Folding and Unfolding

by

Erik D. Demaine

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2001

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Acknowledgments

My time as a graduate student has been the best period of my life so far, and for that wonderful experience I owe many thanks.

I had two excellent advisors, Anna Lubiw and Ian Munro. I started working with Anna after I took her two classes on algorithms and computational geometry during my Master's, which got me excited about both these areas, and even caused me to switch entire fields of computer science, from distributed systems to theory and algorithms. Anna introduced me to Ian when some of our problems in computational geometry turned out to have large data structural components, and my work with Ian blossomed from there. The sets of problems I worked on with Anna and Ian diverged, and both remain my primary interests.

Anna and Ian have had a profound influence throughout my academic career. At the most basic level, they were excellent collaborators, and real joys to work with. They provided key insights and direction on the research side, balanced with humor, inclusion, and friendliness on the social side. And at the same time, they treated me as their equal—another researcher, another friend—just less experienced and needing to be taught about problem-solving techniques, publications, collaboration, and academic politics. It amazes me how much I've learned from them over the past 5 years or so.

But what was perhaps the most important aspect of their advisorship was my freedom—not just freedom to work on whatever problems I wanted (which was important for me), but also freedom to travel everywhere and work with everyone. The latter was *the* essential feature that allowed me to reach where I am now. Their policy of "if you want to work with that person, why not just fly there and visit?" and "sure, I can fund your trip there" was amazing, and critical to my development and way of research.

My collaborators have greatly influenced me, providing new problems to work on together, providing new approaches and insights, and teaching me new techniques for solving problems so that I can apply them myself. It is through my collaborators that I learn most of what I learn now, and that I build my collection of techniques and approaches to problems. This learning is an ongoing process that will never end. Of course, collaboration also brings an enjoyed social aspect to research, without which I could not devote my entire life to research.

My co-authors include Oswin Aichholzer (TU Graz), Esther Arkin (SUNY Stony Brook), Michael Bender (SUNY Stony Brook), David Benoit (InfoInteractive Inc.), Marshall Bern (Xerox PARC), Therese Biedl (U. Waterloo), Prosenjit Bose (Carleton U.), Broňa Brejova (U. Waterloo), David Bremner (U. New Brunswick, Fredericton), Andrej Brodnik (Luleøa Technical U.), Svante Carlsson (Luleøa Technical U.), Eowyn Cenek (U. Waterloo), Timothy Chan (U. Waterloo), Robert Connelly (Cornell U.), Carmen Cortés (U. Sevilla), Martin Demaine (U. Waterloo), Christian Duncan (U. Miami), Vida Dujmović (McGill U.), David Eppstein (U. California, Irvine), Jeff Erickson (U. Illinois, Urbana-Champaign), Martin Farach-Colton (Google Inc.), Sándor Fekete (TU Berlin), Rudolf Fleischer (Hong Kong U. Science and Technology), Greg Frederickson (Purdue U.), Erich Friedman (Stetson U.), Ian Foster (U. Chicago), Mordecai Golin (Hong Kong U. Science and Technology), Angèle Hamel (Laurier U.), George Hart (georgehart.com), Barry Hayes (PlaceWare Inc.), Michael Hoffmann (ETH Zurich), Ferran Hurtado (Polytecnic U. Catalunya), Lars Jacobsen (U. Southern Denmark), Craig Kaplan (U. Washington), Carl Kesselman (U. Southern California), Stephen Kobourov (U. Arizona), Eric Kuo (U. California, Berkeley), Stefan Langerman (McGill U.), Sylvain Lazard (INRIA Lorraine), Alejandro López-Ortiz (U. Waterloo), Anna Lubiw (U. Waterloo), Andrea Mantler (U.

North Carolina, Chapel Hill), Henk Meijer (Queens U.), Joseph Mitchell (SUNY Stony Brook), Pat Morin (McGill U.), Ian Munro (U. Waterloo), Joseph O'Rourke (Smith College), Mark Overmars (Utrecht U.), Belén Palop (U. Rey Juan Carlos), Irena Pashchenko (Stanford U.), Venkatesh Raman (Inst. Mathematical Sciences), Suneeta Ramaswami (Rutgers U.), Steven Robbins (McGill U.), Günter Rote (FU Berlin), Vera Sacristán (Polytecnic U. Catalunya), Robert Sedgewick (Princeton U.), Saurabh Sethia (Oregon State U.), Steven Skiena (SUNY Stony Brook), Marc Snir (IBM Research), Jack Snoeyink (U. North Carolina, Chapel Hill), Michael Soss (McGill U.), Sampalli Srinivas (Dalhousie U.), Ileana Streinu (Smith College), Godfried Toussaint (McGill U.), Helena Verrill (Hannover U.), Tomáš Vinař (U. Waterloo), Ming-wei Wang (U. Waterloo), and Sue Whitesides (McGill U.). I particularly thank Joe Mitchell and Joe O'Rourke for their academic advice, guidance, and encouragement, as well as being excellent collaborators. Sue Whitesides and Anna introduced me to large-scale collaboration, by way of the productive Barbados atmosphere in 1998. This experience in particular had a major impact on my way of research.

My father, Marty Demaine, has greatly influenced me throughout my entire life. Through homeschool he (directly or indirectly) taught me most everything before I entered university, including fundamental aspects like values and what's important in life. Academically, he has always exposed me to fields I know little about, and when I show interest, found resources for both me and him to learn about them. Particular instances of this exposure were my introduction to programming and mathematics, which are now the main things I cherish. He continues to teach me about life and support me in uncountable ways.

My then girlfriend and now good friend, Margaret Dulat, strongly influenced my personal life. She taught me love, emotional maturity, and spontaneity, and showed me what wonderful things exist outside of research and mathematics.

People at Waterloo have always been warm and accepting, in particular of my age (about which I have had trouble elsewhere). Collaboration is easy at Waterloo if you make it happen. Particular thanks to those who attended the University of Waterloo algorithmic problem session and made it a fun and productive time.

Finally, thanks to Wendy Rush for remotely preparing this thesis for submission as I sit here in Cambridge, Massachusetts.

Erik D. Demaine

November 15, 2001

Abstract

The results of this thesis concern folding of one-dimensional objects in two dimensions: planar linkages. More precisely, a planar linkage consists of a collection of rigid bars (line segments) connected at their endpoints. Foldings of such a linkage must preserve the connections at endpoints, preserve the bar lengths, and (in our context) prevent bars from crossing. The main result of this thesis is that a planar linkage forming a collection of polygonal arcs and cycles can be folded so that all outermost arcs (not enclosed by other cycles) become straight and all outermost cycles become convex. A complementary result of this thesis is that once a cycle becomes convex, it can be folded into any other convex cycle with the same counterclockwise sequence of bar lengths. Together, these results show that the configuration space of all possible foldings of a planar arc or cycle linkage is connected.

These results fall into the broader context of folding and unfolding k-dimensional objects in n-dimensional space, $k \leq n$. Another contribution of this thesis is a survey of research in this field. The survey revolves around three principal aspects that have received extensive study: linkages in arbitrary dimensions (folding one-dimensional objects in two or more dimensions, including protein folding), paper folding (normally, folding two-dimensional objects in three dimensions), and folding and unfolding polyhedra (two-dimensional objects embedded in three-dimensional space).

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The area of geometric folding and unfolding is attractive in that problems and even results can be easily understood with little knowledge of mathematics or computer science, yet the solutions are difficult and involve many sophisticated techniques. The general form of a folding problem is to determine how a particular geometric object (e.g., linkage, piece of paper, polyhedron, or protein) can be reconfigured or *folded* according to a few constraints, which depend on the object being folded and the problem of interest. In particular, we are interested in characterizing the existence of foldings, and efficient algorithms for finding foldings, or in proving that such algorithms are impossible.

There is a wide range of folding and unfolding problems; some go back several centuries and are still unsolved, such as unfolding convex polyhedra, while others are more recent, such as protein folding. In the last few years, there has been tremendous progress on many of the fundamental problems in folding and unfolding, yet some of the most important questions still remain open. This leaves the area in an exciting state.

Most folding and unfolding problems are attractive from a pure mathematical standpoint, for the beauty of the problems themselves. Nonetheless, most of the problems have close connections to important industrial applications. Linkage folding has applications in robotics and hydraulic tube bending, and has connections to protein folding. Paper folding has applications in sheet-metal bending, packaging, and air-bag folding. Unfolding polyhedra has applications in manufacturing, particularly sheet-metal bending.

This thesis consists of three main parts. The first part (Chapter 2) surveys the general area of folding and unfolding. The next two parts (Chapters 3 and 4) describe in detail two results on one aspect of folding and unfolding. Together, these results settle one of the major outstanding problems in linkage folding: can every planar configuration of a polygonal chain be reached by continuous motions that preserve the vertex connections and edge lengths and never cause crossings?

More precisely, a *linkage* is an embedded planar graph whose edges represent rigid *bars* and whose vertices represent hinges or joints. In general, linkages are free to fold (i.e., their embedding is free to change) in any way that preserves the lengths of the bars. Linkages have been studied extensively throughout history, in mechanics, discrete geometry, knot theory, molecular biology, polymer physics, and most recently computational geometry. One of the most famous original applications of linkages is converting circular motion (from a motor) into linear motion, the topic of Kempe's well-known 1877 book *How to Draw a Straight Line*

[Kem77].

This thesis is concerned with the connectivity between various configurations of a common linkage. In general, the space of all configurations of a linkage can be arbitrarily complex in the topological sense, homeomorphic to an arbitrary compact real algebraic variety with Euclidean topology [JS99]. However, we are able to show that for a widely studied class of linkages, the space of configurations is connected. Specifically, if the graph of a linkage is a single path, the linkage is called a *polygonal arc*, and if the graph is a single cycle, the linkage is called a *polygonal cycle* or *polygon*; in either case, the linkage is called a *polygonal chain*.

Because motions can be reversed and concatenated, it suffices to ask whether every polygonal chain can be *unfolded* into some canonical configuration. For polygonal arcs, we ask whether they can be *straightened*, that is, folded into the configuration in which every vertex angle is π. For polygonal cycles, we normally ask whether they can be *convexified*, that is, folded into a configuration in which every interior vertex angle is at most π. Of course, then it remains to show that the space of convex configurations of a linkage is connected, that is, every convex configuration can be folded into any other convex configuration with the same clockwise/counterclockwise orientation.

The main result of this thesis (Chapter 4) is that, in the plane, every polygonal arc can be straightened and every polygonal cycle can be convexified.[1] A complementary result of this thesis (Chapter 3) is that once a polygonal cycle is convex, it can be folded into any other convex configuration.[2] Together, these results settle the reachability of all configurations of planar polygonal chains.

Whether every arc in the plane can be straightened, and whether every polygon in the plane can be convexified, have been outstanding open questions for at least a quarter of a century. The problems are natural, so they have arisen independently in a variety of fields, including topology, pattern recognition, and discrete geometry. In the next few paragraphs we detail some contexts in which the problem has appeared, although we are probably not aware of all such contexts.

To our knowledge, the first person to pose the problem of convexifying cycles was Stephen Schanuel, in designing homework problems for a course in algebraic topology.[3] George Bergman learned of the problem from Schanuel during Bergman's visit to the State University of New York at Buffalo in the early 1970's, and suggested the simpler question of straightening arcs.[4] As a consequence of this line of interest, the problems are included in Kirby's *Problems in Low-Dimensional Topology* [Kir97, Problem 5.18].

During the period 1986–1989, Ulf Grenander and the members of the Pattern Theory Group at Brown University explored various problems involving the probabilistic structure when generators (e.g., points and line segments) were transformed by diffeomorphisms (e.g., Euclidean transformations) subject to global constraints (e.g., avoiding intersections).[5] For the purposes of Bayesian image understanding, they were interested in whether the process was *ergodic*, i.e., every configuration could be reached from any other. In particular, they proved this for polygonal cycles in which the roles of angles and lengths are reversed: *angles* are fixed but *lengths*

[1] Joint work with Robert Connelly and Günter Rote.

[2] Joint work with Oswin Aichholzer, Jeff Erickson, Ferran Hurtado, Mark Overmars, Michael Soss, and Godfried T. Toussaint.

[3] Personal communication with Stephen Schanuel, July 2000.

[4] Personal communication with George Bergman, July 2000.

[5] Personal communication with Ulf Grenander, June 2000.

may vary [GCK91, Appendix D, pp. 108–128]. Grenander posed the problems considered here during a seminar at Indiana University in March 1987 [Gre87], and probably also at earlier talks.[6]

In the discrete and computational geometry community, the problems were independently posed by William Lenhart and Sue Whitesides in March 1991 and by Joseph Mitchell in December 1992.[7] Sue Whitesides first posed this problem in a talk in 1991 [LW91a]. In this community the problems were first published in a technical report in 1993 [LW93] and in a journal in 1995 [LW95].

A fairly large group of people was involved in trying to construct and prove or disprove locked polygonal chains, at various times over the past few years. Some of the initial people to form this group were Sue Whitesides, William Lenhart, Joseph Mitchell, Robert Connelly, Sándor Fekete, and Günter Rote. The group grew significantly to include the attendees[8] of the International Workshop on Wrapping and Folding organized by Anna Lubiw and Sue Whitesides at the Bellairs Research Institute of McGill University in February 1998.

Typically, someone in the group would distribute an example that s/he constructed or was given by a colleague. We would try various motions that did not work, and we would often try proving that the example was locked because it appeared so! For some examples, it took several months before we found an unlocking motion. The main difficulty was that "simple" motions that change a few vertex angles at once, while easiest to visualize, seemed to be insufficient for unlocking complex examples. Amazingly, it also seemed that nevertheless there was always a global unlocking motion, and furthermore it was felt that there was a driving principle permitting "blowing up" of the linkage. This notion was formalized by Günter Rote in June 1999 with the idea that perhaps a polygonal chain could be unfolded via a motion that did not decrease the distance between any two vertices.

One of the key meetings that started the solution to the problem is the Monte Verità Conference on Discrete and Computational Geometry in Ascona, Switzerland, organized by Jacob Goodman, Richard Pollack, and Emo Welzl in June 1999. In particular, I first met Robert Connelly and Günter Rote at this conference. Our work continued at the 4th Geometry Festival, an international workshop on Discrete Geometry and Rigidity, in Budapest, Hungary, organized by András Bezdek, Károly Bezdek, Károly Böröczky, and Robert Connelly, in November 1999. Our solution, described in Chapter 4, was announced in January 2000.

In contrast, the result that convex polygonal cycles can be folded into any desired convex configuration was generally assumed by the community, and surprisingly remained unpublished. It surfaced as a hole to be filled during discussions at the 16th Winter Workshop on Computational Geometry: Geometric and Computational Aspects of Molecular Reconfiguration, organized by Godfried Toussaint at the Bellairs Research Institute of McGill University in February 2001. At the workshop we also solved the problem, as described in Chapter 3, using several components from the proof of Cauchy's 1819 rigidity theorem.

Overview. Each chapter of this thesis is self-contained, so reading one does not depend on reading another. In particular, the few needed definitions in early chapters are repeated in later

[6] Personal communication with Allan Edmonds, June 2000.

[7] Personal communications with Sue Whitesides (March 2000) and Joseph Mitchell (January 2000).

[8] Therese Biedl, Martin Demaine, Hazel Everett, Sylvain Lazard, Anna Lubiw, Joseph O'Rourke, Mark Overmars, Steven Robbins, Ileana Streinu, Godfried Toussaint, Sue Whitesides, and myself.

chapters. However, to understand the results of Chapters 3 and 4 in the context of other results in linkage folding, the reader may find it instructive to read the survey up to and including (at least) Section 2.1.2.1. For closely related linkage-folding problems in two and higher dimensions, we suggest reading up to and including (at least) Section 2.1.4.

Chapter 2

Survey

Folding and unfolding problems have been implicit since Albrecht Dürer in the early 1500's [Dür77], but have not been studied extensively in the mathematical literature until recently. Over the past few years, there has been a surge of interest in these problems in discrete and computational geometry. This chapter gives a brief survey of most of the work in this area. Related surveys are given in [Dem00] and [O'R98].

In general, we are interested in how objects (such as linkages, pieces of paper, and polyhedra) can be moved or reconfigured (folded) subject to certain constraints depending on the type of object and the problem of interest. Typically the process of *unfolding* approaches a more basic shape, whereas *folding* complicates the shape. We can also generally define the *configuration space* as the set of all configurations or states of the object, with paths in the space corresponding to motions (foldings) of the object.

The survey in this chapter is subdivided into three sections based on the type of object being folded: linkages, paper, or polyhedra. Unavoidably, areas with which I am more familiar are covered in more detail, and some areas may be omitted entirely, but the intent is to be as complete and balanced as possible. For example, many more problems have been explored in linkage and paper folding than in polyhedron folding, so the length of the sections differ appropriately. On the other hand, this thesis cannot do justice to the wealth of research in protein folding, so only a partial survey appears in Section 2.1.6.

2.1 Linkages

2.1.1 Definitions and Fundamental Questions

A *linkage* or *framework* consists of a collection of rigid line segments (*bars*) joined at their endpoints (*vertices*) to form a particular graph. A linkage can be *folded* by moving the vertices in $\mathbb{R}^d$ in any way that preserves the length of every bar.

Linkages have been studied extensively in the case that bars are allowed to cross; see, for example, [HJW84, JS99, KM95, Kem76, LW95, Sal73, Whi92b]. Such linkages can be very complex, even in the plane. In 1876, Kempe [Kem76] showed that a planar linkage can be built so that a vertex traces an arbitrary polynomial curve. Hopcroft, Joseph, and Whitesides [HJW84] showed that deciding whether a planar linkage can reach a particular configuration is PSPACE-complete. Jordan and Steiner [JS99] proved that there is a linkage whose configuration space is homeomorphic to an arbitrary compact real algebraic variety with Euclidean topology, and thus planar linkages are equivalent to the theory of the reals (solving systems of polynomial inequalities over reals). On the other hand, for a linkage whose graph is just a cycle, all configurations can be reached in Euclidean space of any dimension greater than 2 by a sequence of simple motions [LW95, Sal73], and in the plane there is a simple restriction characterizing which polygons can be inverted in orientation [LW95].

Recently there has been much work on the case that the linkage must remain *simple*, that is, never have two bars cross.[1] This additional constraint defines the type of linkage folding considered in the rest of this thesis. Such linkage folding has applications in hydraulic tube bending [O'R98] and motion planning of robot arms. There are also connections to protein folding in molecular biology; see Section 2.1.6. See also [CDR00, O'R98, Tou99a] for other surveys on linkage folding without crossings.

Perhaps the most fundamental question we can ask about folding linkages is whether it is possible to fold from everywhere to everywhere. That is, is there a folding between any two simple configurations of the same linkage (with matching graphs, combinatorial embeddings, and bar lengths) while preserving the bar lengths and not crossing any bars during the folding? Because folding motions can be reversed and concatenated, this fundamental question is equivalent to whether every simple configuration can be folded into some *canonical configuration*. The definition of canonical configuration depends on the type of linkage under consideration.

In the context of linkages whose edges cannot cross, three general types of linkages are commonly studied, characterized by the structure of their associated graphs (see Figure 2.1): a *polygonal arc* or *open polygonal chain* (a single path); a *polygonal cycle*, *polygon*, or *closed polygonal chain* (a single cycle); and a *polygonal tree* (a single tree).[2] The canonical configuration of an arc is the *straight configuration*, all vertex angles equal to 180°. A canonical configuration of a cycle is a *convex configuration*, planar and having all interior vertex angles less than or equal to 180°. It is relatively easy to show that convex configurations are indeed "canonical" in the sense that any one can be folded into any other; the first published proof of this fact is the topic of Chapter 3. Finally, a canonical configuration of a tree is a *flat configuration*: all vertices lie on a horizontal line, and all bars point "rightward" from a common root. Again it is easy to fold any flat configuration into any other [BDD$^+$01b].

The fundamental questions thus become whether every arc can be straightened, every cycle can be convexified, and every tree can be flattened. The answers to these questions depend on the dimension of the space in which the linkage starts, and the dimension of the space in which the linkage may be folded. Over the past few years, this collection of questions has been completely resolved. Table 2.1 summarizes the answers to these questions when the dimensions of the initial configuration and the folding process are equal. The answers in the columns for arcs and cycles are analogous to the existence of knots tied from one-dimensional string: nontrivial knots exist only in 3D. In contrast, the column for trees in Table 2.1 presents an interesting difference in 2D: while trees in the plane are topologically unknotted, they can be geometrically locked. This observation is some evidence for the belief that the fundamental problems are most difficult in 2D.

[1] Typically, bars are allowed to touch, provided they do not properly intersect. However, requiring bars to touch only at common endpoints does not change the results.

[2] More general graphs have mostly been studied only in the context of allowing bars to cross. Once the class of graphs allows something as general as rigid triangulations, the fundamental question of course has a negative answer.

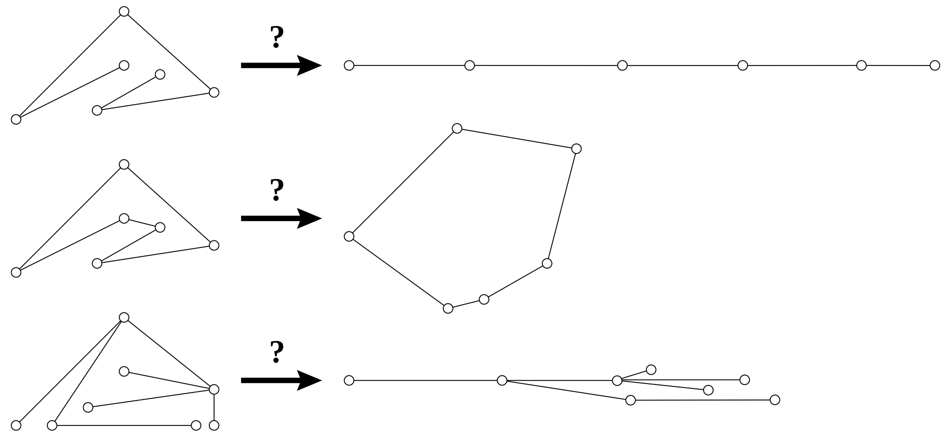


Figure 2.1: The three common types of linkages and their associated canonical configurations. From top to bottom, a polygonal arc $\xrightarrow{?}$ the straight configuration, a polygonal cycle $\xrightarrow{?}$ a convex configuration, and a polygonal tree $\xrightarrow{?}$ a (nearly) flat configuration.

Dimension	Can all arcs be straightened?	Can all cycles be convexified?	Can all trees be flattened?
2	Yes (Chapter 4)	Yes (Chapter 4)	No [BDD+01b]
3	No [CJ98, BDD+01a]	No [CJ98, BDD+01a]	No [from arcs]
4 & above	Yes [CO99, CO01]	Yes [CO99, CO01]	Yes [CO01]

Table 2.1: Answers to fundamental questions for linkage folding without crossings.

The next three subsections describe the historical progress of these results and other results closely related to the fundamental questions, one section per row in Table 2.1. The last two subsections describe other linkage-folding problems and results not directly connected to the fundamental questions. Section 2.1.5 describes several extensions of the notion of Erdős flips arising in the fundamental questions in 3D (Section 2.1.3.1). Section 2.1.6 discusses the connections between linkage folding and protein folding, and describes the most closely related results and open problems; a full survey of protein folding is beyond the scope of this thesis.

2.1.2 Fundamental Questions in 2D

Linkage folding in 2D is the topic of the results in this thesis (Chapters 3 and 4). Section 2.1.2.1 describes the development of the theorems for straightening arcs and convexifying cycles in 2D, the main result of this thesis (Chapter 4). Section 2.1.2.2 explores when the motions from these theorems can be converted into simpler components. Finally, Section 2.1.2.3 discusses the contrary result that not all trees can be flattened.

2.1.2.1 Carpenter's Rule Problem: Polygonal Chains in 2D

The questions of whether every polygonal arc can be straightened in the plane and whether every polygonal cycle can be convexified in the plane have arisen in many contexts over the last quarter-century. In particular, they were posed independently by four groups: Stephen Schanuel and George Bergman in the early 1970's, Ulf Grenander in 1987, William Lenhart and Sue Whitesides in 1991, and Joseph Mitchell in 1992. (Chapter 1 presents a more detailed history.) In the discrete and computational geometry community, the arc-straightening problem has become known as the *carpenter's rule problem* because a carpenter's rule folds like a polygonal arc.

Many people have devoted time to these two problems in the past decade. Most people's initial intuition is that the answers are yes, but describing a precise general motion was difficult. Several people proposed examples of polygonal arcs and cycles that might be "locked" (unstraightenable and unconvexifiable), but eventually every proposed example was unlocked by hand. It was not until early in the year 2000 that the problems were solved by Connelly, Demaine, and Rote [CDR00], with an answer of yes. This result is the topic of Chapter 4.

More generally, the result in Chapter 4 shows that a collection of nonintersecting polygonal arcs and cycles in the plane may be simultaneously folded so that the outermost arcs are straightened and the outermost cycles are convexified. The "outermost" proviso is necessary because arcs and cycles cannot always be straightened and convexified when they are contained in other cycles. The key idea for the solution, introduced by Günter Rote, is to look for *expansive* motions in which no vertex-to-vertex distance decreases. Bars cannot cross before getting closer, so expansiveness allows us to ignore the difficult nonlocal constraint that bars must not cross. Expansiveness brings the problem into the areas of rigidity theory and tensegrity theory, which study frameworks of rigid bars, unshrinkable *struts*, and unexpandable *cables*. Tools from these areas helped show that, *infinitesimally*, arcs and cycles can be folded expansively. These infinitesimal motions are combined by flowing along a vector field defined implicitly by an optimization problem. As a result, the motion is piecewise-differentiable (C^1). In addition, any symmetries present in the initial configuration of the linkage are preserved throughout the motion. Similar techniques show that the area of each cycle increases by this motion and furthermore by any expansive motion. Again, see Chapter 4 for details.

Ileana Streinu [Str00] has demonstrated another expansive motion for straightening arcs and convexifying polygons that is piecewise-algebraic, composed of a sequence of $O(n^2)$ *mechanisms* each with one degree of freedom. Using the half of our proof in Chapter 4 about infinitesimal flexibility (Section 4.3), Streinu proved that each mechanism follows an expansive motion. Because the motion of each mechanism is defined by an algebraic curve, the entire motion can be computed in principle. On the other hand, an approximation to the motion in Chapter 4 is easy to implement, and has resulted in animations such as the one in Figure 2.2. Our motion also has the advantage of being canonical, thereby preserving symmetries in the linkage and proving contractibility of the configuration space. See Section 4.4.4 for a more detailed technical comparison of the two methods.

The pursuit of the arc and cycle problems in 2D inspired research on several related problems, described in the remainder of Section 2.1.

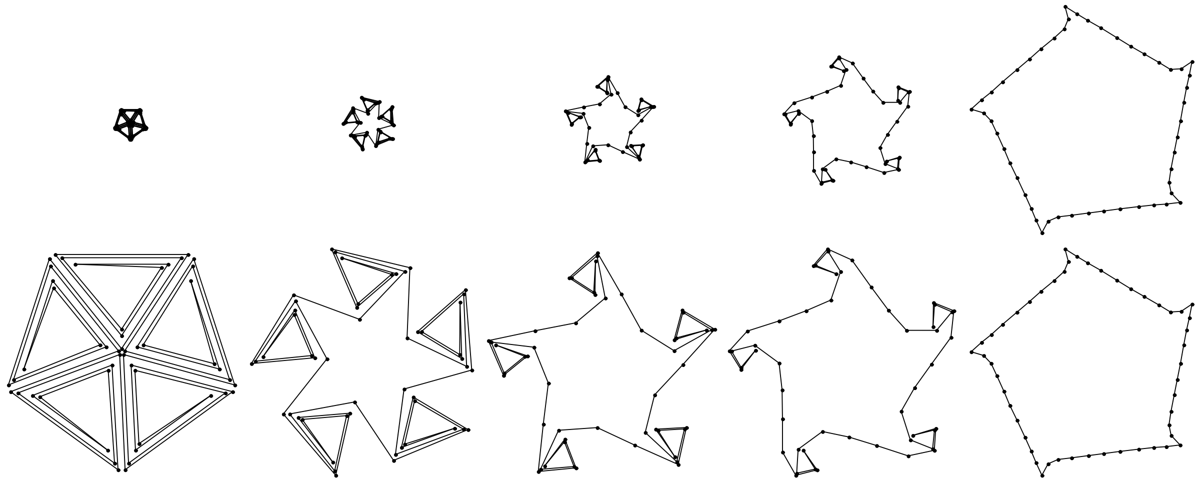


Figure 2.2: Two views of convexifying a "doubled tree" linkage. The top snapshots are all scaled the same, and the bottom snapshots are scaled differently to improve visibility. See `http://db.uwaterloo.ca/~eddemain/linkage/` for more animations.

2.1.2.2 Simple Motions and Special Polygons in 2D

The motions in the previous section are complicated, either involving the solution of a differential equation defined implicitly by an optimization problem (Chapter 4) or a high-degree algebraic curve [Str00]. Sándor Fekete (personal communication, 2000) observed that these expansive motions can be converted into a finite sequence of "simple moves" that avoid crossings but are not expansive. We call a move (motion) *simple* if it changes only $O(1)$ vertex angles. A polygonal arc can be moved while changing only one joint angle, whereas a polygonal cycle in the plane requires changing at least four joint angles at once [BDL$^+$99], because otherwise the joints define a virtual triangle which remains rigid. (In three and higher dimensions, only two joints of a polygonal cycle must rotate at once.)

Fekete observes that an expansive motion can be viewed as a path in the configuration space surrounded by a "tube" of separation that grows in width. More precisely, a *funnel* is a union of a continuum of spheres centered at each point along the curve, and the radii of these spheres increase along the curve. The funnel is a subset of the configuration space, that is, any point within the funnel defines a configuration without crossings. The initial width of the funnel is defined by the initial separation in the linkage, that is, the minimum distance between two nonincident edges. This separation grows during the motion because of expansiveness. Thus, the motion can be replaced by a "staircase" of many small moves, each parallel to a coordinate axis in the configuration space, without violating simplicity. By an appropriate parameterization of configuration space, each such move corresponds to the simplest type of move (in terms of the number of simultaneous joint rotations) for the linkage at hand (1 and 4 for arcs and cycles in 2D).

Unfortunately, the number of simple moves resulting from this argument can be large, depending on the initial separation of the linkage and not just the number of vertices. This seems necessary in general: we conjecture that the number of simple moves to straighten an arc cannot be bounded in terms of its number of vertices.[3] It remains open to obtain a general bound in terms of the encoding size of the input linkage when the coordinates of the vertices are encoded in binary.

In 1999 (before the general arc-and-cycle result), Biedl, Demaine, Lazard, Robbins, and Soss [BDL+99] proved that $O(n^2)$ simple moves suffice to convexify a special class of polygons. Specifically, a polygon is called *monotone* if every vertical line crosses at most two bars. Each of the resulting $O(n^2)$ moves rotates the minimum number 4 of joints at once, and the moves can be computed in $O(n^2)$ time on a real RAM [BDL+99].

In 1998, Everett, Lazard, Robbins, Schröder, and Whitesides [ELR+98] proved that *star-shaped polygons*, polygons whose boundary is visible from a single interior point, can be convexified by a particularly natural motion. Although the motion does not satisfy the definition of "simple" above, it is easy to compute, and matches the intuition of expanding a starshaped polygon from its center. Specifically, the main motion involves moving every second vertex of the polygon at a constant speed along a ray from the point of visibility, and moving each remaining vertex as forced by the constraints imposed by the two incident bars. At certain times this motion is interrupted and a simple motion is applied in order to maintain certain preconditions necessary for the main motion.

