’s theorem.

**Theorem 3** (Theorem 9.3 of [5]). Given a polynomial system \( f : \mathbb{C}^N \to \mathbb{C}^n \), there is a Zariski-open, dense set \( U \subset \mathbb{C}^{k \times n} \) of matrices \( A \) such that \( V(Af) \setminus V(f) \) is either empty or consists of exactly \( C_f \in \mathbb{Z}_{>0} \) irreducible components, each smooth (and hence disjoint) and of dimension \( N - k \). The number \( C_f \) of these irreducible components is independent of \( A \).

We will now give an example of how to calculate a witness superset and corresponding witness set for the union of all 4-dimensional components of some \( V(f) \). Say that \( f : \mathbb{C}^7 \to \mathbb{C}^4 \). Its irreducible components \( X \) can have possible dimensions

\[ \dim X \in \{3, 4, 5, 6\}. \]

To find witness supersets for the components with \( \dim X = 4 \) we would form the following square (7
equations and 7 variables) system of equations:

\[
\begin{bmatrix}
Af \\
Lx
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 & c_{11} \\
0 & 1 & 0 & c_{21} \\
0 & 0 & 1 & c_{31}
\end{bmatrix} \begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
f_4
\end{bmatrix} = \begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6 \\
x_7
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]

Here, the \(4 \times 7\) matrix \(L\) is chosen randomly, as are the three coefficients \(c_{ij}\) in the matrix \(A\). The resulting square system can be solved by homotopy continuation using an appropriate start system and path tracking. The endpoints form a set called the witness superset for the pure 4-dimensional component of \(V(f)\). Witness supersets for the other pure-dimensional components are computed from similar square systems, adjusting the sizes of the matrices \(A\) and \(L\) accordingly. From the witness superset we can form the witness set by

1. Eliminating any solutions coming from the extraneous smooth components arising in Bertini’s theorem by careful evaluation of polynomials against our original system \(f\).

2. Eliminating any solutions which come from higher dimensional components by using the homotopy membership test [5, Section 8.4] and witness sets previously computed for the higher dimensional components.

3. By using the trace test and monodromy [5, Section 10.2], we can further sort the witness set for the pure 4-dimensional component into distinct witness sets for each of the irreducible 4-dimensional components whose union is the pure 4-dimensional component.

6 Setting locations by a change of coordinates

Our goal in this section is to remove rigid motions without changing the space of flexes. Often, when building structures, you attach or fasten them to the ground, or to the wall, or to another structure. In that case, the correct model will treat those nodes as fixed, and not allow them to move at all. For example, if you have a connected structure in \(\mathbb{R}^3\), it is enough to fix three nodes which are not collinear, and there will be no more rigid motions in the space of solutions. Therefore, if the structure you want to model has enough nodes fixed or fastened to the ground, you never have to deal with the rigid motions appearing in the set of valid configurations. The reason is that this deletes node variables from your equations \(g\), and therefore deletes columns of the Jacobian matrix \(dg\). The matrices will be smaller, and the situation simpler.

However, if you are considering a structure which you do not intend to fasten to the ground, for example NASA’s Super Ball Bot,
Theorem 4. Let the affine span of nodes 1 refer to this process as setting locations \( p \) which changes coordinates \((\text{remove the rigid motions but do not change the space of flexes. To see this, note any valid configuration } \hat{x}_{ik})\). We present the general case in the following two theorems.

Proof. We proceed by induction on the number of nodes \( j \). Let the dimension of ambient space be fixed at \( d \). If \( j = 1 \), apply a translation that moves node 1 to the origin. Then its new coordinates satisfy \( \hat{p}_{ik} = 0 \) if \( k \geq 1 \) (all its coordinates are zero). Now say Theorem 4 holds for \( j = d - 1 \). We find a rigid motion that fixes nodes 1, \ldots, \( d - 1 \) and rotates node \( d \) such that \( p_{dd} = 0 \). Let \( H \) be the subspace of \( \mathbb{R}^d \) spanned by the first \( d - 2 \) coordinate axes and \( K \) be the subgroup of \( \text{SO}(\mathbb{R}^d) \) that fixes \( H \). By induction, nodes 1, \ldots, \( d - 1 \) are contained in \( H \). \( K \) is isomorphic to \( \text{SO}(\mathbb{R}^2) \) so we can select a rotation \( r \in K \) that fixes indices 1, \ldots, \( d - 2 \) of every node and rotates node \( d \) such that \( p_{dd} = 0 \) as desired.

