Locked and Unlocked Polygonal Chains in 3D*

T. Biedl† E. Demaine† M. Demaine† S. Lazard‡
A. Lubiw† J. O'Rourke§ M. Overmars¶ S. Robbins||
I. Streinu§ G. Toussaint|| S. Whitesides||

February 20, 2001

Abstract

In this paper, we study movements of simple polygonal chains in 3D. We say that an open, simple polygonal chain can be *straightened* if it can be continuously reconfigured to a straight sequence of segments in such a manner that both the length of each link and the simplicity of the chain are maintained throughout the movement. The analogous concept for closed chains is *convexification*: reconfiguration to a planar convex polygon. Chains that cannot be straightened or convexified are called *locked*. While there are open chains in 3D that are locked, we show that if an open chain has a simple orthogonal projection onto some plane, it can be straightened. For closed chains, we show that there are unknotted but locked closed chains, and we provide an algorithm for convexifying a planar simple polygon in 3D. All our algorithms require only $O(n)$ basic "moves" and run in linear time.

1 Introduction

A *polygonal chain* $P = (v_0, v_1, \ldots, v_{n-1})$ is a sequence of consecutively joined segments (or edges) $e_i = v_i v_{i+1}$ of fixed lengths $\ell_i = |e_i|$, embedded in space.[1] A chain is *closed* if the line segments are joined in cyclic fashion, i.e., if $v_{n-1} = v_0$; otherwise, it is *open*. A closed chain is also called a *polygon*. If the line segments are regarded as obstacles, then the chains must remain *simple* at all times, i.e., self intersection is not allowed. The edges of a simple chain

*This research was initiated at a workshop at the Bellairs Res. Inst. of McGill Univ., Jan. 31–Feb. 6, 1998. This is a revised and expanded version of [BDD+99]. Reseach supported in part by FCAR, NSERC, and NSF.

†University of Waterloo, Waterloo, Canada. {biedl, eddemaine, mldemaine, alubiw}@uwaterloo.ca.
‡INRIA Lorraine, France. lazard@loria.fr.
§Smith College, Northampton, USA. {orourke, streinu}@cs.smith.edu.
¶Utrecht University, The Netherlands. markov@cs.ruu.nl.
||McGill University, Montreal, Canada. {stever, godfried, sue}@cs.mcgill.ca.

[1] All index arithmetic throughout the paper is mod n.

are pairwise disjoint except for adjacent edges, which share the common endpoint between them. We will often use *chain* to abbreviate "simple polygonal chain." For an open chain our goal is to straighten it; for a closed chain the goal is to *convexify* it, i.e., to reconfigure it to a planar convex polygon. Both goals are to be achieved by continuous motions that maintain simplicity of the chain throughout, i.e., links are not permitted to intersect. A chain that cannot be straightened or convexified we call *locked*; otherwise the chain is *unlocked*. Note that a chain in 3D can be continuously moved between any of its unlocked configurations, for example via straightened or convexified intermediate configurations.

Basic questions concerning open and closed chains have proved surprisingly difficult. For example, the question of whether every planar, simple open chain can be straightened in the plane while maintaining simplicity has circulated in the computational geometry community for years, but remains open at this writing. Whether locked chains exist in dimensions $d \geq 4$ was only settled (negatively, in [CO99]) as a result of the open problem we posed in a preliminary version of this paper [BDD$^+$99]. In piecewise linear knot theory, complete classification of the 3D embeddings of closed chains with n edges has been found to be difficult, even for $n = 6$. These types of questions are basic to the study of embedding and reconfiguration of edge-weighted graphs, where the weight assigned to an edge specifies the desired distance between the vertices it joins. Graph embedding and reconfiguration problems, with or without a simplicity requirement, have arisen in many contexts, including molecular conformation, mechanical design, robotics, animation, rigidity theory, algebraic geometry, random walks, and knot theory.

We obtain several results for chains in 3D: open chains with a simple orthogonal projection, or embedded in the surface of a polytope, may be straightened (Sections 2 and 3); but there exist open and closed chains that are locked (Section 4). For closed chains initially taking the form of a polygon lying in a plane, it has long been known that they may be convexified in 3D, but only via a procedure that may require an unbounded number of moves. We provide an algorithm to perform the convexification (Section 5) in $O(n)$ moves.

Previous computational geometry research on the reconfiguration of chains (e.g., [Kan97], [vKSW96], [Whi92]) typically concerns planar chains with crossing links, moving in the presence of obstacles; [Sal73] and [LW95] reconfigure closed chains with crossing links in all dimensions $d \geq 2$. In contrast, throughout this paper we work in 3D and require that chains remain simple throughout their motions. Our algorithmic methods complement the algebraic and topological approaches to these problems, offering constructive proofs for topological results and raising computational, complexity, and algorithmic issues. Several open problems are listed in Section 6.

1.1 Background

Thinking about movements of polygonal chains goes back at least to A. Cauchy's 1813 theorem on the rigidity of polyhedra [Cro97, Ch. 6]. His proof employed a key lemma on opening angles at the joints of a planar convex open polygonal chain. This lemma, now known as Steinitz's Lemma (because E. Steinitz gave the first correct proof in the 1930's), is similar in spirit to our Lemma 5.5. Planar linkages, objects more general than polygonal chains in that a graph structure is permitted, have been studied intensively by mechanical engineers since at least Peaucellier's 1864 linkage. Because the goals of this linkage work are so different from ours, we could not find directly relevant results in the literature (e.g., [Hun78]). However, we have no doubt that simple results like our convexification of quadrilaterals (Lemma 5.2) are known to that community.

Work in algorithmic robotics is relevant. In particular, the Schwartz-Sharir cell decomposition approach [SS83] shows that all the problems we consider in this paper are decidable, and Canny's roadmap algorithm [Can87] leads to an algorithm singly-exponential in n, the number of vertices of the polygonal chain. Although hardness results are known for more general linkages [HJW84], we know of no nontrivial lower bounds for the problems discussed in this paper.

See, e.g., [HJW84], [Kor85], [CH88], or [Whi97] for other weighted graph embedding and reconfiguration problems.

1.2 Measuring Complexity

As usual, we compute the time and space complexity of our algorithms as a function of n, the number of vertices of the polygonal chain. This, however, will not be our focus, for it is of perhaps greater interest to measure the geometric complexity of a proposed reconfiguration of a chain. We first define what constitutes a "move" for these counting purposes.

Define a *joint movement* at v_i to be a monotonic rotation of e_i about an axis through v_i fixed with respect to a reference frame rigidly attached to some other edges of the chain. For example, a joint movement could feasibly be executed by a motor at v_i mounted in a frame attached to e_{i-1} and e_{i-2}. The axis might be moving in absolute space (due to other joint movements), but it must be fixed in the reference frame. Although more general movements could be explored, these will suffice for our purposes. A *monotonic rotation* does not stop or reverse direction. Note we ignore the angular velocity profile of a joint movement, which might not be appropriate in some applications. Our primary measure of complexity is a *move*: a reconfiguration of the chain P of n links to P' that may be composed of a constant number of simultaneous joint movements. Here the constant number should be independent of n, and is small (≤ 4) in our algorithms. All of our algorithms achieve reconfiguration in $O(n)$ moves. One of our open problems (Section 6) asks for exploration of a measure of the complexity of movements.

2 Open Chains with Simple Projections

This section considers an open polygonal chain P in 3D with a simple orthogonal projection P' onto a plane. Note that there is a polynomial-time algorithm to determine whether P admits such a projection, and to output a projection plane if it exists [BGRT96]. We choose our coordinate system so that the xy-plane Π_{xy} is parallel to this plane; we will refer to lines and planes parallel to the z-axis as "vertical." We will describe an algorithm that straightens P, working from one end of the chain. We use the notation $P[i,j]$ to represent the chain of edges $(v_i, v_{i+1}, \ldots, v_j)$, including v_i and v_j, and $P(i,j)$ to represent the chain without its endpoints: $P(i,j) = P[i,j] \setminus \{v_i, v_j\}$. Any object lying in plane Π_{xy} will be labelled with a prime.

Consider the projection $P' = (v'_0, v'_1, \ldots, v'_{n-1})$ on Π_{xy}. Let $r_i = \min_{j \notin \{i-1,i\}} d(v'_i, e'_j)$, where $d(v', e')$ is the minimum distance from vertex v' to a point on edge e'. Construct a disk of radius r_i around each vertex v'_i. The interior of each disk does not intersect any other vertex of P' and does not intersect any edges other than the two incident to v'_i: e'_{i-1} and e'_i; see Fig. 1.

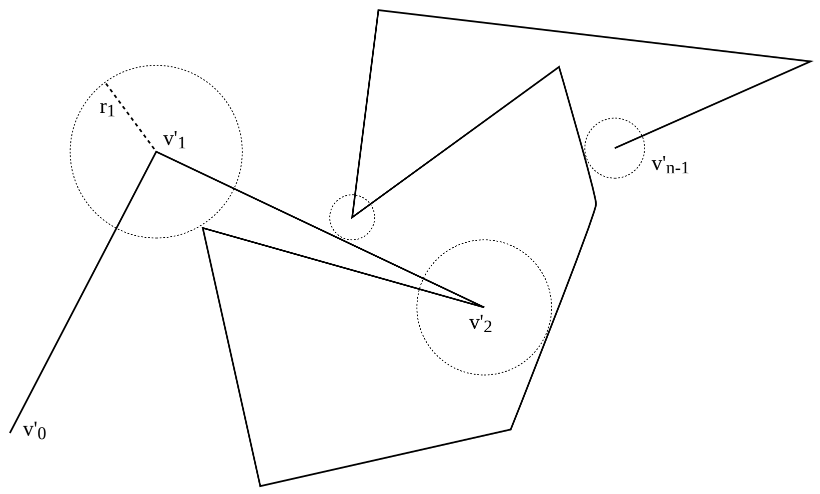


Figure 1: The projection P' of P. Each vertex v'_i is surrounded by an "empty" disk of radius r_i. Several such disks are shown.

We construct in 3D a vertical cylinder C_i centered on each vertex v_i of radius $r = \frac{1}{3}\min_i\{r_i\}$. This choice of r ensures that no two cylinders intersect one another (the choice of the fraction $\frac{1}{3} < \frac{1}{2}$ guarantees that cylinders do not even touch), and no edges of P, other than those incident to v_i, intersect C_i, for all i.

The straightening algorithm proceeds in two stages. In the first stage, the links are squeezed like an accordion into the cylinders, so that after step i all the links of $P_{i+1} = P[0, i+1]$ are packed into C_{i+1}. Let Π_i be the vertical plane containing e_i (and therefore e'_i). After the first stage, the chain is *monotone* in Π_i, i.e., it is monotone with respect to the line $\Pi_i \cap \Pi_{xy}$ in that the intersection of the chain with a vertical line in Π_i is either empty or a single point. In stage two, the chain is unraveled link by link into a straight line. The rest of this section describes the first stage. Let $\delta = r/n$.