2.1.2.3 Trees in 2D

In 1998, Biedl et al. [BDD+01b] showed that not all trees can be flattened in the plane. Their example, shown in Figure 2.3, consists of at least 5 *petals* connected at a central high-degree vertex. Each petal is an arc of three bars, the last of which is "wedged" into the center vertex.

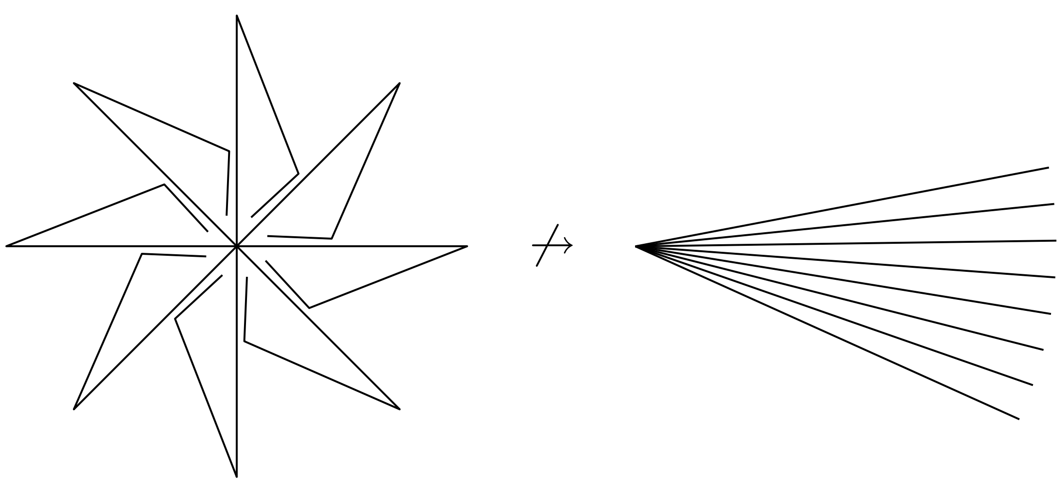


Figure 2.3: The locked tree on the left, from [BDD+01b], cannot be reconfigured into the nearly flat configuration on the right. (Figure 1 of [BDD+01b].)

Intuitively, the argument that the tree is locked is as follows. No petal can be straightened unless enough angular room has been made. But no petal can be reduced to occupy less angular space by more than a small positive number unless the petal has already been straightened. This circular dependence implies that no petal can be straightened, so the tree is locked. The

[3]Personal communication with John Iacono, October 2000.

details of this argument, in particular obtaining suitable tolerances for closeness, are much more involved [BDD+01b]. The key is that each petal occupies a wedge of space whose angle is less than 90°, and thus at least 5 petals are required.

This tree remains locked if we replace the central degree-5 (or higher) vertex with multiple degree-3 vertices connected by very short bars [BDD+01b, full version]. Recently, Connelly, Demaine, and Rote [CDR01] have shown that even a tree with a single degree-3 vertex and the remaining vertices having degrees 1 and 2 can be locked, so the arc-and-cycle result in Chapter 4 is tight. In [CDR01], they also present an extension to rigidity/tensegrity theory that allows proving many planar linkages (e.g., trees) to be locked via linear programming. In particular, they use this method to give short proofs that the tree in Figure 2.3 and the tree with one degree-3 vertex are *strongly locked*, in the sense that sufficiently small perturbations of the vertex positions and bar lengths result in a tree that cannot be moved more than ε in the configuration space for any $\varepsilon > 0$.

2.1.3 Fundamental Questions in 3D

Linkage folding in 3D was initiated earlier, by Paul Erdős in 1935 [Erd35]. His problem and its solution are described in Section 2.1.3.1. Both Sections 2.1.3.1 and 2.1.3.2 deal with linkages that start in the plane. The more general situation, a linkage starting arbitrarily in 3D, is addressed in Section 2.1.3.3. As this problem proves unsolvable in general, additional special cases are addressed in Sections 2.1.3.4 and 2.1.3.5.

2.1.3.1 Flips for Planar Polygons in 3D

Erdős [Erd35] asked whether a particular "flipping" algorithm for folding a planar polygon through three dimensions (preserving bar lengths and simplicity) would convexify the polygon in a finite number of steps. Specifically, a *flip* rotates by 180° a subchain of the polygon called a *pocket*, a region bounded by polygon edges and an edge of the convex hull, the latter defining the axis of rotation. Thus, during the motion, the pocket flips out of the plane, and afterwards the polygon returns to the plane. See Figure 2.4.

Erdős proposed flipping every pocket of the polygon in parallel, but Nagy [Nag39] observed that this can lead to nonsimple polygons; see Figure 2.5. However, by flipping only one pocket at a time, Nagy [Nag39] proved in 1939 that a finite number of flips suffice to convexify any polygon. This problem and result have been rediscovered several times; see [Tou99b, Grü95] for the long history. This problem has also inspired several other problems; we defer details to Section 2.1.5.

Unfortunately, the number of required flips can be arbitrarily large in terms of the number of vertices, even for a quadrangle. This fact was originally proved by Joss and Shannon (1973); see [Grü95, Tou99b, BDD+01a]. Figure 2.6 shows the construction. By making the vertical edge of the quadrangle very short and even closer to the horizontal edge, the angles after the first flip approach the mirror image of the original quadrangle, and hence the number of required flips approaches infinity.

Mark Overmars (February 1998) posed the still-open problem of bounding the number of flips in terms of natural measures of geometric closeness such as the diameter (maximum distance between two vertices), sharpest angle, and the minimum feature size (minimum distance between two nonincident edges).

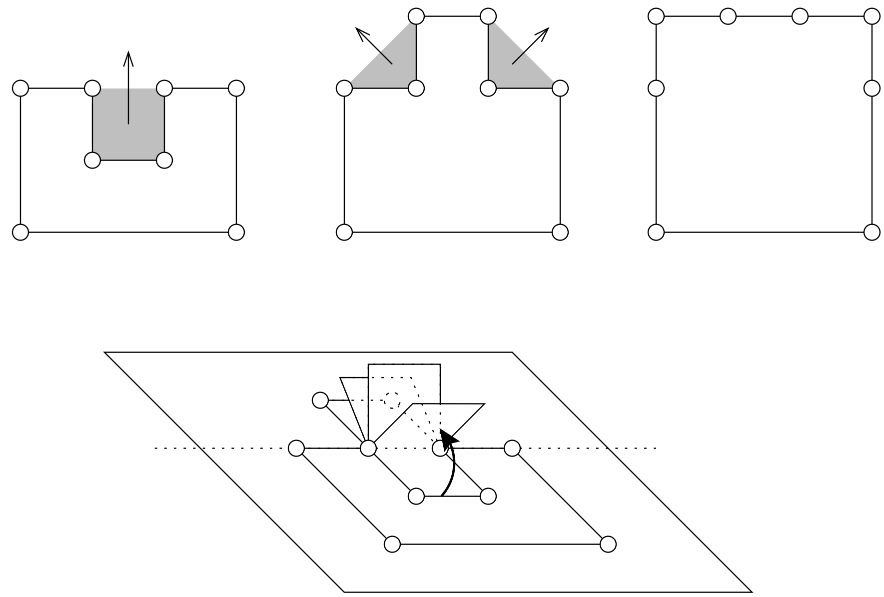


Figure 2.4: Flipping a polygon until it is convex. Pockets are shaded.

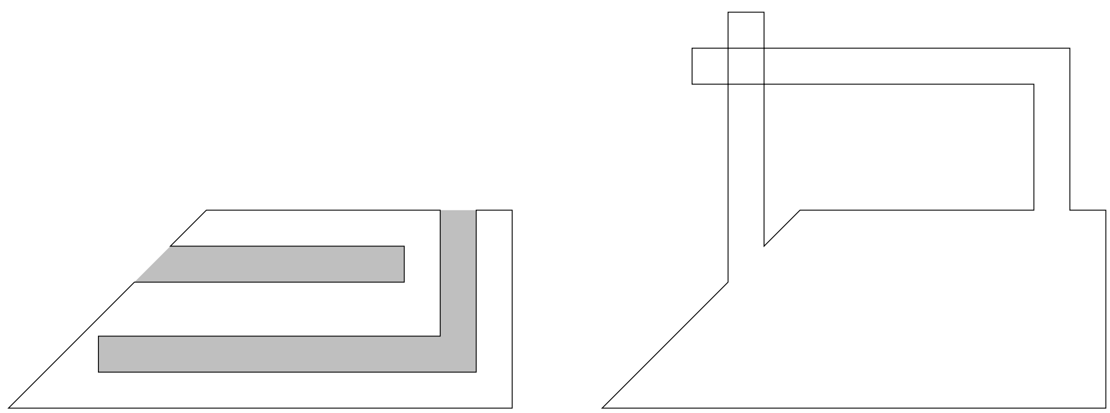


Figure 2.5: Flipping multiple pockets simultaneously can lead to crossings [Nag39].

Another open problem is to determine the complexity of finding the shortest or longest sequence of flips to convexify a given polygon. Weak NP-hardness has been established for the related problem of finding the longest sequence of *flipturns*; see Section 2.1.5.4.

2.1.3.2 Efficient Algorithms for Planar Linkages in 3D

Motivated by the inefficiency of the flip algorithm, two algorithms were developed in 1999 to convexify planar polygons by motions in 3D using a linear number of simple moves [BDD+01a, AGP99]. The basis of these algorithms is to lift the polygon, bar by bar, at all times maintaining a convex chain (or *arch*) lying in a plane orthogonal to the plane containing the polygon; see Figure 2.7. The algorithms differ in the invariant on the form of the arch, how additional bars are added to the arch, and how the invariant is restored. In both cases, the details of the

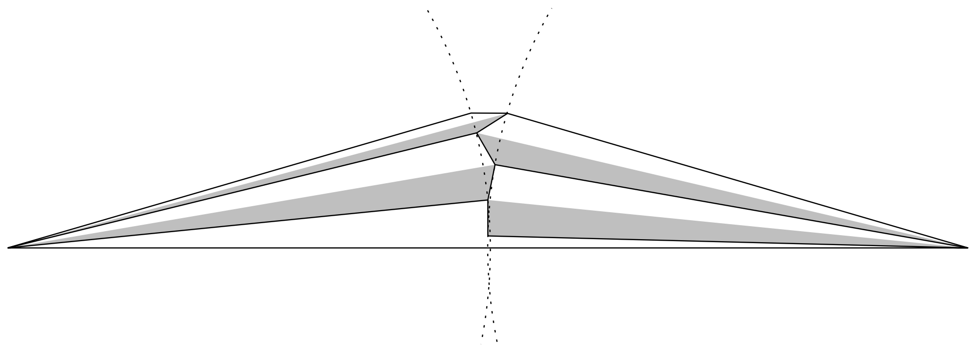


Figure 2.6: Quadrangles can require arbitrarily many flips to convexify [Grü95, Tou99b, BDD+01a].

algorithm are significantly more involved than the overarching idea.

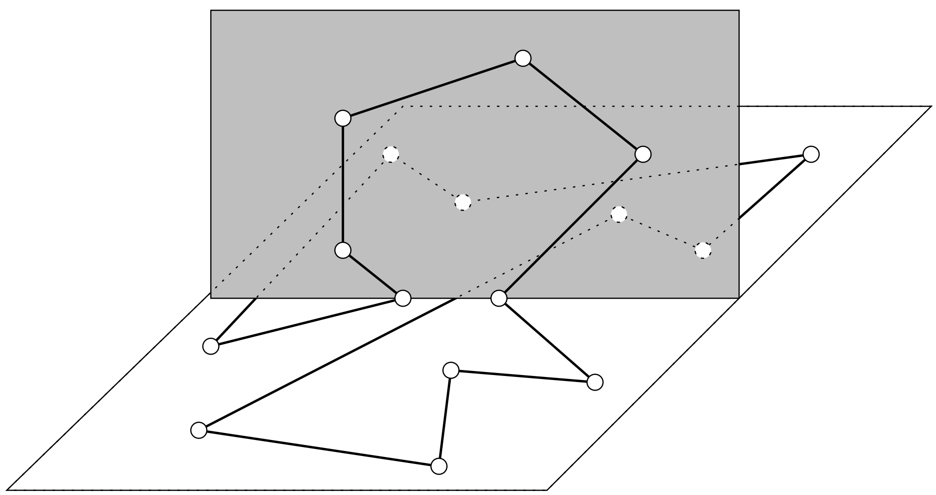


Figure 2.7: A planar polygon partially lifted into a convex arch lying in a vertical plane (shaded). Based on Figure 6 of [BDD+01a].

In contrast, it is relatively easy to straighten a polygonal arc in the plane by motions in 3D [BDD+01a]. The algorithm lifts each bar in turn, from one end of the arc to the other, at all times maintaining a prefix of the arc in a vertical line orthogonal to the plane containing the arc; see Figure 2.8. Each lifting motion causes two joint angles to rotate, so that the lifted vertical line remains vertical, while the rest of the chain remains in its original plane.

Both of these results involve reconfiguring planar linkages by motions in 3D. The initial motivation for this study was the pursuit of the analogous problems involving motions in 2D. Ignoring the complexity of the motions, these results have been superseded by the 2D results in Section 2.1.2.1. However, the motions arising from the algorithms in this section are significantly simpler than those in Section 2.1.2.1, and hence the results of this section remain of interest.

Furthermore, these results can be extended to results impossible in general with planar motions. We observe that an analogous algorithm to the one in [BDD+01a] for lifting a polygonal arc into a vertical line (Figure 2.8) can be used to flatten every planar tree by motions through

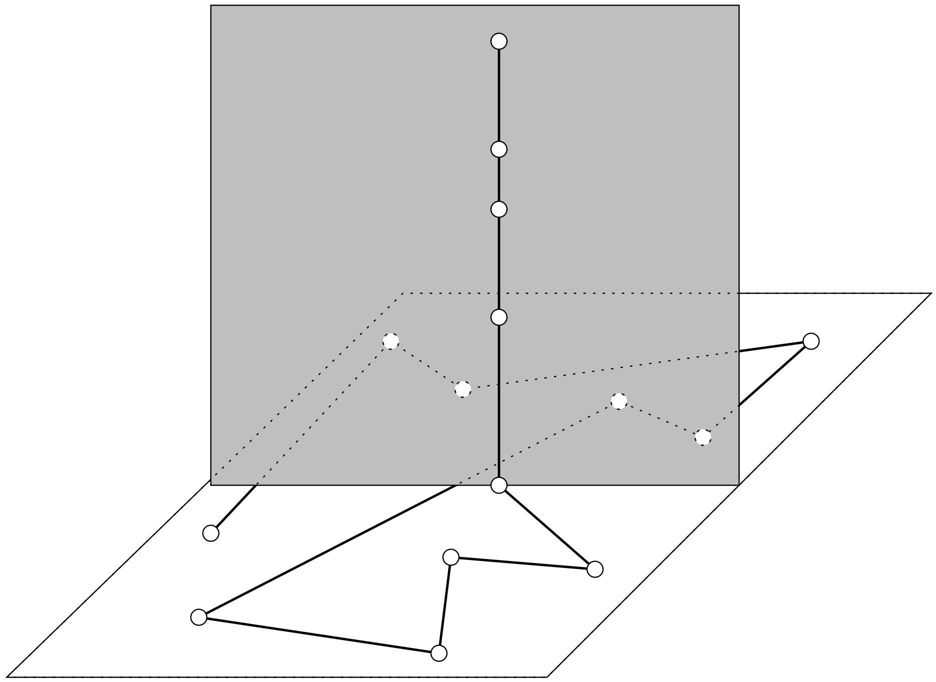


Figure 2.8: A planar arc partially lifted into a vertical line.

3D, working from the leaves of the tree inward. See Section 2.1.3.4 for other generalizations.

2.1.3.3 Almost Knots

What if the linkage starts in an arbitrary position in 3D instead of in a plane? In general, a polygonal arc or an unknotted polygonal cycle in 3D cannot always be straightened or convexified [CJ98, Tou01, BDD$^+$01a]. See Figure 2.9 for an example of a locked arc in 3D. Provided each of the two end bars is longer than the sum s of the middle three bar lengths, the ends of the chain cannot get close enough to the middle bars to untangle the chain. More precisely [BDD$^+$01a], because the ends of the chain remain outside a sphere with radius s and centered at one of the middle vertices, we can connect the ends of the chain with an unknotted flexible cord outside the sphere, and any straightening motion unties the resulting knot, which is impossible without crossings.

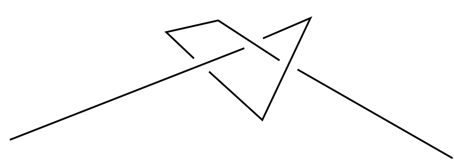


Figure 2.9: A locked polygonal arc in 3D with 5 bars [CJ98, BDD$^+$01a].

It remains open to determine the complexity of deciding whether a polygonal arc or unknotted polygonal cycle can be straightened or convexified. The only known upper bound is an algorithm in PSPACE, via Canny's roadmap algorithm for general motion planning [Can87, Can88].

The next two sections describe special cases of 3D chains, more general than planar chains, that can be straightened and convexified.

2.1.3.4 Chains on Polyhedra

In [BDD$^+$01a], the algorithm for lifting a planar arc into a vertical line, described in Section 2.1.3.2, is generalized to straighten any polygonal arc lying on the surface of a convex polyhedron. The modified invariant is to maintain at all times a prefix of the chain along a ray normal to one of the facets of the polyhedron. Similar to Section 2.1.3.2, we observe that this algorithm can be generalized to polygonal trees lying on the surface of a convex polyhedron.

It remains open whether analogous results can be shown for polygonal cycles, or for more general polyhedra. The case of polyhedral terrains is settled in the next section.

2.1.3.5 Simple Projection

The "almost knottedness" of the example in Section 2.1.3.3 suggests that polygonal chains having simple orthogonal projections can always be straightened or convexified. This fact is established by two papers [BDD$^+$01a, CKM$^+$01]. In addition, there is a polynomial-time algorithm to decide whether a polygonal chain has a simple projection, and if so find a suitable plane for projection [BGRT99].

For a polygonal arc with a simple orthogonal projection, the straightening method is relatively straightforward [BDD$^+$01a]. The basic idea is to process the arc from one end to the other, accumulating bars into a compact "accordion" (x-monotone chain) lying in a plane orthogonal to the projection plane, in which each bar is nearly vertical. Once this accumulation is complete, the planar accordion is unfolded joint-by-joint into a straight arc. We observe that a similar algorithm can be used to fold a polygonal tree with a simple orthogonal projection into a generalized accordion, which can then be folded into a flat configuration.

For a polygonal cycle with a simple orthogonal projection, the convexification method is based on two steps [CKM$^+$01]. First, the projection of the polygon is convexified via the results described in Section 2.1.2.1, by folding the 3D polygon to track the shadow, keeping constant the ascent of each bar. Second, Calvo, Krizanc, Morin, Soss, and Toussaint [CKM$^+$01] develop an algorithm for convexifying a polygon with convex projection. The basic idea is to reconfigure the convex projection into a triangle, and stretch each accordion formed by an edge in the projection. In linear time they show how to compute a motion for the second step that consists of $O(n)$ simple moves each changing at most seven vertex angles. Recently, this motion has been improved [ABB$^+$01] to consist of $O(n)$ simple moves each changing at most four vertex angles, which can be implemented by a larger number of moves each changing just two vertex angles.

2.1.4 Fundamental Questions in 4D and Higher Dimensions

In all dimensions higher than 3, it is known that all knots are trivial; analogously, all polygonal arcs can be straightened, all polygonal cycles can be convexified, and all polygonal trees can be flattened [CO99, CO01]. Intuitively, this result holds because the number of degrees of freedom of any vertex is at least two higher than the dimensionality of the obstacles imposed by any bar. This property allows Cocan and O'Rourke [CO99, CO01] to establish, for example, that the last bar of a polygonal arc can be folded by itself to any target position that is simple.

Cocan and O'Rourke [CO01] show how to straighten an arc using $O(n)$ simple moves that can be computed in $O(n^2)$ time and $O(n)$ space. On the other hand, their method for convexifying a polygon requires $O(n^6)$ simple moves and $O(n^6 \log n)$ time to compute.

An interesting direction for further research is to scale the dimension of the object to be folded together with the dimension of the space in which it is folded. For example, how can solid polygons hinged together at their edges (to form, e.g., an open chain) be folded in 3D (or dimensions higher than 2)? Unlike linkages in 3D, this problem is a generalization of linkages in 2D: linkages in 3D permit additional motions, whereas hinged polygons can be forced to fold like a planar linkage by extruding the linkage orthogonal to the plane. See Figure 2.10.

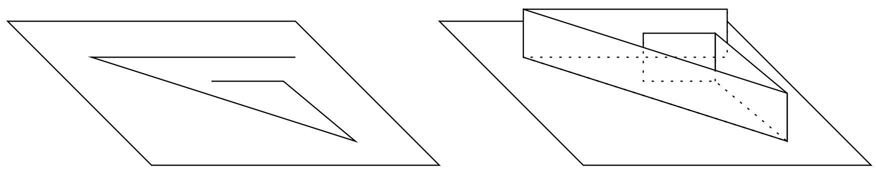


Figure 2.10: Extruding a linkage into an equivalent collection of polygons (rectangles) hinged together at their edges.

On the other hand, not all hinged collections of polygons in 3D have connected configuration spaces. To see this fact, suppose we replace each bar in a 3D open chain (say, the chain in Figure 2.9) by an appropriate thin polygon to build a hinged collection of polygons whose edges include the original chain (as in Figure 2.11). The resulting hinged chain of polygons is only more restrictive in motion than the original 3D open chain, and thus not all hinged chains of polygons have connected configuration spaces. Even stronger, Biedl, Lubiw, and Sun [BLS99, Sun99] showed that even hinged chains of rectangles do not have connected configuration spaces, by considering an orthogonal version of Figure 2.9.

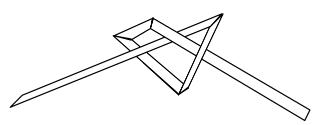


Figure 2.11: Converting the chain in Figure 2.9 into a more-restrictive hinged chain of polygons.

We suggest some problems involving such hinged collections of polygons in the context of polyhedra in Section 2.3.4.

2.1.5 Flips, Deflations, and Flipturns

Erdős's notion of flips described in Section 2.1.3.1 has inspired several directions of research. In Section 2.1.3.2 we described one such direction [BDD$^+$01a, AGP99], connected to the pursuit of the fundamental question for planar polygons in 3D, and reducing the number of moves required. Other directions, without the same connection, are described in this section. See also [Tou99b] for a more detailed survey of this area, including proofs of several results mentioned here. Unless otherwise specified, in this section a *polygon* refers to a polygon in a plane.

2.1.5.1 Flips in Nonsimple Polygons

Flips can be generalized to apply to nonsimple polygons: consider two vertices adjacent along the convex hull of the polygon, splitting the polygon into two chains, and rotate one (either) chain by 180° with respect to the other chain about the axis through the two vertices. Simplicity may not be preserved throughout the motion, just as it may not hold in the initial or final configuration. The obvious question is whether every nonsimple polygon can be convexified by a finite sequence of such flips. Grünbaum and Zaks [GZ98] proved that if at each step we choose the flip that maximizes the resulting sum of distances between all pairs of vertices, then this metric increases per flip, and the polygon becomes convex after finitely many flips. Without sophisticated data structures, computing these flips requires $\Omega(n^2)$ time per flip. Toussaint [Tou99b] proved that a different sequence of flips convexifies a nonsimple polygon, and this sequence can be computed in $O(n)$ time per flip. Recently, Biedl and Demaine [BD01] established that every sequence of flips convexifies a nonsimple polygon. We expect that each flip can be executed in polynomial amortized time using dynamic convex-hull data structures as in [ACD$^+$00].[4]

2.1.5.2 Deflations

A *deflation* [FHM$^+$01, Weg93, Tou99b] is the reverse of a flip. More precisely, a deflation of a polygon should result in a simple polygon that can be flipped into the original polygon. In other words, a deflation is a rotation by 180° about a line meeting the polygon at two vertices and nowhere else, thus separating the chain into two subchains, such that the rotation does not cause any intersections. Hence, after the deflation, this line becomes a line of support (a line extending a convex-hull edge). Wegner [Weg93] proposed the notion of deflations, and their striking similarity to flips led him to conjecture that every polygon can be deflated only a finite number of times. Surprisingly, this is not true: Fevens, Hernandez, Mesa, Soss, and Toussaint [FHM$^+$01] characterized a class of quadrangles whose unique deflation leads to another quadrangle in the class, thus repeating ad infinitum.

2.1.5.3 Pivots

Pivots [ADE$^+$01, Tou99b, Mil94, MS93, MS88, JW90] are a natural generalization of flips to allow rotation by an arbitrary angle about the line connecting any two vertices of the polygon. This is the simplest kind of motion of a linkage in three or higher dimensions. In this context, the polygon need not lie in a plane, and self-intersection is permitted during the motion. Millett [Mil94] gave an algorithm to bring any equilateral polygon to any other, in particular for convexifying an equilateral polygon and for bringing any convex configuration to another other, via a sequence of pivots. The first part (convexification) has a flaw; a replacement algorithm is given by Toussaint [Tou99b] and also applies to nonequilateral polygons in any dimension more than two. Aichholzer, Demaine, Erickson, Hurtado, Overmars, Soss, and Toussaint [ADE$^+$01] proved that the second part, moving between two convex configurations, is also possible for nonequilateral polygons via a sequence of pivots.

Pivots without allowing self-intersection seem for the most part unexplored, although of course then we could not hope to reconfigure arbitrarily. Using an approach similar to Fekete's

[4] Personal communication with Jeff Erickson.

observation described in Section 2.1.2.2, and using the results above to characterize the configuration space, we expect that any motion of a 3D chain in which two nonincident edges never touch can be simulated by a sequence of pivots.

2.1.5.4 Flipturns

Joss and Shannon (1973) were the first to observe that a number of flips unbounded in terms of the number of vertices can be required to convexify a polygon (Figure 2.6). Sadly, their work was never published; see [Grü95] for the story. Joss and Shannon also proposed the operation of a *flipturn* [ACD$^+$00, ABC$^+$00, Grü95], which rotates by 180° a pocket of the polygon, not in 3D about the convex-hull edge, but in 2D around the midpoint of the convex-hull edge. Unlike flips, flipturns are not standard linkage motions because they change the ordering of the edges. However, flipturns have the advantage that they preserve the set of angles and orientations of the edges: each flipturn corresponds to reversing a subsequence of edge angles. Thus flipturns can only result in a permutation of the edge angles (which determines the polygon), and each cyclic permutation can be reached only once because flipturns increase the polygon's area, and therefore at most $(n-1)! - 1$ flipturns can be made before a polygon becomes convex.

This simple $(n-1)!$ bound is an immediate benefit over flips, and surprisingly it remained the best known upper bound on the number of flipturns for a quarter of a century. In 1973, Joss and Shannon conjectured that just $n^2/4$ flipturns sufficed to convexify a polygon. In 1999, Biedl [Bie00] proved a matching lower bound, up to an additive constant, on the length of the longest flipturn sequence. In 2000, Ahn, Bose, Czyzowicz, Hanusse, Kranakis, and Morin [ABC$^+$00] made significant progress: for a modified definition of flipturn that allows ignoring degenerate cases, they prove an upper bound of $n^2/4 - O(1)$ on the length of the longest flipturn sequence. Furthermore, if there are at most s distinct slopes in the polygon, they prove an upper bound of $ns/4 - O(1)$. This bound does not hold for normal flipturns, showing the importance of degenerate flipturns. Using additional analysis of a structure called brackets, Aichholzer et al. [ACD$^+$00] proved an upper bound of $ns - O(n) \leq n^2 - O(n)$ on the length of the longest flipturn sequence in both natural interpretations of degenerate cases.

Aichholzer et al. [ACD$^+$00] have also proved various tighter bounds for orthogonal polygons, for both the shortest and longest sequence, and for three different definitions of flipturns handling degeneracies in different ways. Some of these bounds are tight in the worst case, while others have a constant-factor gap; we omit the details here. Algorithmically, finding the longest flipturn sequence in an orthogonal polygon is weakly NP-hard [ACD$^+$00], while the analogous problem for the shortest flipturn sequence remains open. Aichholzer et al. also develop data structures for maintaining the convex hull subject to flipturns pocket, allowing flipturn sequences to be executed in polylogarithmic time per operation.

The main open problem involving flipturns is whether $\Omega(n^2)$ flipturns are necessary to convexify a polygon if the flipturns are chosen wisely. The best known lower bound is linear.

2.1.6 Protein Folding

Protein folding [CD93, Hay98, ML94] is an important problem in molecular biology because it is generally believed that the folded structure of a protein (the fundamental building block of life) determines its function and behavior.

2.1.6.1 Related Pure Linkage Problems

A protein can be modeled by a linkage in which the vertices represent amino acids and the bars represent bonds connecting them. The bars representing bonds are typically close in length, within a factor less than two. Depending on the level of detail, the protein can be modeled as a tree (more precise) or as a chain (less precise).

An amazing property of proteins is that they fold quickly and consistently to a minimum-energy configuration. Understanding this motion has immediate connections to linkage folding in 3D. One open theoretical question [BDD$^+$01a] arising in this context is whether every equilateral polygonal arc in 3D can be straightened. Cantarella and Johnston [CJ98] have proved that this is true for arcs of at most 5 bars. More generally, can every equilateral polygonal tree in 3D be flattened?

Such questions ask whether what nature does is feasible, in general, without resorting to temporarily disconnecting some bonds. According to some theories, such disconnections do occur; these mathematical questions may shed light on these theories. Another approach in this direction is an open problem of Anna Lubiw [DO99] asking for the minimum number of cuts required to straighten a general n-bar polygonal arc in 3D. The answer is between $n/4 + O(1)$ and $n/2 + O(1)$; see [DLO01] for these and related results.

A more accurate mathematical model of foldings of proteins is not by "normal" linkages whose vertices are universal joints, but rather by *revolute* linkages[5] whose vertices each form a fixed angle. This angular constraint roughly halves the number of degrees of freedom in the linkage; the basic motion is rotating a portion of the linkage around a bar of the linkage. Foldings of such linkages have been explored extensively by Soss and Toussaint [ST00, Sos01]. For example, they prove in [ST00] that it is NP-complete to decide whether a revolute polygonal arc can be flattened (folded into a plane), and in [Sos01] that it is NP-complete to decide whether a revolute polygonal arc can be folded into its mirror image. More positive results analyze the polynomial complexity of determining the maximum extent of a rotation around a bar: Soss and Toussaint [ST00, Sos01] prove an $O(n^2)$ upper bound, and Soss, Erickson, and Overmars [Sos01, SEO01] give a 3SUM-hardness reduction, suggesting an $\Omega(n^2)$ lower bound. The simpler case of revolute chains with a constant number of vertices is studied in [DLO01].

2.1.6.2 H-P Model

So far in this section we have not considered the forces involved in protein folding in nature. There are several models of these forces.

One of the most popular models of protein folding is the hydrophobic-hydrophilic (H-P) model [CD93, Dil90, Hay98], which defines both a geometry and a quality metric of foldings. Basically, proteins are chains of amino acids distinguished into two categories, hydrophobic (H) and hydrophilic (P). A protein folding is an embedding along edges of the square lattice in 2D or the cubic lattice in 3D without self-intersection. The optimum or minimum-energy folding maximizes the number of hydrophobic (H) nodes that are adjacent in the lattice. Intuitively, this metric causes hydrophobic amino acids to avoid the surrounding water.

This combinatorial model is attractive in its simplicity, and already seems to capture several essential features of protein folding such as the tendency for the hydrophobic components to fold to the center of a globular protein [CD93]. While the H-P model is most intuitively defined

[5] This terminology was introduced in [DLO01].

in 3D to match the physical world, in fact it is more realistic as a 2D model for computationally feasible problem sizes. The basic reason for this is that the perimeter-to-area ratio of a short 2D chain is a close approximation to the surface-to-volume ratio of a long 3D chain [CD93, Hay98].

