1-1-2005

Pseudo-Triangulations, Rigidity and Motion Planning

Ileana Streinu
Smith College, istreinu@smith.edu

Pseudo-Triangulations, Rigidity and Motion Planning

Ileana Streinu *

Abstract

We propose a combinatorial approach to planning non-colliding trajectories for a polygonal bar-and-joint framework with n vertices. It is based on a new class of simple motions induced by expansive one-degree-of-freedom mechanisms, which guarantee non-collisions by moving all points away from each other. Their combinatorial structure is captured by *pointed pseudo-triangulations*, a class of embedded planar graphs for which we give several equivalent characterizations and exhibit rich rigidity theoretic properties.

The main application is an efficient algorithm for the *Carpenter's Rule Problem*: convexify a simple bar-and-joint planar polygonal linkage using only non-self-intersecting planar motions. A step of the algorithm consists in moving a pseudo-triangulation-based mechanism along its unique trajectory in configuration space until two adjacent edges align. At the alignment event, a local alteration restores the pseudo-triangulation. The motion continues for $\mathcal{O}(n^3)$ steps until all the points are in convex position.

1 Introduction

We present a combinatorial solution to the *Carpenter's Rule Problem*: how to plan non-colliding reconfigurations of a planar robot arm. The main result is an efficient algorithm for the problem of continuously moving a simple planar polygon to any other configuration with the same edge-lengths and orientation, while remaining in the plane and never creating self-intersections along the way. This is done by first finding motions that convexify both configurations with expansive motions (which never bring two points closer together) and then taking one path in reverse.

All of the constructions are elementary and are based on a novel class of planar embedded graphs called *pointed pseudo-triangulations*, for which we prove a variety of combinatorial and rigidity theoretical properties. More prominently, a pointed pseudo-triangulation with a removed convex hull edge is a one-degree-of-freedom expansive mechanism. If its edges are seen as rigid bars (maintaining their lengths) and are allowed to rotate freely around the vertices (joints), the mechanism follows (for a well defined, finite time interval) a continuous trajectory along which no distance between a pair of points ever decreases. The expansive motion induced by these mechanisms provide the building blocks of our algorithm.

Historical remark. This paper is a systematic, detailed and self-contained presentation of a 10-page conference version [73] which appeared in 2000. In the same conference appeared the independent solution of [30] for the Carpenter's Rule problem, which has since been published as a full paper. Therefore, all the references we give to *previous work* refer to the state of affairs in 2000. The pointed pseudo-triangulations and their special combinatorial and rigidity-theoretical

*Department of Computer Science, Smith College, Northampton, MA 01063, USA. streinu@cs.smith.edu.
This work was partially supported by NSF grant CCR-0105507.

properties, which are the highlight of our solution, found a life of their own since 2000, and a flurry of papers (some extending the results presented here) emerged. For completeness, we also include in the Conclusions a list of references to these recent results.

In the remaining of the introduction, we give an informal high-level preview of the results and their connection with previous work. Precise definitions and complete proofs are given in the rest of the paper. Section 2 contains the definition of pointed pseudo-triangulations and several of their combinatorial characterizations. Sections 3 and 4 prove the rigidity theoretic properties of pointed pseudo-triangulations on which the whole approach relies. Section 5 describes a few simple algorithms for computing pointed pseudo-triangulations of planar point sets and polygons. Section 6 gives the description of the global convexification motion and the complexity analysis of the combinatorial, non-algebraic part of the algorithm. We conclude with some suggestions for further research.

Frameworks and Robot arms. A *bar-and-joint framework* is a graph $G = (V, E)$ embedded in the plane with rigid bars corresponding to the edges (the edge lengths are considered given and fixed). The bars are free to move in the plane around their adjacent joints (vertices), as long as their lengths are preserved. The motions impose no restriction on the non-edges, which may increase or decrease freely. In this general model, edges may cross and slide over each other during the motion, but in this paper we are interested in avoiding collisions and will not allow this. Of particular interest are the *expansive motions*, where no pairwise interdistance between vertices ever decreases during the motion, thus guaranteeing non-collision.

A *chain, linkage* or *robot arm* is a planar framework whose underlying graph is a simple (non-self-intersecting) path, and a *closed chain* is a simple planar polygon. Throughout, n will denote the number of its vertices. *Straightening a chain* refers to moving it continuously until all its vertices lie on one line with non-overlapping edges. *Convexifying a closed chain* means moving it to a position where it forms a simple convex polygon. Other types of frameworks of interest in this paper include Laman graphs and pointed pseudo-triangulations, defined in Section 2.1.

The Carpenter's Rule Problem: Given a closed chain, orient it so that the interior lies to the left when walking along the polygon in the positive direction. While avoiding self-collisions and staying in the plane, we want to continuously reconfigure the chain from an initial to a final configuration with the same orientation. It suffices to show that we can convexify the chain. Then, to move between any two similarly oriented configurations, we will take one path in reverse. Indeed, it is easy to move between two distinct convex positions, see [9]. The Carpenter's Rule Problem asks: *Is it always possible to straighten a planar linkage, or to convexify a planar chain?*

This question has been open since the 1970's. Recently, Connelly, Demaine and Rote [30] have answered it in the affirmative. Their solution still left open the problem: *Find, algorithmically, a finite sequence of simple (finitely described) motions to straighten a linkage, or to convexify a polygon.*

Previous Results on Reconfiguring Linkages. The general techniques for solving motion planning problems based on roadmaps [70, 27, 16, 17] work well on problems with constant number of degrees of freedom, but would yield exponential algorithms in our case. Under various conditions, problems about reconfiguration of linkages range in complexity from polynomial [52] to NP- and even PSPACE-hard, see [43, 84, 44].

The particular problem of straightening bar-and-joint linkages and convexifying polygons has accumulated a distinguished history, with some approaches going back to a question of Erdös

[34]. See Toussaint [79] for a fascinating account. There are abundant connections with work done in the computational biology, chemistry and physics literature and motivated by topics such as protein folding or molecular modeling. When crossings are allowed, Lenhart and Whitesides [52] have shown that the configuration space has at most two connected components and gave a linear algorithm for convexification based on simple motions which displace (relatively) only a constant number of joints at a time. Recent results in the mathematics literature [45, 58, 80] aim at understanding the topology of the configuration space of closed chains, but they allow crossings.

Studying reconfigurations of linkages with non crossing motions has received a recent impetus in [54], and results on planar linkages using spatial motions [19, 11], trees, three- and higher-dimensional linkages [20, 29] have followed. The Carpenter's Rule question, raised in the 1970's in the Topology community by G. Bergman, U. Grenander, S. Schanuel (cf. [48]) and independently in the early 1990's in the Computer Science community by two groups (W. Lenhart and S. Whitesides, resp. J. Mitchell), seems to have first appeared in print in [52] and [48]. It was recently settled by Connelly, Demaine and Rote [30]: all chains can be convexified, all linkages can be straightened. Their approach is based on Rote's seminal idea of using *expansive motions* to guarantee non-collisions. They first prove (using linear programming duality and Maxwell's theorem, specifically a technique originating in Crapo and Whiteley [33] and Whiteley [81]) that there always exists an infinitesimal expansive motion, i.e. one which never decreases any distances. The actual velocities can be found using linear programming. Then they provide a global argument, showing the existence of a continuous deformation obtained by integrating the resulting vector field.

The Main Results. We strengthen and provide an algorithmic extension of the above mentioned Carpenter's Rule result of [30]. We show how to compute a path in configuration space, consisting in at most $\mathcal{O}(n^3)$ simple motions along algebraic curve segments, between any two polygon configurations. Along the way, we obtain a result of independent interest in Rigidity Theory. Namely, we characterize a family of planar infinitesimally rigid, self-stress-free frameworks called *pointed pseudo-triangulations*, which yield one-degree-of-freedom (1dof) *expansive* mechanisms when a convex hull edge is removed.

Overview of the Convexification Algorithm. The convexifying path, seen as the collection of the $2n$ trajectories of the $2n$ coordinates $(x_i, y_i), i = 1, \cdots, n$ of the vertices of the polygon, is a finite sequence of algebraic curve segments (arcs) connecting continuously at their endpoints.

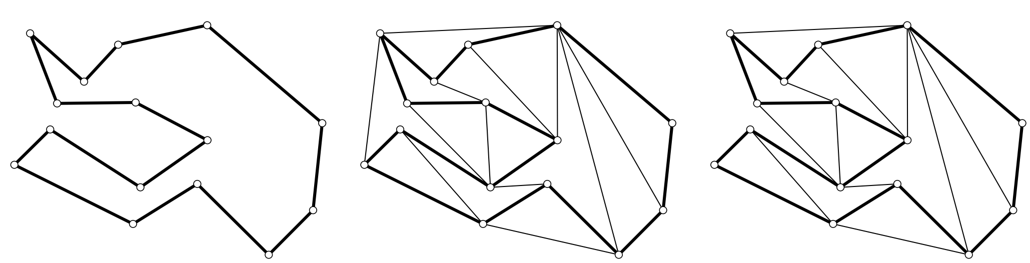

Figure 1: Left: A simple polygon. Middle: one of its pointed pseudo-triangulations. Right: the pointed pseudo-triangulation mechanism obtained by removing a convex hull edge.

Each arc corresponds to the unique free motion of the expansive, one-degree-of-freeedom (1dof) mechanism induced by a planar pointed pseudo-triangulation of the given polygon, where a convex hull edge has been removed and a remaining edge has been pinned down. The mechanism is constructed by adding $n-4$ bars to the original polygon in such a way that there are no crossings, each vertex is incident to an angle larger than π and exactly one convex hull edge is missing. See Fig. 1. We show that this can be done algorithmically in $\mathcal{O}(n)$ time. The mechanism is then set in motion by pinning down one edge and rotating another edge around one of its joints. The framework now moves *expansively*, thus guaranteeing a collision-free trajectory. One step of the convexification algorithm consists in moving this mechanism until two incident edges align, at which moment it ceases to be a pointed pseudo-triangulation. We either freeze a joint (if the aligned edges belong to the polygon) and locally patch a pointed pseudo-triangulation for a polygon with one less vertex, or otherwise perform a local flip of the added diagonals. See Fig. 2.

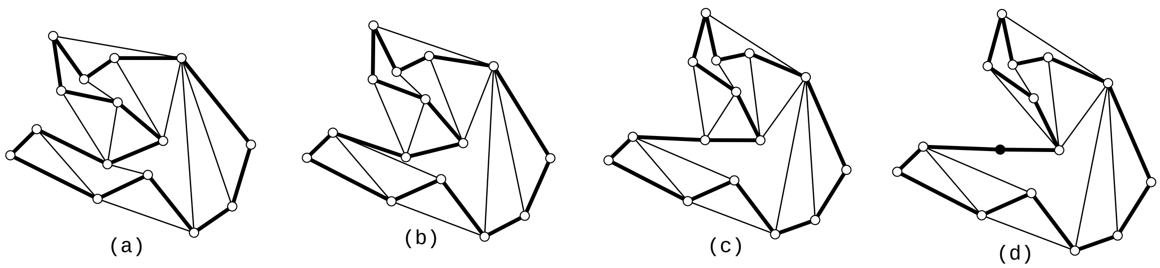

Figure 2: (a) A pointed pseudo-triangulation mechanism just before an alignment event and (b) after the event, when a flip was performed. (c) Continuing the motion, the next event aligns two polygon edges. (d) The aligned vertex (black) is frozen, the pseudo-triangulation is locally restructured and the motion can continue.

There are many ways to construct the initial pointed pseudo-triangulation or to readjust it at an alignment event. For the sake of the analysis, we use a canonical pseudo-triangulation based on shortest-path trees inside the polygon and its pockets. During the convexification process, the total number of bends in the shortest paths, which is bounded by $O(n^2)$ for n active (not frozen) vertices, decreases by at least one at each flip event. There are at most $O(n)$ freeze-events (more precisely, as many as there are reflex vertices). The total number of events, and thus the number of steps induced by simple pseudo-triangulation mechanism motions, is therefore $\mathcal{O}(n^3)$.

Formally stating and proving these results will take the rest of the paper. It will require background material in Rigidity Theory and some intuitions from Oriented Matroids and Visibility. In the context of these fields, we offer next an overview of the theoretical contributions of this paper.

Pointed pseudo-triangulations in the context of Oriented Matroids and Rigidity Theory. Our approach is based on the idea of abstracting (partial) oriented-matroidal properties that hold throughout a portion of a continuous motion. To the best of our knowledge, this idea has not been used before in any other context. This leads to our introduction of the *pointed pseudo-triangulations* of planar point sets (and their further generalization in the

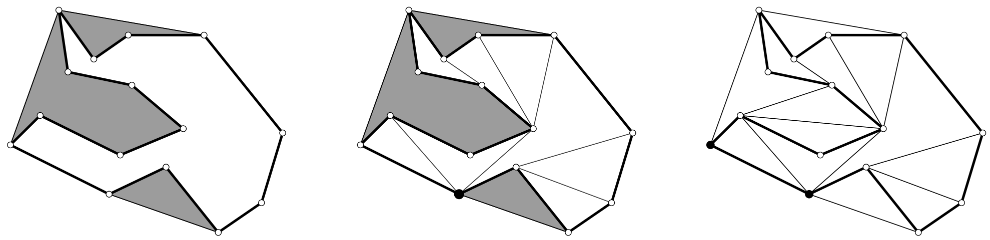

Figure 3: Left: The pockets of the polygon from Fig. 1. Middle: the shortest-path-tree pseudo-triangulation of the interior of the polygon. The source vertex of the tree is black. Right: a complete pointed pseudo-triangulation obtained by taking shortest path trees in all the pockets, and in the interior of the polygon.

companion paper [74] to rank 3 affine oriented matroids, via planar pseudo-point configurations). We emphasize that for these objects, the focus is on *pointedness* (incidence with an angle larger than π), rather than on the partitioning into pseudo-triangular faces (which appears as a consequence), and that they have special properties not shared (and not considered) by other types of pseudo-triangulations previously defined in the literature. Indeed, the combinatorial and rigidity theoretic properties of pointed pseudo-triangulations, as well as their application to expansive mechanisms for finding non-colliding paths in configuration spaces, were discovered by the author and have first appeared in the original conference paper [73], of which the current paper is a comprehensive development.

At the technical, rigidity theoretic level, we prove a new result on the non-existence of self-stresses in bar-and-strut pointed and non-crossing frameworks, which are more complex structures than those arising from polygons. We also prove that pointed pseudo-triangulations are *always* self-stress-free, and therefore (because of having the right number $2n-3$ of edges) infinitesimally rigid. To the best of our knowledge, no other combinatorially-defined class of frameworks was previously shown to possess such strong rigidity theoretical properties. As a consequence, we obtain a generalization of a key lemma in [30] regarding the existence of expansive motions of certain families of linkages, in the strongest way in which this can be done *combinatorially*.

All these preliminary results imply that the configuration space of 1dof pseudo-triangulation frameworks is *smooth* in the neighborhood of a given realization, as long as the combinatorial structure of the embedding (what we call the *combinatorial pseudo-triangulation*) doesn't change. In particular, a generic planar one-degree-of-freedom mechanism (Laman graph with a convex hull removed edge), when embedded as a pointed pseudo-triangulation, always *moves* along a unique, well-defined, one-dimensional trajectory. To the best of our knowledge, this is the first result in the Rigidity Theory literature where the smooth nature of the configuration space is characterized *combinatorially*: in this case, the combinatorics consists not only in the underlying graph structure (a planar Laman graph), but also in the *oriented matroid* properties of the underlying point set on which the Laman graph is embedded (the pointedness of the embedding). Indeed, to contrast the general situation to what we exhibit here and help the reader understand this point, one can easily exhibit examples of infinitesimally flexible frameworks which are *rigid*. In these cases, the infinitesimal motion doesn't extend to a finite motion, and locally the configuration space is an isolated point. Or, one can exhibit examples where the framework is flexible, but lies at a singular point in its configuration space, which complicates the design of which trajectory to follow. Our results imply that this can *never* happen for pointed pseudo-triangulation mechanisms.

Expansive Motions and Pseudo-triangulations in the context of previous work. Our pointed pseudo-triangulations are slight specializations (from smooth obstacles to simple point sets, from emphasis on pseudo-triangular faces to emphasis on pointedness of vertices) of those introduced by Pocchiola and Vegter [62, 63] in their study of the visibility complex and recently applied to kinetic geometric algorithms [1, 15].

A 1dof infinitesimally expansive mechanism obtained from a pointed pseudo-triangulation is a combinatorial abstraction and a canonical representation of one of the many basic feasible solutions of infinitesimally expansive motions, that the linear programming approach of [30] would find for a certain position of the polygon in its configuration space. This idea is further developed in [68].

We characterize pointed pseudo-triangulations in several equivalent ways. Some of these are specialized versions of Laman's $2n-3$ count and Henneberg constructions from combinatorial rigidity [82, 38]. The proof of correctness of our approach derives from these properties, as well as from a generalization, from simple polygons to the wider class of pointed pseudo-triangulation frameworks, of the approach used in [30] based on Linear Programming duality and Maxwell's theorem. This generalization also simplifies the argument needed to guarantee the existence of a global motion, which is now a simple consequence of a basic, fundamental theorem in Calculus, or, alternatively, of the most fundamental theorem in the theory of ordinary differential equations.

References. For background terminology and basic results in rigidity theory, we refer the reader to [66, 82, 83, 38]. In particular, rigidity, infinitesimal (first-order) and generic rigidity, as well as classical results on 2-dimensional rigidity such as Laman's theorem, the Henneberg constructions and Maxwell's Theorem are to be found there. For oriented matroids, see [21], although we won't need more than the intuitions gained through familiarity with the local sequences of [37] (known also as hyperline sequences in [22]), see also [72, 23].

Notation, abbreviations and terminology. Our setting is the Euclidian plane. We abbreviate "counter-clockwise" as *ccw* and "one-degree-of-freedom mechanism" as *1dof mechanism.*

For self-containment, we introduce all the basic terminology and definitions from rigidity theory, graph theory and oriented matroids. Other concepts used in this paper, such as pointed and minimum pseudo-triangulations, combinatorial frameworks, semi-simplicity and expansive 1dof mechanisms are (to the best of our knowledge) new and have not been defined elsewhere. In contrast with the preliminary version of this paper [73], we have settled for a friendlier terminology, and use *pointed* instead of *acyclic* set of vectors, *pointed pseudo-triangulation* instead of acyclic or minimum pseudo-triangulation and *expansive* instead of *monotone* motion or mechanism. See the *Notes* in the Conclusion section for further remarks on the choice of terminology.