Remark 1. We view \( \hat{p}_0 \) as an element of \( \mathbb{R}^N \) where \( N = nd - \binom{d+1}{2} \) by dropping the newly zero coordinates.

Definition 5. We associate to \( \hat{p}_0 \) a system of equations called the fixed member constraints denoted

\[ \hat{g} : \mathbb{C}^N \rightarrow \mathbb{C}^m. \]

As with \( g \), this system enforces the requirement that edges must have constant length and its definition is analogous to that of \( g \). We simply drop the variables corresponding to fixed positions.

Theorem 5. There exists a flex \( p : [0, 1] \rightarrow \mathbb{R}^{nd} \) of the initial configuration \( p_0 \) if and only if there exists a flex \( \widehat{p(t)} : [0, 1] \rightarrow \mathbb{R}^N \) of the initial configuration \( \hat{p}_0 \).
Proof. Order the nodes such that nodes 1, . . . , d span d − 1 dimensions in \( \mathbb{R}^{nd} \). Let \( \pi : \mathbb{R}^{nd} \rightarrow \mathbb{R}^{N} \) be the natural projection and \( \iota : \mathbb{R}^{N} \rightarrow \mathbb{R}^{nd} \) be the natural injection. Let \( R \) be the rigid motion of Theorem 14 such that \( \pi(R(p_0)) = \hat{p}_0 \). First, say we have a flex \( p(t) : [0, 1] \rightarrow \mathbb{R}^{N} \). Then \( R^{-1}(\iota(p(t))) \) is a flex of \( p_0 \) in \( \mathbb{R}^{nd} \). Now suppose we have a flex \( p : [0, 1] \rightarrow \mathbb{R}^{nd} \). By Theorem 14 for every point \( p(t) \), there exists a rigid motion \( r(t) \) which sends \( p(t) \) to another point with \( \hat{r}(t)_{ik} = 0 \) if \( k \geq i \) and \( i \in [d] \). The continuity of \( p(t) \) implies \( r(t) \) is continuous. Thus \( \pi(r(t)p(t)) \) is a flex of \( \hat{p}_0 \). \( \square \)

7 Epsilon local rigidity

In this section we determine whether a given initial configuration \( p_0 \) is \( \varepsilon \)-locally rigid. We fix \( N = nd - \binom{d+1}{2} \).

It is tempting to define a notion of \( \varepsilon \)-locally rigid in the original configuration space \( \mathbb{R}^{nd} \) but since any configuration can be translated by an arbitrary amount, the better notion is applied to the point after coordinate transformation by rigid motions to \( \hat{p}_0 \in \mathbb{R}^{N} \) as in Theorem 14. This motivates the following

**Definition 6.** Let \( p_0 \) be an initial configuration and \( \hat{p}_0 \) be the configuration after setting locations. We say that \( p_0 \) is \( \varepsilon \)-locally rigid if every flex \( p(t) \) of \( \hat{p}_0 \) satisfies \( p(t) \in B_{\varepsilon}(\hat{p}_0) \) for all \( t \in [0, 1] \), where \( B_{\varepsilon}(\hat{p}_0) \) is the open \( \varepsilon \)-ball centered at \( \hat{p}_0 \).

**Remark 2.** In other words, if \( p_0 \) is \( \varepsilon \)-locally rigid, then any positive-dimensional connected component of the real algebraic set \( \mathcal{V}_{\hat{p}_0} \) containing \( \hat{p}_0 \) stays within some ball about \( \hat{p}_0 \) in configuration space. Any flex that may exist can be safely ignored if \( \varepsilon \) is sufficiently small. For all practical purposes, it is as if \( p_0 \) is locally rigid.