Much work has been done on the H-P model [BL98, CD91, CD90, CGP$^+$98, HI95, LD89, LD90, LW91b, UM93a, UM93b, UM93c]. Recently, on the computational side, Berger and Leighton [BL98] proved NP-completeness of finding the optimal folding in 3D, and Crescenzi et al. [CGP$^+$98] proved NP-completeness in 2D. Hart and Istrail [HI95] have developed a 3/8-approximation in 3D and a 1/4-approximation in 2D for maximizing the number of hydrophobic-hydrophobic adjacencies.

Aichholzer, Bremner, Demaine, Meijer, Sacristan, and Soss [ABD$^+$01a] have begun exploring an important yet potentially more tractable aspect of protein folding: can we design a protein that folds stably into a desired configuration? In the H-P model, a protein folds *stably* if it has a unique minimum-energy configuration. So far, Aichholzer et al. [ABD$^+$01a] have proved the existence of stably folding proteins of all lengths divisible by 4, and for closed chains of all possible (even) lengths. The next problem is to characterize these proteins and their foldings.

2.2 Paper

Paper folding (origami) has led to several interesting mathematical and computational questions over the past fifteen years or so. A piece of paper, normally a (solid) polygon such as a square or rectangle, can be folded by any continuous motion that preserves the distances on the surface and does not cause the paper to properly self-intersect. Informally, paper cannot tear, stretch, or cross itself, but may otherwise bend freely. Formally, a folding is a continuum of isometric embeddings of the piece of paper in $\mathbb{R}^3$. However, the use of the term "embedding" is weak: paper is permitted to touch itself provided it does not properly cross itself. In particular, a *flat folding* folds the piece of paper back into the plane, and so the paper must necessarily touch itself. We frequently ignore the continuous motion of a folding and instead concentrate on the final folded state of the paper; in the case of a flat folding, the flat folded state is called a *flat origami*.

Some of the pioneering work in origami mathematics (see Section 2.2.3.1) studies the *crease pattern* that results from unfolding a flat origami, that is, the graph of edges on the paper that fold to edges of a flat origami. Stated in reverse, what crease patterns have flat foldings? Various necessary conditions are known [Hul94, Jus94, Kaw89], but there is little hope for a polynomial characterization: Bern and Hayes [BH96] have shown that this decision problem is NP-hard.

A more recent trend, as in [BH96], is to explore *computational origami*, the algorithmic aspects of paper folding. This field essentially began with Robert Lang's work on algorithmic origami design [Lan96], starting around 1993. Since then, the field of computational origami has grown significantly, in particular in the past two years by applying computational geometry techniques. This section surveys the majority of this work.

This survey is joint work with Martin Demaine. A version will appear in the Proceedings of the 3rd International Meeting of Origami Science, Mathematics, and Education [DD01].

2.2.1 Categorization

Most results in computational origami fit into at least one of three categories: universality results, efficient decision algorithms, and computational intractability results. This categorization applies more generally to folding and unfolding, but is particularly useful for results in computational origami.

A *universality result* shows that, subject to a certain model of folding, everything is possible. For example, any tree-shaped origami base (Section 2.2.2.1), any polygonal silhouette (Section 2.2.2.3), and any polyhedral surface (Section 2.2.2.3) can be folded out of a large-enough piece of paper. Universality results often come with efficient algorithms for finding the foldings; pure existence results are rare.

When universality results are impossible (some objects cannot be folded), the next-best result is an *efficient decision algorithm* to determine whether a given object is foldable. Here "efficient" normally means "polynomial time." For example, there is a polynomial-time algorithm to decide whether a "map" (grid of creases marked mountain and valley) can be folded by a sequence of simple folds (Section 2.2.3.4).

Not all paper-folding problems have efficient algorithms, and this can be proved by a *computational intractability result*. For example, it is NP-hard to tell whether a given crease pattern folds into any flat origami (Section 2.2.3.2), even when folds are restricted to simple folds (Section 2.2.3.4). These results mean that there are no polynomial-time algorithms for these problems, unless some of the hardest computational problems can also be solved in polynomial time, which is generally deemed unlikely.

We further distinguish computational origami results as addressing either *origami design* or *origami foldability*. Basically, in origami design, some aspects of the target configuration are specified, and the goal is to design a suitable target that can be folded out of paper. In origami foldability, the target configuration is unspecified and arbitrary, and rather the initial configuration is specified, specifically the crease pattern possibly marked with mountains and valleys, and the goal is to fold something (anything) using precisely those creases. While at first it may seem that understanding origami foldability is a necessary component for origami design, the results indicate that in fact origami design is easier to solve than origami foldability which is usually intractable.

Our survey of computational origami is partitioned accordingly into Section 2.2.2 (origami design) and Section 2.2.3 (origami foldability). In addition, Section 2.2.4 discusses paper folding one dimension down, which has connections to linkage folding.

2.2.2 Origami Design

We define *origami design* loosely as, given a piece of paper, fold it into an object with certain desired properties, e.g., a particular shape. Most closely related to "traditional" origami design is Lang's TreeMaker work (Section 2.2.2.1), which has brought modern origami design to a new level of complexity. Related to this work is the problem of folding a piece of paper to align a prescribed graph (Section 2.2.2.2), which can be used for a magic trick involving folding and one complete straight cut. Another approach is to design an origami with a specific silhouette or three-dimensional shape (Section 2.2.2.3), although the algorithms developed so far do not lead to practical foldings. A recent specific type of origami is an origami tessellation (Section 2.2.2.4), which "folds" a tiling of the plane. A nonstandard form of origami is to start with a piece of paper that is not flat, but rather the surface of a polyhedron, and the goal is to flatten that surface; this topic will be deferred to Section 2.3 on polyhedra, specifically Section 2.3.5.

2.2.2.1 TreeMaker

TreeMaker is a computer program by Robert Lang that implements the *tree method* for origami design. Some components of this method, such as special cases of the constituent molecules and the idea of disk packing, as well as other methods for origami design, have been explored in the Japanese technical origami community, in particular by Jun Maekawa, Fumiaki Kawahata, and Toshiyuki Meguro. This work has led to several successful designs, but a full survey is beyond the scope of this thesis; see [Lan98]. Here we concentrate on Lang's work [Lan94a, Lan94b, Lan96, Lan98, Lan01]; over the past several years, he has developed the tree method to the point where an algorithm and computer program have been explicitly defined and implemented.

The tree method allows one to design an origami *base* in the shape of a specified tree with desired edge lengths, which can then be folded and shaped into an origami model. See Figure 2.12 for an example. More precisely, the tree method designs a *uniaxial base* [Lan96], which must have the following properties: the base lies above the xy plane, all faces of paper are perpendicular to the xy plane, the projection of the base to the xy plane is precisely where the base comes in contact with the xy plane, and this projection is a tree. The theorem is that every geometric tree (unrooted tree with cyclicly ordered children and prescribed edge lengths) is the projection of a uniaxial base that can be folded from e.g. a square.

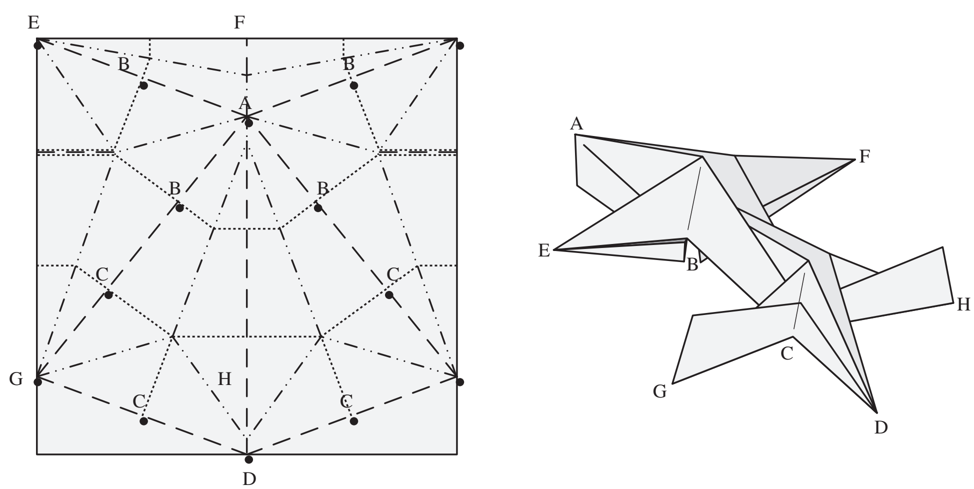


Figure 2.12: Lang's TreeMaker applied to an 8-vertex tree to produce a lizard base. (Figure 2.1.11 of [Lan98].)

The crease pattern for such a base can be found efficiently, certainly in $O(N \operatorname{polylog} N)$ time where N is the number of creases (the output size). It is unclear whether the number of creases required is bounded in terms of the combinatorial complexity of the input, i.e., the number of vertices in the input tree. Optimizing the base to make maximal use of the paper is a difficult nonlinear constrained optimization problem, but the TreeMaker software has shown the viability of existing methods for finding good local optima. Indeed, additional practical constraints can be imposed, such as symmetry in the crease pattern, or that angles of creases are integer multiples of some value (e.g., 22.5°) subject to some flexibility in the edge lengths.

TreeMaker finds a crease pattern that results in the desired uniaxial base. Work on a related problem (see the next section) by Bern, Demaine, Eppstein, and Hayes [BDEH01] suggests a method for finding an appropriate mountain-valley assignment for the crease pattern, and possibly also the resulting folded state.

2.2.2.2 One Complete Straight Cut

Take a piece of paper, fold it flat, make one complete straight cut, and unfold the pieces. What shapes can result? This *fold-and-cut* problem was first formally stated by Martin Gardner in 1960 [Gar95b]. The problem has a much longer history, going back to 1721 in a Japanese puzzle book [Sen21], Betsy Ross in 1777 [Har73], and Houdini in 1922 [Hou22]. See [DDL98] for a more detailed history.

More formally, given a planar graph drawn with straight edges on a piece of paper, can the paper be folded flat so as to map the entire graph to a common line, and map nothing else to that line? The surprising answer is that this is always possible, for any collection of line segments in the plane, forming nonconvex polygons, adjoining polygons, nested polygons, etc. There are two solutions to the problem. The first (partial) solution [DDL98, DDL99b] is based on a structure called the straight skeleton, which captures the symmetries of the graph, thereby exploiting a more global structure to the problem. This solution applies to a large class of instances which we do not describe in detail here. The second (complete) solution [BDEH01] is based on disk packing to make the problem more local, and achieves efficient bounds on the number of creases. See Figures 2.13 and 2.14 for examples of the two methods.

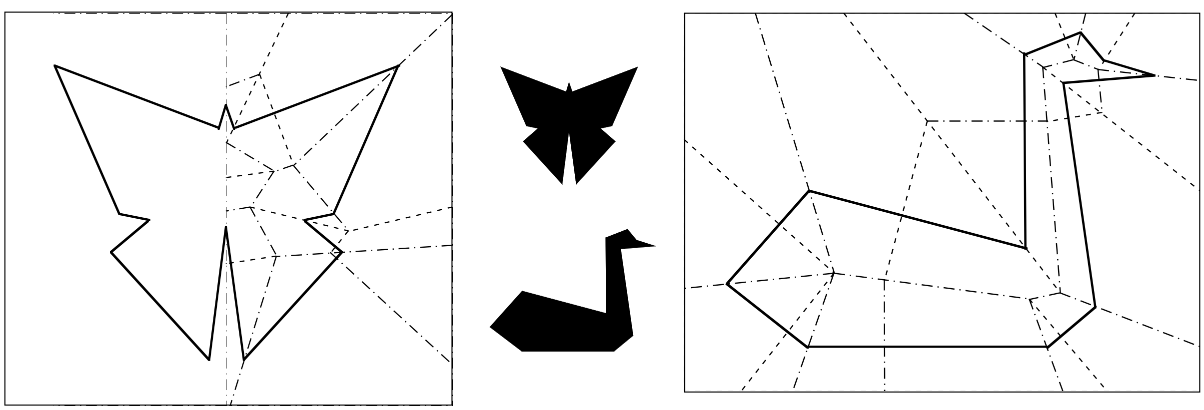


Figure 2.13: Crease patterns for folding a rectangle of paper flat so that one complete straight cut makes a butterfly (left) or a swan (right), based on [DDL98, DDL99b].

While this problem may not seem directly connected to pure paper folding because of the one cut, the equivalent problem of folding a piece of paper to line up a given collection of edges is in fact closely connected to origami design. Specifically, one subproblem that arises in TreeMaker (Section 2.2.2.1) is that the piece of paper is decomposed into convex polygons, and the paper must be folded flat so as to line up all the edges of the convex polygons, and place the interior of these polygons above this line. The fold-and-cut problem is a generalization of this situation to arbitrary graphs: nonconvex polygons, nested polygons, etc. In TreeMaker, there

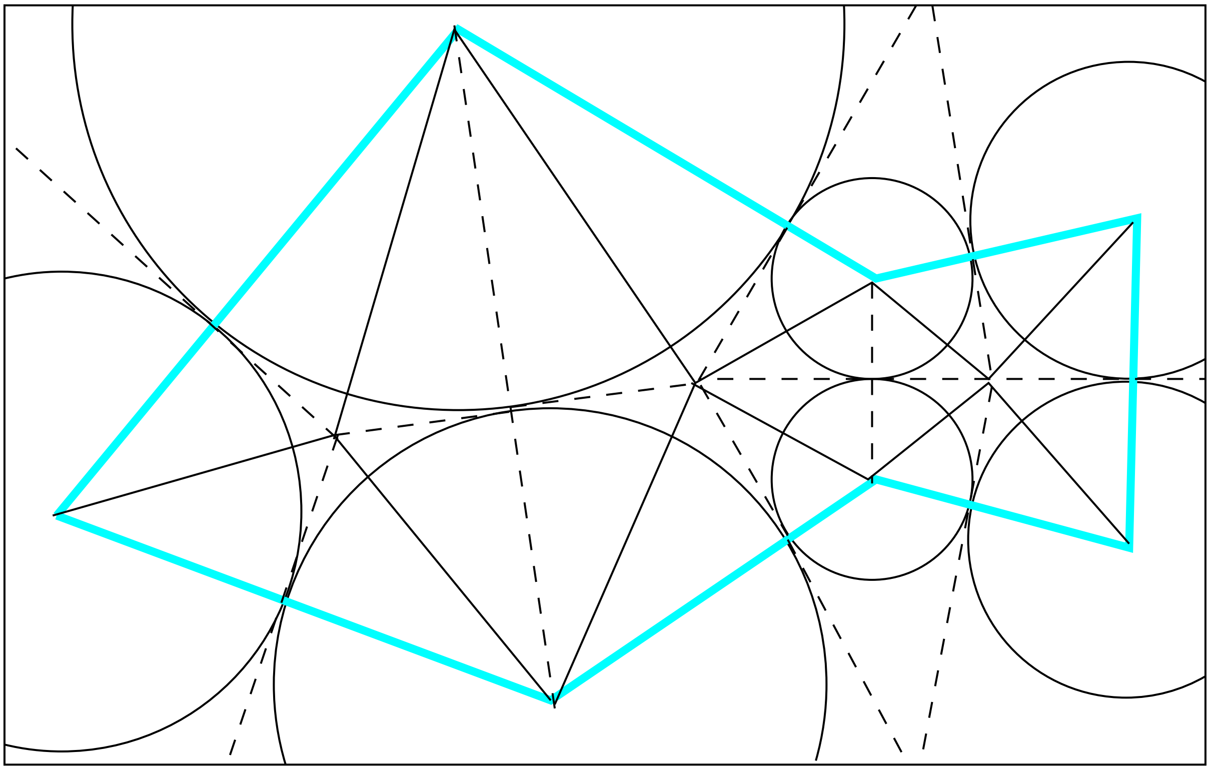


Figure 2.14: Crease pattern for folding a rectangle of paper flat so that one complete straight cut makes a fish (drawn with thick grey segments), based on the method described in [BDEH01]. (Figure 8 of [BDEH01].)

are important additional constraints in how the edges can be lined up, called path constraints, which are necessary to enforce the desired geometric tree. These constraints lead to additional components in the solution called *gussets*.

2.2.2.3 Silhouettes and Polyhedra

A more direct approach to origami design is to impose the exact final shape that the paper should take. More precisely, suppose we specify a particular flat silhouette, or a three-dimensional polyhedral surface, and desire a folding of a sufficiently large square of paper into precisely this object, allowing coverage by multiple layers of paper. For what polyhedral shapes (shapes made up of flat sides) is this possible? This problem is implicit throughout origami design, and was first formally posed by Bern and Hayes in 1996 [BH96]. The surprising answer is “always,” as established by Demaine, Demaine, and Mitchell in 1999 [DDM99, DDM00].

The basic idea of the approach is to fold the piece of paper into a thin strip of paper, and then wrap this strip around the desired shape. This wrapping can be done particularly efficiently using methods in computational geometry. Specifically, three algorithms are described in [DDM00] for this process. One algorithm optimizes paper usage; the amount of paper required can be made arbitrarily close to the surface area of the shape, which is optimal. Another algorithm maximizes the width of the strip subject to some constraints. A third algorithm places the visible *seams* of the paper in any desired pattern forming a decomposition of the sides into convex polygons. In particular, the number and total length of seams can be optimized in polynomial time in most cases [DDM00].

All of these algorithms allow an additional twist: the paper may be colored differently on both sides, and the shape may be two-colored according to which side should be showing. In principle, this allows the design of two-color models similar to the models in Montroll's *Origami Inside-Out* [Mon93]. An example is shown in Figure 2.15.

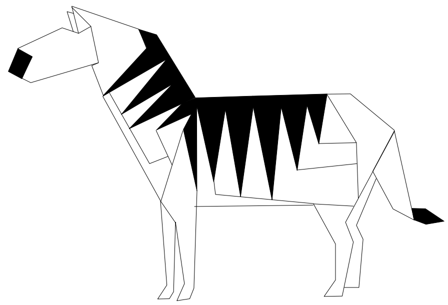


Figure 2.15: A flat folding of a square of paper, black on one side and white on the other side, designed by John Montroll [Mon91, pp. 94–103]. (Figure 1(b) of [DDM00].)

Of course, because of the use on thin strips, none of these methods lead to practical foldings, except for small examples or when the initial piece of paper is a thin strip. Nonetheless, the universality results of [DDM00] open the door to many new problems. For example, how small a square can be folded into a desired object, e.g., a $k \times k$ chessboard? This optimization problem remains open even in this special case, as do many other problems about finding efficient foldings of silhouettes, two-color patterns, and polyhedra.

2.2.2.4 Origami Tessellations

Roughly, an origami tessellation is a flat folding of a piece of paper based on a *tessellation* or *tiling* of the plane [GS87]. One way to make this notion more precise [KY89, Ver98] is to consider the whole plane as the piece of paper and define a *symmetric origami tessellation* to be a flat folding of the plane whose symmetry group is one of the 17 crystallographic groups. To avoid the symmetry restriction and allow aperiodic tilings, one way to define an origami tessellation is as any flat folding of the infinite plane in which no bounded region of the plane contains all of the vertices of the crease pattern. In such a folding, the crease pattern (at least) will necessarily form some kind of tiling.

An origami tessellation can be related to a tiling in various ways. Typically, there are large faces in the flat folded state that remain uncreased, and these faces are precisely the tiles of a tiling. One method for specifying creases around these faces is based on shrinking the tiles and introducing the dual tiling in the resulting gaps, using the notion of a hinged primal-dual tiling (see, e.g., [Wel91]). Bateman [Bat01] has formalized this method to the point of a computer implementation, called Tess.

Some of the key people working on origami tessellations include Alex Bateman [Bat01], Paulo Barreto [Bar94, Pal], S. Fujimoto [Fuj82], Thomas Hull (unpublished), Toshikazu Kawasaki [KY89], Robert Lang (unpublished), Chris Palmer [Pal, Pal94], and Helena Verrill [Ver98].

Unfortunately, much of the work on origami tessellations has not been written formally, so the exact computational results are unclear. Certainly a wide range of origami tessellations have been designed, many using precise algorithms. One of the simplest class of examples are the 11 Archimedian tilings. A characterization of which tilings lead to origami tessellations remains an intriguing open problem.

2.2.3 Origami Foldability

We distinguish origami design from *origami foldability* in which the starting point is a given crease pattern and the goal is to fold an origami that uses precisely these creases. (Arguably, this is a special case of our generic definition of origami design, but we find it a useful distinction.) The most common case studied is when the resulting origami should be flat, i.e., lie in a plane.

2.2.3.1 Local Foldability

For crease patterns with a single vertex, it is relatively easy to characterize flat foldability. Without specified crease directions, a single-vertex crease pattern is flat-foldable precisely if the alternate angles around the vertex sum to 180°; see Figure 2.16. This is known as Kawasaki's theorem [BH96, Hul94, Jus94, Kaw89]. When the angle condition is satisfied, a characterization of valid mountain-valley assignments and flat foldings can be found in linear time [BH96, Jus94], using Maekawa's theorem [BH96, Hul94, Jus94] and another theorem of Kawasaki [BH96, Hul94, Kaw89] about constraints on mountains and valleys. In particular, Hull has shown that the number of distinct mountain-valley assignments of a vertex can be computed in linear time [Hul01].

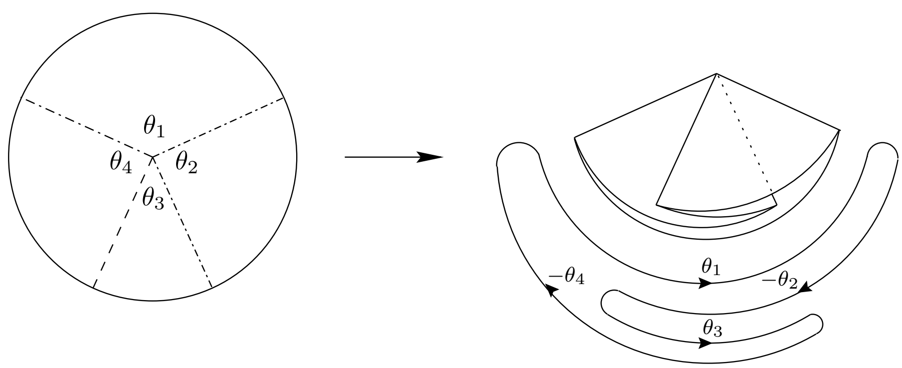


Figure 2.16: A locally flat-foldable vertex: $\theta_1 + \theta_3 + \cdots = \theta_2 + \theta_4 + \cdots = 180°$.

A crease pattern is called *locally foldable* if there is a mountain-valley assignment so that each vertex locally folds flat, i.e., a small disk around each vertex folds flat. Testing local foldability is nontrivial because each vertex has flexibility in its assignment, and these assignments must be chosen consistently: no crease should be assigned both mountain and valley by the two incident vertices. Bern and Hayes [BH96] proved that consistency can be resolved efficiently when it is possible: local foldability can be tested in linear time.

2.2.3.2 Existence of Folded States

Given a crease pattern, does it have a flat folded state? Bern and Hayes [BH96] have proved that this decision problem is NP-hard, and thus computationally intractable. Because local foldability is easy to test, the only difficult part is global foldability, or more precisely, computing a valid *overlap order* of the crease faces that fold to a common portion of the plane. Indeed, Bern and Hayes [BH96] prove that, given a crease pattern and a mountain-valley assignment that definitely folds flat, finding the overlap order of a flat folded state is NP-hard.

2.2.3.3 Equivalence to Continuous Folding Process

In the previous section we have alluded to the difference between two models of folding: the final folded state (specified by a crease pattern, mountain-valley or angle assignment, and overlap order) and a continuous motion to bring the paper to that folded state. Basically all results, in particular those described so far, have focused on the former model: proving that a folded state exists with the desired properties. Intuitively, by appropriately flexing the paper, any folded state can be reached by a continuous motion, so the two models should be equivalent. Only recently has this been proved, by Demaine and Mitchell [DM01], and so far only for rectangular pieces of paper. During their motion, the number of creases is finite at any moment in time, but the overall the number of creases is uncountably infinite. An interesting open problem is whether a finite crease pattern suffices.

The only other paper of which we are aware that proves the existence of continuous folding processes is [DD97]. This paper proves that every convex polygon can be folded into a uniaxial base via Lang's universal molecule [Lan01] without gussets. Furthermore, unlike [DM01], no additional creases are introduced during the motion, and each crease face remains flat. This result can be used to animate the folding process.

2.2.3.4 Map Folding: Sequence of Simple Folds

In contrast to the complex origami folds arising from reaching folded states [DD97, DM01], we can consider the less complex model of simple folds. A *simple fold* (or *book fold*) is a fold by $\pm 180°$ along a single line. Examples are shown in Figure 2.17. This model is closely related to "pureland origami" introduced by Smith [Smi76, Smi93].

We can ask the same foldability questions for a sequence of simple folds. Given a crease pattern, can it be folded flat via a sequence of simple folds? What if a particular mountain-valley assignment is imposed?

An interesting special case of these problems is *map folding* (see Figure 2.17): given a rectangle of paper with horizontal and vertical creases, each marked mountain or valley, can it be folded flat via a sequence of simple folds? Traditionally, map folding has been studied from a combinatorial point of view; see, e.g., [Lun68, Lun71]. Arkin, Bender, Demaine, Demaine, Mitchell, Sethia, and Skiena [ABD$^+$01b] have shown that deciding foldability of a map by simple folds can be solved in polynomial time. If the simple folds are required to fold all layers at once, the running time is at most $O(n \log n)$, and otherwise the running time is linear.

Surprisingly, slight generalizations of map folding are (weakly) NP-complete [ABD$^+$01b]. Deciding whether a rectangle with horizontal, vertical, and diagonal ($\pm 45°$) creases can be folded via a sequence of simple folds is NP-complete. Alternatively, if the piece of paper is

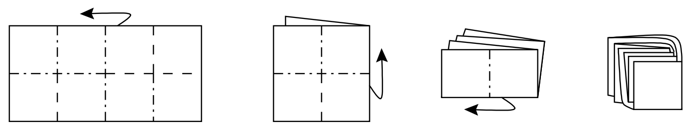


Figure 2.17: Folding a 2×4 map via a sequence of 3 simple folds.

more general, a polygon with horizontal and vertical sides, and the creases are only horizontal and vertical, the same problem becomes NP-complete.

These hardness results are *weak* in the sense that they leave open the existence of a *pseudopolynomial-time* algorithm, whose running time is polynomial in the total length of creases. Another intriguing open problem, posed by Jack Edmonds, is the complexity of deciding whether a map has a flat folded state, as opposed to a folding by a sequence of simple folds. Examples of maps in which these two notions of foldability differ are shown in Figure 2.18.

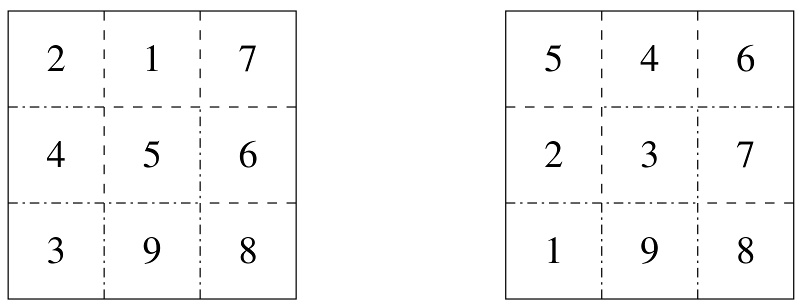


Figure 2.18: Two maps that cannot be folded by simple folds, but can be folded flat. (These are challenging puzzles.) The numbering indicates the overlap order of faces. (Figure 9 of the June 2001 full version of [ABD$^+$01b].)

2.2.4 One-Dimensional Origami

Consider a one-dimensional piece of paper, a line segment, marked at certain points with creases. It is most natural to fold such a piece of paper in the plane; a flat folding ends with the paper back on a line. Every one-dimensional piece of paper has a flat folded state, by alternating the creases between mountain and valley. If the mountain-valley assignment is prescribed, the crease pattern may not be flat-foldable, but this can be tested in linear time [ABD$^+$01b]. Indeed, it is proved in [ABD$^+$01b] that, in one dimension, the existence of a flat folded state is equivalent to foldability via a sequence of simple folds. In particular, any flat folded state can be reached by a continuous folding motion.

A nonflat folded state (at no point touching itself) of a one-dimensional piece of paper can also be reached by a continuous motion while preserving length and avoiding self-intersection. An easy way to prove this [DD97] is by introducing several extra creases, repeatedly “rolling” the folded state into a “cinnamon bun,” then unrolling it into a straight line. No self-intersection will occur provided we introduce creases at intervals of length less than the minimum feature size of the folded state (the minimum distance between two nonincident edges). While this approach applies very generally, e.g., also for trees [DD97], the extra creases are unnecessary for a linear piece of paper. In this case, reaching a folded state is precisely the carpenter’s rule problem discussed in Section 2.1.

2.3 Polyhedra

A standard method for building a model of a polyhedron is to cut out a flat *net* or *unfolding*, fold it up, and glue the edges together so as to make precisely the desired surface. Given the polyhedron of interest, a natural problem is to find a suitable unfolding. On the other hand, given a polygonal piece of paper, we might ask whether it can be folded and its edges can be glued together so as to form a convex polyhedron. These two questions are addressed in Sections 2.3.1 and 2.3.2, respectively. Section 2.3.3 extends different forms of the latter question to nonconvex polyhedra. Section 2.3.4 connects these problems to linkage folding. Section 2.3.5 discusses a different problem related to paper folding.

2.3.1 Unfolding Polyhedra

A classic open problem is whether (the surface of) every convex polyhedron can be cut along some of its edges and unfolded into one flat piece without overlap [She75, O'R98]. Such *edge-unfoldings* go back to Dürer [Dür77], and have important practical applications in manufacturing, such as sheet-metal bending [O'R98, Wan97]. It is widely conjectured that the answer to this question is yes, but all attempts at a solution have so far failed. Experiments by Schevon [Sch89, O'R98] suggest that a random edge-unfolding of a random polytope overlaps with probability 1. Of course, such a result would not preclude, for every polytope, the existence of at least one nonoverlapping edge-unfolding, or even that a large but subconstant fraction of the polytope's edge-unfoldings do not overlap.

Instead of answering this difficult question directly, we can examine to what extent it can be generalized. In particular, define a polyhedron to be *topologically convex* if its 1-skeleton (graph) is the 1-skeleton of a convex polyhedron. Does every topologically convex polyhedron have an edge-unfolding? In particular, every polyhedron made up of convex faces and homeomorphic to a sphere is topologically convex; can they all be edge-unfolded? This problem was posed by Schevon [Sch87].

Bern, Demaine, Eppstein, Kuo, Mantler, and Snoeyink [BDE$^+$01] have shown that the answer to both these questions is no: there is a polyhedron made up of triangles and homeomorphic to a sphere that has no (one-piece, nonoverlapping) edge-unfolding. The polyhedron is shown in Figure 2.19. It consists of four "hats" glued to the faces of a regular tetrahedron, such that only the peaks of the hats have positive curvature, that is, have less than $360°$ of incident material. This property limits the unfoldings significantly, because (1) any set of cuts must avoid cycles in order to create a one-piece unfolding, and (2) a leaf in a forest of cuts can only lie at a positive-curvature vertex of the polyhedron: a leaf at a negative-curvature vertex (more than $360°$ of incident material) would cause local overlap.

The complexity of deciding whether a given topologically convex polyhedron can be edge-unfolded remains open.