2 Combinatorial Properties of Pointed Pseudo-Triangulations

We start by defining *pointed pseudo-triangulations* and derive their main combinatorial properties. At the end of this section, we would have acquired the first piece of evidence that pointed pseudo-triangulations have relevant rigidity theoretic properties: their underlying graphs are minimally generically rigid (Laman) graphs. For this to become an useful algorithmic tool, we'll have to prove later that the rigidity property holds for *any* pseudo-triangular embedding, not just generically.

2.1 Definitions: Pointed Pseudo-Triangulations

Throughout the paper, $G = (V, E)$ will denote a graph with $n = |V|$ vertices and $m = |E|$ edges. The vertex set will be taken as $V = [n] := \{1, 2, \cdots, n\}$.

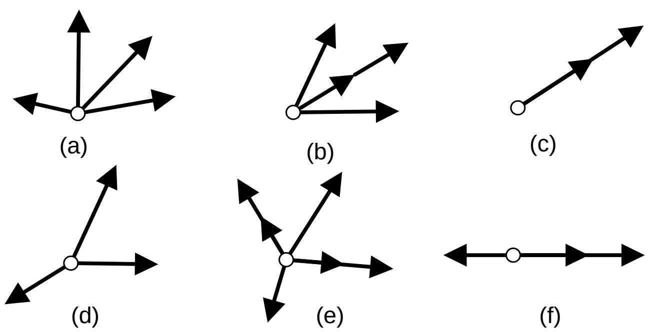

Figure 4: (a), (b) and (c) Pointed (acyclic) and (d), (e) and (f) non-pointed (cyclic) sets of vectors.

Graph Embedding and Planarity. An *embedding* or *drawing* $G(\mathcal{P})$ of the graph G on a set of points $\mathcal{P} = \{p_1, \cdots, p_n\} \subset R^2$ is a mapping of the vertices V to points in the Euclidean plane $i \mapsto p_i \in \mathcal{P}$. The edges $ij \in E$ are mapped to straight line segments $\overline{p_i p_j}$. The embedding $G(\mathcal{P})$ is *planar* if distinct endpoints of edges are mapped to distinct points and edges are mapped to disjoint line segments, except when the edges are incident (in which case, their corresponding segments are allowed to have only one point in common): $\overline{p_i p_j} \cap \overline{p_k p_l} = \emptyset$, for any pair of non-adjacent edges $ij, kl \in E, i, j \notin \{k, l\}$, and $\overline{p_i p_j} \cap \overline{p_i p_k} = \{p_i\}$, for $j \neq k$. A graph G is *planar* if it has a planar embedding.

Pointed Graph Embedding. A set of vectors in $\mathcal{R}^2$ (with a common origin) is *pointed* if it is strictly contained in a half-plane, and *non-pointed* otherwise. Equivalently, some consecutive pair of vectors (in the circular ccw order around the common vertex) spans a *reflex* angle (larger than π). See Fig. 4. Algebraically, this is expressed as the non-existence of a linear combination with all positive coefficients, summing them to zero. A *pointed graph* is an embedded graph such that the edge vectors around each vertex are a pointed set. See Fig. 5.

Simple Polygon. A simple polygon is a *planar embedding* of a cycle graph. In this case, there is a well-defined and connected interior and exterior of the polygon. We will assume that the

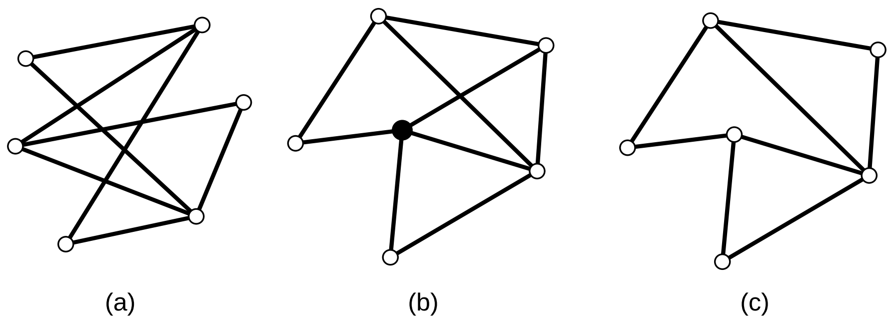

Figure 5: (a) A pointed graph embedding. (b) A non-pointed graph, due to the black non-pointed vertex. (c) A pointed and planar graph.

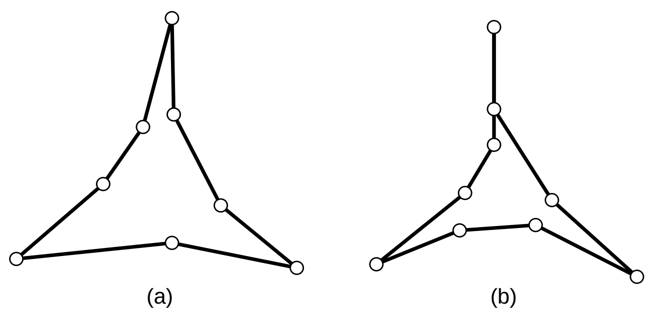

Figure 6: (a) A pseudo-triangle. (b) A semi-simple pseudo-triangle.

labelling of the vertices as $\{1, \cdots, n\}$ is in ccw order, i.e. such that the interior lies to the left when the boundary of the polygon is traversed in increasing order of its labels. *We will work only with simple polygons having no angle equal to* π*, which are therefore* **pointed**. The case when a vertex i is *aligned* (incident to an angle of π) is reduced to the pointed case by the operation of *freezing* the aligned vertex: i is eliminated and its two incident edges $(i-1)i$ and $i(i+1)$ are replaced by a single edge joining the vertices $i-1$ and $i+1$. Throughout, index arithmetic is done mod n in the set $[n] := \{1, \cdots, n\}$.

Pseudo-triangle. A *pseudo-triangle* is a simple (pointed) polygon with exactly three convex vertices, called *corners*. The three corners are on the convex hull of the pseudo-triangle and are joined by three inward convex polygonal chains. In particular, a triangle is a pseudo-triangle. See Fig. 6(a). We also introduce *semi-simple pseudo-triangles* as a special case which allows for some degeneracies: some of the inner convex angles (the corners) may be zero, but none of the inner reflex angles is allowed to be π or 2π. Moreover, we do not allow the two overlapping edges of a zero-angle corner to completely coincide: their other endpoints must be different. See

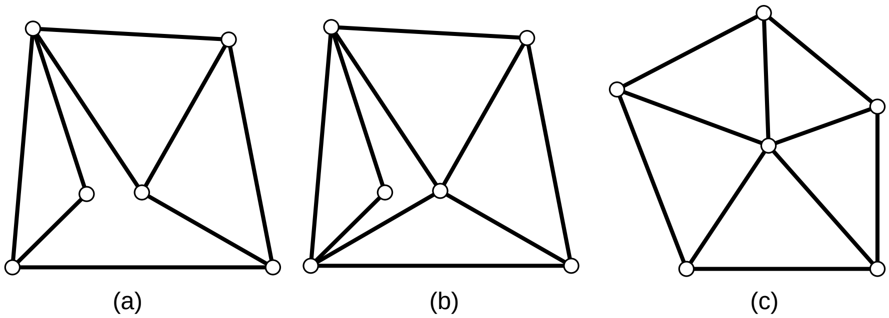

Figure 7: (a) A minimum, pointed pseudo-triangulation. (b) A non-minimum non-pointed pseudo-triangulation which contains a minimum one. (c) A non-minimum non-pointed pseudo-triangulation which does not contain a minimum pseudo-triangulation.

Fig. 6(b).

Pseudo-Triangulation. A pseudo-triangulation is a planar graph embedding whose outer face is convex and all interior faces are pseudo-triangles. A *minimum pseudo-triangulation* has the least number of edges among all pseudo-triangulations on the same point set. A *pointed pseudo-triangulation* is pointed, as an embedded graph. These two definitions will turn out to be equivalent (Theorem 2.3). See Fig. 7.

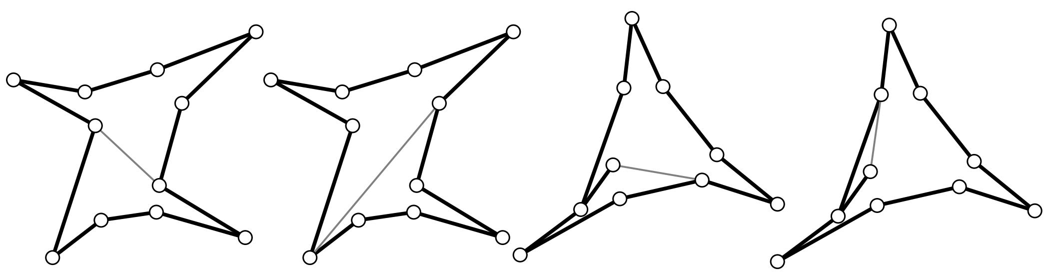

Figure 8: (a) A pseudo-quadrilateral and its two possible subdivisions into two pseudo-triangles, illustrating a diagonal *flip*. (b) A degenerate pseudo-quadrilateral and the associated flip.

Diagonal Flips in Pseudo-Quadrilaterals. More generally, if we focus on the convex vertices of a simple polygon (and call them, for consistency, *corners*) and on the inner convex chains between them, we may refer to the polygon as being a *pseudo-k-gon* if it has exactly k corners. Fig. 8 shows examples of pseudo-quadrilaterals ($k = 4$).

A *tangent* to a convex chain is a line segment with one endpoint on the chain, and whose supporting line contains the chain on one side (doesn't cut through it). A *bitangent* to two convex chains is a line segment whose endpoints are each tangent to one of the chains. Two pseudo-triangles sharing an edge are merged into a pseudo-quadrilateral when the common edge is removed. If the removed edge was incident to a vertex of degree two, the resulting face will have a *dangling* edge (incident to a vertex of degree one). We still can treat it as if it was a sort of degenerate pseudo-quadrilateral: indeed, seen from inside the face, it has exactly four corners. The dangling edge is *doubled* (traversed twice) as we walk around the boundary of the face. Notice that in this case the pseudo-quadrilateral face has an interior angle equal to 2π (at the degree-one vertex): this is an acceptable special situation. In all cases, a different diagonal can be added to produce another partitioning of the pseudo-quadrilateral into two pseudo-triangles. See Fig. 8. The following simple lemma shows that this is always possible.

Lemma 2.1 *A pseudo-quadrilateral can be subdivided by bitangents into two pseudo-triangles in exactly two ways.*

Proof: If we label in ccw order the four corners (convex vertices) as $1, 2, 3$ and 4, there are exactly two geodesic paths inside the polygon joining opposite pairs of these corners (1 and 3, resp. 2 and 4). Each geodesic follows the boundaries of the convex chains except for one line segment, which is a bitangent. □

More rigorous proofs of this and other similar simple properties needed later in the paper can be done using facts about point sets (and graphs embedded on them) that derive from the oriented matroid nature of a point set. These are the subject of the future companion paper [74].

The operation of replacing one of these two bitangents by the other is called a (diagonal) *flip* in a pseudo-quadrilateral. More generally, if any interior (non convex hull) diagonal is removed from a pointed pseudo-triangulation, it always induces a pseudo-quadrilateral and a unique flip.

2.2 Definitions: Laman Graphs and Combinatorial Rigidity

The definitions and results in this section are well known in Rigidity Theory. They are included here for a self-contained presentation of the proofs given in the next section.

Laman graphs. A graph G with n vertices and m edges is a *Laman graph* if $m = 2n - 3$ and every subset of k vertices spans at most $2k - 3$ edges. This is called the definition *by counts* and is one of the many equivalent ways in which Laman graphs can be defined.

Laman graphs are *the* fundamental objects in 2-dimensional Rigidity Theory. Also known as *isostatic* or *generically minimally rigid graphs*, they characterize combinatorially the property that a graph, embedded on a generic set of points in the plane, is infinitesimally rigid (with respect to the induced edge lengths). See Section 3.1 for rigidity theoretic definitions, and [51, 38, 82].

Henneberg constructions for Laman graphs. A Laman graph on n vertices has an inductive construction as follows (see [42, 78]). Start with an edge for $n = 2$. At each step, add a new vertex in one of the following two ways:

- **Henneberg I** (vertex addition): the new vertex is connected via two new edges to two old vertices.

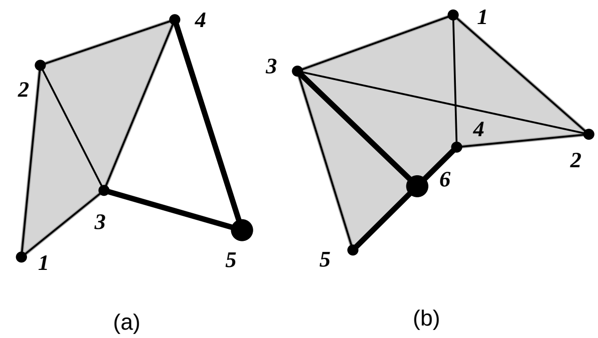

Figure 9: Illustration of the two types of steps in a Henneberg sequence, with vertices labelled in the construction order. The shaded part is the old graph, to which the black vertex is added. (a) Henneberg I for vertex 5, connected to old vertices 3 and 4. (b) Henneberg II for vertex 6, connected to old vertices 3, 4 and 5.

- **Henneberg II** (edge splitting): a new vertex is added on some edge (thus splitting the edge into two new edges) and then connected to a third vertex. Equivalently, this can be seen as removing an edge, then adding a new vertex connected to its two endpoints and to some other vertex.

See Figure 9, where we show drawings with crossing edges, to emphasize that the Henneberg constructions work for *general*, not necessarily *planar* Laman graphs.

The following result is stated by Henneberg [42], and proven by Tay and Whiteley [78].

Lemma 2.2 *A graph is Laman if and only if it has a Henneberg construction.*

In the next section, we will use a similar inductive proof technique to obtain a related inductive construction for pointed pseudo-triangulations (also called, for these historical reasons, a *Henneberg construction*).

2.3 Pointed Pseudo-Triangulations are Laman Graphs

The following theorem exhibits the combinatorial properties of pointed pseudo-triangulations which imply useful rigidity theoretic consequences: they are Laman graphs, and hence generically rigid.

Theorem 2.3 (Characterization of pointed pseudo-triangulations)
Let $G = (V, E)$ be a graph embedded on the set $\mathcal{P} = \{p_1, \cdots, p_n\}$ of points. The following properties are equivalent.

1. **(Minimum Pseudo-Triangulation)** *G is a minimum pseudo-triangulation of $\mathcal{P}$.*

2. **(Pointed Pseudo-Triangulation)** *G is a pointed pseudo-triangulation of $\mathcal{P}$.*

3. **($2n-3$ Pseudo-Triangulation)** *G is a pseudo-triangulation of $\mathcal{P}$ with $2n-3$ edges (and, equivalently, with $n-2$ faces).*

4. **($2n-3$ Planar and Pointed)** *The set of edges E is planar (non-crossing), pointed and has $2n-3$ elements.*

5. **(Maximal Planar and Pointed)** *The edges E of G form a pointed and planar set of segments, and E is maximal (by inclusion) with this property.*

6. **(Planar Pointed Henneberg-type Construction)** *G can be constructed inductively as follows. Start with a triangle. At each iteration, add a new vertex in one of the faces of the already constructed embedded graph (which will be a pointed pseudo-triangulation). Connect in one of the two ways (see Fig. 10):*

   (a) Type 1: (degree 2 vertex) *Join the vertex by two tangents to the already constructed part. If the new vertex is outside the convex hull, the two tangents are uniquely defined. If it is inside an internal pseudo-triangular face, there are three different ways of adding two tangents to the three inner convex chains of the face, out of which two are chosen.*

   (b) Type 2: (degree 3 vertex) *Add two tangents as before. Then choose an edge on the boundary chain between the two tangent points and remove it. This creates a pseudo-4-gon. Re-pseudo-triangulate by adding the unique bitangent different from the one just removed (flip). This edge will be incident to the newly added vertex.*

Moreover, if any of the above conditions is satisfied, then the subgraph induced on any subset of k vertices has at most $2k-3$ edges (the hereditary property*). Therefore G is a* **Laman graph**.

For the proof, we will need some basic definitions and facts, most of which are folklore in Computational Geometry. Therefore we present them in a sketchy manner.

Facts

1. Given a convex hull and an exterior vertex, there exist exactly two tangents from the point to the hull.

2. Given a pseudo-triangle and a vertex interior to it, there exist exactly 3 tangents, all interior to the pseudo-triangle, from the point to the three inner convex chains.

3. Given a pseudo-quadrilateral, there exist exactly two ways of adding a bitangent between two inner convex chains. Each one induces a partitioning of the pseudo-quadrilateral into two pseudo-triangles. This is Lemma 2.1.

4. *(Flips in pointed pseudo-triangulations)* Any internal edge in a pointed pseudo-triangulation can be flipped: the edge is deleted and this joins two pseudo-triangles into a pseudo-quadrilateral, which can then be re-pseudo-triangulated in exactly one other way. See Lemma 2.1 and Fig. 6(c).

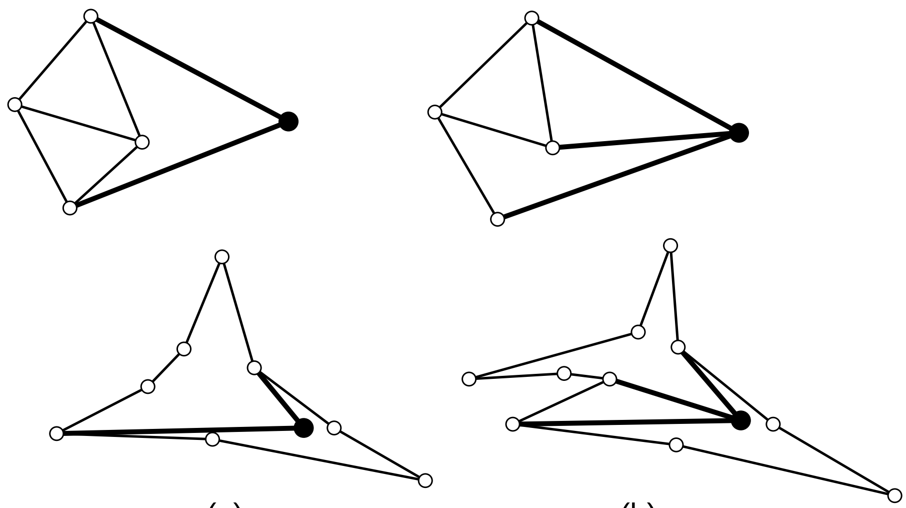

Figure 10: Henneberg steps for pointed pseudo-triangulations. (a) type 1 and (b) type 2. The newly added vertex is black. Top row, when the new vertex is added on the outside face. Bottom row, when it is added inside a pseudo-triangular face. For the type 2 step, a diagonal was previously removed.