2.1 Stage 1

We describe the Step 0 and the general Step i separately, although the former is a special case of the latter.

0. (a) Rotate e_0 about v_1, within Π_0, so that the projection of e_0 on Π_{xy} is contained in e'_0 throughout the motion. The direction of rotation is determined by the relative heights (z-coordinates) of v_0 and v_1. Thus if v_0 is at or above v_1, e_0 is rotated upwards (v_0 remains above v_1 during the rotation); see Fig. 2. If v_0 is lower than v_1, e_0 is rotated downwards (v_0 remains below v_1 during the rotation). The

rotation stops when v_0 lies within δ of the vertical line through v_1, i.e., when v_0 lies in the cylinder C_1 and is very close to its axis. The value of δ is chosen to be r/n so that in later steps more links can be accommodated in the cylinder. Again see Fig. 2.

(b) Now we rotate e_0 about the axis of C_1 away from e_1, until e'_0 and e'_1 are collinear (but not overlapping), i.e., until e_0 lies in the vertical plane Π_1.

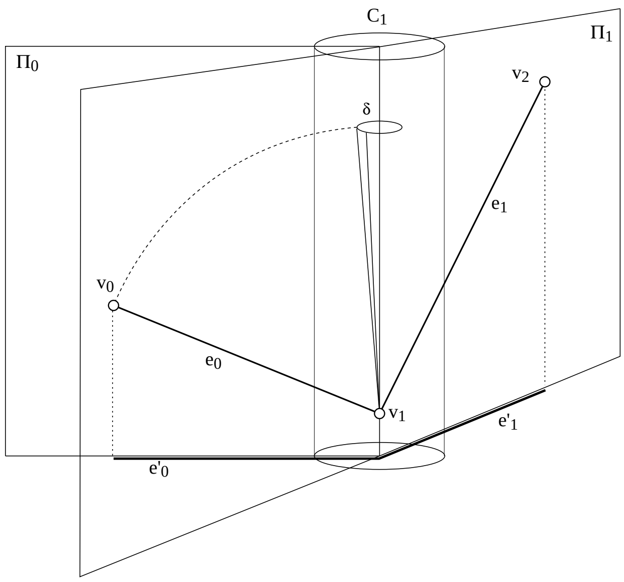


Figure 2: Step 0: (a) e_0 is first rotated within Π_0 into C_1, and then (b) rotated into the vertical plane Π_1 containing e_1.

After completion of Step 0, (v_0, v_1, v_2) forms a chain in Π_1 monotone with respect to the line $\Pi_1 \cap \Pi_{xy}$.

i. At the start of Step $i > 0$, we have a monotone chain $P_{i+1} = P[0, i+1]$ contained in the vertical plane Π_i through e_i, with $P_i = P[0, i]$ in C_i and v_0 within a distance of $i\delta$ of the axis of C_i.

(a) As in Step 0(a), rotate e_i within Π_i (in the direction that shortens the vertical projection of e_i) so that v_i lies within a distance δ of the axis of C_{i+1}. The difference now is that v_i is not the start of the chain, but rather is connected to the chain P_i. During the rotation of e_i we "drag" P_i along in such a way that only joints v_i and v_{i+1} rotate, keeping the joints $v_1, \ldots, v_{i-1}$ frozen. Furthermore, we constrain the motion of P_i (by appropriate rotation about joint v_i) so that it does not undergo a rotation. Thus at any instant of time during the rotation of e_i, the position of P_i remains within Π_i and is a translated copy of the initial P_i. See Fig. 3.

(b) Following Step 0(b), rotate P_{i+1} about the axis of C_{i+1} until e'_i and e'_{i+1} are coplanar.

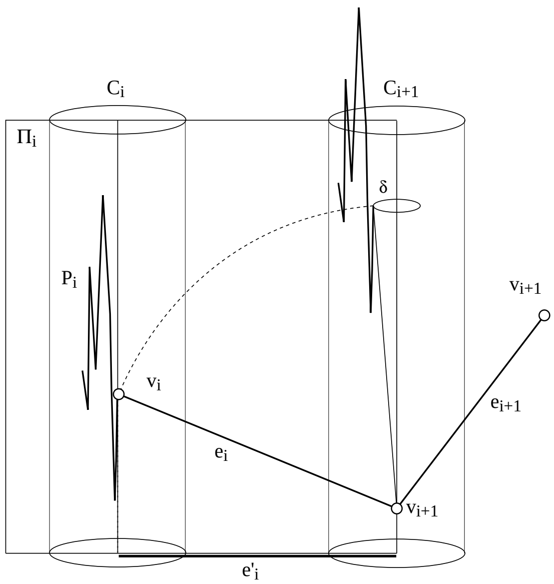


Figure 3: The chain P_i translates within Π_i.

At the completion of Step i we therefore have a chain $P_{i+2} = P[0, i+2]$ in the vertical plane Π_{i+1}, with P_{i+1} in C_{i+1} and v_0 within a distance of $(i+1)\delta$ of its axis. The chain is monotone in Π_{i+1} with respect to the line $\Pi_{i+1} \cap \Pi_{xy}$.

2.2 Stage 2

Now it is trivial to unfold this monotone chain by straightening one joint at a time, i.e., rotating each joint angle to π, starting at either end of the chain. We have therefore established the first claim of this theorem:

Theorem 2.1 *A polygonal chain of n links with a simple orthogonal projection may be straightened, in $O(n)$ moves, with an algorithm of $O(n)$ time and space complexity.*

Counting the number of moves is straightforward. Stage 1, Step i(a) requires one move: only joints v_i and v_{i+1} rotate. Step i(b) is again one move: only v_{i+1} rotates. So Stage 1 is completed in $2n$ moves. As Stage 2 takes $n-1$ moves, the whole procedure is accomplished with $O(n)$ moves.

Each move can be computed in constant time, so the time complexity is dominated by the computation of the cylinder radii r_i. These can be trivially computed in $O(n^2)$ time, by computing each vertex-vertex and vertex-edge distance. However, a more efficient computation is possible, based on the medial axis of a polygon, as follows. Given the projected chain P' in the plane (Fig. 4a), form two simple polygons P_1 and P_2, by doubling the chain from its endpoint v'_0 until the convex hull is reached (say at point x), and from there connecting along the line bisecting the hull angle at x to a large surrounding rectangle, and similarly connecting from v'_{n-1} to the hull to the rectangle. For P_1 close the polygon above P', and below for P_2. See Figs. 4bc. Note that $P_1 \cup P_2$ covers the rectangle, which, if chosen large, effectively covers the plane for the purposes of distance computation.

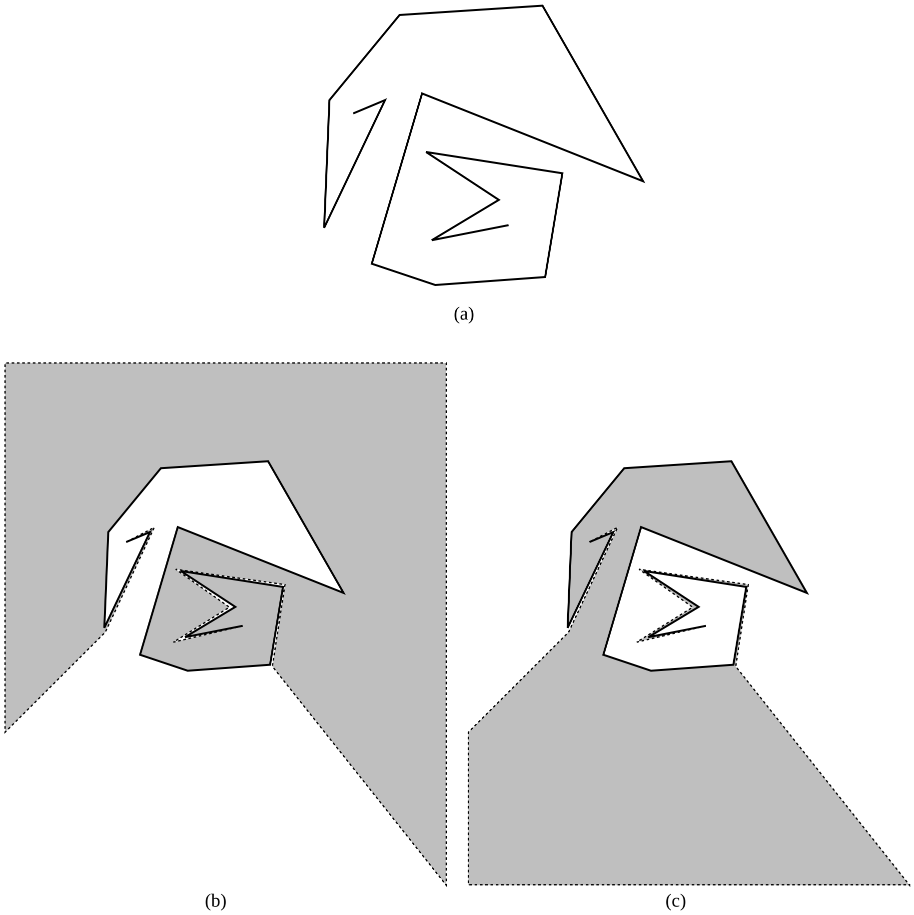


Figure 4: (a) Chain P'; (b) Polygon P_1; (c) Polygon P_2.

Compute the medial axis of P_1 and P_2 using a the linear-time algorithm of [CSW95]. The distances r_i can now be determined from the distance information in the medial axes. For a convex vertex v_i of P_k, its minimum "feature distance" can be found from axis information at the junction of the axis edge incident to v_i. For a reflex vertex, the information is with the associated axis parabolic arc. Because the bounding box is chosen to be large, no vertex's closest feature is part of the bounding box, and so must be part of the chain.

3 Open Chains on a Polytope

In this section we show that any open chain embedded on the surface of a convex polytope may be straightened. We start with a planar chain which we straighten in 3D.

Let P be an open chain in 2D, lying in Π_{xy}. It may be easily straightened by the following procedure. Rotate e_0 within Π_0 until it is vertical; now v_0 projects into v_1 on Π_{xy}. In general, rotate e_i within Π_i until v_i sits vertically above v_{i+1}. Throughout this motion, keep the previously straightened chain $P_i = P[0, i]$ above v_i in a vertical ray through v_i. This process clearly maintains simplicity throughout, as the projection at any stage is a subset of the original simple chain in Π_{xy}. In fact, this procedure can be seen as a special case of the algorithm described in the preceding section.