We now move towards Theorem 7 and Algorithm 1 to decide whether a configuration \( p_0 \) is \( \varepsilon \)-locally rigid.

**Definition 7.** Let \( \hat{g} = [\hat{g}_1, \ldots, \hat{g}_m]^T \) be the fixed member constraints associated to \( \hat{p}_0 \) as in Definition 1. Define the polynomial system \( \hat{g}_\varepsilon : \mathbb{C}^{N} \rightarrow \mathbb{C} \)

\[
\hat{g}_\varepsilon = \hat{g}_1^2 + \cdots + \hat{g}_m^2 + s_\varepsilon^2,
\]

where \( s_\varepsilon \) is defined by

\[
s_\varepsilon = -\varepsilon^2 + \sum_{k=1}^{n} \sum_{i=1}^{n} (x_{ik} - p_{ik})^2.
\]

The system of \( \varepsilon \)-member constraints associated to \( \hat{p}_0 \) is given by \( \hat{g}_\varepsilon(x) = 0 \). We denote the corresponding algebraic set \( \hat{V}_\varepsilon := \{ x \in \mathbb{C}^{N} : \hat{g}_\varepsilon(x) = 0 \} \).

**Lemma 1.** The irreducible components of \( \hat{V}_\varepsilon \) are of dimension exactly \( N - 1 \).

**Proof.** By Theorem 13.4.2 of [33], the possible dimensions of irreducible components \( X \) of an algebraic set \( V(f) \) for \( f : \mathbb{C}^{N} \rightarrow \mathbb{C}^{n} \) are bounded between

\[
N - \text{rank } f \leq \dim X \leq N - 1,
\]

where the rank of \( f \) is the dimension of the closure of its image as a map, or equivalently, the generic rank of its Jacobian. For a single, nonzero polynomial like \( \hat{g}_\varepsilon : \mathbb{C}^{N} \rightarrow \mathbb{C} \) we have that \( N - 1 \leq \dim X \leq N - 1 \). \( \square \)

Below we will prove Theorem 7 which follows from Theorem 5 of [17]. But first we will state formally the assumptions required for the theorem. We also note that Theorem 5 of [17] draws on results from [8, 31] and also from the 1954 paper of Seidenberg [32].

**Assumption 1.** We collect here the following list of assumptions which refer to the homotopy \( H(x, \lambda, t) \) defined in Theorem 8 below.

12
1. Let \( N > k > 0 \) and \( f : \mathbb{R}^N \to \mathbb{R}^{N-k} \) be a polynomial system with real coefficients, with \( V \subset V(f) \) a pure \( k \)-dimensional algebraic set with witness set \( \{f, L, W\} \).

2. Assume that the starting solutions to \( H(x, \lambda, 1) = 0 \) are finite and nonsingular.

3. Assume also that the number of starting solutions is equal to the maximum number of isolated solutions to \( H(x, \lambda, 1) = 0 \) as \( z, \gamma, y, \alpha \) vary over \( \mathbb{C}^{N-k} \times \mathbb{C} \times \mathbb{C}^{N} \times \mathbb{C}^{N-k+1} \). This will be true for a nonempty Zariski open set of \( \mathbb{C}^{N-k} \times \mathbb{C} \times \mathbb{C}^{N} \times \mathbb{C}^{N-k+1} \).

4. Assume all the solution paths defined by \( H \) starting at \( t = 1 \) are trackable. This means that for each starting solution \((x^*, \lambda^*)\) there exists a smooth map \( \xi : (0, 1] \to \mathbb{C}^{N} \times \mathbb{C}^{N-k+1} \) with \( \xi(1) = (x^*, \lambda^*) \) and for all \( t \in (0, 1] \) we have \( \xi(t) \) is a nonsingular solution of \( H(x, \lambda, t) \).