Another intriguing open problem in this area is whether every polyhedron homeomorphic to a sphere has *some* one-piece unfolding, not necessarily using cuts along edges. It is known that every convex polyhedron has an unfolding in this model, allowing cuts across the faces of the polytope. Specifically, the *star unfolding* [AAOS97, AO92] is the set of shortest paths from a common source point to each vertex of the polytope, and the *source unfolding* [MMP87] is the set of points with more than one shortest path to a common source. Both of these unfoldings avoid overlap, the star unfolding being the more difficult case to establish [AO92].

But many nonconvex polyhedra also have such unfoldings. For example, Figure 2.19 illustrates one for the polyhedron described above. Biedl, Demaine, Demaine, Lubiw, Overmars, O'Rourke, Robbins, and Whitesides [BDD+98] have shown how to unfold many orthogonal polyhedra, even with holes and knotted topology, although it remains open whether all orthogonal polyhedra can be unfolded. The only known scenario that prevents unfolding altogether [BDE+01] is a polyhedron with a single vertex of negative curvature (see Figure 2.20), but this requires the polyhedron to have boundary (edges incident to only one face).

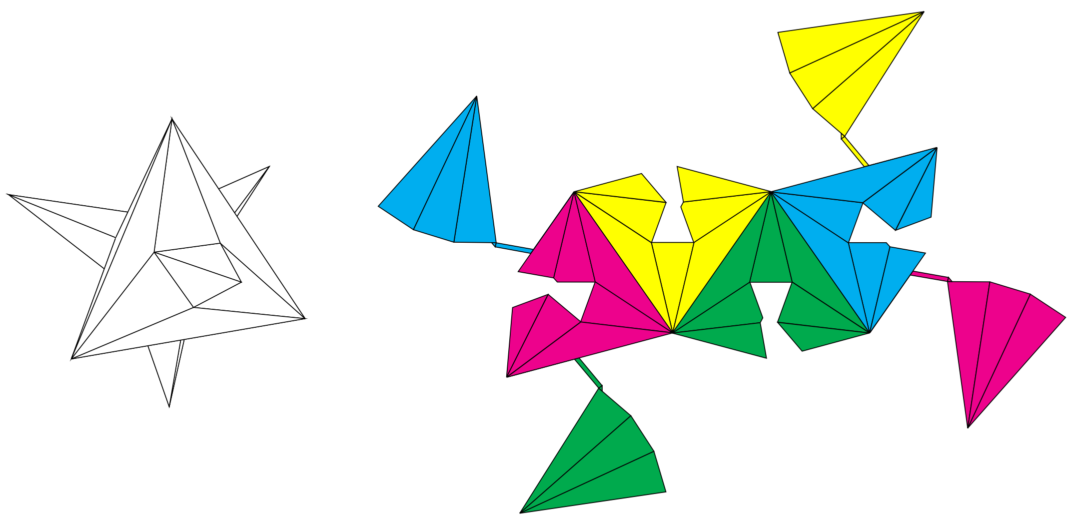


Figure 2.19: (Left) Simplicial polyhedron with no edge-unfolding. (Right) An unfolding when cuts are allowed across faces.

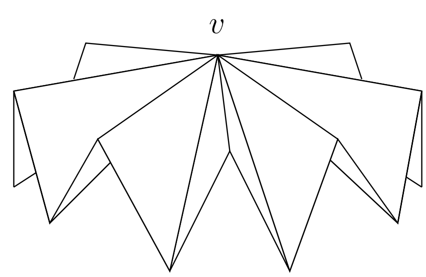


Figure 2.20: A polyhedron with boundary that has no one-piece unfolding even when cuts are allowed across faces. Vertex v has negative curvature, that is, more than 360° of incident material.

A recent approach to both edge-unfoldings of convex polyhedra and general unfoldings of nonconvex polyhedra is the notion of "vertex-unfolding" [DEE+01]; see Figure 2.21. Specifically, a *vertex-unfolding* may only cut along edges of the polyhedron (like an edge-unfolding) but permits the facets to remain connected only at their vertices (instead of at edges as in edge-unfolding). Thus, a vertex-unfolding is connected, but its interior may be disconnected, "pinching" at a vertex.

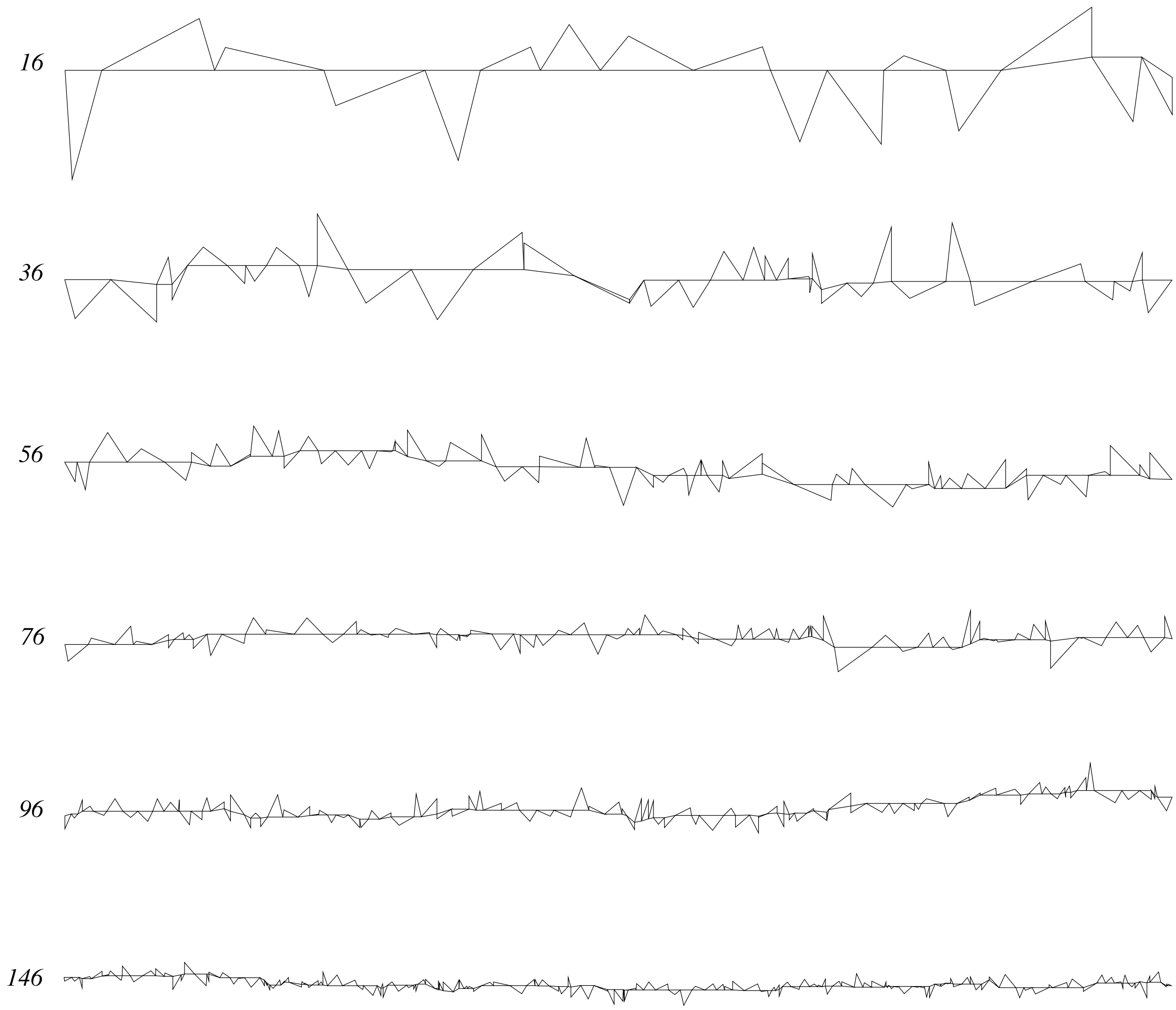


Figure 2.21: Vertex-unfoldings of random convex polyhedra with varying numbers of triangles. (Figure 3 of [DEE$^+$01].)

Demaine, Eppstein, Erickson, Hart, and O'Rourke [DEE$^+$01] have proved that every simplicial manifold (including triangulated polyhedra possibly with boundary, and more generally in any dimension) has a nonoverlapping vertex-unfolding. It remains open to what extent vertex-unfoldings exist for polyhedra with nontriangular faces; for example, does every topologically convex polyhedron have a vertex-unfolding?

2.3.2 Folding Polygons into Convex Polyhedra

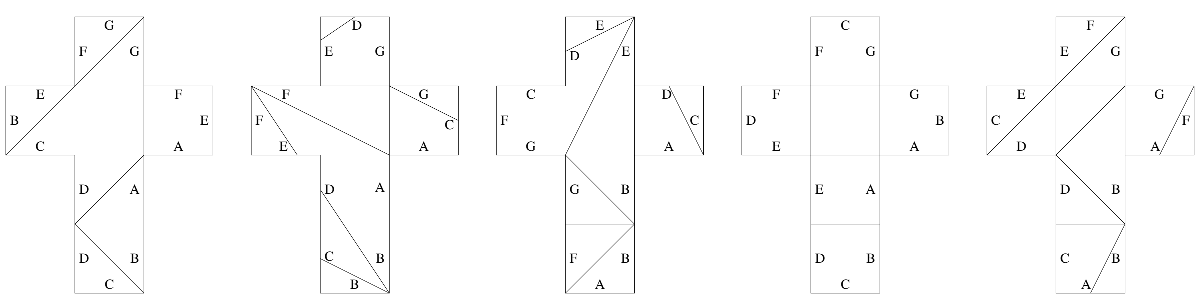
In addition to unfolding polyhedra into simple planar polygons, we can consider the reverse problem of folding polygons into polyhedra. More precisely, when can a polygon have its boundary glued together, with each portion gluing to portions of matching length, and the resulting topological object be folded into a *convex* polyhedron? (There is almost too much flexibility with nonconvex polyhedra for this problem, but see Section 2.3.3 for related problems of interest in this context.) A particular kind of gluing is an *edge-to-edge* gluing, in which each entire edge of the polygon is glued to precisely one other edge of the polygon. The existence of such a gluing requires a perfect pairing of edges with matching lengths.

2.3.2.1 Edge-to-Edge Gluings

Introducing and pioneering this area, Lubiw and O'Rourke [LO96] showed how to test in polynomial time whether a polygon has an edge-to-edge gluing that can be folded into a convex polyhedron, and how to list all such edge-to-edge gluings in exponential time. A key tool in their work is a theorem of the famous Russian geometer A. D. Aleksandrov [Ale50]. The theorem states that a topological gluing can be realized geometrically by a convex polyhedron precisely if the gluing is topologically a sphere, and at most 360° of material is glued to any one point—that is, every point should have nonnegative curvature.

Based on this tool, Lubiw and O'Rourke use dynamic programming to develop their algorithms. There are $\Omega(n^2)$ subproblems corresponding to gluing subchains of the polygon, assuming that the two ends of the subchain have already been glued together. These subproblems are additionally parameterized by how much angle of material remains at the point to which the two ends of the chain glue in order to remain positive curvature. It is this parameterization that causes enumerating all gluings to take exponential time. But for the decision problem of the existence of any gluing, the remaining angle at the ends only needs to be bounded, and only polynomially many subproblems need to be considered.

A particularly surprising discovery from this work [LO96] is that the well-known cross unfolding of the cube can be folded into exactly five convex polyhedra by edge-to-edge gluing: a doubly covered (flat) quadrangle, an (irregular) tetrahedron, a pentahedron, the cube, and an (irregular) octahedron. See Figure 2.22 for crease patterns and gluing instructions. These foldings are the subject of a video [DDL+99a].



(a) Doubly covered quadrangle (b) Tetrahedron (c) Pentahedron (d) Cube (e) Octahedron

Figure 2.22: The five edge-to-edge gluings of the Latin cross [LO96].

2.3.2.2 Non-Edge-to-Edge Gluings

More recently, Demaine, Demaine, Lubiw, and O'Rourke [DDLO00b, DDLO] have extended this work in various directions, in particular to non-edge-to-edge gluings.

In contrast to edge-to-edge gluings, any convex polygon can be glued into a continuum of distinct convex polyhedra, making it more difficult for an algorithm to enumerate all gluings of a given polygon. Fortunately, there are only finitely many *combinatorially distinct* gluings of any polygon. For convex polygons, there are only polynomially many combinatorially distinct gluings, and they can be enumerated for a given convex polygon in polynomial time. This result generalizes to any polygon in which there is a constant bound on the sharpest angle. For general nonconvex polygons, there can be exponentially many ($2^{\Theta(n)}$) combinatorially distinct gluings, but only that many. Again this corresponds to an algorithm running in $2^{O(n)}$ time. Because of the exponential worst-case lower bound on the number of combinatorially distinct gluings, we are justified both here and in the enumeration algorithm of [LO96] to spend exponential time. It remains open whether there is an output-sensitive algorithm, whose running time is polynomial in the number of resulting gluings, or in the number of gluings desired by the user. For non-edge-to-edge gluings, it even remains open whether we can solve in polynomial time the decision question of whether a gluing exists.

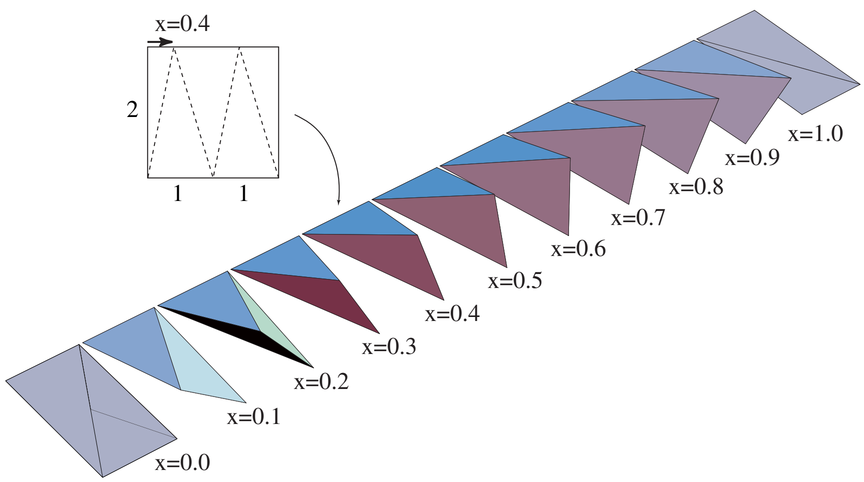


Figure 2.23: A continuum of tetrahedra obtained by gluing a square. (Figure 3 of [DDLO].)

The algorithms for enumerating all non-edge-to-edge gluings have been implemented independently by Anna Lubiw (July 2000) and by Koichi Hirata [Hir00] (June 2000). These programs have been applied to the example of the Latin cross. The surprising result is that there are many more non-edge-to-edge gluings: a total of 85 distinct gluings (43 modulo symmetry) resulting in 21 distinct polyhedra. See [DDLO00a].

2.3.2.3 Constructing Polyhedra

Another intriguing open problem in this area [DDLO] remains relatively unexplored: Aleksandrov's theorem implies that any valid gluing (homeomorphic to a sphere and having nonnegative curvature everywhere) can be folded into a unique convex polyhedron, but how efficiently can this polyhedron be constructed? The key difficulty here is to determine the dihedral angles of the polyhedron, that is, by how much each crease is folded. Finding a (superset of) the creases is straightforward (personal communication with Boris Aronov, June 1998): every edge of the polyhedron is a shortest path between two positive-curvature vertices, so compute all-pairs shortest paths in the polyhedral metric defined by the gluing [CH96, KO00, Kap99].

Recently, Sabitov [Sab00] presented a finite algorithm for this reconstruction problem, with a conjectured running time of around $\Theta(2^{n^2})$. The algorithm is based on the Connelly-Sabitov-Walz Bellows Theorem [CSW97] which characterizes the volume of a polyhedron in terms of the geometry of its surface, independent of how that geometry is embedded in 3-space.

2.3.3 Folding Nets into Nonconvex Polyhedra

Define a *net* to be a connected edge-to-edge gluing of polygons to form a tree structure, the edges shared by polygons denoting creases. An open problem mentioned in Section 2.3.2.3 is deciding whether a given net can be folded into a convex polyhedron using only the given creases. More generally, we can ask whether a given net folds into a nonconvex polyhedron. Now Aleksandrov's theorem and Cauchy's rigidity theorem do not apply, so for a given gluing we are no longer easily guaranteed existence or uniqueness.

Given the dihedral angles associated with creases in the net, it is easy to decide foldability in polynomial time [BLS99, Sun99]: we only need to check that edges match up and no two faces cross. Without the dihedral angles, when does a given net fold into any polyhedron? Biedl, Lubiw, and Sun [BLS99, Sun99] have proved a closely related problem to be weakly NP-complete: does a given orthogonal net (each face is an orthogonal polygon) fold into an orthogonal polyhedron? The difference with this problem is that it constrains each dihedral angle to be $\pm 90°$.

2.3.4 Continuously Unfolding Polyhedra

In the context of paper folding, specifically Section 2.2.3.3, the results in the previous section are essentially about folded states, and not about the continuous process of reaching such states. Assuming the results described in Section 2.2.3.3 [DM01] hold for nonrectangular pieces of paper, a continuous folding process can be obtained, but they will introduce additional creases. In manufacturing applications, it is natural to ask for a folding process of a polygon into a polyhedron (possibly nonconvex), or a continuous unfolding of a polyhedron into a polygon, that keeps the faces rigid and avoids crossings. Does such a motion always exist?

As mentioned in Section 2.1.4, these problems are generalizations of linkage-folding problems. In particular, as shown in Figure 2.10, a hinged collection of rigid polygons in space can be made to act exactly like a linkage in 2D. By closing the arc into a cycle, adding a top and bottom face to the extrusion, and enforcing a particular (strip) unfolding, this hinged collection can be turned into a nonconvex polyhedron, homeomorphic to a sphere, whose continuous unfolding is equivalent to straightening a polygonal arc in the plane (Section 2.1.2.1).

An interesting open special case is all edge-unfoldings (nets) of convex polyhedra. Such motions were implemented in the video showing the 5 edge-to-edge gluings of the cross [DDL+99a] in an obvious manner: affinely interpolate all dihedral angles from flat (180°) to the target angle, so that each angle changes at a constant speed and all angles reach their targets at the same time. This method appears to avoid self-intersection in the examples encountered, but presumably this method does not avoid self-intersection for all unfoldings of all convex polyhedra. When does it work? For example, does this approach work on a large class of unfoldings such as star unfoldings or source unfoldings of convex polyhedra?

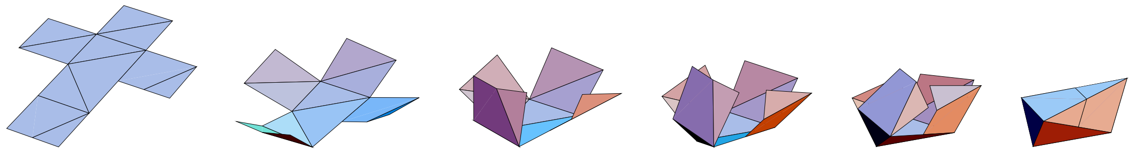


Figure 2.24: Folding the Latin cross into an octahedron, according to the crease pattern in Figure 2.22(e), by affinely interpolating all dihedral angles. (Figure 2 of [DDL+99a].)

One special case that has been explored is orthogonal polyhedra. Specifically, Biedl, Lubiw, and Sun [BLS99, Sun99] have proved that there is an edge-unfolding of an orthogonal polyhedron (which is an orthogonal net) that cannot be folded into the orthogonal polyhedron by a continuous motion that keeps the faces rigid and avoids self-intersection. The basis for their example is the locked polygonal arc in 3D (Figure 2.9), converted into an orthogonal locked polygonal arc in 3D, and then "thickened" into an orthogonal tube. A single chain of faces in the unfolding is what prevents the continuous foldability.

One would expect, analogous to the results described in Section 2.2.3.3 [DM01], that collections of polygons hinged together into a tree can be folded into all possible configurations if we allow additional creases during the motion. However, this extension (equivalent to a polygonal piece of paper) remains open. A particularly interesting version of this question, posed in [BLS99], is whether a finite number of additional creases suffice.

2.3.5 Flattening Polyhedra

Suppose instead of allowing cuts to a polyhedral surface in order to fold it, we treat it as a piece of paper and fold as in origami. We run into the same dichotomy as in Section 2.2.3.2: do we want a continuous motion of the polyhedron, or does a description of the final folded state suffice? If we start with a convex polyhedron, and each face of the crease pattern must remain rigid during the folding, then Connelly's extension [Con80] of Cauchy's rigidity theorem from 1813 [Cau13, Cro97] says that the polyhedron cannot fold at all. Even if we start with a nonconvex polyhedron and keep each face of the crease pattern rigid, the Bellows Theorem [CSW97] says that the volume of the polyhedron cannot change, so foldings are limited. However, if we allow the paper to curve (e.g., introduce new creases) during the motion, as in origami, then folding becomes surprisingly flexible. For example, a cone can be inverted [Con93]; see Figure 2.25.

A natural question [DDL00] is whether every polyhedron can be *flattened*: folded into a flat origami. Intuitively, this can be achieved by applying force to the polyhedral model, but in practice this can easily lead to tearing. There is an interesting connection of this problem to a higher-dimensional version of the fold-and-cut problem from Section 2.2.2.2. Given any polyhedral complex, can $\mathbb{R}^3$ be folded (through $\mathbb{R}^4$) "flat" into $\mathbb{R}^3$ so that the surface of the polyhedral complex maps to a common plane, and nothing else maps to that plane? While the applicability of four dimensions is difficult to imagine, the problem's restriction to the surface of the complex is quite practical, e.g. in packaging: *flatten* the polyhedral complex into a flat folded state, without cutting or stretching the paper.

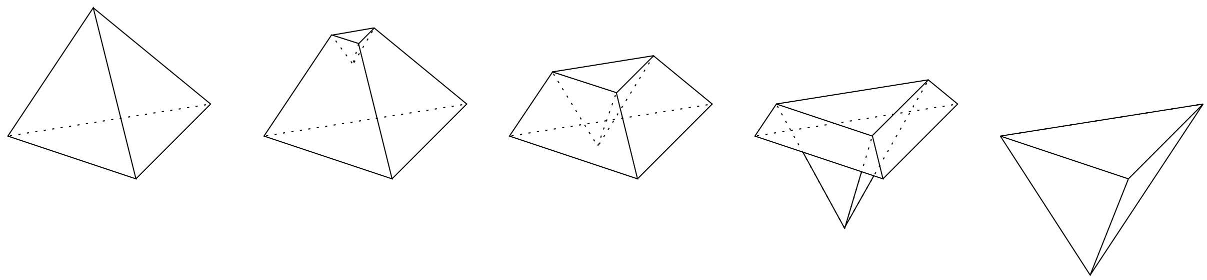


Figure 2.25: Inverting a tetrahedral cone by a continuous isometric motion. Based on Figure 2.5 of [Con93].

The flattening problem remains open if we desire a continuous folding process into the flat state. If we instead focus on the existence of a flat folded state of a polyhedron, then much more is known. Demaine, Demaine, and Lubiw [DDL00] have shown how to flatten several classes of polyhedra, including convex polyhedra and orthogonal polyhedra. See Figure 2.26 for an example. Recently (March 2001), together with Barry Hayes, we have shown that all polyhedra have flat folded states. We conjecture further that every polyhedral complex can be flattened.

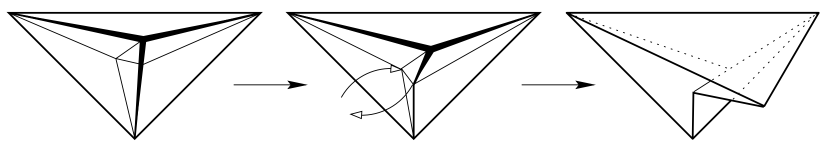


Figure 2.26: Flattening a tetrahedron, from left to right. Note that the faces are not flat in the middle picture.

A natural question is whether the methods of Demaine and Mitchell [DM01] described in Section 2.2.3.3 can be generalized to show that these folded states induce continuous folding motions as in Figure 2.25.

Chapter 3

Reconfiguring Convex Polygons

This chapter proves that there is a motion from any convex polygon to any convex polygon having the same counterclockwise sequence of edge lengths, that preserves the lengths of the edges and keeps the polygon convex at all times. Furthermore, the motion is "direct" (avoiding any intermediate canonical configuration like a subdivided triangle) in the sense that each angle changes monotonically throughout the motion. In contrast, we show that it is impossible to achieve such a result with each vertex-to-vertex distance changing monotonically.

This chapter is joint work with Oswin Aichholzer, Jeff Erickson, Ferran Hurtado, Mark Overmars, Michael Soss, and Godfried T. Toussaint. Versions appear in the 12th Canadian Conference on Computational Geometry [ADE$^+$00] and *Computational Geometry: Theory and Applications* [ADE$^+$01].

3.1 Introduction

First we briefly review the relevant components of Chapter 2. This chapter is concerned with *linkages* modeled by polygons, primarily in the plane, whose vertices represent hinges and whose edges represent rigid bars. A fundamental question about such linkages is whether it is possible to reach every polygon with the same sequence of edge lengths by motions that preserve the edge lengths. Several papers have shown that the answer to this question is yes for various types of polygons; we call this a *universality* result. If edges are allowed to cross each other, then this is true in every dimension [LW95, Sal73]. If edges are not allowed to cross, universality does not hold in general for polygons in 3D [CJ98, BDD$^+$01a], but has been shown for polygons in the plane and motions in 3D [AGP99, BDD$^+$01a], for polygons and motions in the plane (Chapter 4), for polygons in 3D with simple projections [CKM$^+$01], and for all polygons in 4D and higher dimensions [CO99].

All of these papers show universality by proving that every polygon can be *convexified*, that is, moved to a convex (planar) polygon while preserving edge lengths. Convex polygons are used as an intermediate state; because motions can be reversed and concatenated, all that remains is to show that a convex polygon can be moved to every other convex polygon with the same counterclockwise sequence of edge lengths. This fact is established in [LW95] when edges are allowed to cross, and before our work no proof had been published for the case in which edges cannot cross. Although the method in [LW95] may not actually cause edges to cross, this property has not been established, and we instead opt for a simpler solution.

The basic idea in the proof in [LW95] of universality of convex polygons is to show how to reconfigure every convex polygon into another intermediate state, a "canonical triangle." In this chapter, we show that this intermediate state can be avoided. Specifically, a convex polygon can be moved to any other convex polygon with the same counterclockwise sequence of edge lengths in such a way that each vertex angle varies monotonically with time (either never increasing or never decreasing). In this sense, the motion goes directly from the source to the destination. Our motion is also of the simplest type possible [BDL$^+$99]: it can be decomposed into a linear number of *moves*, each of which changes only four joint angles. In contrast, the motion in [LW95] changes up to five joint angles at once.

This chapter is organized as follows. In Section 3.2 we introduce some basic notation that we will use throughout the chapter. Section 3.3 proves the theorem about angle-monotone motions in the plane, using an old lemma of Cauchy and Steinitz. Section 3.4 shows an example in which a different type of monotonicity cannot be achieved.

3.2 Notation

For a polygon P, we denote its vertices by $v_1, \ldots, v_n$ in counterclockwise order, its edges by $e_i = (v_i, v_{i+1})$, and its edge lengths by $\ell_i = |e_i| = |v_i - v_{i+1}|$. Throughout, index arithmetic is modulo n.

A *convex configuration* of edge lengths (positive real numbers) $\ell_1, \ldots, \ell_n$ is a convex polygon with those edge lengths in counterclockwise order. The convex polygon is allowed to have "straight" vertices with angle π, and edges are allowed to overlap. (These assumptions simplify the statement of Lemma 3.1, but are not necessary for our purposes: if the desired configurations avoid straight angles and/or overlap, then so will our motions.) A well-known result characterizes the edge lengths for which convex configurations exist:

Lemma 3.1 (Lemma 3.1 of [LW95]) *The edge lengths $\ell_1, \ldots, \ell_n$ admit a convex configuration precisely if $\ell_i \leq \sum_{j \neq i} \ell_j$ for all i.*

A *motion* or *reconfiguration* is a continuous function from the unit interval $[0, 1]$ (representing time) to a configuration, where each *configuration* is a polygon with the same counterclockwise sequence of edge lengths. An *angle-monotone motion* is a motion in which each vertex angle is a monotone function in time.

3.3 Angle-Monotone Motions

Consider two convex configurations C and C' of the same sequence of edge lengths. We think of C as the source configuration and C' as the destination configuration. Label each angle of C by $+$ if it needs to get bigger in order to match the corresponding angle in C', by $-$ if it needs to get smaller, or by 0 if they already match.

This set up is exactly what arises in the proof of Cauchy's theorem about the rigidity of convex polyhedra [Cau13, Cro97], except that in Cauchy's application the polygon is on the sphere. We also need his key lemma about *sign alternations*: transitions from $+$ to $-$ or from $-$ to $+$ among consecutive vertices along the polygon boundary. Cauchy's original proof of this lemma (in 1813) had an error, noticed and corrected over a century later by Steinitz in 1934 [SR34].

Lemma 3.2 (Cauchy-Steinitz Lemma [SR34, Cro97, SZ67]) *In a $+, -, 0$ labeling that comes from two distinct convex configurations, there are at least four sign alternations.*

See [SR34] for Steinitz's original (complicated) proof, [Cro97] for a simpler proof due to Isaac J. Schoenberg, or [SZ67] for another elementary proof. Here we present the main argument:
Proof (Sketch): Because the configurations are distinct, not all labels are 0. By circularity, the number of alternations between $+$ and $-$ (ignoring 0's) is even. It cannot be zero, because there is no motion of any polygon that increases or decreases all angles. It cannot be two, because then there is a chain of increasing angles and a chain of decreasing angles; the former chain specifies that the ends of the chain should get farther apart, whereas the latter chain specifies the opposite. It is this last part of the argument that needs careful analysis. □

Thus we obtain vertices v_i, v_j, v_k, v_l, in cyclic order around the polygon, whose angles are labeled $+, -, +, -$ in that order; see Figure 3.1. Now we treat the convex polygon as the quadrangle v_i, v_j, v_k, v_l, holding rigid the subchains of the convex polygon connecting those four vertices: $[v_i, v_j]$, $[v_j, v_k]$, $[v_k, v_l]$, and $[v_l, v_i]$. We flex the quadrangle according to the $+, -, +, -$ angular labels, stopping once one angle matches the desired value in C', i.e., a label changes from $+$ or $-$ to 0.

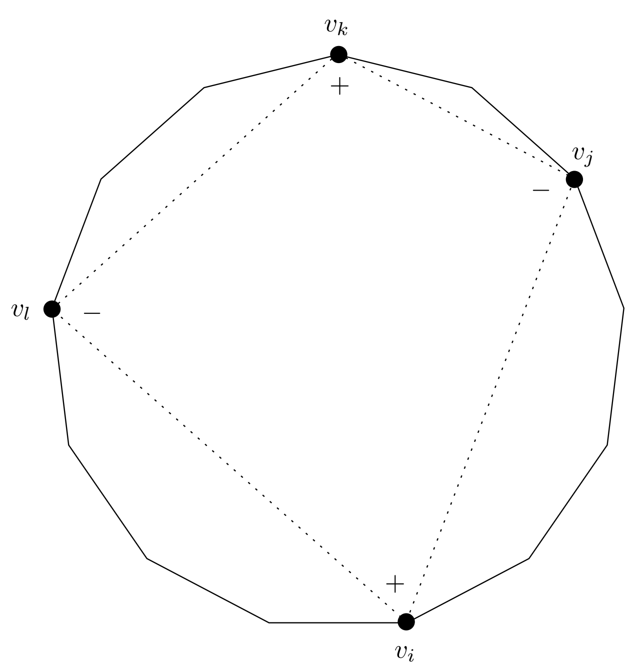


Figure 3.1: Applying a quadrangle motion to a convex polygon by taking vertices labeled $+, -, +, -$ in that order.