We are now ready for the proof of Theorem 2.3.

Proof:

$1 \Leftrightarrow 2 \Leftrightarrow 3$

Let G be a pseudo-triangulation, and let $v = n$, e and f denote its number of vertices, edges and interior faces. Euler's formula gives $v - e + f = 1$. Denote by d_i the degree of vertex i and by c_i the number of convex angles (corners) incident to it. If i is a pointed vertex, $c_i = d_i - 1$, otherwise $c_i = d_i$. Let A be the set of pointed vertices, B the set of non-pointed ("bad") vertices of G and let $b = |B|$ be the total number of non-pointed vertices in G. Then

$$2e = \sum_{i=1}^{n} d_i = \sum_{i \in A}(c_i + 1) + \sum_{i \in B} c_i = \sum_{i=1}^{n} c_i + (n - b)$$

But $\sum_{i=1}^{n} c_i \geq 3f$ since each face has at least three corners (exactly three if it is a pseudo-triangle). Therefore $2e \geq 3f + (n - b)$. Plugging this into Euler's formula yields $e \leq 2n - 3 + b$. When each face is a pseudo-triangle, this holds with equality $e = 2n - 3 + b$ and is minimum iff $e = 2n - 3$ iff $b = 0$, i.e. when G is pointed.

$3 \Leftrightarrow 4$

One direction is obvious. For the other direction, let G be planar, pointed and with $2n - 3$ edges and hence, by Euler's formula, with $n - 2$ faces. A similar argument as above shows that the total number of corners in G is $c = \sum_{i=1}^{n}(d_i - 1) = 2e - n = 3(n - 2)$. Since each interior face in a planar graph has at least 3 inner convex angles, if follows that each must have *exactly* 3 convex angles, hence it is a pseudo-triangle.

$2 \Leftrightarrow 5$

If G is a pointed pseudo-triangulation, each new diagonal added to it will violate either planarity or pointedness of at least one vertex, because there are no common bitangents to the inner convex chains of a pseudo-triangle. Hence G is maximal. To prove the other direction, let G be a planar and pointed graph to which no edge can be added without violating one or the other of these two properties. Obviously G must contain all the convex hull edges, because otherwise they can be added without causing any violation of planarity or pointedness. Assume now that at least one face is not a pseudo-triangle. It is straightforward to show that each face must be topologically a disk, and hence a pseudo-k-gon for some $k \geq 3$. This is because otherwise a "face" would be a polygon with holes, to which we could always add another diagonal (a tangent, say, from a point on the outer boundary to an inner hole) while keeping the graph still pointed and planar. But if a face is a pseudo-k-gon, for some $k > 3$, then we can always add a new bitangent, contradicting maximality.

To prove that G is a *Laman graph*, we use the fact that planarity (non-crossing) and pointedness are hereditary properties: if they are satisfied on G, they are satisfied on any subset of G. We must prove that every subset of k vertices spans at most $2k - 3$ edges. Indeed, if that was not the case for some induced subgraph, then the proof of the equivalences $1 \Leftrightarrow 2 \Leftrightarrow 3$ would imply that it would violate either planarity or pointedness.

$4\ (\Leftrightarrow 3) \Leftrightarrow 6$

Let G be planar, pointed and with $2n - 3$ edges. We will also use the derived fact that any subset of k vertices spans at most $2k - 3$ edges. We work out the Henneberg construction in reverse. Because the number of edges is $2n - 3$, there must exist at least one vertex of degree strictly less than 4. This cannot be 1 or 0, because then the Laman count property would be violated on the subset of the other $n - 1$ vertices. If there exists a vertex of degree 2, its two adjacent edges are tangent to the face obtained by removing them, because of pointedness. For a vertex of degree 3, the two extreme edges adjacent to it (i.e. those adjacent to the reflex angle of the pointed vertex) must be tangents to the object obtained by removing them (again, because of the property of pointedness). The face obtained by removing the third edge (before removing the two extreme ones) is a pseudo-quadrilateral (this follows from the other equivalences), and the addition of the second bitangent recreates a pseudo-triangle. Removing the vertex, the remaining graph satisfies the same properties (because of the *hereditary property*). Hence the argument continues.

The other direction is straightforward: at each step in a Henneberg-type construction, the number of vertices increases by 1, the number of edges by 2 and the graph remains planar and pointed. □

In the rest of this paper we will only be concerned with *pointed* pseudo-triangulations, even if we may omit the word *pointed* to simply the terminology.

Corollary 2.4 *The underlying graphs of pointed pseudo-triangulations are generically minimally rigid graphs.*

Not all generically minimally rigid graphs have embeddings as pseudo-triangulations, because not all are planar graphs. The smallest example is $K_{3,3}$. However, a recent result [41] shows that all minimally rigid planar graphs can be embedded as pseudo-triangulations.

Laman graphs have further nice combinatorial properties, which are inherited by pseudo-triangulations. For completeness, we give them here (but will not make any further use of them.)

Theorem 2.5 (Tree decompositions of pseudo-triangulations)

1. **Two spanning tree decomposition.** *If any edge is doubled, a pseudo-triangulation can be decomposed into two spanning trees.*

2. **2T3 decomposition.** *A pseudo-triangulation can be decomposed into three disjoint trees, so that each vertex is adjacent to exactly two of them.*

Proof: Part 1 is a direct consequence of the result of Lovasz and Yemini [53] for generically minimally rigid graphs (Laman graphs). Part 2 is a consequence of Crapo's theorem [32]. □

3 Rigidity of Pseudo-Triangulations

We have shown that (pointed) pseudo-triangulations are Laman graphs, which are rigid in almost all embeddings ("generically", see [36]). We prove now that pseudo-triangulations have an even more special property: they are *always* infinitesimally rigid.

We start by defining the needed concepts from Rigidity Theory. Although these results are not new, the presentation is. We emphasize the role of the number of edges on the infinitesimal rigidity and self-stress properties of a graph embedding. This perspective, based on rank and orthogonality relations from elementary linear algebra, is then used in the proof of infinitesimal rigidity for pointed pseudo-triangulations.

We have chosen the sequence of facts in such a way that it is clear what properties hold for general frameworks sharing some combinatorial properties with pseudo-triangulations (such as edge counts or planarity). This way, it is easy to extract what is specific to pointed pseudo-triangulations, and generalize to other situations, when needed. Indeed, in the companion paper [74] we generalize some of these properties to the oriented matroid setting.

3.1 Frameworks and Rigidity

Convention. From now on, we will use interchangeably $\mathcal{P} = (p_1, \cdots, p_n) \in (\mathcal{R}^2)^n$, with $p_i = (x_i, y_i)$ or its *flattened* version $p = (x_1, y_1, \cdots, x_n, y_n) \in \mathcal{R}^{2n}$ to stand for elements of either $(\mathcal{R}^2)^n$ or $\mathcal{R}^{2n}$.

Frameworks and Configuration Spaces. A *bar-and-joint framework* (or a *fixed edge-length framework*, or shortly, a *framework*) (G, L) is a graph $G = (V, E)$, $|V| = n$, together with a set of strictly positive weights $L = \{l_e | e \in E\}$, $(l_e \in \mathcal{R}, l_e > 0)$ meant to be used as *edge lengths*. A *realization* $G(\mathcal{P})$ of (G, L) on a set of points $\mathcal{P} = \{p_1, \cdots, p_n\} \subset \mathcal{R}^2$ is a mapping $i \mapsto p_i$ of vertices to points and edges to line segments (i.e. an embedding of G in the plane) so that the length $||p_i - p_j||$ of the segment $\overline{p_i p_j}$ corresponding to the edge $e = ij$ is equal to l_e. Since each realization comes together with a whole set of other realizations obtained from it by translations and rotations, it is convenient to factor out the rigid motions of the whole plane. This can be done, for instance, by *pinning down* an edge, e.g. if $e = 12 \in E$ we will set $x_1 = y_1 = 0,\ x_2 = l_{12},\ y_2 = 0$. The set of all possible realizations of a framework, with rigid motions factored out (e.g. via an arbitrary pinned down edge) is called its *configuration space*. Its topological and differential properties do not depend on the choice of the pinned down element.

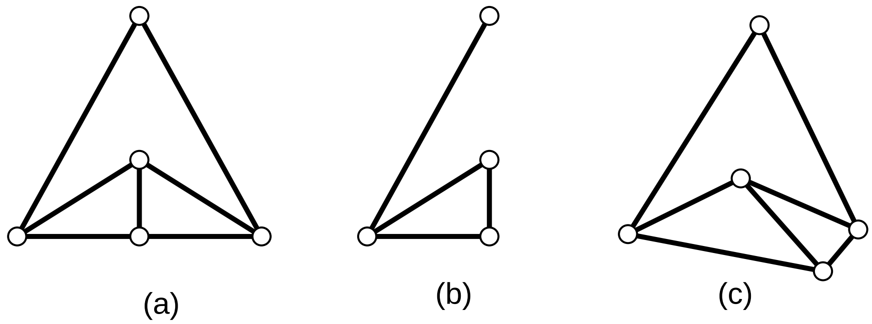

Figure 11: A framework with two distinct embeddings, one of which (a) is rigid and the other (b) is flexible, with one degree-of-freedom. Notice that the symmetry of (a) allows in (b) for some pairs of vertices to be mapped to identical points, and some pairs of edges to be drawn one on top of the other. This allows in (b) for two identical edges, embedded on top of each other, to move with one degree of freedom, as a single dangling edge does. (c) The same graph for other edge lengths has only rigid embeddings.

In this paper we are *not* concerned with questions of realizability, as we will always start with a given embedding, from which the edge lengths are actually computed if needed. Therefore the actual values of the edge lengths are not relevant to our discussion: the configuration space will always be non-empty. To simplify the terminology, from now on we will usually refer to a realization $G(\mathcal{P})$ as a *framework*, and when we actually mean (G, L), we will say it explicitly.

Rigid and flexible frameworks. The configuration space of a (fixed edge-length) framework (G, L) may be disconnected and in general may have a complicated topological structure. The *dimension* of the component of the configuration space in which a framework (realization) $G(\mathcal{P})$ resides is called its *number of degrees of freedom*. If that component is a single, isolated point, the framework $G(\mathcal{P})$ is called *rigid*, otherwise it is *flexible*. We emphasize that rigidity and flexibility refer only to the configuration space component to which *a certain realization* $G(\mathcal{P})$ belongs. Indeed, there exist frameworks (G, L) for which different components may have different dimensions, see Fig. 11.

Minimal Rigidity. A rigid framework is *minimal* if the removal of any edge makes it flexible. Otherwise we say that it is *overbraced*. The example in Fig. 11(c) is minimally rigid. Any extra edge overbraces it.

Generically, the minimum number of bars needed to induce a rigid framework is $m = 2n - 3$. Indeed, the configuration space is given by m quadratic equations in $2n$ variables $\{x_1, y_1, \cdots, x_n, y_n\}$, so $2n$ independent equations are needed to get a discrete set of solutions. We need three equations to eliminate the rigid motions, and the other $2n-3$ must be independent bar-lengths equations of the form $(x_i - x_j)^2 + (y_i - y_j)^2 = l_{ij}^2$. Therefore generically we expect that

$2n-3$ independent edges and edge-length values will typically produce a zero-dimensional configuration space; more than $2n-3$ edges will typically produce an empty configuration space, except for very special, dependent values for some of the edge lengths; and less than $2n-3$ edges will induce flexible frameworks, with a number of degrees of freedom equal to $2n-3-m$. The main difficulty in analyzing rigidity properties of frameworks is due to the occurrence of non-generic situations.

Our main rigidity theoretic results on pseudo-triangulations and pseudo-triangulation mechanisms (sections 3.2 and 4.5) may be interpreted as saying that planarity and pointedness (as in *pointed pseudo-triangulations* and subsets of their edge set) induce *generic frameworks*: their configuration spaces have the generic dimension for the corresponding number of edges.

Infinitesimal Rigidity. An equivalent way of saying that a framework $G(\mathcal{P})$ is *flexible* is that its vertices can be moved *continuously* while preserving the lengths of the edges, such that the motion is not a trivial rigid motion (translation and rotation.) The non-trivial motion is called a *flex* or *reconfiguration* of the framework. It is a continuous curve (one dimensional trajectory) $p(t) = \{p_1(t), \cdots, p_n(t)\}$ in the configuration space going through the point $\mathcal{P}$ giving the framework realization, such that at each moment in time:

$$||p_i(t) - p_j(t)|| = l_{ij} \tag{1}$$

for all edges $ij \in E$. If we assume that the flex $p(t)$ has other good analytic properties (e.g. is differentiable), then taking the derivative of equation (1) we obtain the conditions:

$$\langle p_i(t) - p_j(t), p'_i(t) - p'_j(t)\rangle = 0, \ \forall ij \in E \tag{2}$$

Here $\langle,\rangle$ is the dot product. The first-order derivative $p'(t)$ of the motion $p(t)$, computed at a given moment in time t, is called an *infinitesimal* or *instantaneous motion at time t*. These considerations motivate the following definition.

For a given framework $G(p)$, an *infinitesimal motion* $v = \{v_1, \cdots, v_n\}$ is an assignment of a velocity v_i to each point p_i such that the lengths of the edges are preserved:

$$\langle p_i(t) - p_j(t), v_i(t) - v_j(t)\rangle = 0, \forall ij \in E \tag{3}$$

A framework is *infinitesimally* (or *first-order*) *flexible* if it has an infinitesimal motion, otherwise it is *infinitesimally* (or first-order) *rigid*.

Infinitesimally rigid frameworks are rigid, but the opposite statement may not always be true. See Fig. 12 (a). To capture the relationship between rigidity and infinitesimal rigidity, flexibility and infinitesimal flexibility of $G(\mathcal{P})$, one must investigate the *differential* properties of its configuration space in a neighborhood of the particular point (realization) $G(\mathcal{P})$.

The Edge Map. Given a graph G with n vertices and $|E| = m$ edges, the *edge map* or *rigidity map* $f_G : \mathcal{R}^{2n} \mapsto \mathcal{R}^m$ associates to a point set $p = (p_1, \cdots, p_n) \subset \mathcal{R}^{2n}$ the vector of the squared lengths of the edges $ij \in E$, in some fixed, predefined order:

$$f_G(p_1, \cdots, p_n) = (||p_i - p_j||^2)_{ij \in E} \tag{4}$$

We say that a point $p \in \mathcal{R}^{2n}$ is a *regular* or *generic point* of f_G if the rank of the differential of f_G is maximum at p:

$$\text{rank } df_G(p) = \max\{\text{rank } df_G(q) : q \in \mathcal{R}^{2n}\} \tag{5}$$

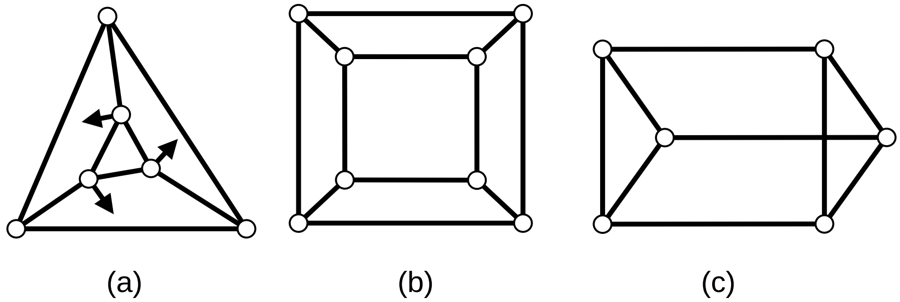

Figure 12: (a) A rigid framework which is not infinitesimally rigid. (b) An infinitesimally flexible framework which is rigid. (c) A generically rigid graph in a flexible embedding.

Otherwise, p is a *singular point.*

If $m \geq 2n-3$, we expect the rank to be $2n-3$ for regular (generic) points. For $m < 2n-3$, the generic rank is m. The rank drops at singular points. The following closer analysis relates it to the infinitesimal rigidity properties of the underlying framework $G(\mathcal{P})$.

The Rigidity Matrix. The rigidity matrix $M_G(\mathcal{P})$ (shortly M) associated to an embedded framework $G(\mathcal{P})$ is the Jacobian matrix $df_G(p)$ of the edge map at the point $p \in \mathcal{R}^{2n}$ corresponding to the embedding $\mathcal{P} = (p_1, \cdots, p_n)$. It has a row for each edge $ij \in E$ and 2 columns for each vertex. The row indexed by the edge $ij \in E$ has 0 entries everywhere, except in the ith and jth group of 2 columns, where the entries are $p_i - p_j$, resp. $p_j - p_i$.

$$\begin{array}{cc} & \begin{array}{ccccccccc} 1 & \cdots & i & & \cdots & & j & \cdots & n \end{array} \\ ij & \begin{pmatrix} & & & & \cdots & & & \\ 0 & \cdots & p_i - p_j & & \cdots & & p_j - p_i & \cdots & 0 \\ & & & & \cdots & & & \end{pmatrix} \end{array}$$

The linear subspace of *infinitesimal motions* $v \in \mathcal{R}^{2n}$ of the framework is the kernel (null space) of M, $kerM = \{v \in \mathcal{R}^{2n} \mid Mv = 0\} \subset \mathcal{R}^{2n}$. A vector of velocities $(v_1, \cdots, v_n)$ will be interchangeably written as a flattened vector $v \in \mathcal{R}^{2n}$. Any such v (not necessarily an infinitesimal motion) yields a set of values associated with the edges of E, $d_{ij} = \langle p_i - p_j, v_i - v_j \rangle$, $\forall ij \in E$. This set is a linear subspace, the image space of M, $ImM = \{d \mid d = Mv\} \subset \mathcal{R}^{|E|}$. The sign of d_{ij} has a physical interpretation: if positive, v is an infinitesimal motion which *expands* the edge, i.e. increases the distance between the vertices i and j. The other two important linear subspaces associated with M are the null space $kerM^T$ of the transposed matrix M^T and its image space $Im\ M^T$. They also have physical interpretations, as follows.