5. Assume that each solution path converges, collecting the endpoints of all solution paths in the sets \( E \) and \( E_1 = \pi(E) \) where \( \pi(x, \lambda) = x \) projects onto the \( x \) coordinates, forgetting the \( \lambda \) coordinates.

**Theorem 6** (Theorem 5 of [17]). Suppose that the conditions in Assumption 1 hold. Let \( z \in \mathbb{R}^{N-k}, \gamma \in \mathbb{C}, y \in \mathbb{R}^{N} - V_{\varepsilon}(f), \alpha \in \mathbb{C}^{N-k+1}, \) and \( H : \mathbb{C}^{N} \times \mathbb{C}^{N-k+1} \times \mathbb{C} \to \mathbb{C}^{2N-k+1} \) be the homotopy defined by

\[
H(x, \lambda, t) = \begin{bmatrix}
\lambda_0 (x - y) + \lambda_1 \nabla f_1(x)^T + \cdots + \lambda_{N-k} \nabla f_{N-k}(x)^T \\
\alpha^T \lambda - 1
\end{bmatrix}
\]

(4)

where \( f(x) = [f_1(x), \ldots, f_{N-k}(x)]^T \). Then

\[
E_1 \cap V \cap \mathbb{R}^N
\]

contains a point on each connected component of \( V_{\varepsilon}(\hat{f}) \) contained in \( V \).

**Theorem 7.** Let \( p_0 \) be an initial configuration and \( \hat{g}_c \) be according to Definition 1 above. Taking \( f = \hat{g}_c \) in Theorem 6 we find that if conditions two through five in Assumption 1 are met, then

\[
E_1 \cap \hat{V}_c \cap \mathbb{R}^N = \emptyset
\]

implies that \( p_0 \) is \( \varepsilon \)-locally rigid.

**Proof.** Take \( f = \hat{g}_c \) and \( V = \hat{V}_c \) in the notation of Theorem 6 above. We have \( N - k = 1 \) and by Lemma 1 all irreducible components are of dimension \( k = N - 1 \). Therefore, the first condition of Assumption 1 is met.

Say \( E_1 \cap \hat{V}_c \cap \mathbb{R}^N = \emptyset \) but \( p_0 \) is not \( \varepsilon \)-locally rigid. Let \( \tilde{p}(t) \) be a flex such that \( \tilde{p}(1) \notin B_c(\hat{p}_0) \) and let

\[
P = \tilde{p}([0, 1])
\]

be the image of \( \tilde{p} \). Then \( (P \cap B_c(\hat{p}_0)) \cap (P \cap B_c(\hat{p}_0)^c) \) is a separation of \( P \) contradicting the continuity of \( \tilde{p} \).

This result suggests the following Algorithm 1.

**Algorithm 1:** Epsilon local rigidity

**Input:** Initial configuration \( p_0 \in \mathbb{R}^{nd} \), edge set \( E \), and choice of \( \varepsilon > 0 \).

**Result:** Boolean \( v \) which is true if items 2, 4, and 5 of Assumption 1 are satisfied. Boolean \( u \) which is true if the set \( E_1 \cap \hat{V} \cap \mathbb{R}^N \) of Theorem 7 above is empty, and false otherwise. Set \( R \) which may be empty or else contains at least one point on each connected component of \( V_{\varepsilon}(\hat{g}_c) \).

1. Apply the rigid motions of Theorem 4 to \( p_0 \) obtaining \( \hat{p}_0 \in \mathbb{R}^N \) for \( N = nd - \binom{d+1}{2} \).
2. Form the systems of equations \( \hat{g}_c \) according to Definition 7.
3. Calculate a witness set \( W \) for the pure \( N - 1 \) dimensional algebraic set \( V(\hat{g}_c) \subset \mathbb{C}^{N} \).
4. Produce \( z, \gamma, y, \alpha \) such that item 5 of Assumption 1 holds.
5. Use the algorithm presented in Section 2.1 of [17] obtaining the boolean \( v \) and the set of real solutions \( R \).
6. If \( R \) is the empty set, set \( u \) as true, else set \( u \) as false.
7. Output the booleans \( v \) and \( u \), and the set \( R \).
Remark 3. An appropriate choice of \( y \in \mathbb{R}^N \setminus V_\mathbb{R}(\hat{g}_\varepsilon) \) could be \( p_0 \) itself, or \( p_0 + \mathcal{N}(0, \sigma^2) \) for some random multivariate Gaussian noise with mean zero and variance \( \sigma^2 \). In step 4 above, if items 2, 3, 4, or 5 of Assumption 1 fail to hold, then generating new and random points \( z, \gamma, y, \alpha \) could be required.