An easy generalization of this "pick-up into a vertical ray" idea permits straightening any open chain lying on the surface of a convex polytope $\mathcal{P}$. The same procedure is followed, except that the surface of $\mathcal{P}$ plays the role of Π_{xy}, and surface normals play the roles of vertical rays. When a vertex v_i of the polygonal chain P lies on an edge e between two faces f_1 and f_2 of $\mathcal{P}$, then the line containing P_i is rotated from R_1, the ray through v_i and normal to f_1, through an angle of measure $\pi - \delta(e)$, where $\delta(e)$ is the (interior) dihedral angle at e, to R_2, the ray through v_i and normal to f_2.

This algorithm uses $O(n)$ moves and can be executed in $O(n)$ time.

Note that it is possible to draw a polygonal chain on a polytope surface that has no simple projection. So this algorithm handles some cases not covered by Theorem 2.1. We believe that the sketched algorithm applies to a class of polyhedra wider than convex polytopes, but we will not pursue this further here.

4 Locked Chains

Having established that two classes of open chains may be straightened, we show in this section that not all open chains may be straightened, describing one locked open chain of five links (Section 4.1). A modification of this example establishes the same result for closed chains (Section 4.2). Both of these results were obtained independently by other researchers [CJ98]. Our proofs are, however, sufficiently different to be of independent interest.

4.1 A Locked Open Chain

Consider the chain $K = (v_0, \ldots, v_5)$ configured as in Fig. 5, where the standard knot theory convention is followed to denote "over" and "under" relations. Let $L = \ell_1 + \ell_2 + \ell_3$ be the total length of the short central links, and let ℓ_0 and ℓ_4 be both larger than L; in particular, choose $\ell_0 = L + \delta$ and $\ell_4 = 2L + \delta$ for $\delta > 0$. (One can think of this as composed of two rigid knitting needles, e_0 and e_4, connected by a flexible cord of length L.) Finally, center a ball B of radius $r = L + \epsilon$ on v_1, with $0 < 2\epsilon < \delta$. The two vertices v_0 and v_5 are exterior to B, while the other four are inside B. See Fig. 5.

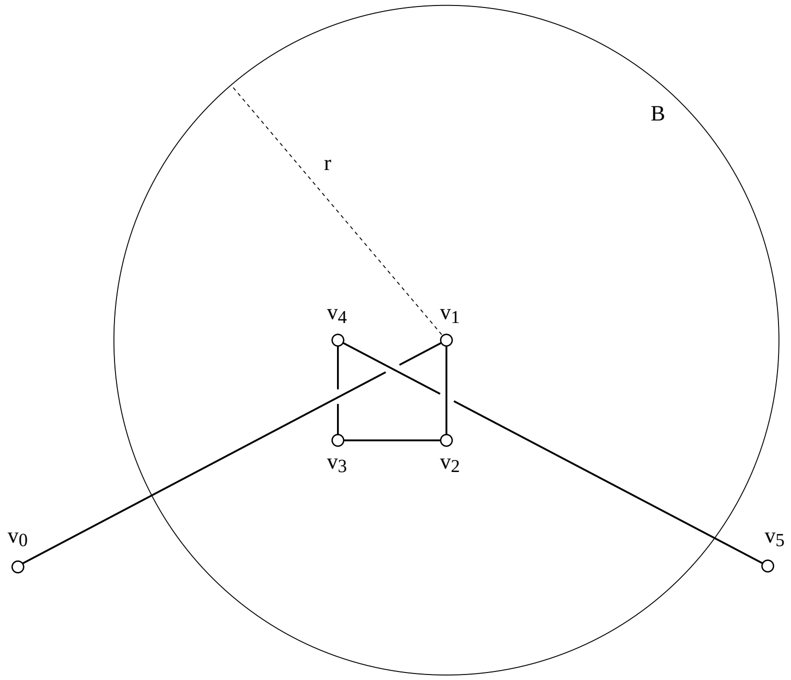


Figure 5: A locked open chain K ("knitting needles"). (The first and last edges e_0 and e_4 are longer than they appear in this view.)

Assume now that the chain K can be straightened by some motion. During the entire process, $\{v_1, v_2, v_3, v_4\} \subset B$ because $L < r$. Of course v_0 remains outside of B because $\ell_0 > r$. Now because $v_4 \in B$ and $\ell_4 = |v_4 v_5| = 2L + \delta$ is more than the diameter $2r = 2(L + \epsilon)$ of B, v_5 also remains exterior to B throughout the motion.

Before proceeding with the proof, we recall some terms from knot theory. The *trivial knot* is an unknotted closed curve homeomorphic to a circle. The *trefoil knot* is the simplest knot, the only knot that may be drawn with three crossings. See, e.g., [Liv93] or [Ada94].

Because of the constant separation between $\{v_0, v_5\}$ and $\{v_1, v_2, v_3, v_4\}$ by the boundary of B, we could have attached a sufficiently long unknotted string P' from v_0 to v_5 exterior to B that would not have hindered the unfolding of P. But this would imply that $K \cup P'$ is the trivial knot; but it is clearly a trefoil knot. We have reached a contradiction; therefore, K cannot be straightened.

4.2 A Locked, Unknotted Closed Chain

It is easy to obtain locked closed chains in 3D: simply tie the polygonal chain into a knot. Convexifying such a chain would transform it to the trivial knot, an impossibility. More interesting for our goals is whether there exists a locked, closed polygonal chain that is *unknotted*, i.e., whose topologically structure is that of the trivial knot.

We achieve this by "doubling" K: adding vertices v_i' near v_i for $i = 1, 2, 3, 4$, and connecting the whole into a chain $K^2 = (v_0, \ldots, v_5, v_4', \ldots, v_1')$. See Fig. 6. Because $K \subset K^2$, the

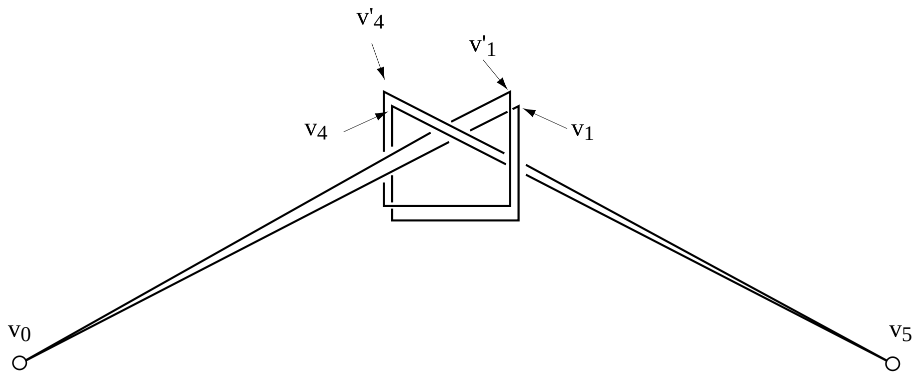


Figure 6: K^2 (K doubled): a locked but unknotted chain.

preceding argument applies when the second copy of K is ignored: any convexifying motion will have the property that v_0 and v_5 remain exterior to B, and $\{v_1, v_2, v_3, v_4\}$ remain interior to B throughout the motion. Thus the extra copy of K provides no additional freedom of motion to v_5 with respect to B. Consequently, we can argue as before: if K^2 is somehow convexified, this motion could be used to unknot $K \cup P'$, where P' is an unknotted chain exterior to B connecting v_0 to v_5. This is impossible, therefore K^2 is locked.

5 Convexifying a Planar Simple Polygon in 3D

An interesting open problem is to generalize our result from Section 2 to convexify a general closed chain. We show now that the special case of a closed chain lying in a plane, i.e., a planar simple polygon, may be convexified in 3D.

Such a polygon may be convexified in 3D by "flipping" out the reflex pockets, i.e., rotating the pocket chain into 3D and back down to the plane; see Fig. 7. This simple procedure was suggested by Erdős [Erd35] and proved to work by de Sz. Nagy [dSN39]. The number of flips, however, cannot be bound as a function of the number of vertices n of the polygon, as first proved by Joss and Shannon [Grü95]. See [Tou99] for the complex history of these results.

We offer a new algorithm for convexifying planar closed chains, which we call the "St. Louis Arch" algorithm. It is more complicated than flipping but uses a bounded number of moves, in fact $O(n)$ moves. It models the intuitive approach of picking up the polygon into 3D. We discretize this to lifting vertices one by one, accumulating the attached links into a convex "arch"[2] A in a vertical plane above the remaining polygonal chain; see Fig. 8. Although the algorithm is conceptually simple, some care is required to make it precise, and to then establish that simplicity is maintained throughout the motions.

[2] We call this the *St. Louis Arch Algorithm* because of the resemblance to the arch in St. Louis, Missouri.

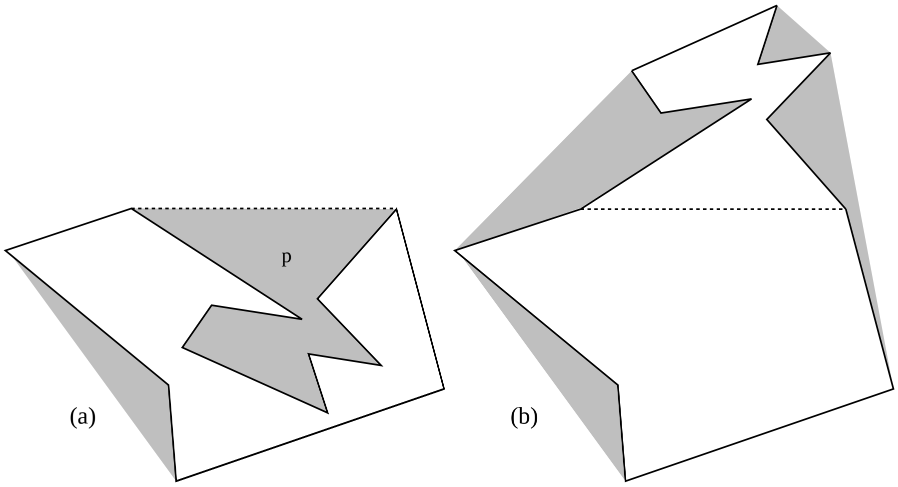


Figure 7: (a) A pocket p; (b) The polygon after flipping p.

Let P be a simple polygon in the xy-plane, Π_{xy}. Let Π_ϵ be the plane $z = \epsilon$ parallel to Π_{xy}, for $\epsilon > 0$; the value of ϵ will be specified later. We use this plane to convexify the arch safely above the portion of the polygon not yet picked up. We will use primes to indicate positions of moved (raised) vertices; unprimed labels refer to the original positions. After a generic step i of the algorithm, $P(0,i)$ has been lifted above Π_ϵ and convexified, v_0 and v_i have been raised to v'_0 and v'_i on Π_ϵ, and $P[i+1, n-1]$ remains in its original position on Π_{xy}. We first give a precise description of the conditions that hold after the ith step. Let $\Pi_z(v_i, v_j)$ be the (vertical) plane containing v_i and v_j, parallel to the z-axis.

H1: Π_ϵ splits the vertices of P into three sets: v'_0 and v'_i lie in Π_ϵ, $v'_1, \ldots, v'_{i-1}$ lie above the plane, and $v_{i+1}, \ldots, v_{n-1}$ lie below it.