Self stresses. A *self-stress* on a framework $G(\mathcal{P})$ is an assignment of scalars w_{ij} to edges $ij \in E$ such that the *forces* along the edges around each vertex, scaled by the self-stresses, are *in equilibrium*: $\forall i \in V$, $\sum_{ij \in E} w_{ij}(p_i - p_j) = 0$. Self stresses form a linear subspace $kerM^T = \{w = (w_{ij})_{ij \in E} \mid M^T w = 0\} \subset \mathcal{R}^{|E|}$, which is orthogonal to ImM:

$$ker M^T \perp Im M \tag{6}$$

A *non-trivial self-stress* is one with at least one non-zero component $w_{ij} \neq 0$ on some edge $ij \in E$.

A vector of scalar values associated to the edges $w = (w_{ij})_{ij \in E}$, not necessarily a self-stress, induces a vector of velocities associated with the vertices, $\{v \ |v = M^T w\} \subset \mathcal{R}^{2n}$. Each v is componentwise a sum of the incident edge vectors, scaled by the w_{ij} factors: $v_i = \sum_{ij \in E} w_{ij}(p_i - p_j)$. They form a linear subspace $Im\ M^T = \{v \mid v = M^T w, w \in \mathcal{R}^{|E|}\} \subset \mathcal{R}^{2n}$, which is orthogonal on the subspace of infinitesimal motions $ker M$:

$$ker M \perp Im M^T \tag{7}$$

The rank relations. The following well-known relations hold between the dimensions of the four linear spaces associated with the $2n \times m$ matrix M:

$$\begin{aligned} \dim ker M + \dim Im M^T &= 2n \\ \dim ker M^T + \dim Im M &= m \\ \dim Im M &= \dim Im M^T \end{aligned} \tag{8}$$

Minimal Infinitesimal Rigidity. The kernel of the rigidity matrix *ker M* always contains the 3-dimensional linear subspace of trivial infinitesimal motions, translations and rotations.

The following Lemma is a simple consequence of the rank relations, and summarizes the main consequence of the lack of self-stress for graphs with $2n - 3$ edges. This is the minimum number of edges that allows for infinitesimal rigidity.

Lemma 3.1 *If $m = 2n - 3$, dim ker $M = 3$ iff dim ker $M^T = 0$. In this case, the framework $G(\mathcal{P})$ is infinitesimally rigid iff it has no self-stress.*

Proof: Using the rank relations (8), we infer that dim *ker* $M = 3$ iff *dim Im* $M^T = 2n - 3$, but *Im* $M^T =$ dim *Im* M, and further equal to $m -$ dim *ker* M^T. Since $m = 2n - 3$, this means dim *ker* $M^T = 0$. □

When the number of edges drops below $2n-3$, non-trivial infinitesimal motions are unavoidable.

Lemma 3.2 *If $m \leq 2n - 4$ then $G(\mathcal{P})$ always has a non-trivial infinitesimal motion.*

When the number of edges exceeds $2n - 3$, self-stress is unavoidable.

Lemma 3.3 *If $m \geq 2n - 2$ then $G(\mathcal{P})$ always has a non-trivial self-stress.*

Generic Minimal Rigidity. We focus now on graphs with exactly $2n-3$ edges. If such a graph has an infinitesimally rigid embedding $G(\mathcal{P})$, then G is not only rigid at $\mathcal{P}$, but also on any point set $\mathcal{P}'$ belonging to a small neighborhood of $\mathcal{P}$. Because of the edge count, this is equivalent to $G(\mathcal{P}')$ being self-stress-free on the same neighborhood. Moreover, G satisfies the property *minimally*: the removal of any edge turns it into a graph which is no longer infinitesimally rigid.

We say that $G(\mathcal{P})$ is a *generic rigid embedding* of a graph G with $2n-3$ edges if it is infinitesimally rigid (or, equivalently, self-stress-free) on an open neighborhood of $\mathcal{P}$. Such a point $\mathcal{P}$ is a regular point of the edge map for G.

A graph G with $2n-3$ edges is called a *generically (minimally) rigid graph* if it has a generic infinitesimally rigid embedding.

Theorem 3.4 (Laman's Theorem [51]) *G is generically minimally rigid iff G is a Laman graph.*

Self-stresses in Planar Frameworks and Maxwell's Theorem. The following beautiful connection between the existence of non-trivial self-stresses and $3d$-liftings of *planar* graphs goes back to Maxwell [55, 56]. Some topological details, as well as the extension to signed stresses, have been fixed later (see [33]). We present here only the simplified formulation that will be used later in our proofs.

A $3d$-lifting of a planar (non-crossing) framework $G(\mathcal{P})$ is an assignment of a *height* $z_i \in \mathcal{R}$ to each vertex i such that when the graph G is *lifted* in $3d$ on the points $(p_i, z_i) := (x_i, y_i, z_i)$, the vertices of each face of G are coplanar. We may assume without loss of generality that the outer face of G stays in the original plane ($z_i = 0$ for vertex p_i on the outer face), and the other vertices are lifted. A lifting is trivial if all z_i's are 0. We are interested only in non-trivial liftings.

Notice that such a lifting (in the particular case of a non-crossing framework embedding presented here) is a polyhedral terrain. The outer (unbounded) face is assumed to be flat at level $z = 0$.

Two faces adjacent along an edge ij span a dihedral angle which, when viewed from below, may be convex, reflex or equal to π. Call the edge a *mountain* if this angle is convex, *valley* if reflex and *flat* if equal to π.

Not all planar graph frameworks admit non-trivial $3d$-liftings. Maxwell's theorem characterizes those which do.

Theorem 3.5 (Maxwell [55, 56], Crapo and Whiteley [33]) *A planar (non-crossing) framework has a non-trivial $3d$ lifting if and only if it has a non-trivial self-stress. Moreover, the correspondence between self-stresses and liftings maps the edges with positive self-stress to mountain edges, those with negative stress to valley edges and those with zero stress to flat edges.*

Bow's construction and Maxwell liftings for non-planar graphs. Maxwell's theorem can be extended to non-planar frameworks by applying a simple trick, called *Bow's construction*, see [57]. Let $G(\mathcal{P})$ be a framework where some of the edges have proper crossings. We will introduce a new vertex for each crossing of two edges. Then we "split" each of the two crossing edges into two new edges, each one having the new vertex as an endpoint. It can be easily proven that the resulting framework inherits the self-stress properties of the original framework, including the signs of the split edges. See [57, 33], and Figure 13(b) and (c).

3.2 Pointed Pseudo-Triangulations are Infinitesimally Rigid

We prove now the main rigidity theoretic property of pointed pseudo-triangulations: they are always infinitesimally rigid. The main tool used in the proof is Maxwell's theorem. The same

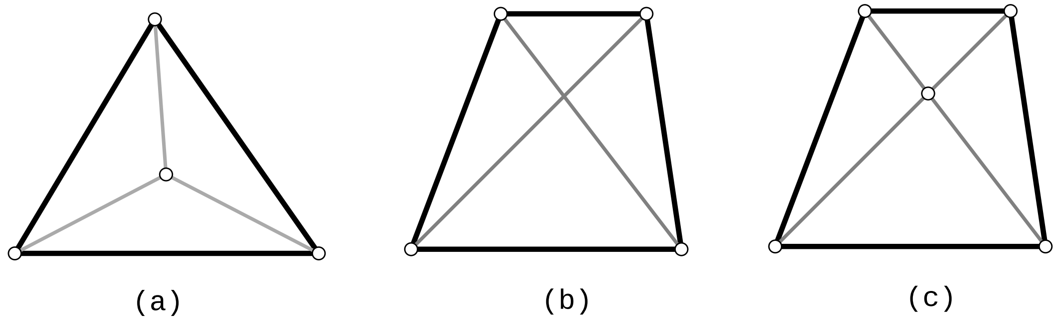

Figure 13: (a) The smallest example of a planar self-stressed graph with no collinear vertices. It lifts to a tetrahedron. The thick, convex hull edges support a positive self-stress, and are valley edges in the lifting. The internal, thin grey edges, support a negative self-stress and lift to mountain edges. (b) The same graph, in a non-planar embedding, and the corresponding self-stresse. (c) Bow's construction applied to the embedding in (b): a new vertex is added at the cross-point of the two internal diagonals. The signs of the self-stresses is inherited from the original graph. The self-stressed *planar* graph in (c) lifts to a pyramid.

tool will be used in the next section to prove that pseudo-triangulations with a convex hull edge removed are expansive mechanisms.

Theorem 3.6 *Pointed pseudo-triangulations are always infinitesimally rigid.*

Let us reformulate the theorem, to emphasize all the properties of pointed pseudo-triangulations that the proof will exhibit.

Theorem 3.7 (Infinitesimal Rigidity of Pointed Pseudo-Triangulations) *A pointed pseudo-triangulation $G(\mathcal{P})$ is a planar Laman graph embedding which is always infinitesimally rigid and self-stress-free.*

In the proof, we will make use of the following simple fact.

Lemma 3.8 *For any lifting into $3d$ at non-constant height of a flat polygon, all vertices of maximum height are corners.*

Proof: All vertices on the convex hull of a polygon are corners. The maximum height is attained on the convex hull of a 3d object, which, for the flat lifting considered here, is exactly the 2d convex hull. See Fig. 14. □

Proof of Theorem 3.7

Because G is a Laman graph, and hence has $2n-3$ edges, we use Lemma 3.1 to prove infinitesimal rigidity by showing that $G(\mathcal{P})$ is self-stress-free.

For the sake of a contradiction assume that there exists a non-trivial self stress $(w_e)_{e\in E}$, $w_e \neq 0$ for some $e \in E$. Then, by Maxwell's Theorem, there exists a non-trivial $3d$-lifting which

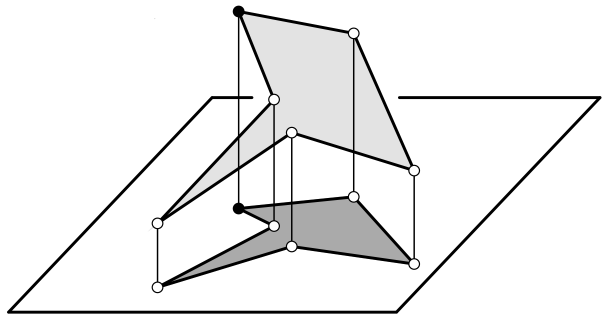

Figure 14: Illustration of Lemma 3.8: the vertex of maximum z-coordinate of the lifted face projects to a vertex on the convex hull of the flat polygon. Reflex vertices cannot lift to maxima.

is not flat on the edges with non-zero self-stress. Consider the points of local maximum height in the lifting. If the set of these points contains only vertices of the convex hull of $G(\mathcal{P})$, it means that the whole lifting lies underneath the $z = 0$ plane. We can simply reverse all the signs to make it above the plane, and assume that we have a local maximum which is not a convex hull vertex and lies above the xy-plane. This is strictly above the xy-plane, since the self-stress is not trivial. We will derive a contradiction by considering one of these local maxima.

Consider the convex hull of the maximum region and a vertex on this convex hull. By Lemma 3.8, this vertex must be a corner of all the faces incident to it, hence its projection is a corner in all incident faces. This contradicts the pointedness of the pseudo-triangulation.

Since the lifting is impossible, the graph is self-stress-free and hence infinitesimally rigid. □

Corollary 3.9 *A pointed pseudo-triangulation $G(\mathcal{P})$ is always infinitesimally rigid, and minimally so: if any edge is removed, it becomes infinitesimally flexible.*

4 Expansive 1dof mechanisms from pseudo-triangulations

In this section we turn our attention to flexible frameworks, specifically those obtained by removing an edge from a Laman graph in a generic embedding. We show that we always obtain a 1dof mechanism: a framework with a one-dimensional configuration space. A pointed pseudo-triangulation with a convex hull edge removed will be called a *pseudo-triangulation mechanism* or shortly a *ppt-mechanism*.

The main theorem of this section is that ppt-mechanisms are *expansive* and smooth for a bounded open interval of their one-dimensional configuration space. This interval is characterized combinatorially by all the realizations of the underlying framework having the same combinatorial structure, more precisely, what we call the *combinatorial pseudo-triangulation* (defined in this section). These pseudo-triangulation mechanisms will be used in Section 6 as the building blocks of the expansive convexifying trajectory of our solution to the Carpenter's Rule problem.

4.1 Preliminaries: Laman graphs with a removed edge

We first investigate the combinatorial structure of graphs obtained by removing an edge from a Laman graph. We show that they have a natural decomposition into disjoint Laman subgraphs, called *rigid components*.

Laman-minus-one graph. This is a graph obtained by removing an edge from a Laman graph. It has $2n-4$ edges, and each subset of k vertices spans at most $2k-3$ edges.

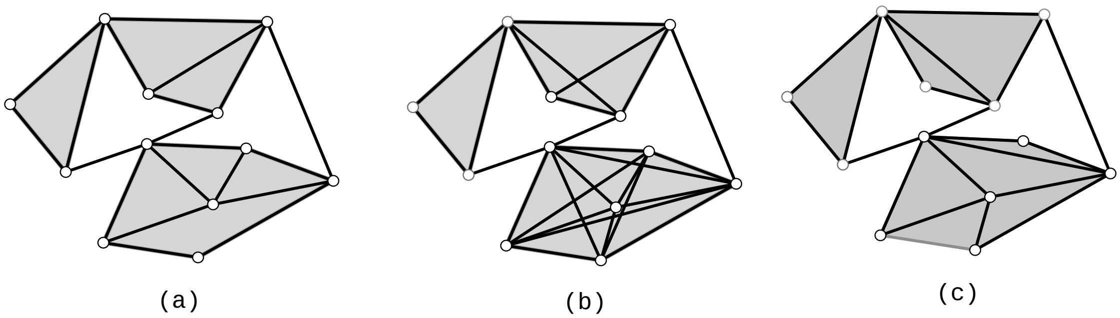

Figure 15: (a) The decomposition of a Laman-minus-one graph into rigid components, six in this case: the three big grey blocks, and the three remaining edges. (b) The collapsed Laman-minus-one graph induced by (a). (c) An equivalent Laman-minus-one graph, differing from (a) on the rigid components (the graph edges are black and thick).

Rigid components in Laman-minus-one graphs. In a Laman-minus-one graph, consider maximal subgraphs $G_R = (V_R, E_R)$ spanning exactly $|E_R| = 2|V_R| - 3$ edges. Since subsets of V_R satisfy the Laman count property, such a subgraph G_R is a Laman graph, called a *rigid component* (r-component) of G.

Lemma 4.1 *The edge set of any Laman-minus-one graph is partitioned into (disjoint) rigid components.*

Proof: A rigid component is well defined, by maximality. Two rigid components can share a vertex, but not more. Indeed, if there is no edge between the two common vertices, the union of the two r-components would violate the Laman count. Otherwise, if there is an edge, maximality would be violated, as in this case the union of the two r-components would be a larger Laman subgraph. □

See Figure 15(a) for an example.

Collapsed Laman-minus-one graphs. Intuitively, in each rigid component all the diagonals, present or missing, are rigid. Hence, although a Laman-minus-one graph will (generically) be flexible, its r-components will move rigidly. If we replace each rigid component in a Laman-minus-one graph by the complete graph, the resulting graph is called a *collapsed Laman-minus-one graph.* See Figure 15(b) for an example.

Equivalent Laman-minus-one graphs. Two Laman-minus-one graphs are called *equivalent* if they differ only in the choice of Laman subgraphs spanning certain rigid components. In other words, the collapsed graphs of two equivalent Laman-minus-one graphs are the same. See Figure 15 (a,c).

4.2 1dof Mechanisms: infinitesimal flexibility and smooth trajectories

We now investigate the rigidity-theoretic properties of Laman-minus-one graphs. Generically, they are flexible, with one-degree-of-freedom. We identify the properties of their smooth intervals, and the unpleasant role of singularities in their configuration spaces. In the next section 4.3 we will use these considerations to justify the investigation of smoothness properties for pseudo-triangulation mechanisms and their role in the convexification algorithm.

Minimal Infinitesimal 1dof Flexibility. Consider a Laman-minus-one graph: it has $2n-4$ edges. The kernel *ker M* of the rigidity matrix M always contains the 3-dimensional linear subspace of trivial infinitesimal motions, translations and rotations.

The proof of the following Lemma follows along the same lines as the proof of Lemma 3.1, and is a simple consequence of the rank relations (8). It shows the basic consequences of the lack of self-stress for such graphs.

Lemma 4.2 *If* $m = 2n-4$, $\dim \ker M = 4$ *iff* $\dim \ker M^T = 0$. *In this case, the framework* $G(\mathcal{P})$ *is infinitesimally flexible with one-degree-of-freedom iff it has no self-stress.*

In particular, for the Lemma 4.2 to be satisfied, G must be a Laman-minus-one graph and each of its r-components must be in generic (self-stress-free) embeddings (as Laman (sub)graphs). But this condition is not sufficient. All the rigid components of the Laman-minus-one graph in Figure 12 (b) are edges, which are always generic (if embedded on two distinct endpoints), but the graph itself is not in a generic embedding, since it has a self-stress.

One-degree-of-freedom mechanisms. The above considerations proved that Laman graphs with a missing edge, in generic embeddings, have a one-dimensional space of infinitesimal motions. Moreover, nearby embeddings are also generic. In fact, the following stronger property holds.

Theorem 4.3 (Main Theorem of 1dof mechanisms) *Let G be a Laman-minus-one graph, and assume that it has a generic (regular, stress-free) embedding $G(\mathcal{P})$. Then the component of its configuration space containing the given generic embedding is one-dimensional, and smooth in the neighborhood of the given generic embedding.*

Proof: The statement is a direct consequence of Proposition 2, page 282 in [12], which relies on the Inverse Function Theorem (see also [13]). Alternatively, it follows from the fundamental theorem of Ordinary Differential Equations, which in turn is a direct consequence of the Implicit Function Theorem. See Chapter 2 in [10], Corollaries 1 (existence) and 2 (uniqueness) on page 93 to the Fundamental Theorem 1 on page 89. □

A framework satisfying the conditions of the previous theorem is called a *1dof mechanism.* A simple example is the Laman graph $K_{3,3}$ with a missing edge from Fig.16(a). In the special embedding of Fig.16 (b), it is the classical Peaucellier linkage, used to convert between circular and linear motion. This embedding is special, in the sense that the edge lengths are chosen to have very specific relationships between them, while in Fig.16(a) they are as general as possible. The previous theorem does not apply to the framework in Fig. 12(b), which is *not* flexible: the component of its configuration space to which it belongs contains only this realization, which is not generic.