To do this we need a lemma about reconfiguring convex quadrangles:

Lemma 3.3 *Given a convex quadrangle v_1, v_2, v_3, v_4, there is a motion that decreases the angles at v_1 and v_3, and increases the angles at v_2 and v_4. The motion can continue until one of the decreasing angles reaches 0 or one of the increasing angles reaches π.*

Proof: We consider the following viewpoint: v_1 is pinned to the plane, and v_3 moves along the directed line from v_1 to v_3 (see Figure 3.2). The motions of v_2 and v_4 are determined by maintaining their distances to v_1 and v_3. Applying Euclid's Proposition I.25[1] [Hea56] to triangle v_1, v_2, v_3, because $|v_1 - v_3|$ is increasing, so is the angle at v_2. Similarly, the angle at v_4 is increasing throughout the motion. Because no angle goes past 0 or π, we maintain a convex quadrangle throughout the motion, so by the Cauchy-Steinitz lemma (Lemma 3.2), there must be at least four sign alternations when compared to any future quadrangle we will visit. This proves that the angles at v_1 and v_3 decrease throughout the motion. □

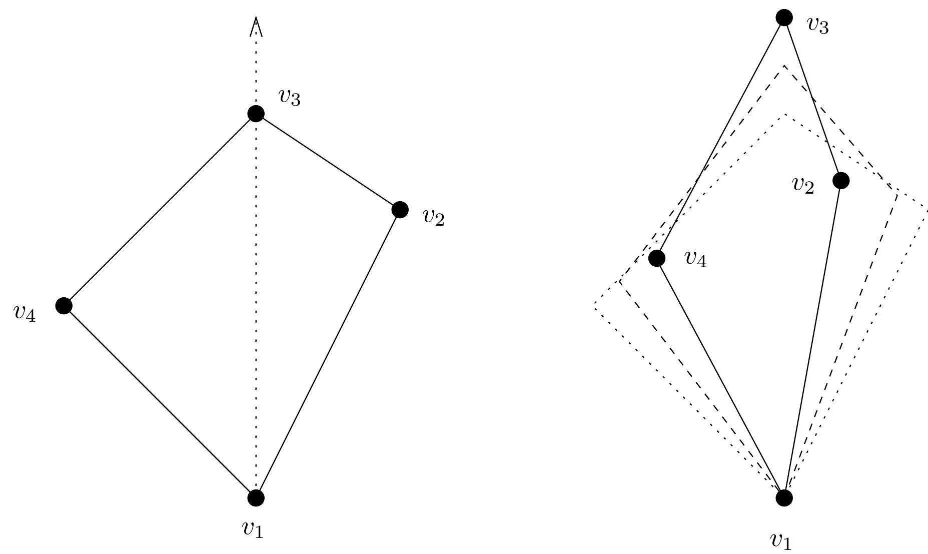


Figure 3.2: Moving a convex quadrangle as in Lemma 3.3.

We are now in the position to prove the main theorem of this chapter:

Theorem 3.4 *Given two convex configurations C, C' of the same edge lengths $\ell_1, \ldots, \ell_n$, there is an angle-monotone motion from C to C' that involves $O(n)$ moves, each of which changes only four vertex angles.*

Proof: Consider the $+, -, 0$ labeling of C to reach C'. By Lemma 3.2, we can find vertices v_i, v_j, v_k, v_l in cyclic order around the polygon, whose angles are labeled $+, -, +, -$ in that order; see Figure 3.1. By specifying the subchains between these vertices to move rigidly, we obtain a convex quadrangle. Move this quadrangle according to Lemma 3.3 until one of the four angles matches the angle in C'. (Because of our stopping condition, no angle will go past 0 or π.) Repeat this process until all angles match. The result is a sequence of motions from C to C'. There are at most n moves, because each motion changes the label of an angle from $+$ or $-$ to 0, and that label persists. □

Note that if the configurations C, C' have no overlapping edges, then they have no zero angles, so by angle-monotonicity, the motion never creates a zero angle, and hence avoids overlapping edges throughout.

[1] "If two triangles have the two sides equal to two sides respectively, but have the base greater than the base, they will also have the one of the angles contained by the equal straight lines greater than the other." [Hea56, p. 297]

Next we prove analogous algorithmic results to Lemma 3.3 and Thereom 3.4:

Proposition 3.5 *On a machine supporting basic arithmetic and roots, any snapshot of the motion in Lemma 3.3 can be computed in constant time, as can the time at which a specified angle reaches desired values in sine and cosine.*

Proof: We can implement the motion as described in the proof of Lemma 3.3, in which v_1 is pinned and v_3 moves along the line through v_1 and v_3. Thus v_3's motion can be described as $v_3 + t(v_3 - v_1)$. Given a particular time t and hence a particular position for v_3, the remaining vertices v_2 and v_4 are each defined by the intersection of two known circles. This circle-intersection predicate involves basic arithmetic and solving a quadratic polynomial, which can be done with basic arithmetic and a square root. The desired root of the quadratic polynomial can be determined by ensuring that v_2 and v_4 stay on the correct side of the line through v_1 and v_3. This concludes the construction of any snapshot of the motion.

Next we describe how to compute when an angle reaches a particular value. By the law of cosines, the cosine of the angle at v_2 is equal to

$$\frac{|p_1 - p_3|^2 - |p_1 - p_2|^2 - |p_2 - p_3|^2}{-2|p_1 - p_2||p_2 - p_3|}.$$

Only the $|p_1 - p_3|$ distance is not a constant, and it is of the form $a + tb$ for constants a and b. Thus, solving for the time at which the cosine of the angle at v_2 becomes a given value only requires solving a quadratic polynomial. The same approach works for v_4.

For v_1 and v_3, we split each angle into two by cutting along the line through v_1 and v_3, and write these angles in terms of the angles at v_2 and v_3 using the law of sines. For example, the sine of the angle at v_1 on the v_2 side is equal to $|v_2 - v_3|/|v_1 - v_3|$ times the sine of the angle at v_2. Solving for the time at which this sine equals a particular value also leads to a quadratic polynomial. Now we combine the two angles at each of v_1 and v_3 using the sum formula $\sin(a+b) = \sin a \cos b + \sin b \cos a$. This multiplication of terms causes the polynomials to blow up to quartic (degree 4), which can still be solved using basic arithmetic, square roots, and cube roots. □

Proposition 3.5 and its proof are similar to Lemma 7 in [BDL+99] about motions of a nonconvex quadrangle.

Proposition 3.6 *All $O(n)$ moves of the motion in Theorem 3.4 can be computed (in the sense of implicitly encoding each quadrangle motion so that Proposition 3.5 can be applied) in $O(n)$ time on a pointer machine supporting basic arithmetic and roots.*

Proof: The first part of the algorithm is keep track of four vertices, v_i, v_j, v_k, v_l, suitable for a quadrangle motion, i.e., labeled $+, -, +, -$. In fact we maintain additional structure: four consecutive maximal blocks I, J, K, L for which all vertices in I and K are labeled either $+$ or 0, and all vertices in J and L are labeled either $-$ or 0. More precisely, we keep track of the first and last vertex not labeled 0 in each block. This structure can be determined initially in linear time by scanning along the polygon's vertices in order. The desired vertices v_i, v_j, v_k, v_l are each defined to be the first vertex not labeled 0 in the corresponding block. When the label of one of them switches to 0, the vertex and the block's first vertex are set to the next vertex not labeled 0 in the block. If no such vertex exists (the block is now empty), we make the following modifications. If I becomes empty, we set it to the block of $+$'s after L. Similarly, if L becomes empty, we set it to the block of $-$'s before I. If K becomes empty, we set it to the block after L, we merge the blocks J and L to produce a new block J, and set L to the block of $-$'s after K. The case of J becoming empty is symmetric.

The second part of the algorithm is to apply the quadrangle motions from Lemma 3.3. This involves computing the time at which the quadrangle motion stops, and then updating the coordinates. We apply Proposition 3.5 four times to determine when each of the angles would reach its desired value, and take the minimum of these times. A fifth application of Proposition 3.5 computes the result of this motion and updates the vertex coordinates. □

3.4 Distance-Monotone Motions

We have shown that an angle-monotone motion between any two convex configurations of a common sequence of edge lengths can be computed in linear time. On the other hand, we prove that a *distance-monotone* motion, in which every distance between a pair of vertices varies monotonically with time, is impossible in general. (This is in direct contrast to convexification of a polygon (Chapter 4), where all distances can be made to increase.) Consider the left and right configurations in Figure 3.3. Because the dotted lines are the same length in both configurations, these distances must be preserved throughout the motion; in other words, the chains v_1, v_2, v_3 and v_4, v_5, v_6 must move rigidly. The problem is thus reduced to moving a quadrangle v_1, v_3, v_4, v_6, which can be moved in only two different ways. Only one motion decreases $|v_1 - v_4|$ and increases $|v_3 - v_6|$ as desired, but then the distance $|v_2 - v_5|$ increases and later decreases. Specifically, the distance in the middle configuration is more than 0.6% larger than the (equal) distances in the left and right configurations.

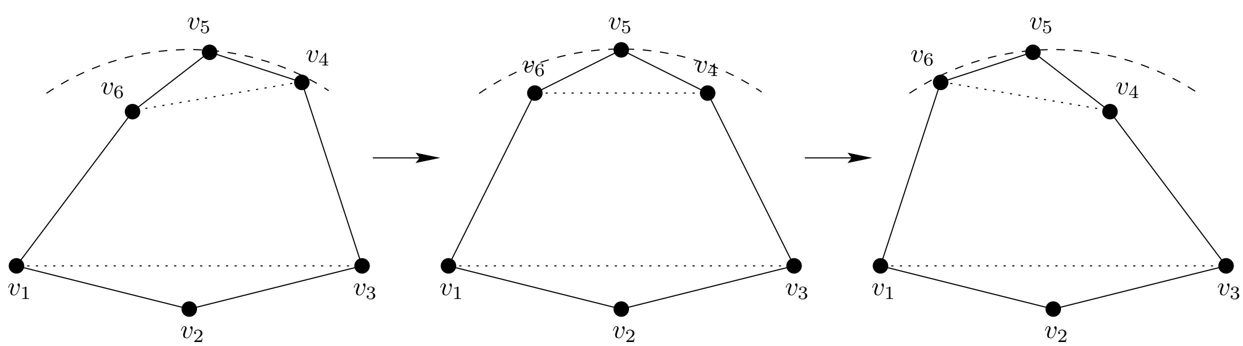


Figure 3.3: (Left and right) An example for which a distance-monotone motion is impossible. (Middle) The transition between $|v_2 - v_5|$ increasing and decreasing.

Chapter 4

Straightening Polygonal Arcs and Convexifying Polygonal Cycles

Consider a planar linkage, consisting of disjoint polygonal arcs and cycles of rigid bars joined at incident endpoints (polygonal chains), with the property that no cycle surrounds another arc or cycle. This chapter proves that the linkage can be continuously moved so that the arcs become straight, the cycles become convex, and no bars cross while preserving the bar lengths. Furthermore, our motion is piecewise-differentiable, does not decrease the distance between any pair of vertices, and preserves any symmetry present in the initial configuration. In particular, this result settles the well-studied carpenter's rule conjecture.

This chapter is joint work with Robert Connelly and Günter Rote. A short version appeared in the 41st Symposium on Foundations of Computer Science [CDR00].

4.1 Introduction

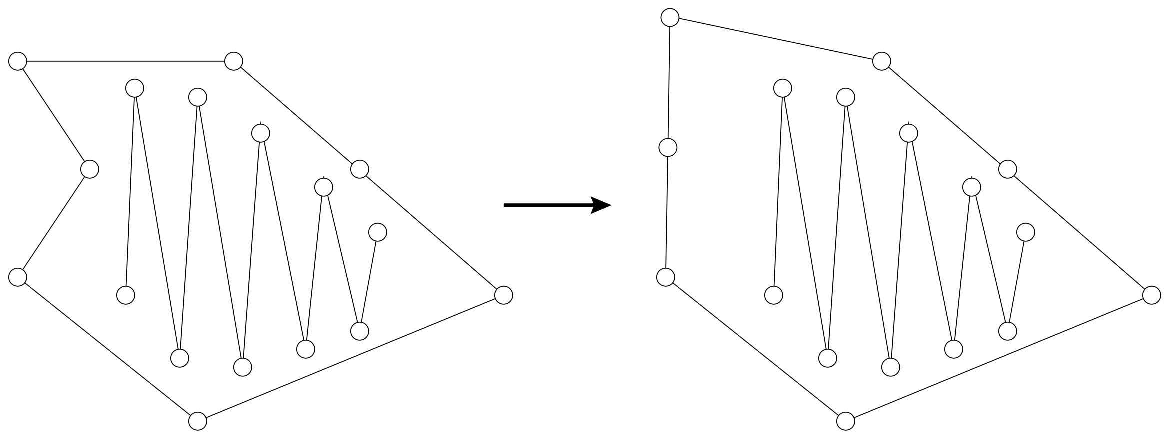
First we briefly review and detail a few notions from Chapters 1 and 2. A planar *polygonal arc* is a collection of line segments in the plane connected in a path.[1] A planar *polygonal cycle* is a polygonal simple closed curve in the plane. We say an arc is *straightened* by a motion if at the end of the motion it lies on a straight line. We say a polygonal cycle is *convexified* by a motion if at the end of the motion it is a convex closed curve. All motions must be *proper* in the sense that no self-intersections are created, and each edge length is kept fixed.

It seems intuitively easy to straighten an entangled chain: just grab the ends and pull them apart. Similarly, a cycle might be opened by blowing air into it until it expands. But these methods have the difficulty that they might introduce singularities, where the arc or cycle intersects itself. Our approach is to use an *expansive* motion in which all distances between two vertices increase. We also show that the area of a polygon increases in such an expansive motion.

We consider the more general situation, which we call an *arc-and-cycle set* A, consisting of a finite number of polygonal arcs and polygonal cycles in the plane, with none of the arcs or cycles intersecting each other or having self-intersections. We say that A is in an *outer-convex*

[1] More formally, an *arc* is a homeomorphic image of the closed interval $[0, 1]$ in the plane, and we are interested in finite embedded polygonal arcs.

configuration if each component of A that is not contained in any cycle of A is either straight (when it is an arc) or convex (when it is a cycle). The best we can hope for, in general, is to bring an arc-and-cycle set to an outer-convex configuration, because components nested within cycles cannot always be straightened or convexified; see Figure 4.1(a).



(a) (b)

Figure 4.1: (a) The nested arc cannot be straightened because there is insufficient space in the containing cycle. (b) Once the containing cycle becomes convex, any expansive motion must move the arc and the cycle rigidly in unison.

We say that a motion of an arc-and-cycle set A is *expansive* if for every pair of vertices of A the distance is monotonically nondecreasing over time, at all times either increasing or staying the same. Note that once a cycle becomes convex, the cycle and any components it contains become a single rigid object in any expansive motion; see Figure 4.1(b). (This fact is a consequence of Cauchy's Lemma 3.2.) Also, once two incident bars become collinear, they will remain so throughout any expansive motion, effectively acting as a single bar. We say that a motion is *strictly expansive* if the distance is constant between two vertices connected by a bar or straight chain of bars, and between two vertices on the boundary of or interior to a common convex cycle, but the distance between all other pairs of vertices monotonically strictly increases over time.

We say that the arc-and-cycle set A has *separated* if there is a line L in the plane such that L is disjoint from A and at least one component of A lies on each side of L. Our main result is the following:

Theorem 4.1 *Every arc-and-cycle set has a piecewise-differentiable proper motion to an outer-convex configuration. Moreover, the motion is strictly expansive until the arc-and-cycle set becomes separated.*

We can also extend this result to insist that the motion be strictly expansive during the entire motion. The definition of this motion is actually even simpler than the one we use for Theorem 4.1, but unfortunately the proof is a great deal more complex. Thus we first prove Theorem 4.1, and then in Section 4.6 prove the extension. Note that when there is just one component in the arc-and-cycle set, there is no difference between Theorem 4.1 and the extension in Section 4.6.

The tools that are applied here for the first time come from the theory of mechanisms and rigid frameworks. Arcs and cycles can be regarded as frameworks. See [AR78, AR79, Con80, Con82, Con93, CW96, CW93, CW82, CW94, GSS93, RW81, Whi84a, Whi84b, Whi87, Whi88, Whi92a] for relevant information about this theory.

Our approach is to prove that for any configuration there is an infinitesimal motion that increases all distances. Because of the nature of the arc-and-cycle set, this implies that there is a motion that works at least for a small expansive perturbation. We then combine these local motions into one complete motion. These notions are described in the rest of this chapter. Section 4.3 proves the existence of infinitesimal motions using the nonexistence of certain stresses, a notion dual to infinitesimal motions for the underlying framework. The analysis of these stresses uses a lifting theorem from the theory of rigidity that was known to James Clerk Maxwell and Luigi Cremona [CW82, CW93, Whi82] in the nineteenth century. Section 4.4 shows how to maneuver through the space of local motions to find a global motion with the desired properties.

4.2 Basics

A *linkage* or *bar framework* $G(\mathbf{p})$ is a finite graph $G = (V, E)$ without loops or multiple edges, together with a corresponding configuration $\mathbf{p} = (\mathbf{p}_1, \ldots, \mathbf{p}_n)$ of n points in the plane, where $\mathbf{p}_i$ corresponds to vertex $i \in V$. (For convenience we assume $V = \{1, \ldots, n\}$.) The edges of G constitute the set E and correspond to the bars in the framework, i.e., the links of a linkage. Arc-and-cycle sets are a particular kind of bar framework in which the graph G is a disjoint union of paths and cycles.

A *flex* or *motion* of $G(\mathbf{p})$ is a set of continuous functions $\mathbf{p}(t) = (\mathbf{p}_1(t), \ldots, \mathbf{p}_n(t))$, defined for $0 \le t \le 1$, such that $\mathbf{p}(0) = \mathbf{p}$ and $\|\mathbf{p}_i(t) - \mathbf{p}_j(t)\|$ is constant for each $\{i, j\} \in E$.[2] We are interested in finding a motion of the arc-and-cycle set with the additional property that it is strictly expansive.

4.2.1 Expansiveness

We begin with some basic properties of expansive motions. Namely, we will show that if a motion expands the distance between all pairs of vertices, it also expands the distance between all pairs of points on the arc-and-cycle framework. One consequence of this property is a key reason why we use expansive motions: they automatically avoid self-intersection. To prove the property, we need the following known basic geometric tool, which will also be useful later on:

Lemma 4.2 *In the plane, suppose that a point* $\mathbf{c}$ *is contained in the closed triangle formed by three points* $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$*, and suppose that another point* $\mathbf{q}_3$ *is chosen farther than* $\mathbf{p}_3$ *is from* $\mathbf{p}_1$ *and* $\mathbf{p}_2$*, i.e.,*

$$\|\mathbf{q}_3 - \mathbf{p}_2\| \ge \|\mathbf{p}_3 - \mathbf{p}_2\| \text{ and } \|\mathbf{q}_3 - \mathbf{p}_1\| \ge \|\mathbf{p}_3 - \mathbf{p}_1\|. \tag{4.1}$$

Then $\mathbf{q}_3$ *is also farther than* $\mathbf{p}_3$ *is from* $\mathbf{c}$*, i.e.,* $\|\mathbf{q}_3 - \mathbf{c}\| \ge \|\mathbf{p}_3 - \mathbf{c}\|$*, with equality precisely if both inequalities of* (4.1) *are equalities.*

[2] Throughout, $\|\mathbf{x}\|$ denotes the Euclidean length (L_2 norm) of a vector $\mathbf{x}$ in $\mathbb{R}^2$.

Proof: Observe that condition (4.1) implies that $\mathbf{q}_3$ is outside the triangle. Now refer to Figure 4.2. We claim that the circular disk C_0 centered at $\mathbf{c}$ and of radius $\|\mathbf{p}_3 - \mathbf{c}\|$ is contained in the union U of the circular disks C_1 centered at $\mathbf{p}_1$ and of radius $\|\mathbf{p}_3 - \mathbf{p}_1\|$, and C_2 centered at $\mathbf{p}_2$ and of radius $\|\mathbf{p}_3 - \mathbf{p}_2\|$. This claim implies the result because $\mathbf{q}_3$ must be outside both C_1 and C_2.

First observe that the circles C_0, C_1, and C_2 all pass through $\mathbf{p}_3$, and that C_0 has smaller radius than C_1 and C_2. If we orient the diagram as in Figure 4.2 so that C_0 is centered above the line $\mathbf{p}_1\mathbf{p}_2$, then the portion of the interior of C_0 that is below $\mathbf{p}_1\mathbf{p}_2$ is contained in the reflection of C_0 through $\mathbf{p}_1\mathbf{p}_2$. Thus, by symmetry, we can restrict attention to the portion of the diagram that is above the line $\mathbf{p}_1\mathbf{p}_2$. Circle C_0 does not penetrate the boundary of U formed by C_2 because it has smaller radius and because its center is on the opposite side of $\mathbf{p}_2\mathbf{p}_3$. Similarly, C_0 does not penetrate the boundary of U formed by C_1, concluding the proof. See also [Con82] for a proof in terms of tensegrities. □

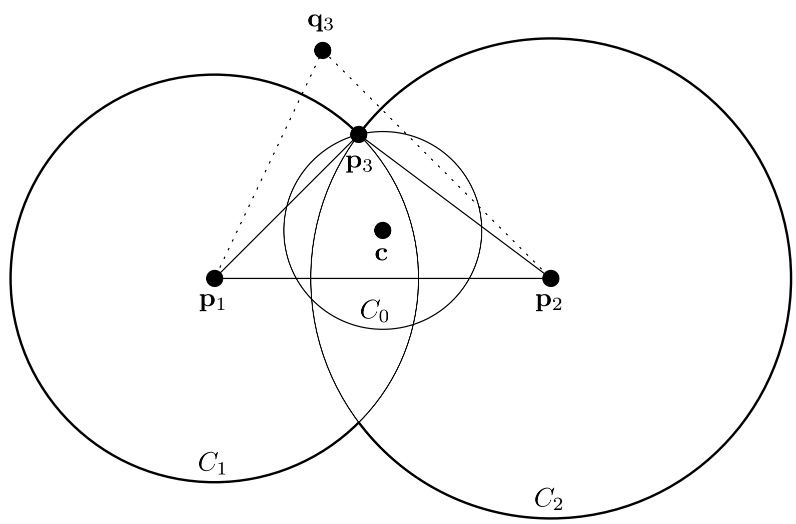


Figure 4.2: Illustration of Lemma 4.2.

Corollary 4.3 *Any expansive motion of an arc-and-cycle set only increases the distance between two points on the arc-and-cycle set (each either a vertex or on a bar). In particular, there can be no self-intersections.*

Proof: Refer to Figure 4.3. First, the result is obvious if the two points are both vertices of the arc-and-cycle set, by definition of expansiveness. Second, consider the distance between a vertex $\mathbf{p}_3$ of the arc-and-cycle set and a point $\mathbf{c}$ on a bar $\mathbf{p}_1\mathbf{p}_2$ of the arc-and-cycle set. Expansiveness implies that $\mathbf{p}_3$ only gets farther from $\mathbf{p}_1$ and $\mathbf{p}_2$, so by Lemma 4.2, $\mathbf{p}_3$ only gets farther from $\mathbf{c}$. Third, consider the distance between $\mathbf{c}$ (again on the bar $\mathbf{p}_1\mathbf{p}_2$) and another point $\mathbf{c}'$ on a bar $\mathbf{p}'_1\mathbf{p}'_2$ of the arc-and-cycle set. Substituting $\mathbf{p}'_1$ and $\mathbf{p}'_2$ as options for $\mathbf{p}_3$ in the previous argument, we know that $\mathbf{p}'_1$ and $\mathbf{p}'_2$ only get farther from $\mathbf{c}$. Applying Lemma 4.2 with $\mathbf{c}$ playing the role of $\mathbf{p}_3$, we obtain that $\mathbf{c}$ can only get farther from $\mathbf{c}'$. □

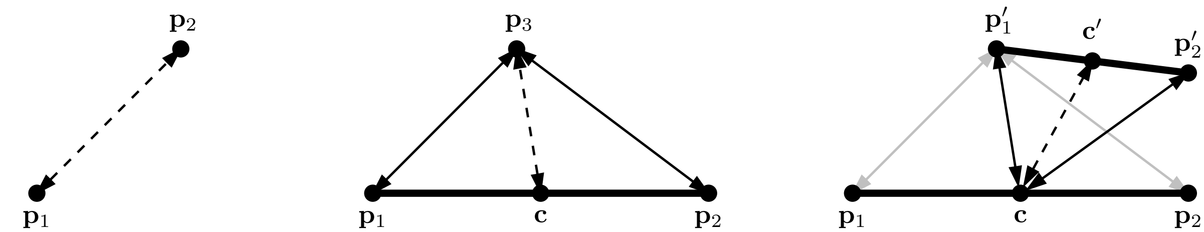


Figure 4.3: Illustration of the three cases of Corollary 4.3: (left) two vertices $\mathbf{p}_1, \mathbf{p}_2$; (middle) one vertex $\mathbf{p}_3$ and one point $\mathbf{c}$ on a bar $\mathbf{p}_1\mathbf{p}_2$; (right) one point $\mathbf{c}$ on a bar $\mathbf{p}_1\mathbf{p}_2$ and another point $\mathbf{c}'$ on a bar $\mathbf{p}'_1\mathbf{p}'_2$. Bold edges denote bars, and arrows denote expansion; dashed arrows are derived from solid arrows.

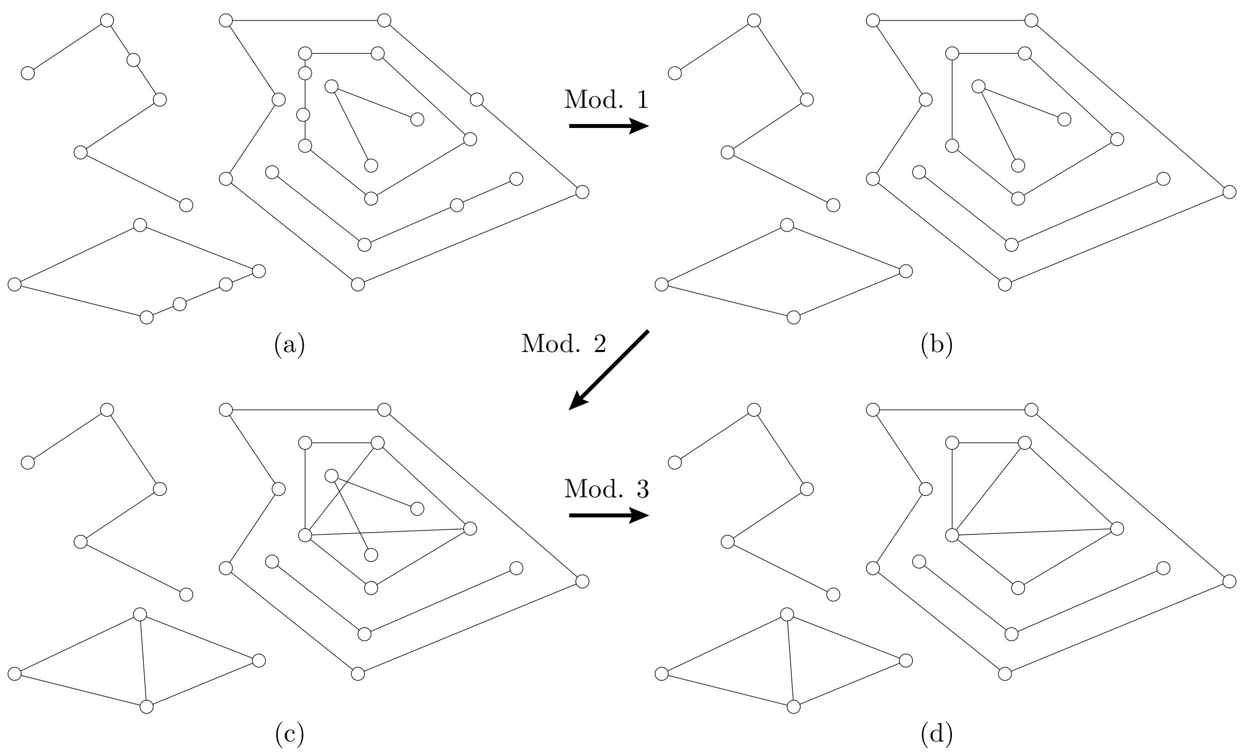


Figure 4.4: (a) Original arc-and-cycle framework. (b) With straight vertices removed. (c) With convex cycles rigidified. (d) With components nested within convex cycles removed.

4.2.2 The Framework $G_A(\mathbf{p})$

Given an arc-and-cycle set A that we would like to move to an outer-convex configuration, we make four modifications to A. The first three modifications simplify the problem by removing a few special cases that are easy to deal with; see Figure 4.4. The fourth modification will bring the problem of finding a strictly expansive motion into the area of tensegrity theory. In the end we will have defined a new framework, $G_A(\mathbf{p})$, which we will use throughout the rest of the proof.

Modification 1: Remove straight vertices. First we show that our arc-and-cycle set can be assumed to have no straight vertices, i.e., vertices with angle π. Furthermore, if during an expansive motion of the arc-and-cycle set we find that a vertex becomes straight, we can proceed by induction. For once the arc-and-cycle set has a straight subarc of more than one bar, we can coalesce this subarc into a single bar, thereby preserving the straightness of the subarc throughout the motion once it becomes straight. This reduces the number of bars and the number of vertices in the framework. By induction, this reduced framework has a motion according to Theorem 4.1, and such a motion extends directly to the original framework. The resulting motion is also strictly expansive by Corollary 4.3.

Modification 2: Rigidify convex polygons. Once a cycle becomes convex, we no longer have to expand it, and in fact it is impossible to expand it, so we hold it rigid from that point on. Of course, we allow a convex cycle to translate or rotate in the plane, but its vertex angles are not allowed to change. This can be directly modeled in the bar framework by introducing bars in addition to the arc-and-set cycle. Specifically, we add the edges of a triangulation of a cycle once that cycle becomes convex. We deal with the contents of the cycle in the next modification.

Modification 3: Remove components nested within convex cycles. The previous modification did not address the fact that components can be nested within cycles. Once a cycle becomes convex, not only can we rigidify it, but we can also rigidify any nested components, and treat them as moving in synchrony with the convex cycle. We do this by removing from the framework any components nested within a convex cycle. Assuming there were some nested components to deal with, this results in a framework with fewer vertices and fewer bars. By induction, this reduced framework has a motion according to Theorem 4.1. This motion can be extended to apply to the original framework by defining nested components to follow the rigid motion of the containing convex cycle (rigid by Modification 2). By the following consequence of Lemma 4.2, the resulting motion is also strictly expansive.

Lemma 4.4 *Extending a motion to apply to components nested within convex cycles preserves strict expansiveness.*

Proof: Consider some vertex $\mathbf{c}$ on a component inside some convex cycle, and a vertex $\mathbf{p}_3$ outside the cycle. We first consider the case that $\mathbf{p}_3$ does not lie inside another convex cycle. Extend the ray from $\mathbf{p}_3\mathbf{c}$ beyond $\mathbf{c}$, and let $\mathbf{p}_1\mathbf{p}_2$ be the edge through which this ray exits the cycle. Thus, $\mathbf{c}$ is in the triangle $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$, so Lemma 4.2 applies, and the distance $\mathbf{p}_3\mathbf{c}$ increases.