H2: The arch $A = P(0,i)$ lies in the plane $\Pi_z(v'_0, v'_i)$, and is convex.

H3: v'_0 and v'_i project onto Π_{xy} within distance δ of their original positions v_0 and v_i. (Here, $\delta > 0$ is a constant that depends only on the input positions; it will be specified later.)

H4: Edges $v_{n-1}v'_0$ and $v'_i v_{i+1}$ connect between Π_{xy} and Π_ϵ.

H5: $P[i+1, n-1]$ remains in its original position in Π_{xy}.

See Fig. 8. A central aspect of the algorithm will be choosing ϵ small enough to guarantee a δ (see H3) that maintains simplicity throughout all movements.

The algorithm consists of an initialization step S0, followed by repetition of steps S1–S4.

5.1 S0

The algorithm is initialized at $i = 2$ by selecting an arbitrary (strictly) convex vertex v_1, and raising $\{v_0, v_1, v_2\}$ in four steps:

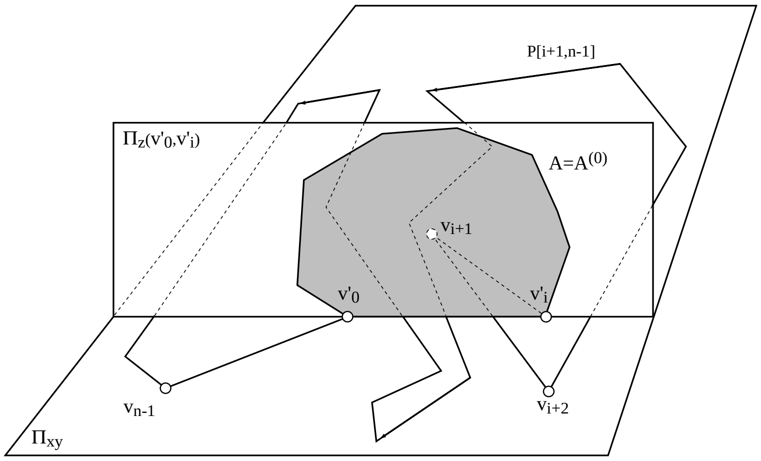


Figure 8: The arch A after the ith step, i.e., after "picking up" $P(0, i)$ into A. (The planes Π_{xy} and Π_ϵ are not distinguished in this figure, nor in Figs. 10 or 11.)

1. Rotate v_1 about the line through $v_0 v_2$ up to Π_ϵ. Call its new position v_1''.
2. Rotate v_0 about the line through $v_{n-1} v_1''$ up to Π_ϵ. Call its new position v_0'.
3. Rotate v_2 about the line through $v_1'' v_3$ up to Π_ϵ. Call its new position v_2'.
4. Rotate v_1'' about the line through $v_0' v_2'$ upwards until it lies in the plane $\Pi_z(v_0', v_2')$. Call its new position v_1'.

So long as the joint at v_1'' is not straight, the 4th step above is unproblematical, simply rotating a triangle from a horizontal to a vertical plane. That this joint does not become straight depends on ϵ and δ, and will be established under the discussion of S1 below. Ditto for establishing that the first three steps can be accomplished without causing self-intersection.

After completion of Step S0, the hypotheses H1–H5 are all satisfied. The remaining steps S1–S4 are repeated for each $i > 2$.

5.2 S1

The purpose of Step S1 is to lift v_i from Π_{xy} to Π_ϵ. This will be accomplished by a rotation of v_i about the line through v_{i-1}' and v_{i+1}, the same rotation used in substeps (2) and (3), and in a modified form in (1), of Step S0. Although this rotation is conceptually simple, it is this key movement that demands a value of ϵ to guarantee a δ that ensures correctness. The values of ϵ and δ will be computed directly from the initial geometric structure of P. Specifying the conditions on ϵ is one of the more delicate aspects of our argument, to which we now turn.

Let α_j be the smaller of the two (interior and exterior) angles at v_j. Also let $\beta_j = \pi - \alpha_j$, the deviation from straightness at joint v_j. We assume that P has no three consecutive collinear vertices. If a vertex is collinear with its two adjacent vertices, we freeze and eliminate that joint. So we may assume that $\beta_j > 0$ for all j.

5.2.1 Determination of δ

As in our earlier Figure 1, the simplicity of P guarantees "empty" disks around each vertex. Here we need disks to meet more stringent conditions than used in Section 2. Let $\delta > 0$ be such that:

1. Disks around each vertex v_j of radius δ include no other vertices of P, and only intersect the two edges incident to v_j.

2. A perturbed polygon, obtained by displacing the vertices within the disks (ignoring the fixed link lengths),

   (a) remains simple, and

   (b) has no straight vertices.

It should be clear that the simplicity of P together with $\beta_j > 0$ guarantees that such a $\delta > 0$ exists. As a technical aside, we sketch how δ could be computed. Finding a radius that satisfies condition (1) is easy. Half this radius guarantees the simplicity condition (2a), for this keeps a maximally displaced vertex separated from a maximally displaced edge. To prevent an angle β_j from reaching zero, condition (2b), displacements of the three points v_{j-1}, v_j, and v_{j+1} must be considered. Let $\ell = \min_j\{\ell_j\}$ be the length of the shortest edge, and let $\beta' = \min_j\{\beta_j\}$ be the minimum deviation from collinearity. Lemma A.1,which we prove in the Appendix, shows that choosing $\delta < \frac{1}{2}\ell\sin(\beta'/2)$ prevents straight vertices.

Let σ be the minimum separation $|v_j v_k|$ for all positions of v_j and v_k within their δ disks, for all j and k. Condition (2a) guarantees that $\sigma > 0$. Note that $\sigma \leq \ell$. Let β be the minimum of all β_j for all positions of v_j within their δ disks. Condition (2b) guarantees that $\beta > 0$. Our next task is to derive ϵ from σ, β, and δ. To this end, we must detail the "lifting" step of the algorithm.

5.2.2 S1 Lifting

Throughout the algorithm, v_0' remains fixed at the position on Π_ϵ it reached in Step S0. During the lifting step, v_{i-1}' also remains fixed, while v_i is lifted. Thus $v_0'v_{i-1}'$, the base of the arch A, remains fixed during the lifting, which permits us, by hypothesis H1, to safely ignore the arch during this step.

We now concentrate on the 2-link chain (v_{i-1}', v_i, v_{i+1}). By H5, $v_i v_{i+1}$ has not moved on Π_{xy}; by H3, v_{i-1}' has not moved horizontally more than δ from v_{i-1}. Let α_i' be the measure in $[0, \pi]$ of angle $\angle(v_{i-1}', v_i, v_{i+1})$, i.e., the angle at v_i measured in the slanted plane determined by the three points. Because $v_i v_{i+1}$ lie on Π_{xy} and v_{i-1}' is on Π_ϵ, $\alpha_i' \neq \pi$ and the chain (v_{i-1}', v_i, v_{i+1}) is kinked at the joint v_i.

Now imagine holding v_{i-1}' and v_{i+1} fixed. Then v_i is free to move on a circle C with center on $v_{i-1}'v_{i+1}$. See Fig. 9. This circle might lie partially below Π_{xy}, and is tilted from the vertical (because v_{i-1}' lies on Π_ϵ). The lifting step consists simply in rotating v_i on C upward until it lies on Π_ϵ; its position there we call v_i'.

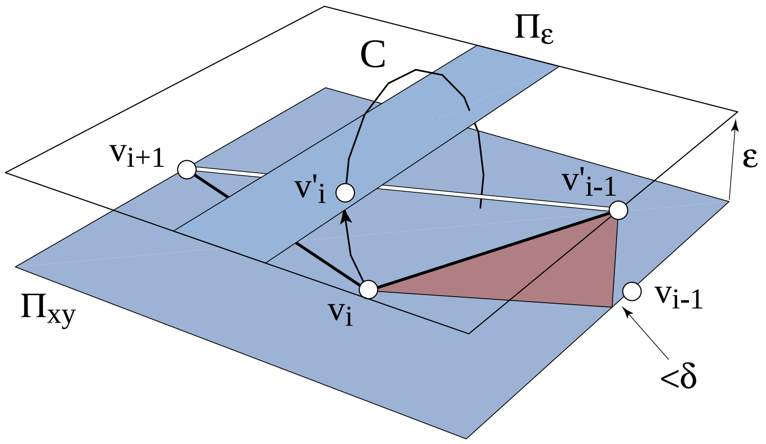


Figure 9: v_i rotates up the circle C until it hits Π_ϵ.

5.2.3 Determination of ϵ

We now choose $\epsilon > 0$ so that two conditions are satisfied:

1. The highest point of C is above Π_ϵ (so that v_i can reach Π_ϵ).

2. v_i' projects no more than δ away from v_i (to satisfy H3).

It should be clear that both goals may be achieved by choosing ϵ small enough. We sketch a computation of ϵ in the Appendix.

The computation of ϵ ultimately depends solely on σ and β—the shortest vertex separation and the smallest deviation from straightness—because these determine δ, and then r and δ_1 and δ_2 and ϵ. Although we have described the computation within Step S1, in fact it is performed prior to starting any movements; and ϵ remains fixed throughout.

As we mentioned earlier, two of the three lifting rotations used in Step S0 match the lifting just detailed. The exception is the first lifting, of v_1 to v_1' in Step S0. This only differs in that the cone axis v_0v_2 lies on Π_{xy} rather than connecting Π_{xy} to Π_ϵ. But it should be clear this only changes the above computation in that the tilt angle ψ is zero, which only improves the inequalities. Thus the ϵ computed for the general situation already suffices for this special case.

5.2.4 Collinearity

We mention here, for reference in the following steps, that it is possible that v_i' might be collinear with v_0' and v_{i-1}' on Π_ϵ. There are two possible orderings of these three vertices along a line:

1. (v_0', v_i', v_{i-1}').

2. (v_0', v_{i-1}', v_i').

The ordering (v_i', v_0', v_{i-1}') is not possible because that would violate the simplicity condition 2(a), as all three vertices project to within δ of their original positions on Π_{xy}, and no vertex comes within δ of an edge.

Despite this possible degeneracy, we will refer to "the triangle $\triangle v_0' v_{i-1}' v_i'$," with the understanding that it may be degenerate. This possibility will be dealt with in Lemma 5.6.

We now turn to the remaining three steps of the algorithm for iteration i. We use the notation $A^{(k)}$ to represent the arch $A = A^{(0)}$ at various stages of its processing, incrementing k whenever the shape of the arch might change.