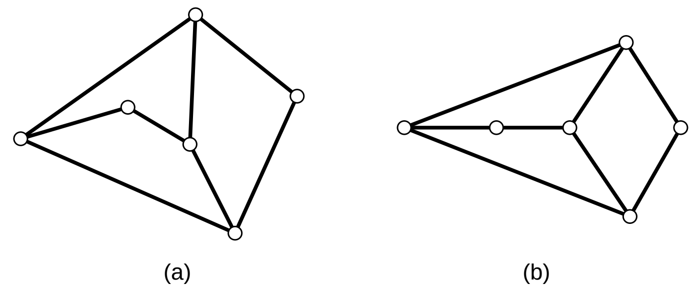

Figure 16: The classical Peaucellier linkage (b), and a general, generic embedding (a) of the underlying graph.

Given a 1dof mechanism $G(\mathcal{P})$, consider the component of its configuration space to which it belongs. We will refer to this component as the *lifetime* or *lifespan* of the mechanism $G(\mathcal{P})$, and to any configuration in the lifetime of a mechanism as a *snapshot*.

Topology of the configuration space of a 1dof mechanism. The topology of the configuration space component of a 1dof mechanism is also relevant. If there are no singular points, it is topologically a circle. But in general it may contain singularities. The simplest example is the four-bar mechanism of Fig.17. Its configuration space is smooth, unless there is a position of alignment for the four joints. This is a consequence of a general theorem of [45] regarding singularities in configuration spaces of planar polygons.

In this case, the one-dimensional configuration space component is topologically a figure-eight instead of being a smooth circle, so at the singular point (the crossing of the two branches of the figure-eight, corresponding to the alignment of the joints of the mechanism), the trajectory is not well determined. The four-bar mechanism from Fig.17 is shown in several embeddings as a pointed pseudo-triangulation with a convex hull edge removed. Its configuration space is smooth everywhere, except at the alignment position, where it is a degenerate pseudo-triangle with all corner angles equal to zero.

Such singularities in the configuration spaces of 1dof mechanisms are troublesome: how do we decide which branch to follow, and how do we enforce algorithmically such a decision? If there are no singularities, however, there are only two directions in which such a mechanism can move. In the next section 4.3 we will prove that pseudo-triangulation mechanisms are smooth as long as the combinatorial structure of the embedding is maintained. This is exactly what the algorithm in section 6 guarantees. Therefore such problems related to singularities will not occur for the mechanisms of interest in this paper.

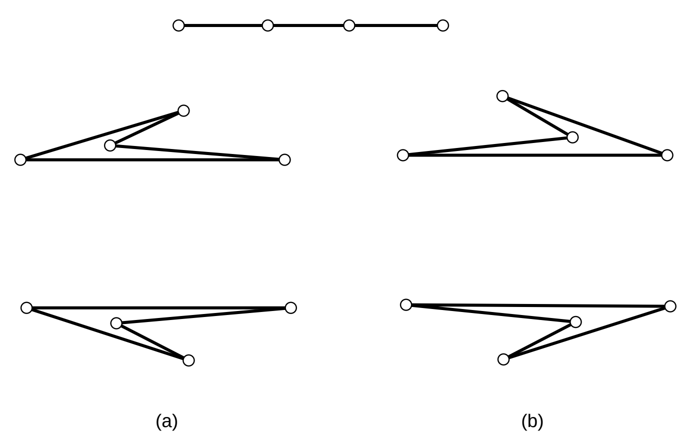

Figure 17: A semi-simple degenerate (collinear) 4-gon and the four possible ways in which a motion could go from the singular aligned position.

4.3 Expansion properties of 1dof Mechanisms

We start looking at the *free diagonals* of a Laman-minus-one mechanism: those which do not maintain rigidly their length, during a motion of the mechanism. They may expand or contract, i.e. increase or decrease in length. In this section, we investigate the expansion pattern of Laman-minus-one mechanisms.

The expansion pattern of a 1dof mechanism. Let $G(\mathcal{P})$ be a generic embedding of a Laman graph without an edge e. The space of infinitesimal motions is one-dimensional. Consider an infinitesimal motion v which *expands* the edge e, i.e. $d_e := \langle p_i - p_j, v_i - v_j \rangle > 0$. Such a motion always exists: take any motion v and if it doesn't expand, replace it with $-v$. The collection $s = (s_{ij})_{ij \in E}$ of the expansion signs of all the edges $ij \in E$, $s_{ij} := sign\langle p_i - p_j, v_i - v_j \rangle$, is called the *expansion pattern* or the *expansion signature* of $G(\mathcal{P})$. The sign is zero on edges of G and on edges *implied* by G, i.e. edges between vertices belonging to the same rigid component of G, and non-zero elsewhere.

The following is a straightforward consequence of continuity and of Theorem 4.3.

Corollary 4.4 *The expansion pattern of a generic 1dof mechanism $G(\mathcal{P})$ is the same for all the realizations in a sufficiently small neighborhood of $\mathcal{P}$.*

Proof: The expansion pattern depends continuously on the points and their velocities, which in turn depend on the rigidity matrix. All these values change continuously in a sufficiently small neighborhood of $\mathcal{P}$, where the rigidity matrix has maximal rank. Hence the signs do not change in a small neighborhood. $\square$

The following simple Lemma shows that the expansion pattern is the same for equivalent Laman-minus-one graphs embedded on the same set of points (as it is for the whole collapsed Laman-minus-one mechanism).

Lemma 4.5 *Let $G(\mathcal{P})$ and $G'(\mathcal{P})$ be generic realizations of equivalent Laman-minus-one graphs on the same point set $\mathcal{P}$. Then $G(\mathcal{P})$ and $G'(\mathcal{P})$ have the same expansion pattern.*

Proof: We replace an infinitesimally rigid component by another one with the same property: the expansion sign is zero on all the edges between vertices in the same rigid component. The space of infinitesimal motions remains the same, and hence so does the expansion sign on all the other edges. □

Theorem 4.6 (Smooth interval of an expansive pattern)
A bar-and-joint framework $G(\mathcal{P})$ based on a Laman-minus-one graph G in a generic embedding is a flexible 1dof mechanism maintaining its expansion pattern in an open interval containing $\mathcal{P}$. The interval is bounded by two special embeddings of G, each one corresponding to either some free edge(s) acquiring a zero-expansion value, or to a singularity (in the component of the configuration space of G containing $G(\mathcal{P})$).

Proof: The proof of Corollary 4.4 shows that the maximal interval consisting of self-stress-free realizations of G with the same expansion pattern as $G(\mathcal{P})$, and containing $G(\mathcal{P})$, is well defined. Its endpoints consist either in points where $G(\mathcal{P})$ acquires a self-stress (and hence the rank of the rigidity matrix drops, and the configuration space becomes singular), or where the expansion sign on some free edge becomes zero. □

Expansive 1dof mechanisms. A 1dof infinitesimal mechanism is (infinitesimally) *expansive* if there exists an infinitesimal motion such that all the free edges increase infinitesimally. I.e., there exists an assignment of velocity vectors v_i to each vertex p_i so that $\langle p_i - p_j, v_i - v_j \rangle \geq 0$, $\forall i, j$: the expansion pattern consists only in 0's and +'s.

Notice that the expansive property is defined *infinitesimally.* A 1dof mechanism $G(\mathcal{P})$ cannot be expansive throughout its lifetime, but only for a certain interval of time. We are interested in *generically infinitesimally expansive* 1dof mechanisms. Indeed, if $G(\mathcal{P})$ is expansive at $\mathcal{P}$, and this is a generic framework, then it will be expansive in a neighborhood of $\mathcal{P}$. Theorem 4.6 implies in this case that the configuration space is smooth and expansive in a neighborhood of $\mathcal{P}$. An *expansive mechanism* is an infinitesimally expansive 1dof mechanism $G(\mathcal{P})$, together with the maximal interval I around $\mathcal{P}$ in which it is infinitesimally expansive. We summarize these considerations in the following theorem, which will be used in section 6..

Theorem 4.7 *Let G be a Laman graph with a missing edge and let $G(\mathcal{P})$ be a generic infinitesimally flexible realization. If $G(\mathcal{P})$ is infinitesimally expansive, then G is an expansive mechanism in an open interval around $G(\mathcal{P})$, bounded by either a singularity or by an embedding where a free edge acquires zero-expansion.*

For example, the Peaucellier linkages in Fig.16 are not expansive, in the given realizations. Fig. 18 illustrates several snapshots in the lifetime of a 1dof mechanism. The first one is a semi-simple pointed pseudo-triangulation, and will be analyzed later. The second is a pointed pseudo-triangulation and is expansive. The last one is not infinitesimally expansive. The first and the third, Fig. 18 (a,c), lie at the boundary of the interval in which the mechanism in Fig. 18 (b) is infinitesimally expansive.

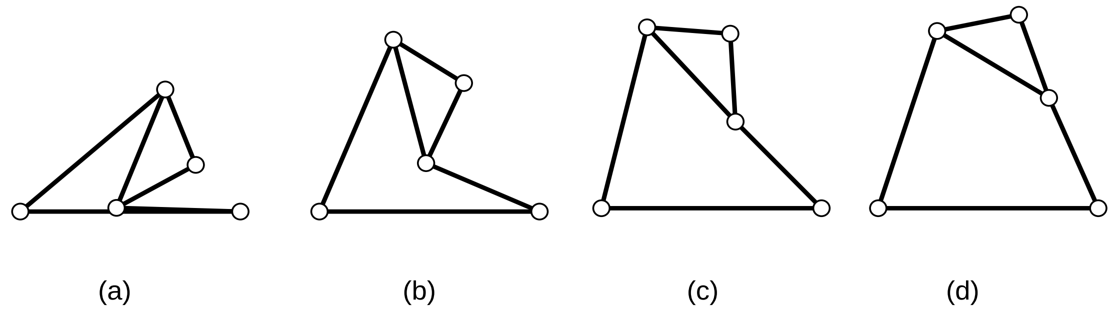

Figure 18: Four snapshots from the lifetime of a 1dof mechanism: (a) and (a) are the limit situations of of (b), which is expansive. (d) is not expansive: some diagonals increase, some decrease. Notice the change in the combinatorial structure from (b) to (d), via (c).

Farkas' Lemma for self-stresses and infinitesimal increases. In section 3.1 we defined the four linear spaces associated to the rigidity matrix and stated two orthogonality conditions (6 and 7). The orthogonality condition (6) between the space of self-stresses *ker* M^T and the space of infinitesimal length increases *Im* M implies that:

$$\sum_{ij\in E} w_{ij}\langle p_i - p_j, v_i - v_j\rangle = 0,\ \forall w = (w_{ij}) \in ker M^T,\ \forall v \in \mathcal{R}^{2n} \tag{9}$$

Let us rewrite it in terms of the values $d_{ij} = \langle p_i - p_j, v_i - v_j\rangle$, the infinitesimal increases.

$$\sum_{ij\in E} w_{ij} d_{ij} = 0,\ \forall w = (w_{ij}) \in ker M^T,\ \forall d_{ij} = \langle p_i - p_j, v_i - v_j\rangle \tag{10}$$

The sum cannot be zero if all the summands are strictly positive. In particular, this implies that there must exist a pair of edges $ij, kl \in E$ such that the signs are different: $w_{ij}d_{ij} > 0 \Leftrightarrow w_{kl}d_{kl} < 0$. We will refer to the sign of w_{ij} as the *self-stress sign*, and to the sign of d_{ij} as the *increase sign*.

Interpreting this condition for all edges proves the following Theorem. Alternatively, this follows from Farkas' Lemma, or Linear Programming duality, see [14].

Theorem 4.8 (Farkas' Lemma for self-stresses) *Let $G(\mathcal{P})$ be a framework, $w \in \ker M^T$ a self-stress on it and v an infinitesimal motion vector which preserves infinitesimally the lengths of some of the edges in G and increases or decreases the others. Denote by $d_{ij} := \langle v_i - v_j, p_i - p_j\rangle$, for $ij \in E$, the infinitesimal increase or decrease value.*

Then there exist at least two edges ij and kl which are not moved rigidly by v, and on which exactly one of the following conditions holds:

1. *Either the self-stress signs on both the edges are the same, $w_{ij}w_{kl} > 0$, and the increase signs differ $d_{ij}d_{kl} < 0$*
2. *Or the self-stress signs differ $w_{ij}w_{kl} < 0$, and the increase signs are the same $d_{ij}d_{kl} > 0$*

We will use the following corollary to prove the expansiveness of pseudo-triangulation mechanisms.

Corollary 4.9 *Let $G(\mathcal{P})$ be a Laman framework with an extra edge and let v be an infinitesimal motion v which acts rigidly on a subset of $2n-4$ edges of G and which changes infinitesimally the lengths of the other two remaining edges. Then $G(\mathcal{P})$ supports a self-stress, and the signs of the self-stress on the two non rigid edges satisfies exactly one of the following two conditions:*

1. *Either both self-stresses have the same sign, and the two edge increases have different signs*
2. *Or the two self-stresses have different signs and the two edge increases have the same sign*

4.4 Definitions: Combinatorial Equivalence of Pseudo-Triangulations

In this section we define what it means for two pseudo-triangulations or two pseudo-triangulation mechanisms to be combinatorially equivalent. We use this concept to characterize the interval of time when a pseudo-triangulation mechanism is expansive, and to identify the moment when it stops being so. Along the way, we define *combinatorial pointed pseudo-triangulations* and mechanisms.

Plane Graphs. A non-crossing embedding of a connected planar graph G partitions the plane into faces (bounded or unbounded), edges and vertices. Their incidences are fully captured by the vertex *rotations*: the ccw circular order of the edges incident to each vertex in the embedding. A *sphere embedding* of a planar graph refers to a choice of a system of rotations (and thus of a facial structure), and is oblivious of an outer face. A *plane graph* is a spherical graph with a choice of a particular face as the outer face.

A (combinatorial) *angle* (incident to a vertex or a face in a plane graph) is a pair of consecutive edges (consecutive in the order given by the rotations) incident to the vertex or face.

Combinatorial frameworks. A *combinatorial framework* $G(M)$ associated to a framework realization $G(\mathcal{P})$ is obtained by retaining (in M) only some combinatorial information from the underlying oriented matroid M of the set of points $\mathcal{P}$. Since in this paper we work only with special types of frameworks, to keep the focus on the main problem we do not give here the general definition, but see the upcoming paper [74]. In our particular case, when the framework is planar, the information M retained from the embedding will be, for each vertex i, the *signed* circular sequence $i : j_1 \cdots j_{n-1}$ of vertex indices $j \neq i$ in which a *directed line* through p_i, rotating ccw around the vertex encounters the adjacent edge vectors $\overrightarrow{p_ip_j}$. An index j corresponding to the edge vector $\overrightarrow{p_ip_j}$ is recorded positively or negatively depending on whether the rotating directed line encounters it in a way that matches its direction or the opposite one. This concept is a specialization of the *local sequences* of Goodman and Pollack [37] or *hyperline sequences* of Bokowski [22] (see also [72] and [23]) and retains (partial) oriented matroid structure from the underlying set of points $\mathcal{P}$. In particular, we can read off from this information the planar nature of an embedding of the framework, the underlying plane graph (rotations) and the pointed or non-pointed nature of the edge vectors at each vertex.

Combinatorial equivalence of planar frameworks. Two planar frameworks are *combinatorially equivalent* if there is a one-to-one correspondence between their vertices preserving edges and faces, the outer face and its orientation, and the underlying partial oriented matroid, i.e.

the circular ccw order of directed lines through the edge vectors around each vertex. This just means that they have the same underlying combinatorial framework (and the embeddings have the same orientation.)

In particular, this defines *combinatorially equivalent pseudo-triangulations*. The underlying combinatorial framework captures the information. We define now an alternative, equivalent but somewhat easier to understand concept, first described and used in [41].

Combinatorial pseudo-triangulation. Let G be a plane connected graph [1]. A *combinatorial pointed pseudo-triangulation (cppt)* of G is an assignment of labels *big* (or *reflex*) and *small* (or *convex*) to the angles of G such that:

(i) Every face except the outer face gets *three* vertices marked *small*. These will be called the *corners* of the face.

(ii) The outer face gets only *big* labels (has no corners).

(iii) Each vertex is incident to exactly one angle labeled *big* and is called *pointed*.

Let $G(\mathcal{P})$ be a pointed pseudo-triangulation (ppt), to which we will associate naturally a combinatorial pointed pseudo-triangulation (cppt) $G(C)$ by forgetting the point coordinates $\mathcal{P}$ and retaining (in C) only the plane graph structure and the label (convex or reflex) of each angle.

Two pointed pseudo-triangulations are *combinatorially equivalent* if they have the same underlying cppt structure.

A fundamental idea of our convexification algorithm described in Section 6 will be to use mechanisms obtained from pointed pseudo-triangulations, and to reconfigure them continuously as long as the combinatorial pseudo-triangulation does not change (where the removed convex hull edge is taken into account when considering the cppt structure). See Figure 18 for an illustration.

4.5 Pointed Pseudo-Triangulation Mechanisms Move Expansively

We are now ready to prove the main rigidity-theoretic result concerning pointed pseudo-triangulation mechanisms: they are expansive 1dof mechanisms for as long as the combinatorial pointed pseudo-triangulation is unchanged. At one of the two endpoints of the expansiveness interval, the configuration space of a ppt-mechanism may sometimes be singular. We show that for certain special situations (which will be the only ones relevant for the convexification algorithm of simple planar polygons), singularities never occur. This will guarantee that the algorithm behaves correctly at those moments.

As a corollary, we obtain that all frameworks which are Laman-minus-one equivalent to a pointed pseudo-triangulation mechanism are also expansive.

Theorem 4.10 (Infinitesimal expansiveness of pseudo-triangulation mechanisms) *A bar-and-joint framework whose underlying graph is obtained by removing a convex hull edge from a pseudo-triangulation is an infinitesimal 1dof expansive mechanism.*

[1] The definition is valid in a more general setting than what we need in this paper, see [41, 61]

Proof: Let G be the underlying graph of the pointed pseudo-triangulation, e_c be the removed convex hull edge (so that $G \setminus \{e_c\}$ is a ppt-mechanism) and e be an arbitrary edge not in G. As a Laman-minus-one graph, $G \setminus \{e_c\}$ has a natural decomposition into rigid components. We must prove that, when the removed convex hull edge e_c is expanded, so does any other free diagonal e (i.e. one which is not part of a rigid component). Indeed, the *missing diagonals* inside rigid components are not changing in length.

Let v be an infinitesimal motion preserving the edge lengths of $G \setminus \{e_c\}$ and increasing the length of the edge e_c, $d_{e_c} > 0$. We want to prove that it also increases the length of the edge e, i.e. $d_e > 0$. Consider the graph $G \cup \{e\}$, with $2n - 2$ edges. It has a non-trivial self-stress. Corollary 4.9 implies that it is sufficient to prove that this self-stress cannot be strictly positive on both e_c and e. Assume for the sake of a contradiction that this happens, i.e. that $G \cup \{e\}$ has a self-stress w with $w_{e_c} > 0$ and $w_e > 0$. We will interpret this in terms of the induced Maxwell lifting to get a contradiction.