Example 2. As an illustrative example, we would like a configuration which we know to be locally rigid, but fails to be infinitesimally rigid so that Theorems 1 and 2 do not apply, and which is also a singular point of the member constraints so that the local real dimension may differ from the local complex dimension computed in \(^{37}\). As a simple example, consider the configuration in Example 1. It is singular, and admits an additional infinitesimal mechanism, hence Theorems 1 and 2 do not apply. However, it is also simple enough that we can see it is locally rigid. Since the triangles among nodes 123 and 124 are both rigid, the only node that could possibly move in a continuous flex is node 5. However, we can also see that node 5 is restricted by node 3 to move in one circle of radius equal to the edge length of edge 35, while by node 4 it is restricted to another similar circle. These circles intersect in exactly one point, the location of node 5 in our initial configuration \( p_0 \). Thus \( p_0 \) is locally rigid.

We implemented the above Algorithm 1 for this example using HomotopyContinuation.jl in julia, obtaining \( \varepsilon \)-local rigidity for \( \varepsilon \in \{0.1, 0.01, 0.001, 0.0001\} \). We generated the point \( y \) by Gaussian noise applied to the original configuration \( p_0 \), and tracked paths, obtaining zero real solutions. For each value of \( \varepsilon \) all paths were trackable and the items in Assumption 1 were satisfied, but none of the resulting solutions were real-valued. Therefore, we can conclude by Theorem 4 that this configuration \( p_0 \) is \( \varepsilon \)-locally rigid for \( \varepsilon \in \{0.1, 0.01, 0.001, 0.0001\} \).

The example we just described is a singular configuration depicted in Example 1 in the illustration of the associated primes on the left. Now consider the singular configuration depicted on the right in that same illustration. This configuration is clearly not locally rigid. However, again it is singular and admits additional infinitesimal mechanisms, hence Theorems 1 and 2 do not apply. In this singular configuration, nodes 3 and 4 coincide, yielding a flexible node 5 which can freely move around a circle centered at nodes 3, 4. This configuration is obviously flexible, and when we run Algorithm 1 we obtain exactly two real solutions (after projecting away the \( \lambda \) components). Upon examination, these two real solutions correspond to node 5 moving in either possible direction along a circle centered at nodes 3, 4, as expected.

8 Producing a discrete flex

In this section we describe an algorithm that repeatedly solves a system of parametrized polynomial equations in order to produce a sequence of real-valued, valid configurations. If a continuous flex of \( p_0 \) exists, then this procedure will produce a discrete sampling of points from that continuous flex. The resulting sequence of configurations may be plotted and animated, yielding easily understandable information for the scientist. In future work, we plan to implement this algorithm in a freely available julia package, utilizing the existing algorithms of the julia package HomotopyContinuation.jl \(^{7}\), which implements polynomial homotopy continuation as discussed in Section 5. A main goal for our package will be ease of use. This is currently under development, but a rough example of its output is shown in Figure 1. These are images of a discrete flex with \( M = 100 \) points computed using homotopy continuation on a cube deforming freely. The cube is obviously flexible, and the goal is to implement software which will find more surprising flexes from other, more complicated examples. We also note that other excellent software exists for homotopy continuation, including \(^{14, 25, 34}\).
Algorithm 2: A discrete flex

Input: Initial configuration $p_0 \in \mathbb{R}^{nd}$, edge set $E$, choice of $\varepsilon_0 > 0$ and $M \in \mathbb{Z}_{>0}$.
Result: A discrete flex in the form of a list $P$ of configurations $p_1, p_2, \ldots, p_M$ to be animated and visualized, or potentially the message of $\varepsilon$-local rigidity for some $\varepsilon = j \cdot \varepsilon_0$.