5.3 S2

After the completion of Step S1, $v_{i-1}' v_i'$ lies in Π_ϵ. We now rotate the arch $A^{(0)}$ into the plane Π_ϵ, rotating about its base $v_0' v_{i-1}'$, away from $v_{i-1}' v_i'$. This guarantees that $A^{(1)} = A^{(0)} \cup \triangle v_0' v_{i-1}' v_i'$ is a planar weakly-simple polygon. Moreover, while $\triangle v_0' v_{i-1}' v_i'$ may be degenerate, the chain $(v_0', \ldots, v_i')$ lies strictly to one side of the line through (v_0', v_{i-1}') and so is simple. See Fig. 10.

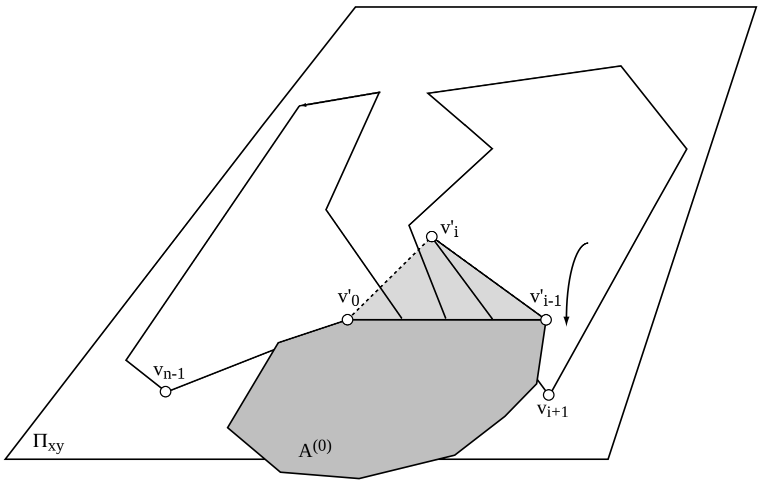


Figure 10: $A^{(1)} = A^{(0)} \cup \triangle v_0' v_{i-1}' v_i'$ lies in the plane Π_ϵ just slightly above Π_{xy}.

5.4 S3

Now that $A^{(1)}$ lies in its "own" plane Π_ϵ, it may be convexified without worry about intersections with the remaining polygon $P[i+1, n-1]$ in Π_{xy}. The polygon $A^{(1)}$ is a "barbed polygon": one that is a union of a convex polygon $(A^{(0)})$ and a triangle $(\triangle v_0' v_{i-1}' v_i')$. We establish in Theorem 5.7 that $A^{(1)}$ may be convexified in such a way that neither v_0' nor v_i' move, and v_0' and v_i' end up strictly convex vertices of the resulting convex polygon $A^{(2)}$.

5.5 S4

We next rotate $A^{(2)}$ up into the vertical plane $\Pi_z(v_0', v_i')$. Because of strict convexity at v_0' and v_i', the arch stays above Π_ϵ. See Fig. 11.

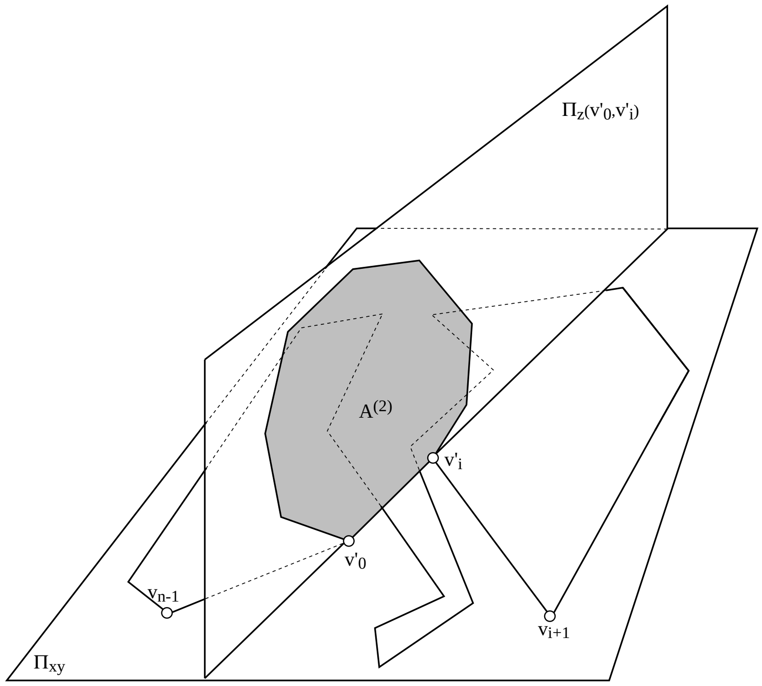


Figure 11: $A^{(2)}$, which has incorporated the edge $v_{i-1}v_i'$ of P, is rotated up into the plane $\Pi_z(v_0', v_i')$.

We have now reestablished the induction hypothesis conditions H1–H5. After the penultimate step, for $i = n-2$, only v_{n-1} lies on Π_{xy}, and the final execution of the lifting Step S1 rotates v_{n-1} about $v_0'v_{n-2}'$ to raise it to Π_ϵ. A final execution of Steps S1 and S2 yields a convex polygon. Thus, assuming Theorem 5.7 in Section 5.7 below, we have established the correctness of the algorithm:

Theorem 5.1 *The "St. Louis Arch" Algorithm convexifies a planar simple polygon of n vertices.*

We will analyze its complexity in Section 5.8.

We now return to Step S3, convexifying a barbed polygon. We perform the convexification entirely within the plane Π_ϵ. We found two strategies for this task. One maintains A as a convex quadrilateral, and the goal of Step S3 can be achieved by convexifying the (nonconvex) pentagon $A^{(1)}$, and then reducing it to a convex quadrilateral. Although this approach is possible, we found it somewhat easier to leave A as a convex $(i+1)$-gon, and prove that $A^{(1)} = A^{(0)} \cup \triangle v_0'v_{i-1}'v_i'$ can be convexified. This is the strategy we pursue in the next two sections. Section 5.6 concentrates on the base case, convexifying a quadrilateral, and Section 5.7 achieves Theorem 5.7, the final piece needed to complete Step S3.

5.6 Convexifying Quadrilaterals

It will come as no surprise that every planar, nonconvex quadrilateral can be convexified. Indeed, recent work has shown that any star-shaped polygon may be convexified [ELR+98a], and this implies the result for quadrilaterals. However, because we need several variations on basic quadrilateral convexification, we choose to develop our results independently, although relegating some details to the Appendix.

Let $Q = (v_0, v_1, v_2, v_3)$ be a weakly simple, nonconvex quadrilateral, with v_2 the reflex vertex. By *weakly simple* we mean that either Q is simple, or v_2 lies in the relative interior of one of the edges incident to v_0 (i.e., no two of Q's edges properly cross). This latter violation of simplicity is permitted so that we can handle a collapsed triangle inherited from step S1 of the arch algorithm (Section 5.2.4).

As before, let α_i be the smaller of the two (interior and exterior) angles at v_i. Call a joint v_i *straightened* if $\alpha_i = \pi$, and *collapsed* if $\alpha_i = 0$. All motions throughout this (5.6) and the next section (5.7) are in 2D.

We will convexify Q with one motion M, whose intuition is as follows; see Fig. 12. Think of the two links adjacent to the reflex vertex v_2 as constituting a rope. M then opens the joint at v_0 until the rope becomes taut. Because the rope is shorter than the sum of the lengths of the other two links, it becomes taut prior to any other "event."

Any motion M that transforms a shape such as Q can take on rather different appearances when different parts of Q are fixed in the plane, providing different frames of reference for the motion. Although all such fixings represent the same intrinsic shape transformation M, when convenient we distinguish two fixings: M_{02}, which fixes the line L containing v_0v_2, and M_{03}, which fixes the line containing v_0v_3.

The convexification motion M is easiest to see when viewed as motion M_{02}. Here the two 2-link chain (v_0, v_1, v_2) and (v_0, v_3, v_2) perform a *line-tracking* motion [LW92]: fix v_0, and move v_2 away from v_0 along the fixed directed line L containing v_0v_2, until v_2 straightens.

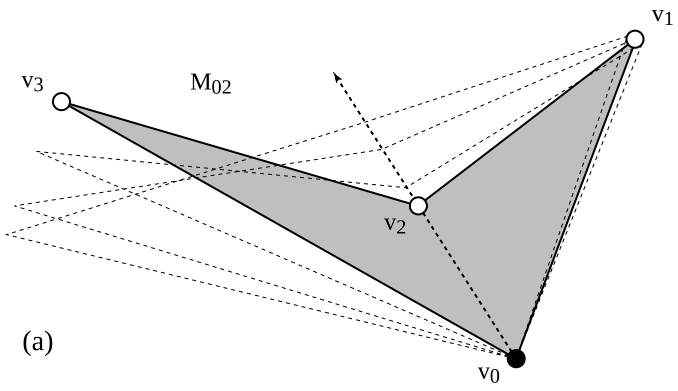
Lemma 5.2 *A weakly simple quadrilateral Q nonconvex at v_2 may be convexified by motion M_{02}, which straightens the reflex joint v_2, thereby converting Q to a triangle T. Throughout the motion, all four angles α_i increase only, and remain within $(0, \pi)$ until $\alpha_2 = \pi$.* See Fig. 12a.

Although this lemma is intuitively obvious, and implicit in work on linkages (e.g., [GN86]), we have not found an explicit statement of it in the literature, and we therefore present a proof in the Appendix (Lemma A.3).

We note that the same motion convexifies a degenerate quadrilateral, where the triangle $\triangle v_0v_1v_2$ has zero area with v_2 lying on the edge v_0v_1. See Fig. 13. As long as we open α_2 in the direction, as illustrated, that makes the quadrilateral simple, the proof of Lemma 5.2 carries through.

The motion M_{02} used in Lemma 5.2 is equivalent to the motion M_{03} obtained by fixing v_0v_3 and opening α_0 by rotating v_1 clockwise (cw) around the circle of radius ℓ_0 centered on v_0. Throughout this motion, the polygon stays right of the fixed edge v_0v_3. See Fig. 12b. This yields the following easy corollary of Lemma 5.2:

Lemma 5.3 *Let $P = Q \cup P'$ be a polygon obtained by gluing edge v_0v_3 of a weakly simple quadrilateral Q nonconvex at v_2, to an equal-length edge of a convex polygon P', such that Q*



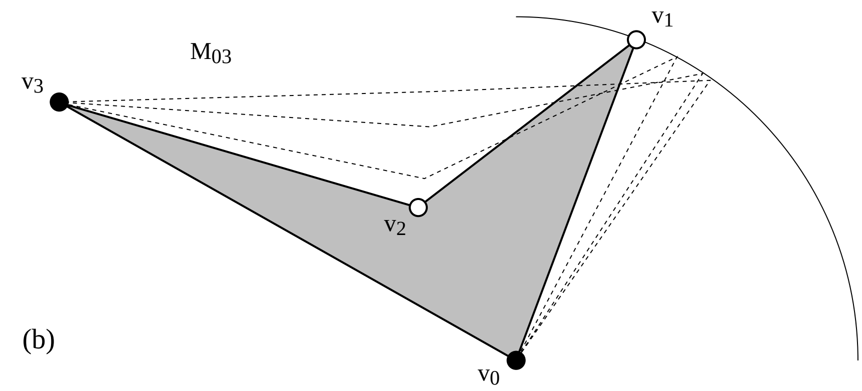


Figure 12: (a) Convexifying a quadrilateral by M_{02}: moving v_2 out the v_0v_2 diagonal; (b) The same motion viewed as M_{03}: opening α_0 with v_0v_3 fixed.