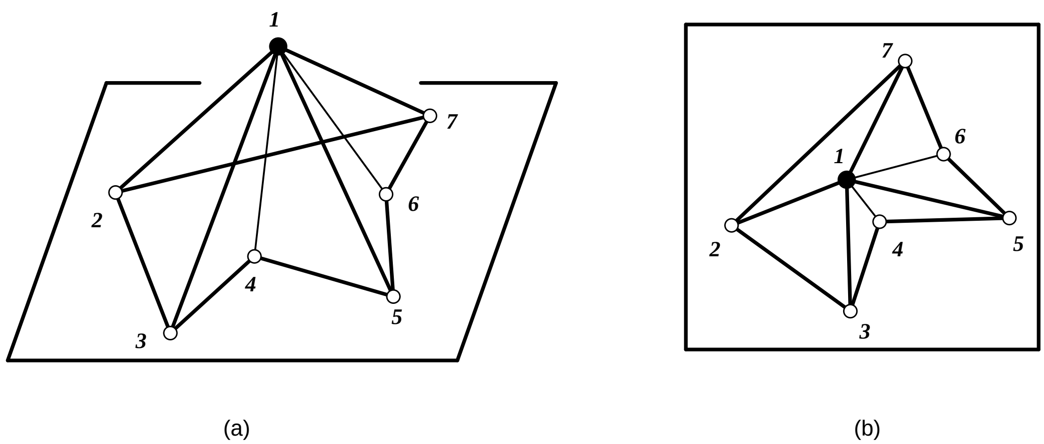

Figure 19: The mountain/valley argument: (a) Cutting just below the vertex of maximum z-coordinate (vertex 1), we get the image in (b). There must be at least 3 mountain edges incident to 1, and their projections are non-pointed. In (a), the mountain edges are 12, 13, 15 and 17.

Consider the framework $G \cup \{e\}$ obtained from G by adding the extra edge e. Since it is no longer a pointed pseudo-triangulation, either an endpoint of this new edge e is non-pointed, or the edge e crosses some other edges of the pseudo-triangulation, or both. If new crossings have been introduced, we will apply Bow's construction to obtain a new planar framework $G'(\mathcal{P}')$, which will have n' vertices and $2n' - 2$ edges. In either case, the new framework is non-pointed only at (one or possibly both of) the endpoints of e or at the crossings of e with other edges. Denote by M the set of non-pointed vertices: the endpoints of e, if non-pointed, and the crossings of e with other edges, if there are such crossings. M is non empty.

The signs of the self-stresses are preserved by Bow's construction. Since we assumed a strictly positive self-stress on e_c and e, this means that both e_c and e are valley edges, in a Maxwell lifting of $G \cup \{e\}$ (and this extends to the split edges in case the Bow's construction has been applied). It means that the only edges which could be mountain edges are the edges of G.

We now apply the same argument as in Theorem 3.7. We look for a vertex of maximum height and conclude that it must be non-pointed, hence only those in M are candidates.

We show first that the maximum cannot be attained (locally) on more than a vertex, i.e. not on an edge or face or larger substructure. Indeed, if an edge would be at maximum height, its two endpoints would have to be in M. But the edges between vertices in M are either e or splittings of e and have negative stress, therefore lift to valleys. Valleys cannot be maxima, obviously.

To complete the proof, let's focus on the vertex of maximum height and call it the *tip*. If the tip is a Bow vertex, then its four incident edges must all be mountain edges, because of their collinearity. These include the split added edge e. But this contradicts the previous argument that only edges of G may be mountain edges.

So we may now assume without loss of generality that the maximum is vertex 1 (see Fig. 19). Because 1 is an isolated maximum (not part of a local maximum subgraph), then all its incident faces are slanted (non-horizontal). Cutting the lifted polyhedral surface with a plane slightly below the tip, we get in this cut plane a simple closed polygon (therefore, with at least three vertices) containing the projection of the tip vertex 1 inside it, and therefore inside its convex hull. The projections on the cut plane of the polyhedral surface edges incident to 1 which are *mountain edges* must join the projection of the tip to the convex hull vertices of the projection in a non-pointed manner (because the projection lies in their convex hull). See Fig. 19 for an illustration. But these are exactly the edges incident to 1 in G, and they form a pointed edge set, hence the contradiction. □

Corollary 4.11 *Let $G(\mathcal{P})$ and $G'(\mathcal{P})$ be generic realizations of equivalent Laman-minus-one graphs. If $G(\mathcal{P})$ is a pointed pseudo-triangulation mechanism, then $G'(\mathcal{P})$ is also a 1dof expansive mechanism.*

Proof: We replace an infinitesimally rigid component by another one with the same property. The space of infinitesimal motions and the expansion pattern do not change. □

Flexible expansiveness of pointed pseudo-triangulation mechanisms. We summarize what we have shown so far.

A bar-and-joint framework $G(\mathcal{P})$ obtained by removing a convex hull edge e from a pseudo-triangulation is a flexible 1dof expansive mechanism for as long as the combinatorial pseudo-triangulation (of $G \cup \{e\}$) doesn't change. The interval of expansiveness is bounded by two special embeddings of $G \cup \{e\}$: one has (at least one) semi-simple pseudo-triangular face (with at least one zero-angle corner), the other has (at least one) aligned pair of extreme edges (an angle of π at a non-corner). A special instance of the latter case occurs when the missing convex hull edge aligns with another convex hull edge.

The interval contains no singular points (of the configuration space), since pseudo-triangulations are infinitesimally rigid. Hence the mechanism is smooth in this interval. It remains to prove (in the next section) that if the semi-simple pseudo-triangular face is not collapsed to a flat polygon, then this extreme embedding is not singular. The other extreme embedding (with an aligned vertex) is never a singularity.

4.6 Pseudo-triangulation mechanisms with semi-simple but not flat pseudo-triangles are non singular

There is only one important thing left to be proven: that singularities cannot occur, for the types of pointed pseudo-triangulation mechanisms used in this paper. More precisely, we have to address the case of those pointed pseudo-triangulations which contain semi simple, but not flat pseudo-triangles. A *flat* pseudo-triangle has all its vertices embedded on a line, and is therefore singular. But such flat pseudo-triangles never appear in the convexification algorithm described in Section 6. The algorithm uses, however, semi-simple pseudo-triangles: these can have some flat corners (possibly more than one), but not be altogether flat. Moreover, even when zero-angle corners occur, the incident edges do not overlap completely. Indeed, this is the definition of semi-simplicity given in Section 2.1.

We show now that these *semi simple pointed pseudo-triangulations* satisfy the same properties as pointed pseudo-triangulations: they are self-stress-free. With a convex hull edge is removed, such an object will appear as the starting point of the expansion interval of the underlying graph. By proving that they are never stressed, we guarantee that the configuration space is smooth at that point, which implies that the motion is well defined.

Theorem 4.12 *Semi simple pseudo-triangulations are infinitesimally rigid.*

Proof: We need only a simple extension of the argument used in the proof of Theorem 3.7, to handle the semi simple pseudo-triangular faces. If we assume that there exists a lifting where the two overlapping edges are lifted to two distinct lines, then there is no way of obtaining a lifted flat face for the semi-simple pseudo-triangle. So the two edges must overlap in the lifting. But then, the face that must be lifted appears in the lifting as a (pure) pseudo-triangle (plus a segment attached to it), with a new corner at the position where the endpoint of one of the two overlapping edges lies on the other edge. Extending the argument used in Theorem 3.7, the only possible placement of a vertex of maximal height would then be at this new corner. But this is impossible, because it belongs to a lifted edge (the larger of the two overlapping ones). The whole edge cannot be at maximum height, because its endpoints are pointed. Hence the contradiction to the existence of a lifting, and of a self-stress. □

Corollary 4.13 *Semi simple pseudo-triangulations with a convex hull removed are self-stress-free, hence non-singular.*

Proof: Removing an edge from a Laman graph in a generic embedding results in a self-stress-free Laman-minus-one graph. □

This concludes the sequence of rigidity theoretic properties of pointed pseudo-triangulations and allows us to use them as building blocks describing simple motions for the convexification algorithm. We summarize their properties in the following Theorem, on which the correctness of the Algorithm described in Section 6 relies.

Theorem 4.14 (Motion of Pointed Pseudo-Triangulation Mechanisms) *A pointed pseudo-triangulation mechanism moves expansively on a unique, well defined (one-domensional) trajectory in its configuration space, from the moment when it is embedded as a semi-simple pseudo-triangulation to the moment when two extreme edges of a vertex (or, as a special case, when one of these is the missing convex hull edge) align.*

Proof: The statement about the beginning of the expansion interval follows from Corollary 4.13. At the other endpoint, since all distances increase, so do all the convex angles. The pointed pseudo-triangulation remains the same, combinatorially, until two extreme edges incident to a vertex align. Notice that one of these two aligned edges may be the missing convex hull edge. □

5 Algorithms for Constructing Pseudo-Triangulations

In this section we investigate several algorithms for constructing pointed pseudo-triangulations of planar point sets and polygons. They rely on the maximality property 5 from Theorem 2.3, which implies that we can extend any non-crossing and pointed edge set to a pointed pseudo-triangulation by adding edges in an arbitrary order, as long as we do not violate these two properties (pointedness and non-crossing). In particular, any simple polygon (which is a pointed and non-crossing edge set) can be extended to a pointed pseudo-triangulation.

5.1 Pseudo-triangulations of planar point sets

The input to a pseudo-triangulation algorithm is a planar point set, not all collinear. The result must be a (possibly semi-simple) pointed pseudo-triangulation. For simplicity, we will assume general position (no three points are collinear). Some of the algorithms we will describe would work without major modifications under less restrictive conditions.

All throughout, we will assume that the pointed pseudo-triangulation is stored as a planar map, e.g. using the quad-edge data structure of [39].

A pseudo-triangulation is called *Henneberg I*, if it can be constructed inductively using only Henneberg I steps, cf. Theorem 2.3(6).

Incremental algorithm. This algorithm assumes that the points are in general position and adds the points one at a time. It produces Henneberg I pseudo-triangulations.

Algorithm 5.1 (Incremental Algorithm)
Input: *A point set $\mathcal{P}$.*
Output: *A pointed pseudo-triangulation of $\mathcal{P}$.*

1. *Start with three arbitrary points, join them in a triangle.*
2. *Add the remaining points one at a time. Each new point falls inside a face of the previously constructed pointed pseudo-triangulation. Take two tangents to the side chains of the pseudo-triangular face to extend the pseudo-triangulation. Readjust the planar graph data structure.*

The complexity of this algorithm is as follows. There are n steps. Each step requires a point location phase, to find the face containing the new point ($O(\log n)$ time), a tangent finding phase (linear time in the size of the face) and an insertion phase, with an update of the planar map data structure. A rough analysis gives a worst case running time of $O(n^2)$.

Sweep line algorithm. Add points in increasing order of their x-coordinates, or in any sweep line order. It is a simpler version of the previous algorithm (the incremental one) and produces pseudo-triangulations that are constructed using only Henneberg I steps on the outer face.

Algorithm 5.2 (Sweep line Algorithm)
Input: *A point set $\mathcal{P}$.*
Output: *A pointed pseudo-triangulation of $\mathcal{P}$*

1. *Sort the points by x coordinate.*
2. *Make a triangle from the first three points.*
3. *Add the remaining points in increasing order of their x-coordinates. At each step, take two tangents to the convex hull of the already constructed pseudo-triangulation.*

The complexity of this algorithm is $O(n \log n)$. The initial sorting takes that much. The rest of the algorithm is identical to the incremental algorithm for constructing the convex hull of a point set, except that here we retain all the intermediate tangents. The amortized analysis for the convex hull algorithm gives linear time total for all the work after the initial sorting phase.

Greedy flip algorithm. The previous algorithms produce only Henneberg I pseudo-triangulations. We may flip some diagonals to obtain other types of pseudo-triangulations. A well developed algorithm is described in Pocchiola and Vegter [63].

Algorithm 5.3 (Greedy Flip Algorithm [63])
Input: *A point set $\mathcal{P}$.*
Output: *A pointed pseudo-triangulation of $\mathcal{P}$.*

1. *Start with a sweep line pseudo-triangulation.*
2. *Flip some edges in a greedy manner: the smallest slope first.*

Complexity. Because the convex hull is part of any pseudo-triangulation of a point set, the well known lower bound of $\Omega(n \log n)$ applies, and hence we cannot do any better than the sweep line algorithm.

5.2 Pseudo-triangulations of simple polygons

We expect to be able to beat $O(n \log n)$ when constructing a pseudo-triangulation of a polygon. Indeed, we show that this is possible if we rely on the linear time polygon triangulation algorithm of Chazelle [28]. For a practical implementation, we also describe some other alternative approaches (or we rely on other sub-optimal polygon triangulation algorithms).

Incremental Algorithm for polygons. The pseudo-triangulation is constructed incrementally, adding one edge of the polygon at a time, in ccw order, starting with a vertex on the convex hull. This insures that the last step of the algorithm will reuse a previously inserted edge and won't necessitate any additional deletions. At each step of the insertion we add a vertex, a polygon edge and one other additional edge. However, we might have to displace or modify several other edges, depending on whether the pointedness condition, the planarity condition, or both are violated by the insertion of the new polygon edge.

We show how to modify some of the added pseudo-triangulation edges using an argument which we call the *rubber band argument with snapping*, which locally modifies (in worst case linear time) the edges adjacent to the new polygon edge to preserve pointedness and planarity. The resulting worst case complexity is $O(n^2)$ time.

Algorithm 5.4 (Incremental Algorithm for Polygons)
Input: *A simple polygon $\mathcal{P}$ given by its point set, in ccw order around its boundary.*
Output: *A pointed pseudo-triangulation of $\mathcal{P}$*

1. *Start from a vertex on the convex hull of the polygon and the first two edges to obtain a triangle.*

2. *Add the vertices one at a time, together with a new polygon edge.*

3. *If pointedness is violated, split the existing edges incident to the last vertex into two parts, using the new edge as a divider. Move half of them (the part containing only added edges, not the other polygon edge incident to the vertex) to the new endpoint. Recurse, to maintain planarity and pointedness.*

4. *If planarity is violated, it is because the new edge is cutting through several added edges. Subdivide these crossed edges into two, and recurse to maintain planarity and pointedness on each half. Some of these new edges will coincide.*

See Figure 20 for an example. The correctness proof would show that at step 4, only one new edge is inserted. We omit the details of the implementation and analysis, because we won't rely on this approach.

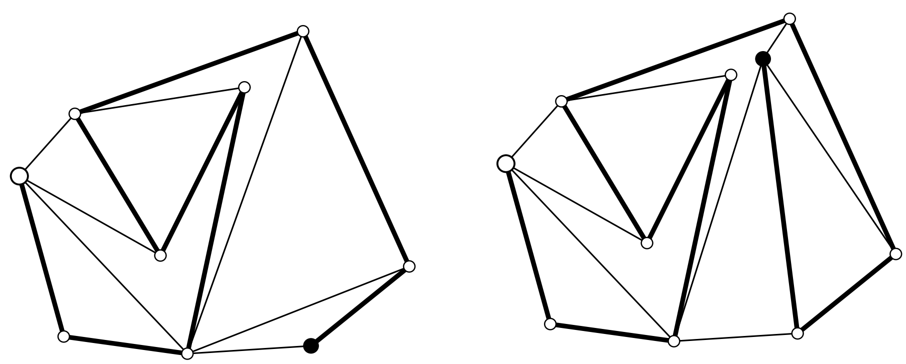

Figure 20: A typical step, violating planarity, of the incremental algorithm for computing the pseudo-triangulation of a polygon. The first vertex is white and big, and the last inserted vertex is black. The polygon edges are thick and the new diagonals are thin.

Shortest Path (Geodesic) Tree algorithm. The *reduced shortest path tree* inside a polygon, rooted at a vertex i, computes the shortest paths from a convex hull vertex i to all the other vertices, such that adding the edges of the path doesn't produce (at its endpoint) a non-pointed vertex. For this to happen, the destination of the path must be a reflex vertex of the polygon. It suffices to compute all the paths ending at convex vertices. The collection of edges of the reduced shortest path tree is planar and pointed. If we compute it for the polygon and all its pockets (determined by the complement of the polygon with respect to its convex hull), we obtain a pointed pseudo-triangulation of the point set of the polygon, which includes the polygon edges.

All the steps of this algorithm have already been described in the literature. To compute the convex hull of the polygon and identify its pockets takes linear time. We triangulate in linear time using Chazelle's [28] algorithm. Inside the polygon and in each of the pockets, a shortest path tree from a vertex using the triangulation, is done in linear time using the algorithm in [40]. We retain only the paths leading to convex vertices.

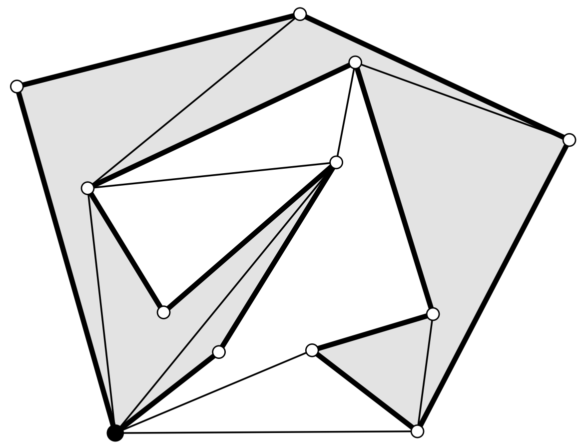

Figure 21: The shortest path tree pseudo-triangulation of a polygon. The polygon edges are thick and the new diagonals are thin. The root of the tree, for the polygon and its unique pocket, is black.

6 The Convexification Algorithm

We are now ready to present the main results of the paper, Algorithm 6.1 for convexifying a simple planar polygon with expansive motions, and its correctness proof, Theorem 6.4.

To convexify a polygon, our approach is to first construct a pointed pseudo-triangulation, then turn it into an expansive mechanism by removing a convex hull edge, and then move it until the end of its expansive interval. At that point we apply a local flip to obtain another pseudo-triangulation mechanism, and continue like this until reaching a convex position.