1. Initialize a list with one element $P = [p_0]$ to be filled with more points $p_j \in \mathbb{R}^N$ as in $p_1, p_2, \ldots, p_M$ if the algorithm succeeds.

2. for $j$ in $1:M$ do
   3. Set $\varepsilon := j \cdot \varepsilon_0$.
   4. Apply Algorithm 1 with inputs $p_0, E, \varepsilon$, collecting the outputs $u, v, R$.
   5. if $u = 1$ then
      6. Output the current list $P$ and the message that $p_0$ is $\varepsilon$-locally rigid.
   7. else
      8. Collect the output set $R$ and store it as $R_j = R$.
      9. Set $p_j := \text{argmin}\{\text{dist}(p_{j-1}, w) : w \in R_j\}$.
     10. Append $p_j$ to the list $P$.
   11. end
12. end
13. Return the list $P$ as well as an animation of each of its $M$ configurations displayed in $\mathbb{R}^d$ if $d = 2, 3$.

There are many possible alterations of the above algorithm, which we plan to explore in our implementation. First, the 2-homogeneous structure should be exploited in generating start systems. Second, a parameter homotopy could be used after obtaining the first new solution $p_1$. Consider the square system of equations

$$F_{y,\epsilon}(x_1, \ldots, x_N, \lambda_0, \lambda_1) = \begin{bmatrix} \hat{g}_\epsilon(x) \\ \lambda_0 (x - y) + \lambda_1 \nabla \hat{g}_\epsilon \\ \alpha_0 \lambda_0 + \alpha_1 \lambda_1 - 1 \end{bmatrix} : \mathbb{C}^{N+2} \rightarrow \mathbb{C}^{N+2}.$$  

After solving and finding a new configuration $p_1$ at radius $\varepsilon$ away from $p_0$, this means we have obtained solutions $(x, \lambda)$ to $F_{y,\epsilon}$ for some specific $y \in \mathbb{R}^N$ and $\varepsilon > 0$, where one of these solutions has $x = p_1$. We could then consider the homotopies perturbing $y$ to $y'$ or perturbing $\varepsilon$ to $\varepsilon'$ as in

$$H_y(x, \lambda, t) = F_{(1-t)y+t_y,\epsilon}(x, \lambda)$$  \hspace{1cm} (5)

or

$$H_\epsilon(x, \lambda, t) = F_{y, (1-t)\epsilon+t_\epsilon}(x, \lambda),$$  \hspace{1cm} (6)
either of which would produce new and relevant configurations. In particular, using $H_\varepsilon$ can allow us to
generate new solutions for expanding (or contracting) $\varepsilon$-balls about $p_0$, slightly modifying line 4 of Algorithm

**Remark 4.** In the homotopies above we have removed the usual factor $\gamma$. This allows real-valued solutions
like $p_1$ to stay real-valued along the parameter homotopy. This will succeed unless we cross the discriminant,
so we can expect success for $|\varepsilon - \varepsilon'|$ small. For small perturbations $\varepsilon$ to $\varepsilon'$ we can numerically follow a
continuous flex and every step of the path-tracking process will compute new and real-valued configurations
sampled from that continuous flex. More precisely, starting from a non-singular solution to $F_{y,\varepsilon}$ we can continue to a non-singular solution of $F_{y,\varepsilon'}$ for $|\varepsilon - \varepsilon'|$ small. Even if we do cross the discriminant, it may
be due to phenomena involving complex solutions elsewhere, and thus not affect our particular real-valued solution.