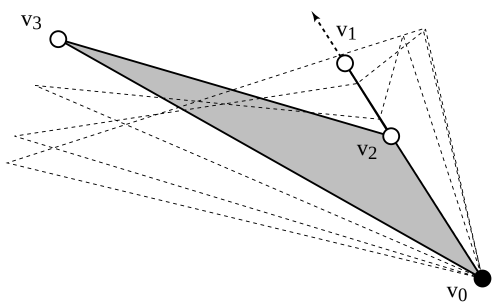


Figure 13: Motion M_{02} also convexifies a weakly simple quadrilateral.

and P' are on opposite sides of the diagonal v_0v_3. Then applying the motion M_{03} to Q while keeping P' fixed, maintains simplicity of P throughout.

5.6.1 Strict Convexity

Motion M converts a nonconvex quadrilateral into a triangle, but we will need to convert it to a strictly convex quadrilateral. This can always be achieved by continuing M_{02} beyond the straightening of α_2.

Lemma 5.4 *Let $Q = (v_0, v_1, v_2, v_3)$ be a quadrilateral, with (v_1, v_2, v_3) collinear so that $\alpha_2 = \pi$, and such that $\triangle v_0v_1v_3$ is nondegenerate. As in Lemma 5.3, let $P = Q \cup P'$ be a convex polygon obtained by gluing P' to edge v_0v_3 of Q, with v_0 and v_3 strictly convex vertices of P. The motion M_{02} (moving v_2 along the line determined by v_0v_2) transforms Q to a strictly convex quadrilateral Q' such that $Q' \cup P'$ remains a convex polygon* (See Fig. 14.)

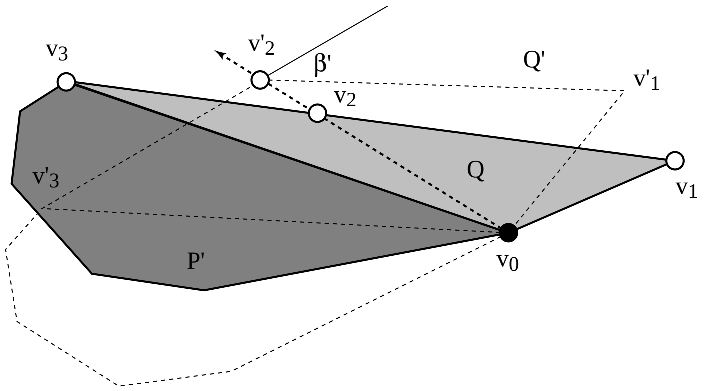


Figure 14: Converting Q to the strictly convex quadrilateral Q' via M_{02}. Attachment P' is carried along rigidly.

Proof: Because v_0 and v_3 are strictly convex vertices, and v_1 must be strictly convex because Q is a nondegenerate triangle, all the interior angles at these vertices are bounded away from π. By assumption, they are also bounded away from 0. Thus there is some freedom of motion for v_2 along the line determined by v_0v_2 before the next event, when one of these angles reaches 0 or π. $\square$

A lower bound on $\beta' = \pi - \alpha_2'$, the amount that v_2 can be bent before an event is reached, could be computed explicitly in $O(1)$ time from the local geometry of $Q \cup P'$, but we will not do so here.

5.7 Convexifying Barbed Polygons

Call a polygon *barbed* if removal of one ear $\triangle abc$ leaves a convex polygon P'. $\triangle abc$ is called the *barb* of P. Note that either or both of vertices a and c may be reflex vertices of P. In order to permit $\triangle abc$ to be degenerate (of zero area), we extend the definition as follows.

A weakly simple polygon (Section 5.6, Figure 13) is *barbed* if, for three consecutive vertices a, b, c, deletion of b (i.e., removal of the possibly degenerate $\triangle abc$) leaves a simple convex polygon P'. Note this definition only permits weak simplicity at the barb $\triangle abc$.

The following lemma (for simple barbed polygons) is implicit in [Sal73], and explicit (for star-shaped polygons, which includes barbed polygons) in [ELR+98b], but we will need to subsequently extend it, so we provide our own proof.

Lemma 5.5 *A weakly simple barbed polygon may be convexified, with $O(n)$ moves.*
Proof: Let $P = (v_0, v_1, \ldots, v_{n-1})$, with $\triangle v_0 v_{n-2} v_{n-1}$ the barb. See Fig. 15.

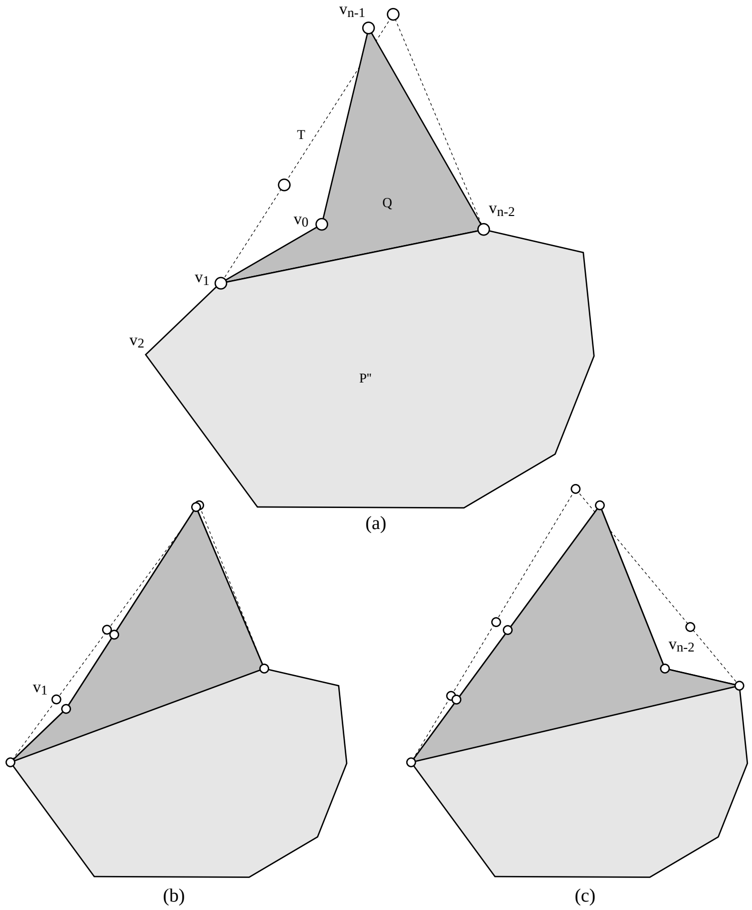


Figure 15: (a) A barbed polygon with barb $\triangle v_0 v_{n-2} v_{n-1}$. The nonconvex quadrilateral Q is transformed to T, resulting in a new barbed polygon $T \cup P''$. (b) and (c) show the remaining convexification steps.

The proof is by induction. Lemma 5.2 establishes the base case, $n = 4$, for every quadrilateral is a barbed polygon. So assume the theorem holds for all barbed polygons of up to $n - 1$ vertices.

If both v_0 and v_{n-2} are convex, P is already convex and we are finished. So assume that P is nonconvex, and without loss of generality let v_0 be reflex in P. It must be that $v_1 v_{n-2}$ is a diagonal, as it lies within the convex portion of P. Let $Q = (v_0, v_1, v_{n-2}, v_{n-1})$

be the quadrilateral cut off by diagonal $v_1 v_{n-2}$, and let $P'' = (v_1, \ldots, v_{n-2})$ be the remaining portion of P, so that $P = Q \cup P''$. Q is nonconvex at v_0.

Lemma 5.3 shows that motion M (appropriately relabeled) may be applied to convert Q to a triangle T by straightening v_0, leaving P'' unaffected. At the end of this motion, we have reduced P to a polygon P' of one fewer vertex. Now note that T is a barb for P' (because P'' is convex): $P' = T \cup P''$. Apply the induction hypothesis to P'. The result is a convexification of P.

Each reduction uses one move M, and so $O(n)$ moves suffice for P. $\square$

Note that although each step of the convexification straightens one reflex vertex, it may also introduce a reflexivity: v_1 is convex in Fig. 15a but reflex in Fig. 15b. We could make the procedure more efficient by "freezing" any joint as soon as it straightens, but it suffices for our analysis to freeze each straightened reflex vertex, thenceforth treating the segment on which it lies as a single rigid link.

As is evident in Fig. 15c, the convexification leaves a polygon with several vertices straightened. One of the edges e of the barbed polygon is the base of the arch A from Section 5.2.2. If either of e's endpoints are straightened, then part of the arch will lie directly in the plane Π_e, and could cause a simplicity violation during the S1 lifting step. Therefore we must ensure that both of e's endpoints are strictly convex:

Lemma 5.6 *Any convex polygon with a distinguished edge e can be reconfigured so that that both endpoints of e become strictly convex vertices.*

Proof: Suppose the counterclockwise endpoint v_2 of e has internal angle $\alpha = \pi$; see Fig. 16. Let v_1 be the next strictly convex vertex in clockwise order before v_2 (it may be that v_1 is the other endpoint of e), and v_3, v_0 be the next two strictly convex vertices adjacent to v_2 counterclockwise. Let $Q = (v_0, v_1, v_2, v_3)$. Then apply Lemma 5.4 to Q to convexify v_2 via

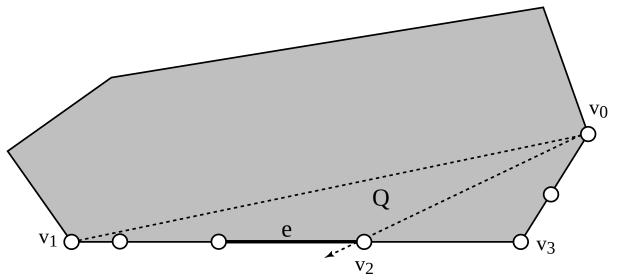


Figure 16: Making one endpoint of e strictly convex.

motion M_{02}. Apply the same procedure to the other endpoint of e if necessary. $\square$

Using Lemma 5.5 to convexify the barbed polygon arch, and Lemma 5.6 to make its base endpoints strictly convex, yields:

Theorem 5.7 *A weakly simple barbed polygon may be convexified in such a way that the endpoints of a distinguished edge are strictly convex.*

This completes the description of the St. Louis Arch Algorithm, as $A^{(1)} = A^{(0)} \cup \triangle v_0' v_{i-1}' v_i'$ is a barbed polygon, and Step S4 may proceed because of the strict convexity at the arch base endpoints.