Reconfigurable frameworks. Two realizations $G(\mathcal{P})$ and $G(\mathcal{P}')$ of the same framework (G, L) are said to be *reconfigurable* into one another, if they belong to the same component of the realization space. We want to reconfigure two simple polygonal frameworks, without producing self-intersections along the way. For this, the two polygons must be similarly oriented.

Reconfiguring more general frameworks. The algorithm can be applied to collections of open and closed polygons, none inside the other, or to more general collections of objects consisting in a pointed and non-crossing collection of edges. In this last case, the expansive, non-colliding motion will continue only as long as planarity and pointedness are maintained by the ability to flip an added edge (see below).

6.1 The Convexification Algorithm

The main algorithmic result of the paper can now be described.

Algorithm 6.1 (The Pseudo-Triangulation Convexification Algorithm)
Input: *A simple planar polygon on the point set $\mathcal{P}$.*
Output: *A convexification of $\mathcal{P}$, and a sequence of motions, each one given by a pseudo-triangulation mechanism, describing the convexification process.*

1. *Compute an initial pseudo-triangulation of $\mathcal{P}$, and remove an arbitrary convex hull edge which is not a polygon edge.*
2. *Repeat until a convex position is reached:*
   (a) *Pin down an arbitrary edge. Move the pseudo-triangulation mechanism along its unique trajectory, in the expansive direction, until an alignment event occurs.*
   (b) *At the alignment event, perform either a flip or a freeze operation (defined below). If necessary, reconfigure locally the pseudo-triangulation (see below how), and continue.*

See the examples from Figures 1, 2 and 3.

In the rest of this section, we give the missing definitions, the correctness proof and the time analysis.

6.2 Alignment Events

Each expansive mechanism induced by a pseudo-triangulation can be moved as long as the framework stays pointed. Pointedness is violated when two extreme edges at a vertex align. We now show how to reconfigure the mechanism and continue the motion after an alignment event. The overall motion is obtained by gluing together the trajectories corresponding to the motions happening between two events.

When two bars become collinear, we have to recompute a new pseudo-triangulation. Theorem 6.2 proves that this can be done either with local changes and keeping the same number of vertices in the original polygon, or by decreasing by at least one the number of vertices of the polygon and applying induction.

Theorem 6.2 (Gluing trajectories at alignment events) *When two edges align, one of the following three cases may occur:*

1. **Freeze event.** *Two adjacent edges of the polygon become collinear. In this case, we freeze the joint, eliminating one vertex of the polygon.*
2. **Flip event.** *Two adjacent added diagonals or one diagonal and an edge of the polygon become collinear. In this case, we perform a flip in the pseudo-triangulation to obtain a pseudo-triangulation with a semi simple face.*

3. **Convex hull** *false* **event.** *The* missing convex hull edge *and either a diagonal or a polygonal edge align. In this case, we continue the motion and do not consider this a proper event. Note that the* missing edge *will change combinatorially, as does the convex hull.*

An example is depicted in Fig. 22.

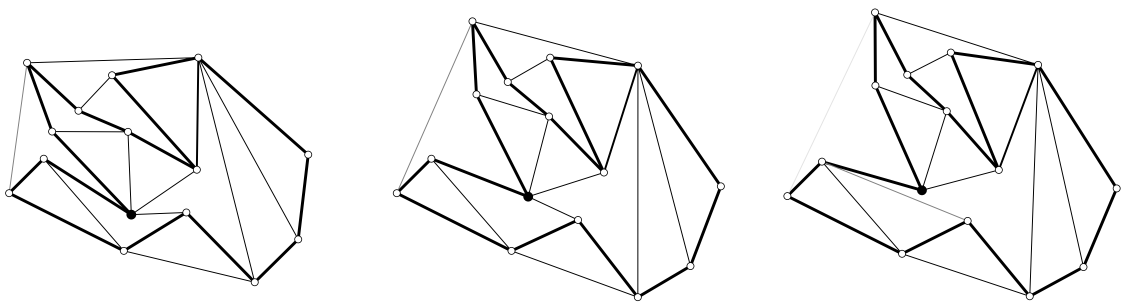

Figure 22: Patching the pseudo-triangulation by a local flip when two bars (not both of which are polygon edges) align.

The above theorem is not stated in full generality, to avoid cluttering the overall picture with details. In particular, several vertices may straighten simultaneously, but the same type of flipping would work in each case.

Other problems occur when we freeze two aligned edges of the polygon. In this case we must get rid of the other diagonals (if any) adjacent to the vertex, which can also be done by local changes (but may involve linearly many edges). Occasionally this operation rigidifies the framework: then we must pick up another convex hull edge to remove from the convex hull. All these details are solved in a straightforward manner.

Theorem 6.3 (Termination and complexity analysis) *The convexification of a polygon terminates in at most $\mathcal{O}(n^3)$ steps, where a* step *is the continuous motion induced by* one *pseudo-triangulation mechanism, between two alignment (flip or freeze) events.*

Proof: With some care in the patching strategy, it can be shown that no combinatorial pseudo-triangulation will occur twice in the convexifying motion. Since there are finitely many combinatorial pseudo-triangulations, the algorithm terminates.

A more careful accounting refines the analysis and gives at most $\mathcal{O}(n^3)$ steps. We do so by bounding the number of flip events between freeze events, which are exactly the number $r = \mathcal{O}(n)$ of reflex vertices in the polygon.

Consider the number of bends in all the shortest paths forming the pseudo-triangulation, inside the polygon and its pockets. Since there are linearly many shortest paths accounted for (we consider maximal paths only, not subpaths), and each can have at most linearly many bends, the crude bound on this number is $\mathcal{O}(n^2)$. In certain situations, the bound may even be attained, for instance when an added diagonal flips along the edges of a big $O(n)$-sized rigid (and therefore convex) component. The key observation is that a local flip decreases the number

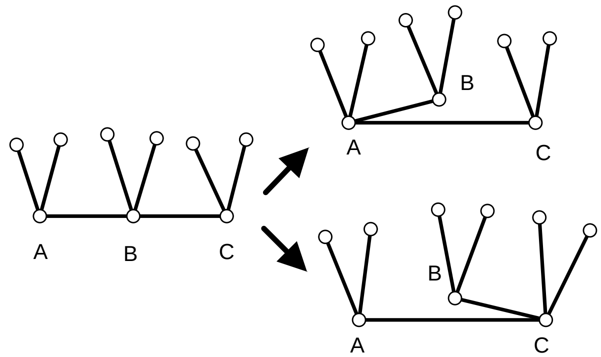

Figure 23: When the two aligned edges are not polygon edges, there are two possible diagonal flips. A flip reduces the number of bends in geodesic paths using the vertex involved in the alignment.

of bends by at least one (or more, depending on how many geodesic paths go through the vertex where the event happened). Hence the number of steps is at most $\mathcal{O}(rn^2) = \mathcal{O}(n^3)$. □

The main result is now a consequence of all the results proven so far:

Theorem 6.4 (Main Result: Convexification of Planar Chains with Expansive Motions) *Every planar polygon can be convexified with at most $\mathcal{O}(n^3)$ motions. Each motion is induced by a 1dof expansive mechanism constructed from a pseudo-triangulation with a deleted hull edge, which is continuously reconfigured expansively until two adjacent edges align. At that point a local flip of the diagonals restores a pseudo-triangulation. The complete trajectory in configuration space is a sequence of simple curves, each one naturally parametrized by a rotating edge in the plane. The first pseudo-triangulation can be computed efficiently in linear time and updated in (at most) linear time per step.*

7 Conclusions

More than just giving a particular solution to the Carpenter's Rule problem, our combinatorial approach to planning non-colliding motions based on pseudo-triangulations has several conceptual consequences. We conclude with a brief discussion of further directions of research and remaining open questions or applications to be considered, as well as related papers, resources and recent work on pseudo-triangulations and their applications.

Note on multiple papers. While working on the full version of [73], I realized that it would be too much material for a single self-contained paper. I decided to split it into three parts, of which the current paper should be read first.

The second part, describing some of the numerical issues of the algorithm implementation, has appeared in [75]. Here's a short summary of these considerations. The algorithm we presented works in steps, but from the configuration in the beginning of a step to the one at the end, there is no constant-time algebraic computation. Indeed, finding the coordinates of the points at the alignment events involves solving a system of quadratic equations in $2n-3$ equations with that many unknowns. Solving them exactly many lead, in the worst case, to exponentially many real solutions (see [24]), from which the solution corresponding to the actual position during the motion would have to be isolated based on combinatorial considerations. Moreover, there is no simple algorithm for finding all of these solutions, as elimination-based solvers work in exponential time. In practice, the systems are solved numerically. See also the Note below on the model of computation and complexity.

The third part [74] (in preparation) will deal with the observation (and its consequences) that certain geometric facts used in the current paper have natural extensions, proofs and applications in the purely combinatorial setting of rank 3 affine oriented matroids (pseudo point sets). We explore, in particular, to what extent is the rigidity matroid *orientable* in a combinatorially recognizable way (based on the oriented matroid of the underlying point set). Indeed, pointed pseudo-triangulation mechanisms induce instances of combinatorially recognizable signed co-circuits. In particular, certain simple geometric facts used in the current paper can be precisely derived in this context.

Note on terminology. In the conference version of this paper, I used *acyclic* instead of *pointed*. The terminology of *acyclic vector sets* comes from oriented matroid theory (see [21]), an approach which provided the guiding line in the search for combinatorial properties of simple motions induced by mechanisms. More precisely, for a set of acyclic vectors there is no linear combination with positive, not all zero coefficients that sums them to zero, while in the cyclic case there is always one. The decision to change the terminology was motivated by the obvious potential of confusion with the entrenched graph-theoretic concept of acyclic graph.

Note on the Model of Computation and Complexity. Computationally, our problem has as input an ordered list of $2n$ real numbers, the coordinates of the n vertices of the polygon. The desired output is a finite description of the convexifying motion, and ultimately a set of $2n$ real coordinates for the polygon vertices in the final convex state.

The moving vertices follow continuous curve segments during this motion, which can be parametrized by time. Ideally, one would like to express all the computations leading to parametric descriptions of these $n-2$ vertex trajectories in the real RAM model, the computational model of choice in Computational Geometry (see [64]). However, as is the case for other algorithms in Real Algebraic Geometry (see [17]), not all numerical data can be computed explicitly. The positions of the polygon vertices at the event points are given implicitly as solutions to systems of polynomial equations. Theoretically, one may attempt to perform the costly (exponential time) exact algebraic elimination (Gröbner basis algorithms) to reduce the system to a single one-variable polynomial equation, but then one still would have to find the roots of this polynomial numerically (i.e. not in the RAM model). In practice, and since we must rely on numerical methods anyway for root finding, everything is computed approximately via numerical methods, see [75].

In [73], the number of steps in the convexification algorithm was hastily estimated as $O(n^2)$ instead of $O(n^3)$.

Animations and web resources. A web page (constructed with the help of my students Elif Tosun and Beenish Chaudry) containing animations and other graphical illustrations of the approach described in this paper, can be found on the author's web page at:

`http://cs.smith.edu/~streinu/Research/Motion/motion.html`

7.1 Open Questions on Reconfiguring Chains

Several questions directly connected to our approach remain still open at the time of this writing.

In practical implementations of our algorithm, we have observed that the number of events is often linear. We can easily produce an example where a shortest-path pseudo-triangulation approach would encounter a quadratic number of flip-events between two freeze events. Take a pseudo-triangulation containing a rigid (and hence convex) component with linearly many vertices, and design shortest paths from a vertex to linearly many other vertices, which are obstructed by the large convex component and must bend around it. It it not hard to design a pseudo-triangulation mechanism where all these $O(n)$ paths would unfold one by one, for a total of $O(n^2)$ steps, before any freeze event would happen. It is also easy to design a freeze event which would readjust the pseudo-triangulation by introducing all these bends again. However, neither the shortest path tree pseudo-triangulation, nor the patching strategy for the pseudo-triangulation (at a freeze event) are the only available options. It is possible that other choices of pseudo-triangulations would lead to faster convexification algorithms.

Open Problem 1 *Is there a different strategy of choosing pseudo-triangulation mechanisms, leading to asymptotically fewer events?*

Another potential source of simplification is in the complexity of the flip operation.

Open Problem 2 *Is it possible to perform a constant time flip at an alignment event, while maintaining an overall small number of events during the convexification algorithm?*

The most intriguing remaining question is whether the approach based on expansive motions and pseudo-triangulations can be extended to non-crossing polygons which are not necessarily simple. To define the problem, let's introduce some additional terminology. Related problems, and more general self-touching linkages have also been recently considered in [31].

Two edges are non-crossing if their segments do not have an interior point in common, otherwise they are said to cross properly. Non-crossing edges may be touching, either overlapping, or having an endpoint of one edge lying on another edge. We view each edge as oriented and having two "sides" (left or right, with respect to a given direction), so that when two edges are touching or overlapping, we can specify on which side they meet. For example, we think of the two edges 12 and 45 in Fig. 24 (oriented from 1 to 2, and from 4 to 5) as touching on the left of both 12 and 45.

A *non-crossing polygon* has no properly crossing edges, although the edges may be touching in complicated ways. Therefore, we also need to add to the definition of the polygon a *touching pattern*, defined in such a way that would allow some simple, non-crossing unfolding reconfiguration (not necessarily maintaining rigidly the lengths of the edges). The touching pattern indicates which side of each edge is in contact with which side of another edge, in a consistent fashion. A touching pattern is *consistent* if there exists a (topological) simple realization of the polygon (not necessarily with straight-line segments, but simple, i.e. non-touching), which is consistent with the sidedness relation.

For instance, if the touching pattern of the two edges 12 and 45 in Fig. 24 was defined such that 12 was in contact on its right side with the right side of edge 45, then that wouldn't have been considered a non-crossing touching pattern, as the two touching edges would have had to slide over each other to get "on the other side", in order to become a simple polygon.

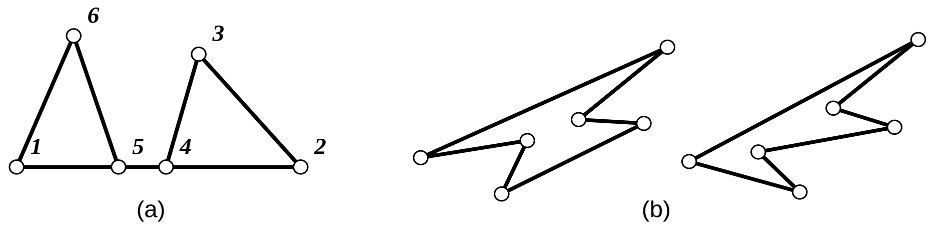

Figure 24: (a) A non-crossing, but not simple polygon. (b) Two possible polygonal patterns (given by the above/below relationships between edges) compatible with the same flat polygon.

Open Problem 3 *Is it possible to convexify every non-crossing polygon (with any consistent touching pattern)?*

The special situation when all the edges are collinear is the only case when the configuration space of a polygon has singularities, see [45]. This creates additional questions, regarding which are the combinatorial possibilities (given by touching patterns) for an opening flat polygon. For the simple example of a four-gon, see Fig. 17. Figure 24 (b) suggests two possible opening patterns of a flat six-gon.

Open Problem 4 *Given a flat n-gon and a consistent touching pattern, is there a configuration, infinitesimally close to the the flat singular position, which opens the polygon in a manner compatible with the given pattern?*

Open Problem 5 *Are all the consistent simple touching patterns of a flat polygon realizable by infinitesimally close configurations (with the same edge lengths)?*

Related questions refer to finding (possibly semi-algebraic) conditions that would distinguish between opening motions along trajectories with distinct combinatorial patterns, when moving away from a singularity point.

7.2 Recent work and Open Questions on Pseudo-Triangulations

Since the conference abstract was published [73], several questions regarding the combinatorial properties of pseudo-triangulations have been addressed. Without making a claim at giving a complete listing, we refer here to some of these developments, as well as other interesting remaining open questions.

Open Problem 6 *How many pseudo-triangulations of a point set or polygon are there?*

Partial results are given in [65, 8, 6]. A related question is to enumerate efficiently all pseudo-triangulations. Current approaches include [18, 26, 7]. A remaining questions is:

Open Problem 7 *Find a simple algorithm to enumerate pseudo-triangulations with constant amount of work per new enumerated object.*

More generally, it is natural to ask whether there is an interesting polytope whose 1-skeleton is the graph of pseudo-triagulations. Solutions have appeared for pointed pseudo-triangulations [68] and general (not necessarily pointed) ones, [60], [2], [4]. An interesting remaining question is:

Open Problem 8 *Is there a concept of a regular pseudo-triangulation, as it is for triangulations ([35])?*

Since pseudo-triangulations can be naturally defined in the coordinate-free context of oriented matroids, we are also interested in those combinatorial properties that would distinguish them (as a collection) from the ones realizable in the Euclidean plane. In [41], it was shown that all planar Laman graphs are realizable as pointed pseudo-triangulations, even when an additional combinatorial pseudo-triangular structure was enforced. Additional properties of combinatorial pseudo-triangulations were studied in [61].

Special types of pseudo-triangulations were studied in [47, 5] and [69]. Further applications in visibility [71] and kinetic data structures for collision detection have appeared in [50, 49]. A recent application to single-vertex origami and unfoldings of spherical linkages is given in [77]. Surfaces from pseudo-triangulations and reciprocal diagrams of pseudo-triangulations were studied in [59] and [3].

One of the most intriguing remaining open question is to extend this work in higher dimensions.

Open Problem 9 *Extend combinatorial expansive motions and pointed pseudo-triangulations to three and higher-dimensions.*

Some partial work in the direction of combinatorial conditions for expansive motions in three dimensions is contained in [77].

From the point of view of Rigidity Theory, pointed pseudo-triangulations are a class of planar minimally rigid graphs. There is an analogous theory of rigidity with fixed edge-directions (rather than fixed edge-lengths). In some recent work of the author [76], pointed pseudo-triangulation mechanisms have been shown to possess interesting properties in this model, too.

All the pictures in this paper have been produced using the software package Cinderella [67], which supports motions of 1dof mechanisms and even animates them. But not all pointed pseudo-triangulation mechanisms seem to admit a Cinderella construction.

Open Problem 10 *Prove that not all pointed pseudo-triangulations admit a ruler-and-compass construction. Characterize those which do.*

Lastly, two software implementations of pseudo-triangulations are freely available [25, 46].