For two points $\mathbf{c}_1$ and $\mathbf{c}_2$ in two different cycles C_1 and C_2, we extend the ray $\mathbf{c}_1\mathbf{c}_2$ to identify the edge $\mathbf{p}_1\mathbf{p}_2$ on C_2 where the ray leaves C_2. From the first part of the proof we conclude that $\mathbf{c}_1\mathbf{p}_1$ and $\mathbf{c}_1\mathbf{p}_2$ increase, and by Lemma 4.2, the distance $\mathbf{c}_1\mathbf{c}_2$ increases. □

Modification 4: Add struts. In order to model the expansive property we need, we apply the theory of tensegrity frameworks, in which frameworks can consist of both bars and "struts." In contrast to a bar, which must stay the same length throughout a motion, a *strut* is permitted to increase in length, or stay the same length, but cannot shorten. The last modification adds a strut between nearly every pair of vertices in the framework. The exceptions are those vertices already connected by a bar, and vertices on a common convex cycle, because in both cases we cannot hope to strictly increase the distance.

Final framework: $G_A(\mathbf{p})$. The above modifications define a tensegrity (bar-and-strut) framework $G_A(\mathbf{p})$ in terms of the arc-and-cycle set A. Specifically, assume that A has no straight vertices (Modification 1) and no components nested within convex cycles (Modification 3). We call such an arc-and-cycle set *reduced*. We define the set of bars, B, to consist of the bars from the arc-and-cycle set together with the bars forming the rigidifying triangulation of each convex cycle (Modification 2). The set S of struts consists of all vertex pairs that are not connected by a bar in B and which do not belong to a common convex cycle (Modification 4). See Figure 4.5 for an example of A and the resulting bar-and-strut framework $G_A(\mathbf{p})$. (The rightmost framework $G'_A(\mathbf{p}')$ will be defined later.)

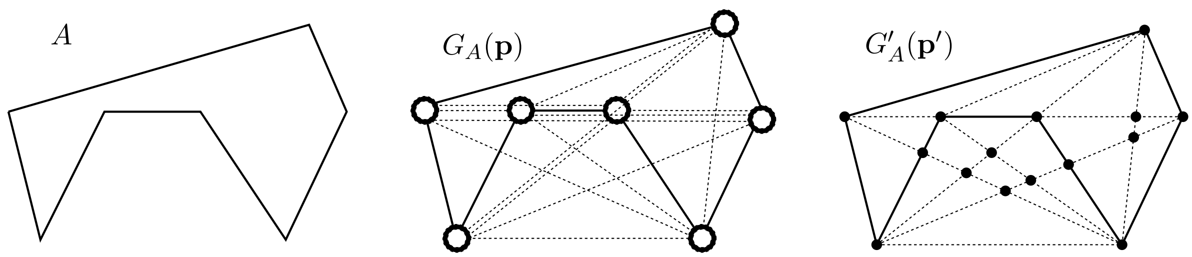


Figure 4.5: Construction of the frameworks $G_A(\mathbf{p})$ and $G'_A(\mathbf{p}')$. Solid lines denote bars, and dashed lines denote struts.

Our goal in the proof of Theorem 4.1 is to find a motion such that all bars maintain their length, while all struts strictly increase in length, in other words, a motion of $G_A(\mathbf{p})$ that is *strict* on all struts.

Thus, we want to find a motion $\mathbf{p}(t)$ for $0 \le t \le 1$ such that $\mathbf{p}(0) = \mathbf{p}$ and

$$\begin{aligned} \tfrac{d}{dt}\|\mathbf{p}_j(t) - \mathbf{p}_i(t)\| &= 0 \quad \text{for } \{i,j\} \in B, \\ \tfrac{d}{dt}\|\mathbf{p}_j(t) - \mathbf{p}_i(t)\| &> 0 \quad \text{for } \{i,j\} \in S. \end{aligned}$$

Differentiating the *squared* distances $\|\mathbf{p}_j(t) - \mathbf{p}_i(t)\|^2 = (\mathbf{p}_j(t) - \mathbf{p}_i(t)) \cdot (\mathbf{p}_j(t) - \mathbf{p}_i(t))$ and denoting the velocity vectors by $\mathbf{v}_i(t) := \frac{d}{dt}\mathbf{p}_i(t)$, we obtain the following equivalent conditions.

$$\begin{aligned} (\mathbf{v}_j(t) - \mathbf{v}_i(t)) \cdot (\mathbf{p}_j(t) - \mathbf{p}_i(t)) &= 0 \quad \text{for } \{i,j\} \in B, \\ (\mathbf{v}_j(t) - \mathbf{v}_i(t)) \cdot (\mathbf{p}_j(t) - \mathbf{p}_i(t)) &> 0 \quad \text{for } \{i,j\} \in S. \end{aligned}$$

Intuitively, the first-order change in the distance between vertex i and j is modeled by projecting the velocity vectors onto the line segment between the two vertices; see Figure 4.6.

4.2.3 Infinitesimal Motions

A *strict infinitesimal motion* or *strict infinitesimal flex* $\mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_n)$ specifies the first derivative of a strictly expansive motion at time 0. In other words, it assigns a velocity vector $\mathbf{v}_i$ to each vertex i so that it preserves the length of the bars to the first order, and strictly

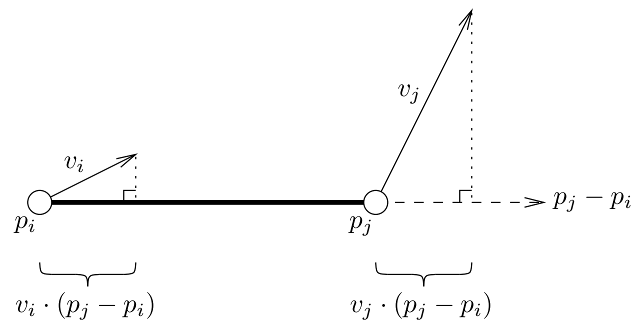


Figure 4.6: The dot product $(\mathbf{v}_j(t)-\mathbf{v}_i(t))\cdot(\mathbf{p}_j(t)-\mathbf{p}_i(t))$ is zero if the distance between $\mathbf{p}_i$ and $\mathbf{p}_j$ stays the same to the first order, positive if the distance increases, and negative if the distance decreases.

increases the length of struts to the first order. More precisely, a strict infinitesimal motion must satisfy

$$\begin{aligned}(\mathbf{v}_j - \mathbf{v}_i)\cdot(\mathbf{p}_j - \mathbf{p}_i) = 0 \text{ for } \{i,j\} \in B,\\ (\mathbf{v}_j - \mathbf{v}_i)\cdot(\mathbf{p}_j - \mathbf{p}_i) > 0 \text{ for } \{i,j\} \in S,\end{aligned} \tag{4.2}$$

where $\mathbf{p}_i$ denotes the initial position of vertex i.

In the next section, we prove that such a strict infinitesimal motion always exists. In Section 4.4 we show how this leads to motions for small amounts of time. These motions are then shown to continue globally until the configuration reaches an outer-convex configuration.

4.3 Local Motion

Recall that an arc-and-cycle set is called *reduced* if adjacent collinear bars have been coalesced, and components nested within convex cycles have been removed. In this section, we prove the following:

Theorem 4.5 *For any reduced arc-and-cycle set A there is an infinitesimal motion $\mathbf{v}$ of the corresponding bar-and-strut framework $G_A(\mathbf{p})$ satisfying* (4.2)*.*

4.3.1 Equilibrium Stresses

The equations and inequalities in (4.2) form a linear feasibility problem that is common for tensegrity frameworks. In order to solve this problem it is helpful to apply linear-programming duality and consider the equivalent dual problem. We discuss the duality first in terms of equilibrium stresses in tensegrity frameworks, and later reconnect it to linear-programming duality.

A *stress* in a framework $G(\mathbf{p})$ is an assignment of a scalar $\omega_{i,j} = \omega_{j,i}$ to each edge $\{i,j\}$ of G (a bar or strut). Intuitively, a positive scalar means that the edge is pushing on its two endpoints by an equal amount, a negative scalar means that the edge is pulling on its endpoints by an equal amount, and a zero scalar means that the edge induces no force. The whole stress is denoted by $\omega = (\ldots, \omega_{i,j}, \ldots)$. We say that the stress ω is an *equilibrium stress* if each vertex

i of G is in equilibrium, i.e., stationary subject to the the forces from the incident edges:

$$\sum_{j\,:\,\{i,j\}\in B\cup S} \omega_{ij}(\mathbf{p}_j - \mathbf{p}_i) = 0 \tag{4.3}$$

We say that the stress ω is *proper* if furthermore, for all struts $\{i,j\}$, $\omega_{i,j} \geq 0$. That is, struts can carry only zero or positive stress. There is no sign condition for bars: they can carry zero, positive, or negative stress. Thus, only bars can carry negative stress. (Terminology and sign conditions are not uniform in the literature. An equilibrium stress is also called a *self-stress* or simply a *stress*. All stresses that we deal with are equilibrium stresses.)

A trivial example of an equilibrium stress is the *zero stress* in which every edge is assigned a scalar of zero. All other stresses are called *nonzero*. To prove Theorem 4.5, we use the following duality principle connecting nonzero equilibrium stresses and infinitesimal motions:

Lemma 4.6 *If the only proper equilibrium stress in a bar-and-strut framework is the zero stress, then the framework has an infinitesimal motion.*

This equivalence is a standard technique in the theory of rigidity. See [CW96, Theorem 2.3.2] for a similar statement. For completeness, we give a brief proof here based on linear programming duality:

Proof: To make it easier to take the dual of the linear feasibility problem defined by (4.2), we write a linear program in standard form. First we add an otherwise pointless objective function of $0 = 0 \cdot \mathbf{v}$. Then we rescale the velocities $\mathbf{v}$ in (4.2) to obtain the following equivalent linear program:

$$\begin{array}{ll} \text{minimize} & 0 \cdot \mathbf{v} \\ \text{subject to} & (\mathbf{v}_j - \mathbf{v}_i) \cdot (\mathbf{p}_j - \mathbf{p}_i) = 0 \text{ for } \{i,j\} \in B, \\ & (\mathbf{v}_j - \mathbf{v}_i) \cdot (\mathbf{p}_j - \mathbf{p}_i) \geq 1 \text{ for } \{i,j\} \in S, \end{array} \tag{4.4}$$

We wish to show that the framework has an infinitesimal motion, which is equivalent to this linear program having a feasible solution, that is, an optimal solution of value 0. By linear-programming duality (the Farkas lemma), it suffices to show that the dual linear program

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{\{i,j\}\in S} \omega_{i,j} & \\ \text{subject to} & \displaystyle\sum_{j:\{i,j\}\in B\cup S} \omega_{i,j}(\mathbf{p}_j - \mathbf{p}_i) = 0 & \text{for } i \in V, \\ & \omega_{i,j} = \omega_{j,i} & \text{for } \{i,j\} \in B \cup S, \\ & \omega_{i,j} \geq 0 & \text{for } \{i,j\} \in S, \end{array} \tag{4.5}$$

has an optimal solution of value 0. This linear program specifies precisely the constraints of a proper equilibrium stress. Thus, it suffices to show that every proper equilibrium stress has $\omega_{i,j} = 0$ for all $\{i,j\} \in S$. In particular, it suffices to show that every proper equilibrium stress is identically zero. □

The important consequence of this lemma is that, in order to prove the desired Theorem 4.5, it suffices to prove the following:

Theorem 4.7 *The framework $G_A(\mathbf{p})$ corresponding to a reduced arc-and-cycle set A has only the zero proper equilibrium stress.*

4.3.2 Planarization

To prove that only the zero equilibrium stress exists (i.e., to prove Theorem 4.7), we use another tool in rigidity called the Maxwell-Cremona theorem. Before we can apply this tool, we need to transform the framework $G_A(\mathbf{p})$ into a planar framework $G'_A(\mathbf{p}')$. (Refer to the framework on the right of Figure 4.5.) We introduce new vertices at all intersection points between edges of $G_A(\mathbf{p})$, and subdivide the bars and struts accordingly. Any multiple edges resulting from this operation are merged. We define the resulting framework $G'_A(\mathbf{p}')$ to have bars precisely covering the bars of $G_A(\mathbf{p})$. All the other edges of $G'_A(\mathbf{p}')$ are struts. $G'_A(\mathbf{p}')$ is *planar* in the sense that two edges (bars or struts) intersect only at a common endpoint.

A natural concern is that the added vertices in this modification introduce additional freedom in finding infinitesimal motions, so they may not transfer directly to infinitesimal motions in the original framework. Nonetheless, the planar framework $G'_A(\mathbf{p}')$ is effectively equivalent to the original framework $G_A(\mathbf{p})$ in the sense of equilibrium stresses. Indeed, the following stronger statement holds. We call a stress *outer-zero* if the only edges that carry a nonzero stress are edges of convex cycles and edges interior to convex cycles. Otherwise, an edge exterior to all convex cycles carries a nonzero stress, and we call the stress *outer-nonzero*.

Lemma 4.8 *If $G_A(\mathbf{p})$ has a nonzero proper equilibrium stress ω, then $G'_A(\mathbf{p}')$ has an outer-nonzero proper equilibrium stress ω'.*

Proof: During the modifications to $G_A(\mathbf{p})$ that made $G'_A(\mathbf{p}')$, we modify ω to make ω' as follows. When we subdivide an edge $\{i,j\}$ with stress $\omega_{i,j}$, each edge $\{k,l\}$ of the subdivision gets the stress $\omega_{i,j}\|p_i - p_j\|/\|p_k - p_l\|$. (The ratio of lengths is necessary because $\omega_{i,j}$ is a weight, and the actual force comes from scaling by the length of the edge $\{i,j\}$; see Equation (4.3).) When merging several edges, we add the corresponding stresses. The resulting stress is in equilibrium because edges meet in opposing pairs at the added vertices, and because summation preserves force. The stress is also proper because a strut in $G'_A(\mathbf{p}')$ is made up only of struts from $G_A(\mathbf{p})$, and the sum of nonnegative numbers is nonnegative. It only remains to check that positive and negative stresses do not completely cancel during the merging process, and that the stress is furthermore outer-nonzero.

First we prove that some strut $\{i,j\}$ of $G_A(\mathbf{p})$ carries a positive stress. In other words, $G_A(\mathbf{p})$ cannot be stressed only on its bars; in particular, a framework consisting exclusively of arcs, cycles, and triangulated convex cycles cannot carry a nonzero stress. This follows because, in any such bar framework, there is a vertex $\mathbf{v}$ with degree at most two; in particular, every triangulated convex cycle has a degree-two vertex (an ear). Because the framework is reduced, the two bars incident to $\mathbf{v}$ are not parallel, so these two bars cannot carry stress while satisfying equilibrium at $\mathbf{v}$. Removing them and proceeding inductively with the rest of the framework, we conclude that the stress is zero on the whole bar framework. Hence, the bars alone cannot carry a nonzero stress, so some strut $\{i,j\} \in G_A(\mathbf{p})$ must have a nonzero stress.

The conditions of Theorem 4.5 enforce that no angles at vertices of the arc-and-cycle set are π or 0: an angle of π would create a straight subarc of two bars (contradicting the assumption that the framework is reduced), and an angle of 0 would violate simplicity. Thus, no strut of $G_A(\mathbf{p})$ is completely covered by bars. Therefore, for the strut $\{i,j\}$ of $G_A(\mathbf{p})$ that carries a positive stress, some portion of it in $G'_A(\mathbf{p}')$ will also have a positive stress, because a positive stress can only be canceled by a stress on a bar. In particular, ω' must be nonzero.

Furthermore, if the strut $\{i, j\}$ is exterior to all convex cycles in A, we have that ω' is outer-nonzero. Now suppose that $\{i, j\}$ is partially interior to convex cycles in A (by construction, the strut cannot be entirely within convex cycles of A). Then there is a portion of $\{i, j\}$ with the property that it is incident to a convex cycle and exterior to all convex cycles in A. This portion must be uncovered by bars, because no bar in A has this property, and the only additional bars in $G_A(\mathbf{p})$ are interior to convex cycles. Hence, the corresponding strut in $G'_A(\mathbf{p})$ carries a positive stress, so ω' is outer-nonzero in all cases. □

Thus, to prove that the original framework $G_A(\mathbf{p})$ has only the zero proper equilibrium stress, it suffices to prove that the planar framework $G'_A(\mathbf{p}')$ has only outer-zero proper equilibrium stresses.

4.3.3 Maxwell-Cremona Theorem

To prove that only outer-zero equilibrium stresses exist, we employ the Maxwell-Cremona correspondence between equilibrium stresses in planar frameworks and three-dimensional polyhedral graphs that project onto these frameworks. More precisely, a *polyhedral graph* or *polyhedral terrain* Γ comes from lifting a planar framework into three dimensions—that is, assigning a z coordinate (positive or negative) to each vertex in the framework—such that the vertices of each face of the framework (including the exterior face) remain coplanar. Thus, each face of the framework lifts to a planar polygon in 3-space. The polyhedral surface Γ is then the graph of a piecewise-linear continuous function of two variables that is linear on the faces determined by $G'_A(\mathbf{p}')$.

Consider an edge $\{i, j\}$ in a planar framework, separating faces F and F'. We distinguish whether this edge lifts in Γ to a "valley," "mountain," or "flat" edge according to its dihedral angles; see Figure 4.7. More formally, let $z(\mathbf{p}) = \mathbf{a} \cdot \mathbf{p} + b$ and $z(\mathbf{p}) = \mathbf{a}' \cdot \mathbf{p} + b'$ be the two linear functions specifying the graph Γ on F and F', respectively. Thus, $\mathbf{a}$ and $\mathbf{a}'$ are vectors in (the dual space of) $\mathbb{R}^2$, and b and b' are real numbers. A straightforward calculation reveals that the vector $\mathbf{a}' - \mathbf{a}$ in $\mathbb{R}^2$ must be perpendicular to the edge $\{i, j\}$:

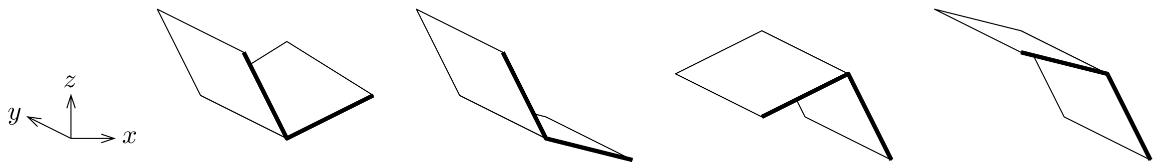
$$\mathbf{a}' - \mathbf{a} = \omega_{i,j}\mathbf{e}_{i,j}^{\perp} \tag{4.6}$$

where $\mathbf{e}_{i,j}^{\perp}$ is a vector in $\mathbb{R}^2$ of the same length as the vector $\mathbf{p}_j - \mathbf{p}_i$, perpendicular to it, and pointing from F towards F'. We call the edge $\{i, j\}$ a *valley* if $\omega_{i,j} > 0$, a *mountain* if $\omega_{i,j} < 0$, and *flat* if $\omega_{i,j} = 0$.

These definitions are illustrated in Figure 4.7. In particular, if two sides of an edge both "go up" in z, as in Figure 4.7(a), then the edge is a valley; however, the converse does not hold (so a valley might not carry water), as shown in Figure 4.7(b). Similarly, if the two sides of an edge both "go down" in z (Figure 4.7(c)), then the edge must be a mountain, but not all mountains have this property (Figure 4.7(d)).

Theorem 4.9 (Maxwell-Cremona Theorem) *(i) For every polyhedral graph Γ that projects to a planar bar framework $G(\mathbf{p})$, the stress ω defined by* (4.6) *forms an equilibrium stress on $G(\mathbf{p})$.*

(ii) For every proper equilibrium stress ω in a planar framework $G(\mathbf{p})$, $G(\mathbf{p})$ can be lifted to a polyhedral graph Γ such that (4.6) *holds for all edges. In particular, edges with positive stress lift to valleys, edges with negative stress lift to mountains, and edges with no stress lift to flat edges. Furthermore, Γ is unique up to addition of affine-linear functions.*



(a) Valley (b) Valley (c) Mountain (d) Mountain

Figure 4.7: Definition of valley and mountain in a polyhedral terrain Γ. The thick edges indicate the xz projection.

A proof of this result can be found in [CW94, HK92, Whi82], which follows the idea suggested above. Another point of view [Glu74] is that the stresses are a scaling of the angular momentum vectors of the function that lifts from the plane to the graph.

4.3.4 Main Argument

The zero equilibrium stress corresponds to a *trivial* polyhedral graph in which all faces are coplanar (i.e., defined by a single linear function). More generally, an outer-zero equilibrium stress corresponds to an *outer-flat* polyhedral graph that is flat on every edge exterior to all convex cycles. Therefore, to prove that all equilibrium stresses of the planar framework are outer-zero, and hence prove Theorem 4.7, it suffices to show that all polyhedral graphs projecting to the planar framework are outer-flat.

More precisely, consider any polyhedral graph Γ that projects to the planar framework $G'_A(\mathbf{p}')$ with the property that all struts are lifted to valleys or flat edges (because struts can carry only positive or zero stress), and bars are lifted to valleys, mountains, or flat edges. We need to show that nonflat edges can only appear within or on the boundary of convex cycles. Because we may add an arbitrary affine-linear function, we may conveniently assume that the exterior face of Γ is on the xy-plane. Thus the problem is to show that Γ does not lift off the xy-plane any vertex of $G'_A(\mathbf{p}')$ except possibly vertices interior to convex cycles of A.

One simple fact that we will need is the following:

Lemma 4.10 *Any mountain in the polyhedral graph Γ projects to a bar in the planar framework $G'_A(\mathbf{p}')$.*

Proof: A strut can only carry nonnegative stress, so by Theorem 4.9 it can only lift to a valley or a flat edge. □

We now come to the heart of our proof, the proof of Theorem 4.11. It is here that we finally show that the stress must be outer-zero, by looking at the maximum of any Maxwell-Cremona lift. The following statement immediately implies Theorem 4.7 and hence Theorem 4.5:

Theorem 4.11 *Let M denote the region in the xy-plane where the z value attains its maximum in the polyhedral graph Γ. Then M contains every face of the planar framework $G'_A(\mathbf{p}')$ that is exterior to all convex cycles.*

M is a nonempty union of faces, edges, and vertices of the planar framework $G'_A(\mathbf{p}')$. Consider the boundary ∂M, which may be empty if M fills the whole plane. Because points in M lift to maximum height, all edges of ∂M must lift to mountains. Thus by Lemma 4.10, all

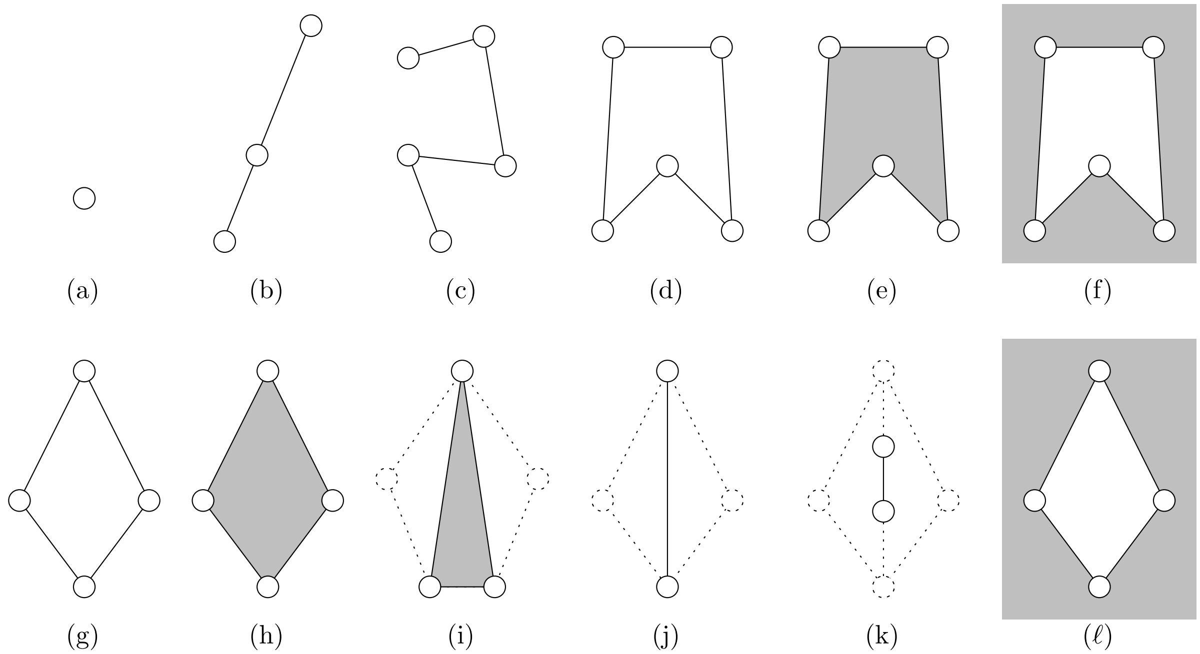


Figure 4.8: Hypothetical connected components of ∂M and their relation to M. Solid lines are edges of ∂M; white regions are absent from M; and shaded regions are present in M. (a) An isolated vertex. (b) A straight subarc. (c) A nonstraight subarc. (d) A nonconvex cycle. (e) A nonconvex cycle and its local interior. (f) A nonconvex cycle and its local exterior. (g–k) Various situations with a convex cycle. (ℓ) The only possible case: A convex cycle, its local exterior, and possibly some of its interior.

edges of ∂M must be bars in the framework. Hence, ∂M consists of disjoint vertices, paths of edges, and complete cycles of the arc-and-cycle set, together with a subset of the triangulations of the convex components. Figure 4.8 shows typical cases of all possibilities. We will show that the only case in Figure 4.8 that can actually occur is (ℓ), in which ∂M includes a convex cycle and M includes the local exterior of that cycle (and possibly some of its interior).

Our main technique for arriving at a contradiction in all cases except (ℓ) is that of *slicing* the polyhedral graph. Consider a plane Π that is parallel to the xy-plane and just below the maximum z coordinate of Γ. (By "just below" we mean that Π is above all vertices of Γ not at the maximum z coordinate.) Now take the intersection of Π with the surface Γ, and project this intersection to the xy-plane. The resulting set X is shown in Figure 4.9 for the various cases.

The set X captures several properties of the polyhedral graph Γ. First note that because X is the boundary of a small neighborhood of M in the plane, it is a disjoint union of cycles. It is also polygonal. Each edge of X corresponds to a face of Γ, and each vertex of X corresponds to an edge of Γ. The *angle* at a vertex of X (on the side interior to M) determines the type of edge corresponding to that vertex: the angle is π (straight) if the edge is flat, less than π (convex) if the edge is a mountain, and more than π (reflex) if the edge is a valley.

The basic idea is to show that X has "many" convex angles, and apply Lemma 4.10 to prove that the framework has "too many" bars. The key fact underlying the proof is that the

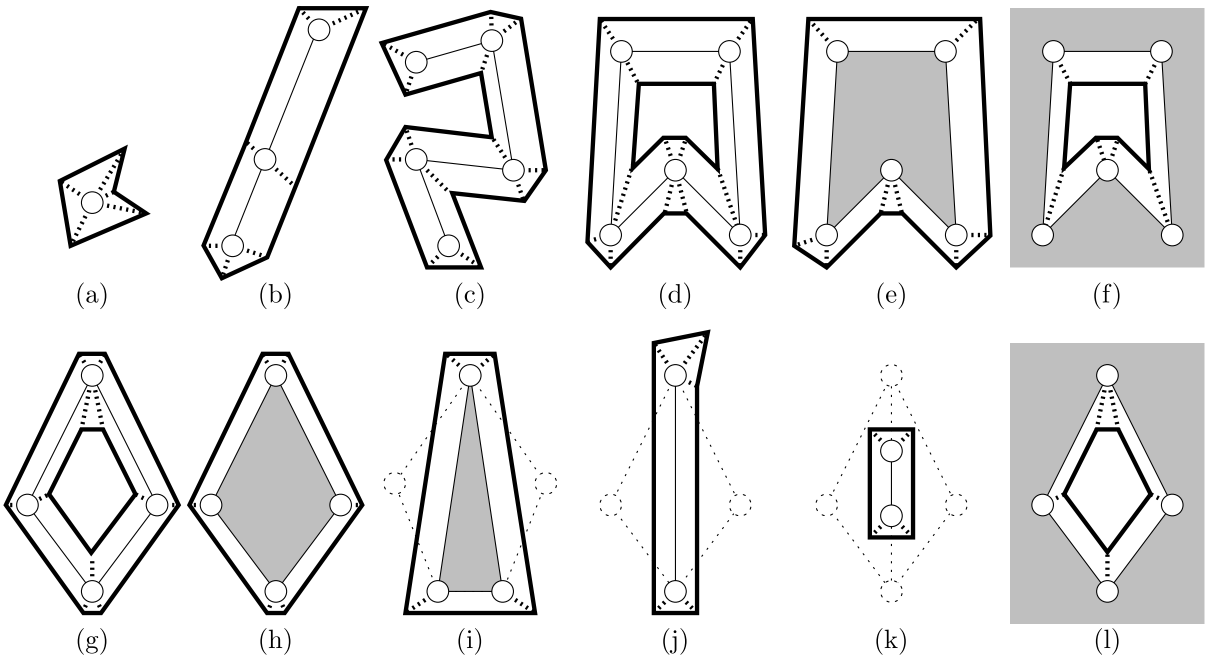


Figure 4.9: Slicing the polyhedral graph Γ just below the maximum z coordinate, in each case corresponding to those in Figure 4.8. Thick lines denote the slice intersection X, and thick dotted lines denote the corresponding edges in the polyhedral graph Γ.

original arc-and-cycle set has maximum bar-degree two: every vertex is incident to at most two bars. In the planar framework $G'_A(\mathbf{p}')$, only vertices $\mathbf{v}$ of convex cycles can have bar-degrees greater than two, and these bars are contained in a convex wedge from $\mathbf{v}$.

Our proof deals with all cases at once. To illustrate the essence of the proof, we first describe it for a subcase of case (a) in which one component of ∂M is a single vertex $\mathbf{v}$ that does not belong to a convex cycle. In this case, one component of X is a planar polygonal cycle P that is star-shaped around $\mathbf{v}$, that is, every point on the boundary of P is visible from $\mathbf{v}$. In particular, P is a planar polygon with positive area and no self-intersections. Every such polygon has at least three convex vertices. (To see this, define the *turn angle* at a vertex to be π minus the interior angle, so it is positive for convex angles and negative for reflex angles, and always strictly between $-\pi$ and π. Because the turn angles of a planar polygon sum to 2π, and the maximum turn angle of a vertex is $< \pi$, every polygon has at least three vertices with positive turn angles.) These three convex vertices correspond to three mountains in Γ, all incident to a common vertex v. By Lemma 4.10, there are three bars incident to v, contradicting the maximum-degree-two property for vertices not on convex cycles. Therefore, this subcase of case (a) cannot exist.

The general reason that cases (a–k) cannot exist is the following:

Lemma 4.12 *Let $\mathbf{v}$ be a vertex on the boundary of M, and let $b_1, \ldots, b_k$ be the bars incident to $\mathbf{v}$ in cyclic order. Consider a small disk D around $\mathbf{v}$.*

(1) *If there is an angle of at least π at $\mathbf{v}$ between two consecutive bars, say b_i and b_{i+1}, then the pie wedge P of D bounded by b_i and b_{i+1} belongs to M (see Figure* 4.10*).*

(2) *If there are no bars or only one bar incident to* $\mathbf{v}$*, i.e.,* $k \leq 1$*, then the entire disk* D *belongs to* M*.* (*This can be viewed as a special case of* (1).)