**Remark 5.** In particular, letting $\varepsilon \to 0$ we can attempt to follow any real-valued points $p_1$ on the $\varepsilon$-sphere
towards $p_0$. If there is a continuous flex of $p_0$ we can expect one of the real-valued solutions on the $\varepsilon$-sphere
to move towards $p_0$ as $\varepsilon \to 0$. The collection of points computed along the way are a discrete flex of $p_0$,
having been sampled from a continuous flex. We can also follow the discrete flex away from $p_0$ by letting $\varepsilon$
increase, computing points in a parameter homotopy as above.

**Remark 6.** It is tempting to use monodromy or the trace test on each of the $p_j$ to ensure they are points on the
same irreducible component. However, even if they are on the same irreducible component, they could be
on distinct connected components of the real algebraic set. Therefore, applying the trace test or monodromy
are necessary but not sufficient conditions for our discrete flex to be sampled from the same connected component of $V_{\varepsilon}(\bar{g})$. This option could be added to Algorithm 2 explicitly, but this would require witness
sets for each irreducible component of $V(\bar{g})$ be computed, whereas here we are only assuming computation of a witness set for the pure $(N-1)$-dimensional component of $V(\bar{g}_\varepsilon)$.

### References

[1] Timothy Abbott, Reid Barton, and Eric Demaine. Generalizations of Kempe’s Universality Theorem. Master’s Thesis, MIT, 2009.

[2] L. Asimow and B. Roth. The rigidity of graphs. II. J. Math. Anal. Appl., 68(1):171–190, 1979.

[3] Philippe Aubry, Fabrice Rouillier, and Mohab Safey El Din. Real solving for positive dimensional systems. J. Symbolic Comput., 34(6):543–560, 2002.

[4] Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler. Bertini: Software for numerical algebraic geometry. Available at bertini.nd.edu with permanent doi: dx.doi.org/10.7274/R0H41PB5.

[5] Daniel J. Bates, Andrew J. Sommese, Jonathan D. Hauenstein, and Charles W. Wampler. Numerically Solving Polynomial Systems with Bertini. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2013.

[6] Daniel A. Brake, Jonathan D. Hauenstein, and Andrew J. Sommese. Numerical local irreducible decomposition. In Mathematical aspects of computer and information sciences, volume 9582 of Lecture Notes in Comput. Sci., pages 124–129. Springer, [Cham], 2016.

[7] Paul Breiding and Sascha Timme. Homotopycontinuation.jl: A package for homotopy continuation in julia. In James H. Davenport, Manuel Kauers, George Labahn, and Josef Urban, editors, Mathematical Software – ICMS 2018, pages 458–465, Cham, 2018. Springer International Publishing.

[8] R. Connelly and S.D. Guest. Frameworks, Tensegrities and Symmetry. To appear, 2020.

[9] Robert Connelly. The rigidity of certain cabled frameworks and the second-order rigidity of arbitrarily triangulated convex surfaces. Adv. in Math., 37(3):272–299, 1980.
[10] Robert Connelly and Herman Servatius. Higher-order rigidity—what is the proper definition? Discrete Comput. Geom., 11(2):193–200, 1994.

[11] Barry H. Dayton and Zhonggang Zeng. Computing the multiplicity structure in solving polynomial systems. In ISSAC’05, pages 116–123. ACM, New York, 2005.

[12] M. Dehn. Über die Starrheit konvexer Polyeder. Math. Ann., 77(4):466–473, 1916.

[13] Florian Geiß and Frank-Olaf Schreyer. A family of exceptional Stewart-Gough mechanisms of genus 7. Contemporary Mathematics, page 221–234, 2009.

[14] Grigore Gogu. Mobility of mechanisms: a critical review. Mech. Mach. Theory, 40(9):1068–1097, 2005.

[15] Jacob E. Goodman, Joseph O’Rourke, and Csaba D. Tóth, editors. Handbook of discrete and computational geometry. Discrete Mathematics and its Applications (Boca Raton). CRC Press, Boca Raton, FL, 2018. Third edition of [ MR1730156].

[16] M. Grüber. Getriebelehre: eine Theorie des Zwanglaufes und der ebenen Mechanismen. Springer, 1917.