Acknowledgements. This work was possible due to the stimulating research environment I encountered at the McGill University Workshops on Folding and Unfolding Problems in Computational Geometry, Barbados 1998 and 2000, the Monte Verita 1999 Discrete and Computational Geometry Workshop in Ascona, Switzerland, and the Geometry Festival, an International Workshop on Discrete Geometry and Rigidity in Budapest, Hungary, Nov. 1999. I am grateful to all the organizers for their invitation (András Bezdek, Károly Bezdek, Károly Böröczy, Bob Connelly, Jacob E. Goodman, Anna Lubiw, Ricky Pollack, Emo Welzl and Sue Whitesides) and to the many participants (too many to list here, but see [19], [20]) for enthusiastical conversations on this and related problems.

I am particularly indebted to Günter Rote for explaining to me his ideas, finalized in [30], on the use of expansive motions and the linear programming approach to the Carpenter's Rule problem.

I acknowledge useful conversations on various aspects of Rigidity Theory with Bob Connelly, Henri Crapo, Ruth Haas, Brigitte and Herman Servatius and Walter Whiteley. Many thanks to Ciprian Borcea for pointing several connections with the mathematical literature and for stimulating conversations on aspects of rigidity theory pertaining to real algebraic geometry.

I acknowledge partial NSF support for organizing the Barbados workshop on pseudo-triangulations, Jan. 26-Feb 2, 2001, where some of the open problems raised in the preliminary version of this paper [73] have been solved.

References

[1] Pankaj Agarwal, Julien Basch, Leonidas Guibas, John Hershberger, and Li Zhang. Deformable free space tilings for kinetic collision detection. *International Journal of Robotics Research*, 2003. Preliminary version appeared in Proc. 4th International Workshop on Algorithmic Foundations of Robotics (WAFR), 2000.

[2] O. Aichholzer, F. Aurenhammer, P. Brass, and H. Krasser. Pseudo-triangulations from surfaces and a novel type of edge flip. *SIAM Journal on Computing*, 32:1621–1653, 2003.

[3] O. Aichholzer, F. Aurenhammer, P. Brass, and H. Krasser. Spatial embedding of pseudo-triangulations. In *Proc. 19^{th} Ann. ACM Symp. Computational Geometry*, volume 19, pages 144–153, San Diego, California, USA, 2003.

[4] O. Aichholzer, F. Aurenhammer, and H. Krasser. Adapting (pseudo)-triangulations with a near-linear number of edge flips. In *Lecture Notes in Computer Science 2748, Proc. 8th International Workshop on Algorithms and Data Structures (WADS)*, volume 2748, pages 12–24, 2003.

[5] O. Aichholzer, M. Hoffmann, B. Speckmann, and C. D. Tóth. Degree bounds for constrained pseudo-triangulations. In *Proc. 15th Annual Canadian Conference on Computational Geometry CCCG 2003*, pages 155–158, Halifax, Nova Scotia, Canada, 2003.

[6] O. Aichholzer, D. Orden, F. Santos, and B. Speckmann. On the number of pseudo-triangulations of certain point sets. In *Proc. 15th Annual Canadian Conference on Computational Geometry CCCG 2003*, pages 141–144, Halifax, Nova Scotia, Canada, 2003.

[7] O. Aichholzer, G. Rote, B. Speckmann, and I. Streinu. The zigzag path of a pseudo-triangulation. In *Lecture Notes in Computer Science 2748, Proc. 8th International Workshop on Algorithms and Data Structures (WADS)*, volume 2748, pages 377–389, 2003.

[8] Oswin Aichholzer, Franz Aurenhammer, Hannes Krasser, and Bettina Speckmann. Convexity minimizes pseudo-triangulations. In *Proc. 14th Canad. Conf. Comp. Geom.*, 2002. Full version to appear in *Computational Geometry: Theory and Applications*, 2004.

[9] Oswin Aichholzer, Erik D. Demaine, Jeff Erickson, Ferran Hurtado, Mark Overmars, Michael A. Soss, and Godfried T. Toussaint. Reconfiguring convex polygons. *Computational Geometry: Theory and Applications*, 20(1–2):85–95, October 2001.

[10] Vladimir I. Arnol´d. *Ordinary Differential Equations*. Springer Verlag, New York, 1992. Translated from Russian, 3rd edition 1984.

[11] Boris Aronov, Jacob E. Goodman, and Richard Pollack. Convexifying a planar polygon in $\mathcal{R}^3$. Manuscript, October 1999.

[12] L. Asimow and B. Roth. The rigidity of graphs. *Transactions of the American Mathematical Society*, 245:279–289, 1978.

[13] L. Asimow and B. Roth. The rigidity of graphs II. *Journal of Mathematical Analysis and Applications*, 68(1):171–190, 1979.

[14] Achim Bachem and Walter Kern. *Linear Programming Duality, an Introduction to Oriented Matroids*. Springer Verlag, 1992.

[15] Julien Basch, Jeff Erickson, Leonidas Guibas, John Hershberger, and Li Zhang. Kinetic collision detection between two simple polygons. In *Proc. 10th ACM-SIAM Symp. Discr. Algorithms (SODA)*, pages 102–111, 1999.

[16] Saugata Basu, Richard Pollack, and Marie-Francoise Roy. Computing roadmaps of semi-algebraic sets on a variety. *Journal AMS*, 13:55–82, 2000. Prelim. version in Proc. 28th ACM Symp. Theory of Computing, pp. 168-173, 1996.

[17] Saugata Basu, Richard Pollack, and Marie-Francoise Roy. *Algorithms in Real Algebraic Geometry*. Algorithms and Computation in Mathematics. Springer Verlag, New York, 2003.

[18] Sergei Bespamyatnikh. Enumerating pseudo-triangulations in the plane. In *Proc. 14th Canad. Conf. Comp. Geom.*, pages 162–166, 2002. To appear in *Computational Geometry: Theory and Applications*, 2004.

[19] Therese Biedl, Erik Demaine, Martin Demaine, Sylvain Lazard, Anna Lubiw, Joseph O'Rourke, Steve Robbins, Ileana Streinu, Godfried Toussaint, and Sue Whitesides. Locked and unlocked polygonal chains in three dimensions. *Discrete & Computational Geometry*, 26(3):269–281, October 2001.

[20] Therese Biedl, Erik Demaine, Martin Demaine, Sylvain Lazard, Anna Lubiw, Joseph O'Rourke, Steve Robbins, Ileana Streinu, Godfried Toussaint, and Sue Whitesides. On reconfiguring tree linkages: Trees can lock. *Discrete Applied Mathematics*, 117:293–297, 2002. Preliminary version appeared in *Proceedings of the 10th Canadian Conference on Computational Geometry*, abs:4-5, 1998.

[21] Anders Björner, Michel Las Vergnas, Bernd Sturmfels, Neil White, and Günter Ziegler. *Oriented Matroids*. Cambridge University Press, 1999.

[22] Jürgen Bokowski. Oriented matroids. In Peter Gruber and Jörg Wills, editors, *Handbook of Convex Geometry*, volume A, B, pages 555–602. North Holland, 1993.

[23] Jürgen Bokowski, Susanne Mock, and Ileana Streinu. On the Folkman-Lawrence topological representation theorem for oriented matroids in rank 3. *European J. of Combinatorics*, 22(5):601–615, 2001.

[24] Ciprian Borcea and Ileana Streinu. The number of embeddings of minimally rigid graphs. *Discrete and Computational Geometry*, 31:287–303, 2004. A preliminary version appeared in Proc. ACM Symp. Comput. Geometry, Barcelona, 2002.

[25] Hervé Brönnimann. The pseudo triangulation template toolkit. http://geometry.poly.edu/pstoolkit/, 2001.

[26] Hervé Brönnimann, Lutz Kettner, Michel Pocchiola, and Jack Snoeyink. Counting and enumerating pseudo-triangulations with the greedy flip algorithm. In *Proc. 13th Canad. Conf. Comp. Geom.*, 2001.

[27] John Canny. *Complexity of Robot Motion Planning.* MIT Press, Cambridge, MA, 1988.

[28] Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6(5):485–524, 1991.

[29] Roxana Cocan and Joseph O'Rourke. Polygonal chains cannot lock in 4d. *Discrete & Computational Geometry*, 20(3):105–129, November 2001.

[30] Robert Connelley, Erik Demaine, and Günter Rote. Straightening polygonal arcs and convexifying polygonal cycles. In *Proc. 41st Annual Symp. on Found. of Computer Science (FOCS), Redondo Beach, California*, pages 432–452, 2000. Journal version in *Discrete and Computational Geometry*, 30(5), 205-239, 2003.

[31] Robert Connelly, Erik D. Demaine, and Günter Rote. Infinitesimally locked self-touching linkages with applications to locked trees. In Jorge Calvo, Kenneth Millett, and Eric Rawdon, editors, *Physical Knots: Knotting, Linking, and Folding of Geometric Objects in* R^3, pages 287–311. American Mathematical Society, 2002. Collection of papers from the Special Session on Physical Knotting and Unknotting at the AMS Spring Western Section Meeting, Las Vegas, Nevada, April 21–22, 2001.

[32] Henri Crapo. Two-tree-three decompositions. unpublished manuscript, 1990.

[33] Henri Crapo and Walter Whiteley. Plane self-stresses and projected polyhedra I: the basic pattern. *Structural Topology*, 20:55–77, 1993.

[34] Paul Erdös. Problem number 3763. *American Mathematical Monthly*, 46(627), 1935.

[35] Israel Gel´fand, Mikhail Kapranov, and Andrei Zelevinsky. Newton polytopes of principal a-determinants. *Soviet Math. Doklady*, 40:278–281, 1990.

[36] H. Gluck. Almost all simply connected closed surfaces are rigid. In *Geometric Topology*, number 438 in Lecture Notes in Mathematics, pages 225–239. Springer Verlag, Berlin, 1975.

[37] Jacob E. Goodman and Richard Pollack. Allowable sequences and order types in discrete and computational geometry. In János Pach, editor, *New Trends in Discrete and Computational Geometry.* Springer Verlag, 1993.

[38] Jack Graver, Brigitte Servatius, and Herman Servatius. *Combinatorial Rigidity*. Graduate Studies in Mathematics vol. 2. American Mathematical Society, 1993.

[39] Leonidas Guibas and Jorje Stolfi. Primitives for the manipulation of general subdivisions and the computation of voronoi diagrams. *ACM Transactions on Graphics*, 4:74–123, 1985.

[40] Leonidas J. Guibas, J. Hershberger, D. Leven, Micha Sharir, and R. E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.

[41] Ruth Haas, David Orden, Günter Rote, Francisco Santos, Brigitte Servatius, Herman Servatius, Diane Souvaine, Ileana Streinu, and Walter Whiteley. Planar minimally rigid graphs and pseudo-triangulations. In *ACM Symp. Computational Geometry (SoCG)*, pages 154–163, San Diego, California, June 2003.

[42] L. Henneberg. *Die graphische Statik der starren Systeme*. Johnson Reprint 1968. Leipzig, 1911.

[43] John Hopcroft, Deborah Joseph, and Sue Whitesides. On the movement of robot arms in 2-dimensional bounded regions. *SIAM Journal of Computing*, 14:315–333, 1985.

[44] Deborah Joseph and W.H. Platinga. On the complexity of reachability and motion planning questions. In *Proceedings ACM Symposium on Computational Geometry (SoCG)*, pages 62–66, 1985.

[45] Micha Kapovich and John Millson. On the moduli spaces of polygons in the euclidean plane. *Journal of Differential Geometry*, 42(1):133–164, 1995.

[46] Lutz Kettner. Pseudo triangulation workbench. http: //www.cs.unc.edu/ Research/ compgeom/ pseudoT/, 2001.

[47] Lutz Kettner, David Kirkpatrick, Andrea Mantler, Jack Snoeyink, Bettina Speckmann, and Fumihiko Takeuchi. Tight degree bounds for pseudo-triangulations of points. *Comp. Geom. Theory. Applications*, 25:1–12, 2003.

[48] Rob Kirby. Problems in low-dimensional topology. `http://www.math.berkeley.edu/~kirby`, 1995.

[49] David Kirkpatrick, Jack Snoeyink, and Bettina Speckmann. Kinetic collision detection for simple polygons. *International Journal of Computational Geometry and Applications*, 12(1 and 2):3–27, 2002.

[50] David Kirkpatrick and Bettina Speckmann. Kinetic maintenance of context-sensitive hierarchical representations for disjoint simple polygons. In *Proc. 18th ACM Symposium on Computational Geometry (SoCG)*, pages 179–188, 2002.

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

[52] William J. Lenhart and Sue Whitesides. Reconfiguring closed polygonal chains in euclidian d-space. *Discrete & Computational Geometry*, 13:123–140, 1995.

[53] Laszlo Lovasz and Y. Yemini. On generic rigidity in the plane. *SIAM J. Algebraic and Discrete Methods*, 3(1):91–98, 1982.

[54] Anna Lubiw and Sue Whitesides, editors. *Workshop on Folding and Unfolding*, Barbados, 1998. McGill University Bellairs Institute. Informal proceedings.

[55] James Clerk Maxwell. On reciprocal figures and diagrams of forces. *Philosphical Magazine*, 27:250–261, 1864.

[56] James Clerk Maxwell. On reciprocal figures, frameworks and diagrams of forces. *Transactions of the Royal Society Edinburgh*, 26:1–40, 1870.

[57] James Clerk Maxwell. On bow's method of drawing diagrams in graphical statics, with illustrations from peaucellier's linkage. *Cambridge Phil. Soc. Proc.*, 2:407–414, 1876.

[58] R. James Milgram and Jeff C. Trinkle. The geometry of configuration spaces for closed chains in two and three dimensions. *Homology Homotopy and Applications*, 2003. to appear.

[59] David Orden, Guenter Rote, Francisco Santos, Brigitte Servatios, Herman Servatius, and Walter Whiteley. Non-crossing frameworks with non-crossing reciprocals. http://arxiv.org/abs/math.MG/0309156, 2003.

[60] David Orden and Francisco Santos. The polytope of non-crossing graphs on a planar point set. http://arxiv.org/abs/math.CO/0302126, February 2003.

[61] David Orden, Brigitte Servatius, Herman Servatius, and Francisco Santos. Combinatorial pseudo-triangulations. http://arxiv.org/abs/math.CO/0307370, July 2003.

[62] Michel Pocchiola and Gert Vegter. Computing the visibility graph via pseudo-triangulations. In *Proceedings of the* 11*th annual Symposium on Computational Geometry (SoCG)*, pages 248–257. ACM Press, 1995.

[63] Michel Pocchiola and Gert Vegter. Topologically sweeping visibility complexes via pseudo-triangulations. *Discrete & Computational Geometry*, 16(4):419–453, Dec. 1996.

[64] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, NY, 1985.

[65] Dana Randall, Günter Rote, Francisco Santos, and Jack Snoeyink. Counting triangulations and pseudo-triangulations of wheels. In *Proceedings 13th Canad. Conf. Comput. Geom.*, pages 149–152, 2001. http://compgeo.math.uwaterloo.ca/ cccg01/proceedings, citeseer.nj.nec.com/randall01counting.html.

[66] András Recski. *Matroid Theory and Its Applications.* Springer Verlag, 1989.

[67] Jürgen Richter-Gebert and Ulrich Kortenkamp. *Cinderella, an interactive software package for Geometry.* Springer Verlag, 1999.

[68] Günter Rote, Francisco Santos, and Ileana Streinu. Expansive motions and the polytope of pointed pseudo-triangulations. In Janos Pach Boris Aronov, Saugata Basu and Micha Sharir, editors, *Discrete and Computational Geometry - The Goodman-Pollack Festschrift*, Algorithms and Combinatorics, pages 699–736. Springer Verlag, Berlin, 2003.

[69] Günter Rote, Cao An Wang, Lusheng Wang, and Yinfeng Xu. On constrained minimum pseudo triangulations. In *Proceedings 9th Annual International Conference on Computing and Combinatorics, COCOON 2003, Big Sky, MT, USA, July 25-28, 2003*, volume 2697 of *Lecture Notes in Computer Science*, pages 445–454. Springer, 2003.

[70] Jack Schwartz and Micha Sharir. On the piano mover's problem II: general techniques for computing topological properties of real algebraic manifolds. *Adv. in Appl. Math.*, 4:298–351, 1983.

[71] Bettina Speckmann and Csaba Tóth. Allocating vertex π-guards in simple polygons via pseudo-triangulations. In *Proc. ACM-SIAM Symp. Discrete Algorithms (SODA)*, pages 109–118, 2003. To appear in *Discrete and Computational Geometry*, 2004.

[72] Ileana Streinu. Clusters of stars. In *Proceedings of the thirteenth annual Symposium on Computational Geometry*, pages 439–441. ACM Press, 1997.

[73] Ileana Streinu. A combinatorial approach to planar non-colliding robot arm motion planning. In *ACM/IEEE Symposium on Foundations of Computer Science*, pages 443–453, 2000.

[74] Ileana Streinu. Combinatorial pseudo-triangulations and partial oriented matroids. Draft, December 2003.

[75] Ileana Streinu. Combinatorial roadmaps in configuration spaces of simple planar polygons. In Saugata Basu and Laureano Gonzalez-Vega, editors, *Proceedings of the DIMACS Workshop on on Algorithmic and Quantitative Aspects of Real Algebraic Geometry in Mathematics and Computer Science*, pages 181–206. DIMACS, 2003.

[76] Ileana Streinu. Parallel redrawing pseudo triangulation mechanisms. Manuscript, 2003.

[77] Ileana Streinu and Walter Whiteley. Single-vertex origami and 3-dimensional expansive motions. To appear, Proc. Japan Conf. Discrete and Comp. Geometry, Tokai Univ., Oct. 2004.

[78] T.S. Tay and Walter Whiteley. Generating isostatic graphs. *Structural Topology*, 11:21–68, 1985.

[79] Godfried Toussaint. The erdös-nagy theorem and its ramifications. In *Proceedings of the Canadian Conference on Computational Geometry*, 1999.

[80] Jeff C. Trinkle and R. James Milgram. Complete path planning for closed kinematic chains with spherical joints. *International Journal of Robotics Research*, 21(9):773–789, Dec. 2002.

[81] Walter Whiteley. Motions and stresses of projected polyhedra. *Structural Topology*, 7:13–38, 1982.

[82] Walter Whiteley. Some matroids from discrete applied geometry. In J. Oxley J. Bonin and B. Servatius, editors, *Matroid Theory*, volume 197 of *Contemporary Mathematics*, pages 171–311. American Mathematical Society, 1996.

[83] Walter Whiteley. Rigidity and scene analysis. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 893–916. CRC Press, Boca Raton New York, first edition, 1997.

[84] Sue Whitesides. Algorithmic issues in the geometry of planar linkage movement. *Australian Computer Journal, Special Issue on Algorithms*, 24(2):42–50, 1992.