5.8 Complexity of St. Louis Arch Algorithm

It is not difficult to see that only a constant number of moves are used in steps S0, S1, S2, and S4. Step S3 is the only exception, which we have seen in Lemma 5.5 can be executed in $O(n)$ moves. So the resulting procedure can be accomplished in $O(n^2)$ moves. The algorithm actually only uses $O(n)$ moves, as the following amortization argument shows:

Lemma 5.8 *The St. Louis Arch Algorithm runs in $O(n)$ time and uses $O(n)$ moves.*

Proof: Each barb convexification move used in the proof of Lemma 5.5 constitutes a single move according to the definition in Section 1.2, as four joints open monotonically (cf. Lemma 5.2). Each such convexification move necessarily straightens one reflex joint, which is subsequently "frozen." The number of such freezings is at most n over the life of the algorithm. So although any one barbed polygon might require $\Omega(n)$ moves to convexify, the convexifications over all n steps of the algorithm uses only $O(n)$ moves. Making the base endpoint angles strictly convex requires at most two moves per step, again $O(n)$ overall.

Each step of the algorithm can be executed in constant time, leading to a time complexity of $O(n)$. Again we must consider computation of the minimum distances around each vertex to obtain δ (Section 5.2.1), but we can employ the same medial axis technique used in Section 2 to compute these distances in $O(n)$ time. □

Note that at most four joints rotate at any one time, in the barb convexification step.

6 Open problems

Although we have mapped out some basic distinctions between locked and unlocked chains in three dimensions, our results leave many aspects unresolved:

1. What is the complexity of deciding whether a chain in 3D can be unfolded?

2. Theorem 2.1 only covers chains with simple orthogonal projections. Extension to perspective (central) projections, or other types of projection, seems possible.

3. Can a closed chain with a simple projection always be convexified? None of the algorithms presented in this paper seem to settle this case.

4. Find unfolding algorithms that minimize the number of simultaneous joint rotations. Our quadrilateral convexification procedure, for example, moves four joints at once, whereas pocket flipping moves only two at once.

5. Can an open chain of unit-length links lock in 3D? Cantarella and Johnson show in [CJ98] that the answer is NO if $n \leq 5$.

Acknowledgements

We thank W. Lenhart for co-suggesting the knitting needles example in Fig. 5, J. Erickson for the amortization argument that reduced the time complexity in Lemma 5.8 to $O(n)$, and H. Everett for useful comments.

A Appendix

A.1 Computation of ϵ

Here we detail a possible computation of ϵ, as needed in Section 5.2.3.

The smallest radius r for the circle C is determined by the minimum angle β (the smallest deviation from straightness) and the shortest edge length ℓ. In particular, $r \geq \ell \sin(\beta/2)$; see Figure 17a,b. Here it is safe to use the β from the plane Π_{xy} because the deviation from straightness is only larger in the tilted plane of $\triangle v_{i+1}, v_i, v'_{i-1}$ (cf. Fig. 9), and we seek a lower bound on r.

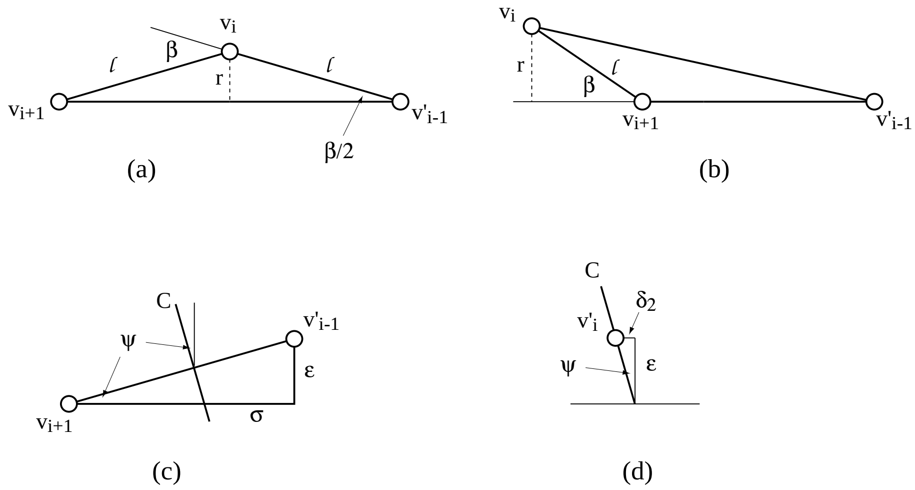


Figure 17: Determination of smallest circle radius r: (a) $r \geq \ell \sin(\beta/2)$; (b) $r \geq \ell \sin \beta$. So $\ell \sin(\beta/2)$ is a lower bound. Determination of largest circle tilt ψ: (c) $\cos \psi \geq \sigma/\sqrt{\sigma^2 + \epsilon^2}$. (d) Determination of δ_2: $\delta_2 \leq \epsilon \tan \psi$.

The tilt ψ of the circle leaves the top of C at least at height $r \cos \psi$. Because $|v'_{i-1} v_{i+1}| \geq \sigma$, the tilt angle must satisfy $\cos \psi \geq \sigma/\sqrt{\sigma^2 + \epsilon^2}$; see Figure 17c. Thus to meet condition (1), we should arrange that

$$\ell \sin(\beta/2) \frac{\sigma}{\sqrt{\sigma^2 + \epsilon^2}} > \epsilon$$

which can clearly be achieved by choosing ϵ small enough, as ℓ, β, and σ are all constants fixed by the geometry of P.

Turning to condition (2) of Section 5.2.3, the movement of v'_i with respect to v_i can be decomposed into two components. The first is determined by the rotation along C if that circle were vertical. This deviation is no more than $\delta_1 = r(1 - \cos \phi)$, where ϕ is the lifting rotation angle measured at the cone axis $v'_{i-1} v_{i+1}$. Because $\sin \phi \leq \epsilon/r$, this leads to $\delta_1 \leq r \left[1 - \sqrt{1 - (\epsilon/r)^2}\right]$. The second component is due to the tilt of the circle, which is $\delta_2 = \epsilon \tan \psi \leq \epsilon^2/\sigma$; see Figure 17d. The total displacement is no more than $\delta_1 + \delta_2$. Now it is clear that as $\epsilon \to 0$, both $\delta_1 \to 0$ and $\delta_2 \to 0$. Thus for any given δ, we may choose ϵ such that $\delta_1 + \delta_2 < \delta$.

A.2 Straightening Lemma

The following lemma is used to determine δ in Section 5.2.1.

Lemma A.1 *Let ABC be a triangle, with $|AB| \geq \ell$, $|BC| \geq \ell$, and $\beta \leq \angle ABC \leq \pi - \beta$. Then for any triangle $A'B'C'$ whose vertices are displaced at most δ from those of $\triangle ABC$, i.e.,*

$$|AA'| < \delta, \;\; |BB'| < \delta, \;\; |CC'| < \delta,$$

$\angle A'B'C' < \pi$.

Proof: Let a be the point on BA a distance $\ell/2$ from B, and let c be the point on BC a distance $\ell/2$ from B. Let L be the line containing ac. Set $\theta = \angle Bac = \angle acB$, and $\phi = \angle aBc = \pi - 2\theta$. Because $\phi = \angle ABC$, the assumptions of the lemma give $\beta \leq \pi - 2\theta \leq \pi - \beta$, or $\beta/2 \leq \theta \leq (\pi - \beta)/2$. The distance $d(A, L)$ from A to L satisfies

$$\begin{aligned} d(A, L) &\geq (\ell/2)\sin\theta && (1)\\ &\geq (\ell/2)\sin(\beta/2) && (2)\\ &> \delta\,. && (3) \end{aligned}$$

The exact same inequality hold for the distances $d(B, L)$ and $d(C, L)$, because the relevant angle is θ is each case, and the relevant hypotenuse is $\geq \ell/2$ in each case.

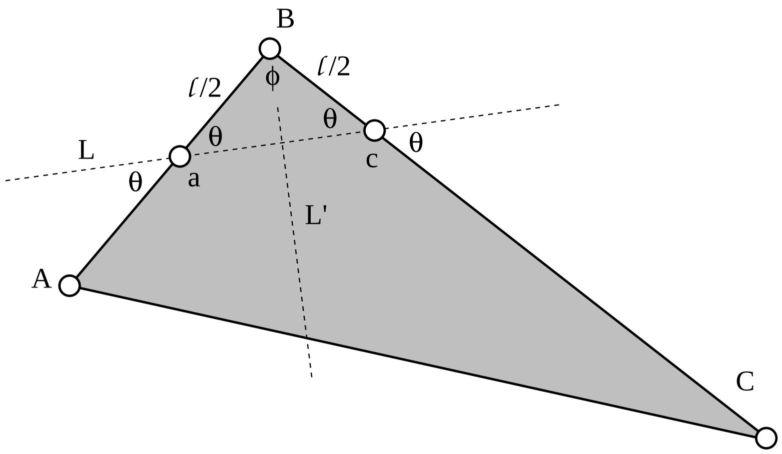


Figure 18: A and C are separated by L from B, and by L' from each other.

Now suppose the three vertices A', B', C' each move no more than δ from A, B, and C respectively. Then L continues to separate A' and C' from B', by the above argument.

□

A.3 Quadrilateral Convexification

The next results are employed in Section 5.6 on convexifying quadrilaterals. We need the following lemma that states that the reflex joint of a quadrilateral can be straightened in the first place. Let $Q = v_0v_1v_2v_3$ be a four-bar linkage with v_2 a reflex joint.

Lemma A.2 *A non-convex four-bar linkage can be convexified into a triangle by straightening its reflex joint.*

Proof: Let $ray(v_0, v_2)$ be the ray starting at v_0 in the direction of v_2 and refer to Figure 19. Without loss of generality let v_0 be the origin, $ray(v_0, v_2)$ the positive x-axis and assume the sum of the link lengths $(l_0 + l_1)$ is smaller than $(l_2 + l_3)$. Assume that v_2 is translated continuously along the x-axis in the positive direction until it gets stuck. Since v_2 cannot move further it follows that joints v_0, v_1 and v_2 all lie on the x-axis and joint v_1 has been straightened. This implies the new interior angle of v_2, $\gamma < \pi$. But before the motion v_2 was a reflex angle with $\gamma > \pi$. Since the angles change continuously there must exist a point during the motion at which $\gamma = \pi$. □

Lemma A.3 *When $d(v_0, v_2)$ is increased, all joints of the linkage open, that is, the interior angles of the convex joints and the exterior angle of the reflex joint all increase.*

Proof: We will show that if v_2 is moved along $ray(v_0, v_2)$ in such a way that the distance $d(v_0, v_2)$ is increased by some positive real number ϵ, no matter how small, while v_2 remains reflex, then all joints open. First note that by Euclid's Proposition 24 of Book I, v_1 and v_3 open, that is, their interior angles increase. Secondly, note that if the interior angle at v_0 opens then so does the exterior angle at v_2 (and vice-versa) by applying Euclid's Proposition 24 to distance $d(v_1, v_3)$. Hence another way to state the theorem in terms of distances only is: in a non-convex four-bar linkage the length of the interior diagonal increases if, and only if, the length of the exterior diagonal increases. It remains to show that increasing $d(v_0, v_2)$ increases the angle at v_0.