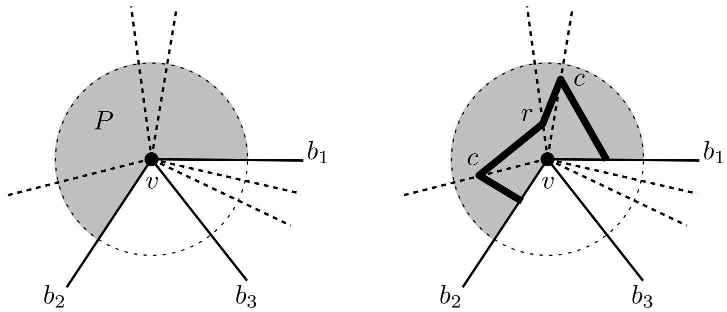


Figure 4.10: (Left) Illustration of Lemma 4.12: solid lines are bars, dotted lines are struts, and the shaded pie wedge P must be contained in M. (Right) Illustration of the proof; the thick lines form the portion of X inside P, and the symbols c and r denote convex and reflex vertices, respectively.

Proof: (1) Because there are no bars in the pie wedge P, and hence no edges of ∂M in P, P must be completely contained in or disjoint from M. Assume to the contrary that P is disjoint from M. Then the intersection of the slice X with the pie wedge P is a star-shaped polygonal arc around $\mathbf{v}$ starting from a point on b_i and ending at a point on b_{i+1}. By the properties of X, convex vertices on this arc correspond to mountains emanating from $\mathbf{v}$, and reflex vertices correspond to valleys emanating from $\mathbf{v}$. Because the angle of the pie wedge P is at least π, the arc must have at least one convex vertex in P. (The turn angles along the arc must sum to a positive number, so some vertex must have a positive turn angle.) By Lemma 4.10, there must be a bar in P, a contradiction.

(2) If $k = 1$, the bars b_i and b_{i+1} coincide, and the same proof applies. The star-shaped polygonal arc becomes a star-shaped polygonal cycle, which must have at least two convex vertices not on $b_i = b_{i+1}$. If $k = 0$, X also has a star-shaped polygonal cycle around $\mathbf{v}$, which must have at least three convex vertices, yet $\mathbf{v}$ has no incident bars. $\square$

Note that this lemma applies to every vertex in our planar framework $G'_A(\mathbf{p}')$, because every vertex either has bar-degree at most two or is a vertex of a convex cycle, and in either case there is a nonconvex angle between two consecutive bars.

One can immediately verify that the examples shown in Figure 4.9(a–k) contradict Lemma 4.12. For example, applying the lemma to any vertex of ∂M shows that M should contain a positive two-dimensional area incident to that vertex. This immediately rules out cases (a–d), (g), (j), and (k).

A general proof is also easy with Lemma 4.12 in hand:

Proof (Theorem 4.11): Consider first a degree-0 or degree-1 vertex $\mathbf{v}$ in ∂M. (Such a point would appear when M has a component that is an isolated point or an arc of bars.) Because Lemma 4.12 applies to every vertex of the framework, we know that some positive two-dimensional area in the vicinity of $\mathbf{v}$ belongs to M, contradicting that $\mathbf{v}$ has degree 0 or 1 in ∂M. This rules out cases (a–c) and (j–k).

It follows that ∂M is a union of cycles. A component of ∂M can be of two kinds:

(1) If it is formed from the edges of a convex cycle and its triangulation, Lemma 4.12 applies to any vertex in it, and we conclude that M contains the face of the framework immediately exterior to the cycle. This rules out cases (g–i).

(2) If it consists of a complete nonconvex cycle, we can apply Lemma 4.12 to some convex vertex and to some reflex vertex (they must both exist), and we conclude that M contains both the face of the framework immediately interior and the face immediately exterior to the cycle. This rules out cases (d–f).

In the end, the only faces of the framework that can be missing from M are those interior to convex cycles (case (ℓ)). This completes the proof of Theorem 4.11 and of Theorems 4.7 and 4.5. ☐

4.4 Global Motion

In this section, we combine the infinitesimal motions into a global motion, thereby proving Theorem 4.1, the main theorem. In overview, Theorem 4.5 establishes the existence of *some* direction of motion $\mathbf{v}$. We select a unique vector $\mathbf{v} = f(\mathbf{p})$ for each configuration $\mathbf{p}$ as the solution of a convex optimization problem (4.7–4.9). We then set up the differential equation

$$\frac{d}{dt}\mathbf{p}(t) = f(\mathbf{p}(t)).$$

The solution of this differential equation moves the linkage to a configuration where an angle between two bars becomes straight. At this point we merge the two bars and continue with the reduced framework that has one vertex less. This procedure is iterated until the framework is outer-convex and no further expansive motion is possible.

It is convenient for the proof of Theorem 4.1 to effectively pin an edge in the configuration. Choose any edge, say $\{\mathbf{p}_1, \mathbf{p}_2\}$, that is a bar. During the motion we will arrange matters so that this bar is stationary.

We now go into the details of the proof. We use the following nonlinear minimization problem to define a unique direction $\mathbf{v}$ for every configuration $\mathbf{p}$ of a reduced arc-and-cycle set.

$$\text{minimize} \quad \sum_{i\in V} \|\mathbf{v}_i\|^2 + \sum_{\{i,j\}\in S} \frac{1}{(\mathbf{v}_i - \mathbf{v}_j)\cdot(\mathbf{p}_i - \mathbf{p}_j) - \|\mathbf{p}_j - \mathbf{p}_i\|} \tag{4.7}$$

$$\text{subject to} \quad (\mathbf{v}_j - \mathbf{v}_i)\cdot(\mathbf{p}_j - \mathbf{p}_i) > \|\mathbf{p}_j - \mathbf{p}_i\|, \quad \text{for } \{i,j\}\in S \tag{4.8}$$

$$(\mathbf{v}_j - \mathbf{v}_i)\cdot(\mathbf{p}_j - \mathbf{p}_i) = 0, \quad \text{for } \{i,j\}\in B \tag{4.9}$$

$$\mathbf{v}_1 = \mathbf{v}_2 = 0 \tag{4.10}$$

The restrictions (4.8) place a uniform constraint on the growth of the struts S: the derivative of the length of each strut must be larger than 1. Since the system (4.2) is homogeneous, the system (4.8–4.9) is feasible for any choice of right-hand sides in (4.8). This particular right-hand side has been chosen for convenience in the proof.

The objective function (4.7) includes the norm of $\mathbf{v}$ as a quadratic term, plus a barrier-type penalty term that keeps the solution away from the boundary (4.8) of the feasible region. This penalty term is necessary to achieve a smooth dependence of the solution on the data. The objective function is strictly convex because it is a sum of strictly convex functions, of the form x^2 for a variable x, and convex functions, of the form $1/f(x_1, x_2, x_3, x_4)$ where f is an affine function in four variables that is guaranteed to be positive. Because the objective function is strictly convex, and it goes to infinity if $\mathbf{v}$ increases to infinity or approaches the boundary of condition (4.8), there is a unique solution $\mathbf{v}$ for every $\mathbf{p}$; we denote this solution by $f(\mathbf{p})$.

The function $f(\mathbf{p})$ is defined on an open set $U \subset \mathbb{R}^{2n}$ that is characterized by the conditions of Theorem 4.5: no angles are 0° or 180°, no vertex touches a bar, and at least one cycle is nonconvex or at least one open arc is not straight.

4.4.1 Smoothness

We will show that f is differentiable in the domain U. This follows from the stability theory of convex programming under equality constraints, as applied to parametric optimization problems of the type

$$\min\{\, g(p,x) : x \in \Omega(p) \subseteq \mathbb{R}^n,\ A(p)x = b(p)\,\} \tag{4.11}$$

where $A(p)$ is an $m \times n$ matrix and $b(p)$ is an m-vector. The objective function g, the domain $\Omega(p)$, and the linear constraints (A, b) depend on a parameter p that ranges over an open region $U \subseteq \mathbb{R}^k$.

For such an optimization problem, the following lemma establishes the smooth dependence of the solution vector on the problem-definition data $A(p)$ and $b(p)$. Recall that the *Hessian* H_g of a function $g(x_1, \ldots, x_m)$ is the $m \times m$ matrix whose (i,j)th element is $\frac{\partial^2 g}{\partial x_i \partial x_j}$, and the gradient ∇g is the m-element vector whose ith element is $\frac{\partial g}{\partial x_i}$.

Lemma 4.13 *Suppose that the following conditions are satisfied in the optimization problem* (4.11).

(a) *The objective function $g(p,x)$ is twice continuously differentiable and strictly convex as a function of $x \in \Omega(p)$, with a positive definite Hessian H_g, for every $p \in U$.*

(b) *The domain $\Omega(p)$ is an open set, for every $p \in U$.*

(c) *The rows of the constraint matrix $A(p)$ are linearly independent, for every $p \in U$.*

(d) *The problem-definition data $A(p)$ and $b(p)$ and the gradient ∇g of g with respect to x are continuously differentiable in p, for $p \in U$.*

(e) *The optimum point $x^*(p)$ of the problem* (4.11) *exists for every $p \in U$ (and is unique, by* (a)*).*

Then $x^*(p)$ is continuously differentiable in U.

Proof: The proof is based on the implicit function theorem and follows the standard lines of the proof in this area; cf. [BS74, in particular Section 4] or [Fia76, Theorem 2.1] for more general theorems where inequalities are also allowed. From (a) and (e) it follows that x^* can be found as part of the unique solution (x^*, λ) of the system of equations $h(p, x, \lambda) = 0$ that represents the first-order necessary optimality conditions. Specifically, λ is a k-element vector of Lagrange multipliers, and $h\colon U \times \mathbb{R}^{n+k} \to \mathbb{R}^{n+k}$ is given by

$$h = \begin{pmatrix} \nabla g - \lambda^{\mathrm{T}} A^{\mathrm{T}} \\ Ax - b \end{pmatrix}.$$

The implicit function theorem guarantees the local existence of $x(p)$ (and $\lambda(p)$) as a solution of $h(p, x(p), \lambda(p)) = 0$ in a neighborhood of x if the Jacobian $J = \partial h / \partial(x, \lambda)$ is an invertible matrix for every $p \in U$. Moreover, differentiability of $x(p)$ is ensured if h is continuously differentiable. The Jacobi matrix is given by

$$J = \frac{\partial h(p, x, \lambda)}{\partial(x, \lambda)} = \begin{pmatrix} H_g & A^{\mathrm{T}} \\ A & 0 \end{pmatrix}.$$

Differentiability of h follows from assumption (d); we only have to check that J is invertible. By assumption (a), H_g is positive definite and hence invertible. Thus

$$\det J = \det H_g \cdot \det(-A H_g^{-1} A^{\mathrm{T}}).$$

By assumption (c), A has full row rank, so the matrix AA^{T} is positive definite; because H_g is positive definite, so is $A H_g^{-1} A^{\mathrm{T}}$. Therefore $\det J \neq 0$. □

Lemma 4.14 *f is differentiable on U.*

Proof: The objective function is the sum of the quadratic function $\sum \|\mathbf{v}_i\|^2$, which has a positive definite (constant) Hessian, and additional convex terms, and therefore assumption (a) of Lemma 4.13 holds, as well as the second half of assumption (d). The feasible domain Ω is defined by the inequalities (4.8), and because the inequalities are strict, Ω is an open set, so assumption (b) holds. The problem-definition data A and b are defined by the linear constraints (4.9) and (4.10). Both are clearly continuously differentiable, verifying the first half of assumption (d). Assumption (e) follows from the existence of an infinitesimal motion (Theorem 4.5).

It only remains to check assumption (c), the linear independence of the equations (4.9). We have to check whether the system (4.9) has a solution for any choice of right-hand sides other than 0. As argued in the proof of Lemma 4.8, there is always a vertex i that is incident to at most two bars $\{i, j\}, \{i, k\}$, and moreover, these two bars cannot be parallel. The corresponding unknown vector $\mathbf{v}_i$ appears in at most two equations in which the scalar products with two vectors $\mathbf{p}_i - \mathbf{p}_j$ and $\mathbf{p}_i - \mathbf{p}_k$ are taken; because these vectors are not parallel, there is always a solution for $\mathbf{v}_i$ regardless of the values of the other variables. □

4.4.2 Solving the Differential Equation and Proving Theorem 4.1

Differentiability of f on U is sufficient to ensure that the initial-value problem

$$\frac{d}{dt}\mathbf{p}(t) = f(\mathbf{p}(t)), \quad \mathbf{p}(0) = \mathbf{p}_0 \tag{4.12}$$

has a (unique) *maximal solution* $\mathbf{p}(t)$, $0 \leq t < T$, that cannot be extended beyond some positive bound $T \leq \infty$; see for example [Wal96, Section II.XXI]. This means that one of three cases occurs:

(a) $\mathbf{p}(t)$ exists for all t, i.e., $T = \infty$.

(b) T is finite, and $\mathbf{p}(t)$ becomes unbounded as $t \to T$.

(c) T is finite, and $\mathbf{p}(t)$ approaches the boundary of U as $t \to T$.

The last case (c) is the case we want: at the boundary of U, some angle becomes straight, and we can reduce the linkage. Case (a) is the situation that most intuitively must be avoided: the motion of the framework slows down and never reaches the limit of an outer-convex configuration. Case (b) is another situation that must be avoided, arising only with multiple disjoint components: the components could repel and fly away from each other faster than they straighten or convexify, thus never reaching an outer-convex configuration.

Case (a) can be excluded very easily. By assumption, the bar-and-strut framework $G_A(\mathbf{p})$ has some strut $\{i, j\}$ between two points in the same component of the bar framework; their distance increases at least with rate 1, by (4.8), but it is bounded from above because i and j are linked by a sequence of bars. It follows that the solution cannot exist indefinitely and T must be finite.

If there is a line L that separates the components of the arc-and-cycle set A, this partitions A into two nonempty sets, so by keeping the two sets separated they can be treated independently and recursively. Unfortunately, the guarantee for the expansive property between different members of the partition is lost. But for the purposes of proving Theorem 4.1 we may assume that there is no such separation. Then the sum of the maximum diameters of each of the components of A is a uniform a-priori bound on the diameter of A for all time. This eliminates case (b).

Thus we are left with case (c) only. We show that $\mathbf{p}(t)$ converges as $t \to T$. Observe that all pairwise distances of vertices $\mathbf{p}(t)$ are monotonically increasing, and by condition (4.10) $\mathbf{p}_1$ and $\mathbf{p}_2$ are fixed during the motion. Thus, all other vertices are determined up to reflection (through $\mathbf{p}_1\mathbf{p}_2$), so the whole configuration is determined up to reflection. Thus $\mathbf{p}(t) \to \mathbf{p}$ for some configuration $\mathbf{p}$ as $t \to T$. The configuration $\mathbf{p}$ is on the boundary of U and thus must have some vertex with a straight angle. Then we inductively continue with a simpler linkage. This completes the proof of Theorem 4.1. $\square$

In this proof, the easy exclusion of possibility (b), that the motion becomes unbounded, depends crucially on the fact that the diameter of A is bounded, and the motion is stopped as soon as there is a separating line. Boundedness is valid even without this precaution, as stated in the following lemma.

Lemma 4.15 (Boundedness Lemma) *Let $\mathbf{p}(t)$ be the motion given by the differential equation* (4.12)*, where $\mathbf{v} = f(\mathbf{p})$ is given as the solution of the optimization problem* (4.7–4.9)*. Then the motion of every vertex i is bounded:*

$$\|\mathbf{p}_i(t) - \mathbf{p}_i(0)\| \le \int_0^t \|\mathbf{v}_i(t)\|\, dt \le K_{B,S,\mathbf{p}_0}(t),$$

where $K_{B,S,\mathbf{p}_0}(t)$ is an explicit function of t that depends only on the combinatorial structure of the arc-and-cycle set (B and S) and on the initial configuration $\mathbf{p}_0$.

Note that the definition of $\mathbf{v}$ does not involve the pinning constraints (4.10). The lemma implies that it is not necessary to treat separated components separately. The proof of the lemma is complicated, and it is given in Section 4.6.

4.4.3 Alternative Approaches

There are many ways to select a local motion $\mathbf{v}$ among the feasible local motions whose existence is guaranteed by Theorem 4.5. We have chosen one possibility in equations (4.7–4.10) that is most convenient for the proof.

As a possible alternative approach, we might consider a *linear* programming problem, with some arbitrary artificial linear objective function $\mathbf{c}$, and some linear normalization condition to ensure boundedness, pinning down some bar $(i_1, i_2) \in B$:

$$\begin{aligned}
\text{minimize} \quad & \textstyle\sum_{i \in V} \mathbf{c}_i \cdot \mathbf{v}_i \\
\text{subject to} \quad & (\mathbf{v}_j - \mathbf{v}_i) \cdot (\mathbf{p}_j - \mathbf{p}_i) = 0 \text{ for } \{i,j\} \in B, && (4.13) \\
& (\mathbf{v}_j - \mathbf{v}_i) \cdot (\mathbf{p}_j - \mathbf{p}_i) \geq 0 \text{ for } \{i,j\} \in S, && (4.14) \\
& \textstyle\sum_{i \in V} \mathbf{d}_i \cdot \mathbf{v}_i = 1, && (4.15) \\
& \mathbf{v}_{i_1} = \mathbf{v}_{i_2} = 0, && (4.16)
\end{aligned}$$

We have given up *strict* expansiveness in (4.14), The set of vectors given by (4.13), (4.14), and (4.16) forms a polyhedral cone C. Theorem 4.5 guarantees that there are nonzero solutions. One can check that the pinning constraints (4.16) ensure that the cone is pointed. The idea is now to use an extreme ray of the cone C for the motion. A vector $\mathbf{d}$ can be found which ensures that the feasible set (4.13–4.16) is a bounded set. Any basic feasible solution of the linear program will correspond to an extreme ray of the cone C. It will have a few inequalities of (4.14) fulfilled with equality. The resulting framework obtained by inserting "artificial" bars corresponding to the nonbasic inequalities of (4.14), will have a unique vector of velocities $\mathbf{v}$ subject to the normalization constraint (4.15). This means that the framework is a *mechanism*, allowing one degree of freedom; as the mechanism follows this forced motion, all nonfixed distances will increase, at least for some time.

So one follows the paradigm of parametric linear programming: The optimal basic feasible solution will continue to remain feasible as the coefficients $\mathbf{p}_i$ in the constraints (4.14) change smoothly. At some point, one of these constraints will threaten to become violated: this is the time to make a *pivot*, exchanging one of the artificial bars for a new one which allows the motion to be continued.

The above discussion has ignored several issues, such as possible degeneracy of the linear program. However, this approach might be more attractive from a conceptual, as well as practical, point of view.

Recently, Streinu [Str00] has found a class of such mechanisms, so-called *pseudo-triangulations*. These structures have several nice properties; for example, they form a planar framework of bars. Streinu [Str00] has proved that a polygonal arc can be opened by a sequence of at most $O(n^2)$ motions, where each motion is given by the mechanism of a single pseudo-triangulation.

4.4.4 Comparison of Approaches

The approach based on mechanisms might avoid some of the numerical difficulties associated with solving the optimization problem (4.7–4.9). For example, consider a spiral n-bar arc winding around a unit square in layers of thickness ε (Figure 4.11). Basically, a strut deep inside the spiral cannot increase in length quickly before an outer strut increases significantly. But in the solution of (4.8–4.9), the inner strut lengths must increase at unit speed; a rough estimate shows that this causes the outermost vertex to move with an *exponential* speed of at least $(1/\varepsilon)^{n/4}$, as $\varepsilon \to 0$. On the other hand, the "natural" solution of unwinding the spiral one bar at a time fits nicely into the setup of mechanisms and the parametric linear program approach.

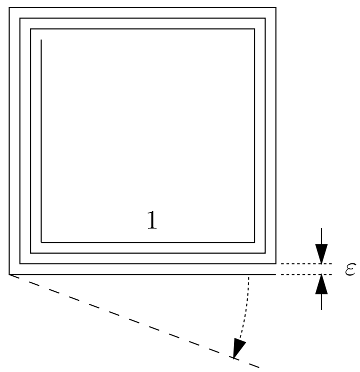


Figure 4.11: An arc that is numerically difficult to unfold.

Our proof has certain nonconstructive aspects: the direction $\mathbf{v}$ of movement is specified implicitly as the solution of an optimization problem, and the global motion arises as the solution of a differential equation. Both of these items are numerically well-understood, and our approach lends itself to a practical implementation. Indeed, we implemented our approach to produce animations such as Figure 2.2. However, this does not necessarily lead to a finite algorithm in the strict sense. The optimization problem (4.7–4.9), having an objective function which is rational, can in principle be solved exactly by solving the system $h(p, x, \lambda) = 0$ of algebraic equations as in Lemma 4.13. The differential equation cannot be solved explicitly, but it may be possible to bound the convergence and solve the differential equation up to a given error bound.

Because the motions of a mechanism are described by algebraic equations, Streinu's algorithm leads to a finite algorithm for a digital computer, at least in principle. It remains to be seen how a practical implementation competes with our approach; in any case, as an algorithm for a direct realization of the motion by a mechanical device, Streinu's algorithm appears attractive.

On the other hand, our nonlinear programming approach might be preferable because it produces a "canonical" movement. As a consequence of this, any symmetry of the starting configuration is preserved (see Corollary 4.16).

4.5 Additional Properties and Related Problems

4.5.1 Symmetry

We show that the deformation that we have defined in Section 4.4 preserves any symmetries that the original configuration might have. We say that the arc-and-cycle set A has some group H of congruences of the plane as *symmetry group* if each element of H permutes the vertices and edges of A.

Corollary 4.16 *If an arc-and-cycle set has a symmetry group H, then there is a piecewise-differentiable proper motion to an outer-convex configuration; the motion is expansive until the linkage becomes separated, and the symmetry group H is preserved during the motion.*

Proof: Because A has finitely many vertices and edges, H must be finite, so the Affine Fixed Point Theorem implies that there must be a point fixed by all elements of H. Let this point be the origin, and let $\mathbf{p}_1$ be any vertex of the configuration distinct from the origin. Consider the infinitesimal motion defined by the conditions (4.7), (4.8), and (4.9) but not (4.10). There is a unique solution $\mathbf{v}$ to this minimization problem. This solution must be symmetric with respect to the symmetry group H. If not, then the action of some element of H takes $\mathbf{v}$ to another distinct solution say $\mathbf{v}'$, contradicting the uniqueness of the solution. There is now a unique infinitesimal rotation that we can add to $\mathbf{v}$ so that $\mathbf{p}_1$ and $\mathbf{v}_1$ are parallel. This still maintains the symmetry of the infinitesimal motion $\mathbf{v}$. Now it is clear that the limit exists as before in the proof of Theorem 4.1, and the symmetry of H is preserved. □

4.5.2 Increasing Area

A natural question is whether every expansive motion increases the area bounded by each polygonal cycle. The answer turns out to be yes, but the proof is difficult from elementary methods. A simple example that helps motivate why this problem is nontrivial is an obtuse triangle: if the base edge increases in length (as a strut) and the others remain the same length (as bars), then the area decreases. The cycle of bars in a polygonal cycle is therefore crucial but difficult to exploit except with our theory of expansive motions.

First we show how to extend any given expansive infinitesimal motion to any point in the plane, which is of interest in its own right.

Lemma 4.17 *Consider an infinitesimal motion $\mathbf{v}$ on points $\mathbf{p}_1, \ldots, \mathbf{p}_n$ in $\mathbb{R}^d$, and suppose that the motion is* expansive*, i.e., $(\mathbf{p}_i - \mathbf{p}_j) \cdot (\mathbf{v}_i - \mathbf{v}_j) \geq 0$ for all i, j. Then the infinitesimal motion $\mathbf{v}$ can be extended to another point $\mathbf{p}_0$ in $\mathbb{R}^d$ and remain expansive. Furthermore, the new expansiveness inequalities are all strict unless $\mathbf{p}_0$ is in the convex hull of a subset of points on which the original infinitesimal motion is* trivial*, i.e., describes a rigid motion.*

Proof: We have two proofs of this statement: one proof is based on a direct calculation; the second proof is not so straightforward, but it provides some geometric insight.

We first consider the case where we want to prove strict expansiveness. By the Farkas lemma, the desired inequalities $(\mathbf{v}_0 - \mathbf{v}_i) \cdot (\mathbf{p}_0 - \mathbf{p}_i) > 0$, can be fulfilled by some unknown vector $\mathbf{v}_0$ if and only if the dual system

$$\sum_{i=1}^{n} \lambda_i(\mathbf{p}_0 - \mathbf{p}_i) = 0 \tag{4.17}$$

$$\sum_{i=1}^{n} \lambda_i \mathbf{v}_i \cdot (\mathbf{p}_0 - \mathbf{p}_i) \geq 0 \tag{4.18}$$

$$\lambda_i \geq 0$$

has no solution except the trivial solution $\lambda \equiv 0$. Suppose for contradiction that a nontrivial solution λ exists. Without loss of generality, we may assume

$$\sum_{i=1}^{n} \lambda_i = 1.$$

Then we get from (4.17) a representation of $\mathbf{p}_0$ as a convex combination

$$\mathbf{p}_0 = \sum_{i=1}^{n} \lambda_i \mathbf{p}_i. \tag{4.19}$$

Substituting this into (4.18) yields

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j (\mathbf{v}_i \cdot \mathbf{p}_j) - \sum_{i=1}^{n} \lambda_i (\mathbf{v}_i \cdot \mathbf{p}_i) \geq 0. \tag{4.20}$$

On the other hand, multiplying the given inequalities

$$\mathbf{v}_i \cdot \mathbf{p}_i - \mathbf{v}_i \cdot \mathbf{p}_j - \mathbf{v}_j \cdot \mathbf{p}_i + \mathbf{v}_j \cdot \mathbf{p}_j \geq 0 \tag{4.21}$$

by $-\lambda_j \lambda_j / 2$ and summing them over $i, j = 1, \ldots, n$ (including the trivial cases for $i = j$) yields

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j (\mathbf{v}_i \cdot \mathbf{p}_j) - \sum_{i=1}^{n} \lambda_i (\mathbf{v}_i \cdot \mathbf{p}_i) \leq 0. \tag{4.22}$$

By the assumption of the lemma, we have $\lambda_i > 0$ in (4.19) for at least two points $\mathbf{p}_i$ and $\mathbf{p}_j$ whose distance expands strictly. This means that the corresponding strict inequality in (4.21) will hold in (4.22) too, a contradiction to (4.20). This finishes the case when $\mathbf{p}_0$ does not lie in the convex hull of some points which move rigidly.

In the other case, nonstrict expansiveness can be shown by a variation of the above argument. Alternatively, we can appeal to Lemma 4.4 (or its higher-dimensional extension) and let the point $\mathbf{p}_0$ move rigidly with the rigid point set in whose convex hull it lies. The resulting motion is expansive; the distance from $\mathbf{p}_0$ to the other points will expand strictly, with the obvious exception of the points with which it moves rigidly. This concludes the first proof of the lemma.

The other proof establishes a connection to tensegrity frameworks and is more intuitive. However, we have to go through a sequence of steps to reduce the statement of the lemma to the basic case that the points $\mathbf{p}_1, \ldots, \mathbf{p}_n$ form the vertices of a simplex that contains the point $\mathbf{p}_0$ in its interior.

We proceed by induction on the dimension d. There is nothing to prove in case $d = 0$. So we assume the statement for $0, \ldots, d-1$ and $d \geq 1$. The desired inequalities $(\mathbf{v}_0 - \mathbf{v}_i) \cdot (\mathbf{p}_0 - \mathbf{p}_i) > 0$ define open half spaces

$$H_i = \{\mathbf{v}_0 \mid \mathbf{v}_0 \cdot (\mathbf{p}_0 - \mathbf{p}_i) > \mathbf{v}_i \cdot (\mathbf{p}_0 - \mathbf{p}_i)\}.$$

This finite collection of half spaces is nonempty precisely if every set of $d+1$ of them is nonempty by Helly's theorem [DGK63]. So we first consider any subset of $d+1$ points $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{d+1}$ of

$\mathbf{p}$. We consider the case when $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{d+1}$ are affine-independent, i.e., they are the vertices of a d-dimensional simplex σ in $\mathbb{R}^d$.

If $\mathbf{p}_0$ is outside σ, then choose $\mathbf{v}_0$ in a direction along a normal to a hyperplane separating $\mathbf{p}_0$ and σ, pointing away from σ. If the magnitude of $\mathbf{v}_0$ is large enough, then the desired inequalities will be satisfied.

If $\mathbf{p}_0$ is interior to σ, suppose that the inequalities defined above do not have a solution. Then if we look at the complementary half-spaces defined by

$$H_i^- = \{\mathbf{v}_0 \mid \mathbf{v}_0 \cdot (\mathbf{p}_0 - \mathbf{p}_i) \leq \mathbf{v}_i \cdot (\mathbf{p}_0 - \mathbf{p}_i)\},$$

they do have a solution. Let $\mathbf{v} = (\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{d+1})$ be a solution to those new inequalities. Now $\mathbf{v}$ is an infinitesimal motion of the tensegrity that is obtained by having *cables* from $\mathbf{p}_0$ (whose lengths can only shrink) and the rest all struts as before. But it is easy to show that this tensegrity has no infinitesimal motion in $\mathbb{R}^d$. (For example, apply Theorem 5.2(c) of [RW81] observing that the underlying bar framework has no infinitesimal motion, and there must be a proper stress that is nonzero on all struts and cables. An explicit calculation of the proper stress is given in [BC99]. The calculation is similar to the calculations (4.19)–(4.22) in the first part of the proof, with $\lambda_i\lambda_j$ being interpreted as stress.) So $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{d+1}$ must be a trivial infinitesimal motion, which can only happen if the motion is trivial on all of σ.

If $\mathbf{p}_0$ lies in a proper face of σ or the vertices $\mathbf{p}, \mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{d+1}$ are not affine independent, they lie in an affine $(d-1)$-dimensional space S with all the points of $\mathbf{p}$ either in S or in one of the open half-spaces defined S. Orthogonally project into S all of the $\mathbf{v}_i$ that correspond to $\mathbf{p}_i$ that are in S. By the induction hypothesis, there is a vector $\mathbf{v}_S$ in S such that it together with the vertices in S is infinitesimally expansive with respect to the projected $\mathbf{v}_i$ and hence the $\mathbf{v}_i$ themselves. It is strict unless $\mathbf{p}_0$ is in the convex hull of a simplex, where the infinitesimal motion is trivial. Let $\mathbf{v}_N$ be a large vector perpendicular to S pointing away from any other points in $\mathbf{p}$. If $\mathbf{v}_N$ is large enough, $\mathbf{v}_0 = \mathbf{v}_S + \mathbf{v}_N$ together with the other vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{d+1}$ will be expansive.

Thus in any case we see that there is a strict solution to the inequalities for $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{d+1}$ or $\mathbf{p}_0$ is in the convex hull of points where the infinitesimal motion acts as bars for all surrounding points. This concludes the second proof of the lemma. $\square$

Now we apply this lemma to prove that the area of a polygonal cycle increases by any expansive motion. Using the lemma inductively, we can extend an expansive motion to any finite set of points. Specifically, we apply Lemma 4.17 to the vertices of an appropriately chosen triangulation of the region bounded by a polygonal cycle. (The triangulation introduces new vertices in addition to the vertices of the polygonal cycle.) The following result can be found in [BGR88]. (See also [BMR95, Epp97] for faster algorithms.)

Lemma 4.18 *Any simple closed polygonal curve in the plane can be triangulated, introducing extra vertices, such that all the triangles are nonobtuse, i.e., every angle is at most $\pi/2$.*

There has been some interest in providing acute triangulations and subdivisions (in contrast to nonobtuse triangulations) of various planar polygonal objects. For example, the column of Martin Gardner [Gar60] (see also [Gar95a] and [Man60]) asks for a dissection of an obtuse triangle, in which one angle equals $\pi/2$, into acute triangles, in which every angle is strictly less than $\pi/2$. But we do not know of a result guaranteeing an acute triangulation for a general polygon. Fortunately, the following is sufficient for the area-expanding property that we need:

Lemma 4.19 *Let* $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ *be an infinitesimal motion of a nonobtuse triangle* $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ *such that for* $i \neq j$,

$$(\mathbf{v}_i - \mathbf{v}_j) \cdot (\mathbf{p}_i - \mathbf{p}_j) \geq 0. \tag{4.23}$$

Then the infinitesimal change in the area of triangle $\mathbf{p}$ *is always nonnegative. Furthermore, the infinitesimal change in the area is positive unless the infinitesimal flex* $\mathbf{v}$ *is trivial, i.e., no inequality in* (4.23) *is strict, or* $\mathbf{p}$ *is a right triangle and only the hypotenuse has a strict inequality in* (4.23).