[17] Jonathan D. Hauenstein. Numerically computing real points on algebraic sets. Acta Applicandae Mathematicae, 125(1):105–119, Sep 2012.

[18] Jonathan D. Hauenstein and Andrew J. Sommese. What is numerical algebraic geometry? Journal of Symbolic Computation, 79:499 – 507, 2017. SI: Numerical Algebraic Geometry.

[19] Alexander Heaton. Nonlinear algebra via tensegrity structures. arXiv e-prints, page arXiv:1908.08392, August 2019.

[20] J.M. Hervé. Analyse structurelle des mécanismes par groupe des déplacements. Mechanism and Machine Theory, 13(4):437 – 450, 1978.

[21] Miranda Holmes-Cerfon, Louis Theran, and Steven J. Gortler. Almost-rigidity of frameworks, 2019.

[22] A. Karger. Architecture Singular Parallel Manipulators, pages 445–454. Springer Netherlands, Dordrecht, 1998.

[23] K. Kutzbach. Mechanische leitungsverzweigung, ihre gesetze und anwendungen. Maschinenbau, der Betrieb, 1929.

[24] G. Laman. On graphs and rigidity of plane skeletal structures. J. Engrg. Math., 4:331–340, 1970.

[25] 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(2-3):109–133, 2008.

[26] F. S. Macaulay. The algebraic theory of modular systems. Cambridge Mathematical Library. Cambridge University Press, Cambridge, 1994. Reprinted reprint of the 1916 original, With an introduction by Paul Roberts.

[27] Mateusz Michałek and Bernd Sturmfels. Invitation to Nonlinear Algebra. 2019. https://personal-homepages.mis.mpg.de/michalek/NonLinearAlgebra.pdf.

[28] Hilda Pollaczek-Geiringer. Über die gliederung ebener fachwerke. ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik, 7:58–72, 1927.

[29] Hilda Pollaczek-Geiringer. Zur gliederung raumlicher fachwerke. ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik, 12:369–376, 1932.

[30] Zvi Rosen, Jessica Sidman, and Louis Theran. Algebraic matroids in action, 2018.

[31] F. Rouillier, M.-F. Roy, and M. Safey El Din. Finding at least one point in each connected component of a real algebraic set defined by a single equation. J. Complexity, 16(4):716–750, 2000.
[32] A. Seidenberg. A new decision method for elementary algebra. *Ann. of Math. (2)*, 60:365–374, 1954.

[33] Andrew J Sommese and Charles W Wampler. *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific, 2005.

[34] G. Strang. *Computational Science and Engineering*. Wellesley-Cambridge Press, 2007.

[35] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 8.8)*, 2019. https://www.sagemath.org.

[36] Jan Verschelde. Algorithm 795: Phcpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Trans. Math. Softw.*, 25(2):251–276, June 1999.

[37] Charles W. Wampler, Jonathan D. Hauenstein, and Andrew J. Sommese. Mechanism mobility and a local dimension test. 2011.

[38] Charles W. Wampler and Andrew J. Sommese. Numerical algebraic geometry and algebraic kinematics. *Acta Numerica*, 20:469–567, 2011.

[39] Walter Whiteley. Matroids and rigid structures. In *Matroid applications*, volume 40 of *Encyclopedia Math. Appl.*, pages 1–53. Cambridge Univ. Press, Cambridge, 1992.

[40] Walter Whiteley. Some matroids from discrete applied geometry. In *Matroid theory (Seattle, WA, 1995)*, volume 197 of *Contemp. Math.*, pages 171–311. Amer. Math. Soc., Providence, RI, 1996.

[41] K. Wohlhart. *Kinematotropic Linkages*, pages 359–368. Springer Netherlands, Dordrecht, 1996.

[42] Zhonggang Zeng. The closedness subspace method for computing the multiplicity structure of a polynomial system. In *Interactions of classical and numerical algebraic geometry*, volume 496 of *Contemp. Math.*, pages 347–362. Amer. Math. Soc., Providence, RI, 2009.