Before proceeding let us take care of the value of ϵ. While there is no problem selecting ϵ too small, we must ensure it is not too big, for otherwise when we increase $d(v_0, v_2)$ by ϵ the linkage may become convex. From Lemma A.2 we know that as $d(v_0, v_2)$ is increased the linkage will become a triangle at some point when joint v_2 straightens, at which time $d(v_0, v_2)$ will have reached its maximum value, say l. Using the law of cosines for this triangle we obtain

$$l^2 = l_2^2 + l_3^2 - l_2\{[(l_1 + l_2)^2 + l_3^2 - l_0^2]/(l_1 + l_2)\}.$$

Therefore if we choose ϵ such that

$$\epsilon < l - d(v_0, v_2),$$

then we ensure that v_2 remains reflex.

It is convenient to analyse the situation with link v_3v_0 as a rigid frame of reference rather than the $ray(v_0, v_2)$. Therefore let both v_0 and v_3 be fixed in the plane. Then as $d(v_0, v_2)$ increases, from Euclid's Proposition 24 it follows that v_1 rotates about v_0 along the *fixed* circle $C(v_0, l_0)$ centered at v_0 of radius l_0, v_2 rotates about v_3 on the *fixed* circle $C(v_3, l_2)$ centered at v_3 with radius l_2, and $ray(v_0, v_2)$ rotates about v_0.

Denote the initial configuration by $Q = v_0v_1v_2v_3$ and the final configuration after $d(v_0, v_2)$ is increased by ϵ by $Q' = v_0u_1u_2v_3$. In other words v_1 has moved to u_1, v_2 has moved to u_2 and $ray(v_0, v_2)$ has moved to $ray(v_0, u_2)$. Since the exterior angle at v_2 is less than π and link v_3v_2 rotates in a counterclockwise manner this motion causes u_2 to penetrate the interior of the shaded circle $C(v_1, l_1)$ centered at v_1 with radius l_1. Furthermore, u_2 cannot overshoot this shaded circle and find itself in its exterior after having penetrated it, for this would imply the joint u_2 is convex, which is impossible for the value of ϵ we have chosen. Now, since u_2 is in the interior of the shaded disk $C(v_1, l_1)$ and the radius of this disk is l_1 it follows that the distance $d(u_2, v_1)$ is less than the link length l_1. Let us therefore extend the segment $u_2 v_1$ along the $ray(u_2, v_1)$ to a point u_2' so that $d(u_2, u_2') = l_1$. Note that the figure shows the situation when u_2' lies in the exterior of $C(v_0, l_0)$. If u_2' lies on $C(v_0, l_0)$ it yields u_1 imediately. If u_2' lies in the interior of $C(v_0, l_0)$ then the arc u_1, u_2', u_1' in the figure would be in the interior of $C(v_0, l_0)$. But of course u_1, the new position of v_1, must lie on the circle $C(v_0, l_0)$. To compute the possible locations for u_1 we rotate segment $u_2 u_2'$ about u_2 in both the clockwise and counterclockwise directions to intersect the circle $C(v_0, l_0)$ at points u_1 and u_1', respectively. Since u_2 lies on $ray(v_0, u_2)$ it follows that u_1' lies to the left of $ray(v_0, u_2)$. But the two links $v_0 u_1$ and $u_1 u_2$ must remain to the right of $ray(v_0, u_2)$ because the links are not allowed to cross each other. Therefore u_1' cannot be the final position of link v_1 and the latter must move to u_1. Now since

$$d(u_2, u_2') = l_1 > d(u_2, v_1),$$

it follows that u_1 lies clockwise from v_1. Therefore link $v_0 v_1$ has rotated clockwise with respect to v_0 and since link $v_0 v_3$ is fixed the interior angle at v_0 has increased, proving the theorem. □

References

[Ada94] C. C. Adams. *The Knot Book*. W. H. Freeman, New York, 1994.

[BDD+99] T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, I. Streinu, G. Toussaint, and S. Whitesides. Locked and unlocked polygonal chains in 3D. In *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, pages 866–867, January 1999.

[BGRT96] P. Bose, F. Gomez, P. Ramos, and G. T. Toussaint. Drawing nice projections of objects in space. In *Graph Drawing (Proc. GD '95)*, volume 1027 of *Lecture Notes Comput. Sci.*, pages 52–63. Springer-Verlag, 1996.

[Can87] J. Canny. *The Complexity of Robot Motion Planning*. ACM – MIT Press Doctoral Dissertation Award Series. MIT Press, Cambridge, MA, 1987.

[CH88] G. Crippen and T. Havel. *Distance Geometry and Molecular Conformation*. Research Studies Press, Letchworth, UK, 1988.

[CJ98] J. Cantarella and H. Johnston. Nontrivial embeddings of polygonal intervals and unknots in 3-space. *J. Knot Theory Ramifications*, 7:1027–1039, 1998.

[CO99] R. Cocan and J. O'Rourke. Polygonal chains cannot lock in 4d. In *Proc. 11th Canad. Conf. Comput. Geom.*, 1999. Extended abstract. Full version: LANL/CoRR paper cs.CG/9908005.

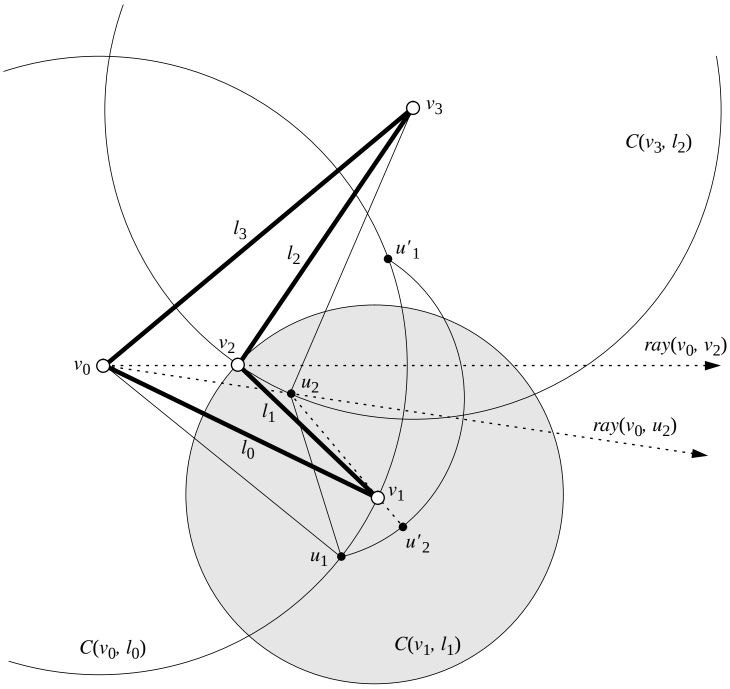


Figure 19: Line tracking holding link v_0v_3 fixed.

[Cro97] P. Cromwell. *Polyhedra.* Cambridge University Press, 1997.

[CSW95] F. Chin, J. Snoeyink, and C.-A. Wang. Finding the medial axis of a simple polygon in linear time. In *Proc. 6th Annu. Internat. Sympos. Algorithms Comput.*, volume 1004 of *Lecture Notes Comput. Sci.*, pages 382–391. Springer-Verlag, 1995.

[dSN39] B. de Sz. Nagy. Solution to problem 3763. *Amer. Math. Monthly*, 46:176–177, 1939.

[ELR+98a] H. Everett, S. Lazard, S. Robbins, H. Schröder, and S. Whitesides. Convexifying star-shaped polygons. In *Proc. 10th Canad. Conf. Comput. Geom.*, 1998.

[ELR+98b] H. Everett, S. Lazard, S. Robbins, H. Schröder, and S. Whitesides. Convexifying star-shaped polygons. In *Proc. 10th Canad. Conf. Comput. Geom.*, pages 2–3, 1998.

[Erd35] P. Erdős. Problem 3763. *Amer. Math. Monthly*, 42:627, 1935.

[GN86] C. C. Gibson and P. E. Newstead. On the geometry of the planar 4-bar mechanism. *Acta Applicandae Mathematicae*, 7:113–135, 1986.

[Grü95] B. Grünbaum. How to convexify a polygon. *Geombinatorics*, 5:24–30, July 1995.

[HJW84] J. E. Hopcroft, D. A. Joseph, and S. H. Whitesides. Movement problems for 2-dimensional linkages. *SIAM J. Comput.*, 13:610–629, 1984.

[Hun78] K. H. Hunt. *Kinematic geometry of mechanisms.* Oxford University Press, 1978.

[Kan97] V. Kantabutra. Reaching a point with an unanchored robot arm in a square. *Internat. J. Comput. Geom.*, 7(6):539–550, 1997.

[Kor85] J. U. Korein. *A Geometric Investigation of Reach.* ACM distinguished disserations series. MIT Press, 1985.

[Liv93] C. Livingston. *Knot Theory.* The Mathematical Association of America, 1993.

[LW92] W. J. Lenhart and S. H. Whitesides. Reconfiguration with line tracking motions. In *Proc. 4th Canad. Conf. Comput. Geom.*, pages 198–203, 1992.

[LW95] W. J. Lenhart and S. H. Whitesides. Reconfiguring closed polygonal chains in Euclidean d-space. *Discrete Comput. Geom.*, 13:123–140, 1995.

[Sal73] G. T. Sallee. Stretching chords of space curves. *Geometriae Dedicata*, 2:311–315, 1973.

[SS83] J. T. Schwartz and Micha Sharir. On the "piano movers" problem II: General techniques for computing topological properties of real algebraic manifolds. *Adv. Appl. Math.*, 4:298–351, 1983.

[Tou99] G. T. Toussaint. The Erdős-Nagy theorem and its ramifications. In *Proc. 11th Canad. Conf. Comput. Geom.*, 1999. Extended abstract.

[vKSW96] M. van Kreveld, J. Snoeyink, and S. Whitesides. Folding rulers inside triangles. *Discrete Comput. Geom.*, 15:265–285, 1996.

[Whi92] S. H. Whitesides. Algorithmic issues in the geometry of planar linkage movement. *Australian Comput. J.*, pages 42–50, 1992.

[Whi97] W. Whiteley. Rigidity and scene analysis. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 49, pages 893–916. CRC Press LLC, Boca Raton, FL, 1997.