Proof: Let the lengths of the sides of the triangle be denoted by a, b, c, and let the area of the triangle be denoted by A. If we differentiate Heron's formula

$$16A^2 = 2(a^2b^2 + a^2c^2 + b^2c^2) - (a^4 + b^4 + c^4)$$

and rearrange terms, denoting derivatives by a', b', c', A', we get

$$8AA' = (b^2 + c^2 - a^2)aa' + (a^2 + c^2 - b^2)bb' + (a^2 + b^2 - c^2)cc'. \tag{4.24}$$

We can regard aa', bb', cc' as the left hand side of (4.23). Each of the terms in parentheses in (4.24) is nonnegative because $\mathbf{p}$ is nonobtuse. Thus $A' \geq 0$.

If $\mathbf{p}$ is an acute triangle and at least one of $a' > 0$ or $b' > 0$ or $c' > 0$, then (4.24) is positive, and thus $A' > 0$. Suppose $\mathbf{p}$ is a right triangle and c represents the length of the hypotenuse. If at least one of $a' > 0$ or $b' > 0$, then (4.24) is positive, so $A' > 0$. □

Note that with a right triangle it is possible for the first derivative of the length of the hypotenuse to be positive while the first derivative of the length of the two legs is 0, and in this case the first derivative of the area will still be 0. This is the reason for the condition on the legs of the triangle.

Theorem 4.20 *Any smooth expansive noncongruent motion of a simple closed polygonal curve* C *in the plane, fixing the lengths of its edges, must increase the area of the interior of* C *during the motion.*

Proof: Consider the vector field $\mathbf{v}_t$, $0 \leq t \leq 1$ defined as the derivative at each vertex of C at time t. Apply Lemma 4.18 to find a triangulation T of the area bounded by C with all triangles nonobtuse. Apply Lemma 4.17 to extend the vector field to the vertices of T.

To get a strictly increasing area, we have to show that the triangulation T has an acute triangle with an edge interior to C, or a right triangle with a leg interior to C. Otherwise, T would be a single triangle, or it would exclusively consist of right triangles with both legs on C, hence it would be a convex quadrilateral with two opposite corners having right angles. These cases are excluded because a triangle or a convex quadrilateral (or any convex polygon) does not have an expansive noncongruent motion.

So we have established that T must have an acute triangle with an edge interior to C or a right triangle with a leg interior to C. Because the motion is expansive and the derivative of at least one of those lengths must be positive for all but a finite number of times, the derivative of the area of at least one of those triangles must be strictly positive, and they all are nonnegative by Lemma 4.19. So the derivative of the area bounded by C must be positive for all but a finite number of times $0 \leq t \leq 1$. Thus the area must strictly increase. □

4.6 Expansive Motion for Several Components

The motion described so far is expansive in each component, but may not expand the distance between vertices on different components once those components separate. In this final section we prove the following stronger version of Theorem 4.1 that guarantees a strictly expansive motion for multiple components even after the components separate. This result is attractive because it allows us to specify a canonical motion all the way to completion. Unfortunately, the proof is long.

Theorem 4.21 *Every arc-and-cycle set has a piecewise-differentiable proper motion to an outer-convex configuration that is strictly expansive throughout the motion.*

Proof: As mentioned in Section 4.4.2, the motion is defined in the same way as for Theorem 4.1, except that the explicit pinning constraint (4.10) is removed from the definition of $\mathbf{v} = f(\mathbf{p})$.

As in the case of Theorem 4.1, we know that the initial value problem (4.12) has a unique maximal solution $\mathbf{p}(t)$, $0 \le t < T$, that cannot be extended beyond T, and we have $T < \infty$. We have to show that, for $t \to T$, the configuration $\mathbf{p}(t)$ converges to a configuration on the boundary of U. From the Boundedness Lemma (Lemma 4.15 in Section 4.4.2), whose proof is given below, we conclude that the monotone limit

$$K_i := \lim_{t \to T} \int_0^t \|\mathbf{v}_i\| \, dt$$

exists. Therefore, for all $t_2 \ge t_1 \ge t_0$, we have

$$\|\mathbf{p}_i(t_2) - \mathbf{p}_i(t_1)\| \le \int_{t_1}^{t_2} \|\mathbf{v}_i(t)\| \, dt \le K_i - \int_0^{t_0} \|\mathbf{v}_i(t)\| \, dt.$$

The last bound goes to zero as $t_0 \to T$, and therefore $\mathbf{p}_i(t)$ converges. The limit configuration must lie on the boundary of the domain U because otherwise the solution of (4.12) could be extended beyond T. □

The rest of this section proves the only remaining aspect: the Boundedness Lemma (Lemma 4.15 in Section 4.4.2). That is, we would like to show that, under the expansive motion given by the optimization problem (4.7–4.9), the motion of individual points is limited.

Before embarking on the proof, let us consider a possibly difficult situation, which the proof has to handle. This example may aid the reader in appreciating the arguments of the proof.

We know that the distances change monotonically, and therefore their total change is bounded if the distances remain bounded. However, it is conceivable that the points move back and forth without increasing the distances too much. Consider three collinear points $\mathbf{p}_1$, $\mathbf{p}$ and $\mathbf{p}_2$ on a horizontal line; see Figure 4.12. If the point $\mathbf{p}$ wiggles vertically up and down with a small and decreasing amplitude, one can maintain an expansive motion where all distances remain bounded and yet $\mathbf{p}$ moves on a path of infinite length. More precisely, we move $\mathbf{p}$ repeatedly up and down, each time returning to its original location, with the amount of vertical motion decreasing harmonically: 1, 1/2, 1/3, 1/4, etc. Suppose the distance between $\mathbf{p}$ and both $\mathbf{p}_1$ and $\mathbf{p}_2$ is initially $d_1 = 1$, and after $k-1$ wiggles is $d_k \ge 1$. Then when $\mathbf{p}$ moves up $1/k$ units, the distance between $\mathbf{p}$ and $\mathbf{p}_i$ increases from d_k to

$$d_{k+1} = \sqrt{d_k^2 + 1/k^2} = d_k + \frac{1}{2d_k k^2} + O\left(\frac{1}{d_k^3 k^4}\right) \le d_k + \frac{1}{k^2} + O\left(\frac{1}{k^4}\right).$$

When $\mathbf{p}$ moves back down $1/k$ units, we move $\mathbf{p}_1$ left and move $\mathbf{p}_2$ right by this amount, to keep the motion expansive. After infinitely many wiggles, the distance between $\mathbf{p}$ and $\mathbf{p}_i$ is at most

$$\sum_{k=1}^{\infty}\left(\sqrt{d_k^2 + 1/k^2} - d_k\right) \leq \sum_{k=1}^{\infty}\left(\frac{1}{k^2} + O\left(\frac{1}{k^4}\right)\right) = O(1)$$

while the total motion of $\mathbf{p}$ is

$$\sum_{k=1}^{\infty}\frac{1}{k} = \infty.$$

This conceivable situation must be avoided.

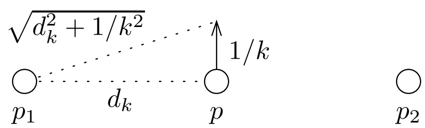


Figure 4.12: Wiggling point $\mathbf{p}$ vertically up and down by harmonic distances allows $\mathbf{p}$ to travel infinitely while pushing $\mathbf{p}_1$ and $\mathbf{p}_2$ away by only a bounded amount.

We may get arbitrarily close to this situation if all points are very nearly collinear. We can however get a quantitative estimate of the motion by placing bounds on the minimum width and on the diameter of the point set. If the angles in the triangle $\mathbf{p}_1\mathbf{p}\mathbf{p}_2$ are not arbitrarily small, the motion of $\mathbf{p}$ relative to $\mathbf{p}_1$ and $\mathbf{p}_2$ can be bounded in terms of the increase of pairwise distances.

4.6.1 Bounding the motion in terms of angles and distance changes

We give two geometric lemmas that allow us to bound the motions in terms of quantities that can only change monotonically.

Lemma 4.22 *Let $\mathbf{p}_1, \ldots, \mathbf{p}_n$ be the sequence of vertices of a component with the edges of fixed length $\|\mathbf{p}_j - \mathbf{p}_{j+1}\| = \ell_{j,j+1}$ for $j = 1, \ldots, n-1$. (These constraints do not include the edge from $\mathbf{p}_n$ to $\mathbf{p}_1$ in the case of a polygonal cycle.) Say $\mathbf{p}_1$ and $\mathbf{p}_2$ are pinned, i.e., $\mathbf{v}_1 = \mathbf{v}_2 = 0$. Let θ_j be the internal angle at vertex j $(0 < \theta_j < \pi$; we disregard the orientation of the angles). For any motion that preserves the lengths $\ell_{j,j+1}$, let θ'_j denote the derivative of θ_j with respect to time, and let $\mathbf{v}_k$ denote the derivative of $\mathbf{p}_k$ with respect to time. Then, for $k = 2, \ldots, n$,*

$$\|\mathbf{v}_k\| \leq (\ell_{1,2} + \cdots + \ell_{k-1,k}) \cdot (\theta'_2 + \cdots + \theta'_{k-1})$$

Proof: Using complex notation,

$$\frac{\mathbf{p}_{j+1} - \mathbf{p}_j}{\ell_{j,j+1}} = \frac{\mathbf{p}_{j-1} - \mathbf{p}_j}{\ell_{j-1,j}} \cdot \exp(\pm\theta_j i).$$

Taking derivatives gives

$$\frac{\mathbf{v}_{j+1} - \mathbf{v}_j}{\ell_{j,j+1}} = \frac{\mathbf{v}_{j-1} - \mathbf{v}_j}{\ell_{j-1,j}} \cdot \exp(\pm\theta_j i) + \frac{\mathbf{p}_{j-1} - \mathbf{p}_j}{\ell_{j-1,j}} \cdot \exp(\pm\theta_j i) \cdot (\pm\theta'_j i).$$

Taking the norm of both sides yields

$$\frac{\|\mathbf{v}_{j+1}-\mathbf{v}_j\|}{\ell_{j,j+1}} = \frac{\|\mathbf{v}_{j-1}-\mathbf{v}_j\|}{\ell_{j-1,j}} \cdot \underbrace{|\exp(\pm\theta_j i)|}_{=1} + \underbrace{\frac{\|\mathbf{p}_{j-1}-\mathbf{p}_j\|}{\ell_{j-1,j}}}_{=1} \cdot \underbrace{|\exp(\pm\theta_j i)|}_{=1} \cdot |\theta_j' i| = \frac{\|\mathbf{v}_{j-1}-\mathbf{v}_j\|}{\ell_{j-1,j}} + |\theta_j'|.$$

Applying this fact recursively,

$$\frac{\|\mathbf{v}_{j+1}-\mathbf{v}_j\|}{\ell_{j,j+1}} \leq \frac{\|\mathbf{v}_{j-1}-\mathbf{v}_j\|}{\ell_{j-1,j}} + |\theta_j'| \leq \cdots \leq |\theta_2'| + \cdots + |\theta_j'|,$$

and thus

$$\|\mathbf{v}_{j+1}-\mathbf{v}_j\| \leq \ell_{j,j+1} \cdot (|\theta_2'| + \cdots + |\theta_j'|) \leq \ell_{j,j+1} \cdot (|\theta_2'| + \cdots + |\theta_{k-1}'|),$$

for $j \leq k$. Applying induction we obtain the desired result. □

The above lemma works for only a single component. For disconnected components, we want to bound the speed of motion $\mathbf{v}$ in terms of distance changes. However, it is always possible to move the point set as a whole without changing any distances at all. Therefore we will consider a modified motion in which one vertex $\mathbf{p}_1$ is fixed ("pinned"). Another vertex $\mathbf{p}_2$ remains on a fixed line through $\mathbf{p}_1$, preventing the complete set from spinning. This modification corresponds to adding a rigid motion of the point set $\mathbf{p}$ to the given velocities $\mathbf{v}$. It does not change the lengths ℓ_{ij} or their derivatives, $(\mathbf{v}_i - \mathbf{v}_j) \cdot (\mathbf{p}_i - \mathbf{p}_j)$.

The *width* of a set of points is the minimum width of a strip (infinitely long rectangle) containing the set.

Lemma 4.23 *Consider a set $\mathbf{p}_1, \ldots, \mathbf{p}_n$ of $n \geq 2$ points with distances $\ell_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|$, moving with velocities $\mathbf{v}_i = \mathbf{p}_i'$ under an expansive motion: $\ell_{ij}' \geq 0$. Denote the initial width by w. Without loss of generality, we assume that the diameter D is equal to ℓ_{12}. We normalize the motion by pinning $\mathbf{p}_1$ and the direction of the axis from $\mathbf{p}_1$ to $\mathbf{p}_2$. In other words, we assume that $\mathbf{v}_1 = 0$ and that $\mathbf{v}_2$ is parallel to $\mathbf{p}_2 - \mathbf{p}_1$. Then, for every $k \in \{1, \ldots, n\}$,*

$$\|\mathbf{v}_k\| \leq C_0 \cdot \sum_{i<j} \ell_{ij}',$$

where $C_0 = 666D^3/w^3$.

The proof of this lemma is tedious, and delayed to Section 4.6.5 at the end of this chapter.

4.6.2 Constant Quantities

We define the following constants. We denote by D_0 the maximum sum of the bar lengths in a single component of the arc-and-cycle set. This is an upper bound on the diameter of a component. As we shall see, $D_0 6^{n-1}$, denoted henceforth as $D_{\max}$, is an upper bound on the distance between two points in the same cluster of components (a notion defined in the next subsection). Let m_0 ($0 < m_0 \leq 1$) be a lower bound on the distance between any two different components throughout the motion (achieved either as a distance between two vertices or as a distance from a vertex to an edge). Let m_1 ($0 < m_1 \leq 1$) be a lower bound on the distance between any two vertices (in the same component or in different components) throughout the motion. Because these distances can only increase during the expansive motion, we just need to choose m_0 and m_1 smaller than the minimum distance between two different components or between two vertices in the initial configuration.

The constant $C = 5328 D_{\max}^3 / m_0^3 = 8{\cdot}666 D_{\max}^3 / m_0^3$ will play an important role in Lemma 4.27 and subsequently.

4.6.3 Combining components into clusters

In the proof we distinguish between components that are "close" to each other or even entangled within each other and components that are separated. Namely, we call two nonempty sets of points A and B *well-separated* if they can be separated by a line, and moreover, the two inner common tangents separating A and B form an angle less than 60°. See Figure 4.13(a).

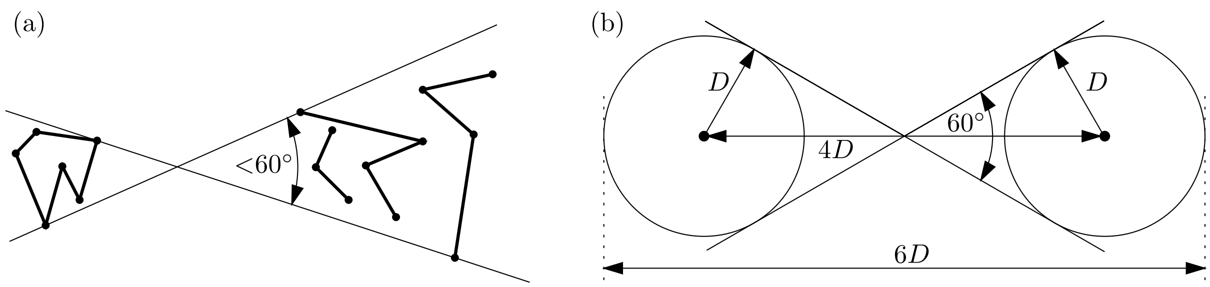


Figure 4.13: (a) Two well-separated clusters. (b) Inductive proof of Lemma 4.24. $D = D_0 6^{k-2}$ denotes the bound on the diameter of the two sub-clusters.

We now define a clustering procedure that will combine the components into clusters. (This procedure is similar to bottom-up hierarchical clustering [Eve93, HJ97], except that we pay no attention to the order in which we merge clusters.) We start with each component as a separate cluster. Whenever two clusters are not well-separated, we merge them into a single cluster. We repeat this until we end up with a single cluster or with a set of clusters that are pairwise well-separated (because otherwise we would perform another merge). The order in which we merge clusters has no effect on the outcome, because merging clusters may destroy well-separated pairs but never creates them, and hence only creates mergeable pairs and never destroys them. However, the clustering may change over time as the framework moves.

Lemma 4.24 *A cluster with k components has diameter at most $D_0 6^{k-1}$.*

Proof: By induction on k. Because a single component is connected, the case $k = 1$ follows. A cluster with $k \geq 2$ components results from merging two clusters of at most $k-1$ components that are not well-separated. By induction, these two clusters are contained in two disks of radius at most $D_0 6^{k-2}$. If the distance between the disk centers were bigger than $4 \cdot D_0 6^{k-2}$, the two sets would be well-separated; see Figure 4.13(b). Hence the union of the two circles, which contains the cluster, has diameter at most $(2 + 4) \cdot D_0 6^{k-2} = D_0 6^{k-1}$. □

So the quantity $D_{\max} = D_0 6^{n-1}$, which has already been introduced, is indeed an upper bound on the maximum possible distance between two points in the same cluster.

Lemma 4.25 *A cluster with at least two components has width at least $m_0/2$.*

Proof: Assume for contradiction that a cluster with at least two components lies in a strip of width less than $m_0/2$, which we can without loss of generality assume to be horizontal, as in Figure 4.14. Consider any two components in the cluster. By the definition of m_0, the distance between these two components is at least $m_0 > m_0/2$. Hence, there must be a vertical line separating the two components. Consider one of the two inner common tangents. The segment between the tangent points has length at least m_0 and lies in the strip. Therefore the angle between the tangent and the horizontal axis is less than 30°, and the same is true for the other tangent. It follows that any two components in the strip are well-separated, and therefore they do not belong to the same cluster. □

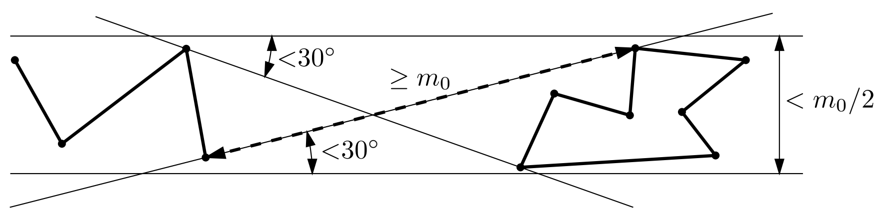


Figure 4.14: Proof of Lemma 4.25: two components in a thin strip.

Lemma 4.26 *Let $\mathbf{p}_1 \in V_1$ and $\mathbf{p}_2 \in V_2$ be points in two well-separated clusters V_1 and V_2, and let $\bar{\mathbf{c}}^{(1)}$ and $\bar{\mathbf{c}}^{(2)}$ denote the center of gravity of the vertices in V_1 and V_2, respectively. Then the angle between the vectors $\mathbf{p}_2 - \mathbf{p}_1$ and $\bar{\mathbf{c}}^{(2)} - \bar{\mathbf{c}}^{(1)}$ is at most 60°.*

Proof: Immediate from the definition of well-separated, because the center of gravity of a cluster is within the convex hull of the cluster. □

4.6.4 Rounding up the proof

Using Lemmas 4.22 and 4.23, we can bound all velocities $\mathbf{v}_i$ in terms of the sum of ℓ'_{ij} over all pairs $(\mathbf{p}_i, \mathbf{p}_j)$ in the *same* cluster (plus some θ' terms to accommodate single-component clusters). These quantities cannot grow indefinitely: we have an upper bound on the diameter of a cluster.

Lemma 4.27 *Let V_1 denote the set of vertices of a cluster and $T_1 \subseteq V_1$ the set of degree-2 vertices, with angles θ_j for $j \in T_1$, $0 < \theta_j < \pi$. Let ℓ_{ij} denote the distance between vertices $i, j \in V_1$. Given an expansive motion of all vertices, let θ'_j denote the derivative of θ_j with respect to time, $j \in T_1$, let ℓ'_{ij} denote the derivative of ℓ_{ij} with respect to time, and let $\mathbf{v}_j$ denote the derivative of a vertex $\mathbf{p}_j$ with respect to time, $j \in V_1$. If the motion is pinned as in Lemma 4.22 (for the case of a single-component cluster) or Lemma 4.23 (for a multi-component cluster), then, for every $k \in V_1$,*

$$\|\mathbf{v}_k\| \leq C \cdot \sum_{i,j \in V_1} \ell'_{ij} + D_0 \cdot \sum_{i \in T_1} \theta'_i$$

where $C = 5328 D_{\max}^3 / m_0^3 = 8 \cdot 666 D_{\max}^3 / m_0^3$.

Proof: The first term handles a cluster with at least two components by Lemma 4.23. We have width $w \geq m_0/2$ by Lemma 4.25 and diameter $D \leq D_{\max}$ by Lemma 4.24. The second term accounts for a single-component cluster and follows from Lemma 4.22. □

Everything so far has been generic to expansive motions. Now we will use the precise form of (4.7–4.9).

Lemma 4.28 *For an arc-and-cycle set, let $T \subseteq V$ denote the set of degree-two vertices, and let θ_j denote the angle at vertex $j \in T$, $0 < \theta_j < \pi$. For a motion $\mathbf{p}_j(t)$ of this arc-and-cycle set, let $\ell_{ij} = \|\mathbf{p}_i(t) - \mathbf{p}_j(t)\|$ denote distances between vertices, let $D(t) = \max\{1, \max_{i,j\in V} \ell_{ij}(t)\}$ denote the diameter at time t raised to be at least one, and let $\bar{\ell}_{ij}(t) = \min\{\ell_{ij}(t), D_{\max}\}$ denote the "truncated" distances. Then the solution $\mathbf{v}_i$ of the minimization problem* (4.7–4.9) *(defining a motion of the arc-and-cycle set) is bounded by*

$$\|\mathbf{v}_k\| \leq \frac{6n}{m_0 m_1} \left(C \sum_{i,j\in V} \bar{\ell}'_{ij} + D_0 \sum_{i\in T} \theta'_i + 2 \right) \cdot D(t)$$

Proof: The strategy of the proof is as follows:

1. Start with $\mathbf{v}_i$ given by the optimization problem (4.7–4.9).
2. In each cluster separately, "normalize" ("pin") $\mathbf{v}_i$ as indicated in Lemma 4.27, yielding "well-bounded" $\mathbf{v}_i$'s for each cluster.
3. Combine these pinned motions into an overall expansive motion by adding a vector to each cluster that ensures that the clusters will fly away from each other. We have control over those vectors, and we guarantee that the overall motion is still well-bounded (bounded in terms of the sum of ℓ'_{ij} over all pairs i, j in the same cluster).
4. Thus we have shown the existence of a well-bounded solution. Using the minimizing property of the optimization problem we conclude that its solution is also well-bounded.

Here are the details. Assume that there are K clusters with vertex sets $V_1, \ldots, V_K$. We first consider the motion $\hat{\mathbf{v}}^{(r)}$ that is obtained from $\mathbf{v}$ by restricting it to a cluster V_r and pinning it as in Lemma 4.22 or 4.23. We know that

$$\|\hat{\mathbf{v}}_k^{(r)}\| \leq \underbrace{C \cdot \sum_{i,j\in V_r} \bar{\ell}'_{ij} + D_0 \cdot \sum_{i\in T_r} \theta'_i}_{\text{denoted by } H^{(r)}} \tag{4.25}$$

for all $k \in V_r$.

By construction, all distance increases $\|\hat{\mathbf{v}}_i^{(r)} - \hat{\mathbf{v}}_j^{(r)}\|$ satisfy the conditions (4.8–4.9) for i, j in the same cluster V_r. We will now add a vector to the motion of each cluster in order to ensure that the expansion (4.8) also holds for points in different clusters. Intuitively, we want the clusters to "fly apart," but in a controlled manner.

Let $\bar{\mathbf{c}}^{(r)}$ denote the center of gravity of cluster r. We set

$$\bar{\mathbf{v}}_k := 2 \cdot \hat{\mathbf{v}}_k^{(r)} + E \cdot (\bar{\mathbf{c}}^{(r)} - \mathbf{p}_1) \tag{4.26}$$

for $k \in V_r$, with an arbitrarily chosen fixed vertex $\mathbf{p}_1$ and where E is defined by

$$E = 2\left(H^{(1)} + H^{(2)} + \cdots + H^{(K)} + 2\right)/m_0 > 4.$$

Note that $m_0 E/2$ is the value in parentheses in the bound of Lemma 4.28 that we want to prove. For vertices in the same cluster, we get $\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j = 2(\hat{\mathbf{v}}_i^{(r)} - \hat{\mathbf{v}}_j^{(r)})$, and hence the constraint

$$(\mathbf{v}_i - \mathbf{v}_j) \cdot (\mathbf{p}_i - \mathbf{p}_j) \geq \|\mathbf{p}_i - \mathbf{p}_j\| \tag{4.8}$$

is "twice over-fulfilled":

$$(\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j) \cdot (\mathbf{p}_i - \mathbf{p}_j) \geq 2 \cdot \|\mathbf{p}_i - \mathbf{p}_j\|$$

So we get the penalty term

$$\left[(\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j) \cdot (\mathbf{p}_i - \mathbf{p}_j) - \|\mathbf{p}_i - \mathbf{p}_j\|\right]^{-1} \leq \left[\|\mathbf{p}_i - \mathbf{p}_j\|\right]^{-1} \leq 1/m_1. \tag{4.27}$$

We claim that this inequality also holds for points i and j in two different clusters V_r and V_s. We have

$$\begin{aligned}
(\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j) \cdot (\mathbf{p}_i - \mathbf{p}_j) &= (\hat{\mathbf{v}}_i^{(r)} - \hat{\mathbf{v}}_j^{(s)}) \cdot (\mathbf{p}_i - \mathbf{p}_j) + E\left(\bar{\mathbf{c}}^{(r)} - \bar{\mathbf{c}}^{(s)}\right) \cdot (\mathbf{p}_i - \mathbf{p}_j) \\
&\geq \left|E\left(\bar{\mathbf{c}}^{(r)} - \bar{\mathbf{c}}^{(s)}\right) \cdot (\mathbf{p}_i - \mathbf{p}_j)\right| - \left|\left(\hat{\mathbf{v}}_i^{(r)} - \hat{\mathbf{v}}_j^{(s)}\right) \cdot (\mathbf{p}_i - \mathbf{p}_j)\right| \\
&\geq E \cdot \underbrace{\frac{1}{2}\|\bar{\mathbf{c}}^{(r)} - \bar{\mathbf{c}}^{(s)}\| \cdot \|\mathbf{p}_i - \mathbf{p}_j\|}_{\text{by Lemma 4.26}} - (H^{(r)} + H^{(s)}) \cdot \|\mathbf{p}_i - \mathbf{p}_j\| \\
&\geq \|\mathbf{p}_i - \mathbf{p}_j\| \left[Em_0/2 - (H^{(r)} + H^{(s)})\right] \\
&\geq 2 \cdot \|\mathbf{p}_i - \mathbf{p}_j\|
\end{aligned}$$

by the definition of E. So $\bar{\mathbf{v}}$ is indeed a feasible solution to (4.8–4.9), and (4.27) holds for all struts $\{i,j\} \in S$. From (4.25) and $\|\bar{\mathbf{c}}^{(r)} - \mathbf{p}_1\| \leq D(t)$ we obtain the bound

$$\|\bar{\mathbf{v}}_k\| \leq 2H^{(r)} + E \cdot D(t) \leq Em_0 + E \cdot D(t) \leq 2E \cdot D(t)$$

because we defined $D(t)$ to be at least 1.

From this and from (4.27), the objective function (4.7) for the solution $\bar{\mathbf{v}}$ that we have constructed satisfies

$$\begin{aligned}
&\sum_{i \in V} \|\bar{\mathbf{v}}_i\|^2 + \sum_{\{i,j\} \in S} \left[(\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j) \cdot (\mathbf{p}_i - \mathbf{p}_j) - \|\mathbf{p}_j - \mathbf{p}_i\|\right]^{-1} \leq n \cdot (2E \cdot D(t))^2 + n^2/m_1 \\
&\quad \leq n^2 \cdot (2E \cdot D(t))^2 \cdot \frac{1}{m_1} + \frac{n^2}{m_1} \cdot E^2 \cdot D(t)^2 = \frac{5n^2}{m_1} \cdot E^2 \cdot D(t)^2 \leq \left(\frac{3n}{m_1} \cdot E \cdot D(t)\right)^2.
\end{aligned} \tag{4.28}$$

It follows that the optimal solution $\mathbf{v}$, from which we started, must also satisfy this inequality. Therefore we obtain,

$$\sum_{i \in V} \|\mathbf{v}_i\|^2 \leq \sum_{i \in V} \|\mathbf{v}_i\|^2 + \sum_{\{i,j\} \in S} \left[(\mathbf{v}_i - \mathbf{v}_j) \cdot (\mathbf{p}_i - \mathbf{p}_j) - \|\mathbf{p}_j - \mathbf{p}_i\|\right]^{-1} \leq \left(\frac{3n}{m_1} \cdot E \cdot D(t)\right)^2.$$

and hence, for every $k \in V$,

$$\|\mathbf{v}_k\| \le \sqrt{\sum \|\mathbf{v}_i\|^2} \le \frac{3n}{m_1} \cdot E \cdot D(t) \qquad \square$$

Corollary 4.29 *$D(t) \le D(0)e^{F(t)}$ where $F(t) = \frac{12n}{m_0 m_1}(Cn^2 D_{\max} + D_0 n\pi + 2t)$.*

Proof: By noting the relation $D'(t) \le 2\max_{k\in V} \|\mathbf{v}_k\|$, we can apply the previous lemma to obtain an upper bound on $D'(t)/D(t) = \frac{d}{dt}\log D(t)$:

$$\frac{D'(t)}{D(t)} \le \frac{6nE}{m_1} \le \frac{12n}{m_0 m_1}\left(C \sum_{i,j\in V} \bar{\ell}'_{ij} + D_0 \sum_{i\in T} \theta'_i + 2\right).$$

Integrating and noting the bounds $\bar{\ell}_{ij} \le D_{\max}$ and $\theta_i \le \pi$, we obtain

$$\log D(t) - \log D(0) = \int_0^t \frac{D'(t)}{D(t)} dt \le \frac{12n}{m_0 m_1}\left(Cn^2 D_{\max} + D_0 n\pi + 2t\right). \qquad \square$$

We finally establish the result of the Boundedness Lemma (Lemma 4.15).

Corollary 4.30 *The path length of a vertex $\mathbf{p}_k$ until time T is bounded by*

$$\int_{t=0}^{T} \|\mathbf{v}_k\|\, dt \le \frac{D(0)F(T)}{2} e^{F(T)}$$

Proof: This follows from Lemma 4.28 and the previous corollary:

$$\int_0^T \|\mathbf{v}_k(t)\|\, dt \le \int_0^T \frac{3nE}{m_1} D(t)\, dt \le \int_0^T \frac{3nE}{m_1} dt \cdot D(T) \le \frac{F(T)}{2} \cdot D(0) \cdot e^{F(T)} \qquad \square$$

Remark: By a different choice of shifting vectors in (4.26) instead of $E \cdot \bar{\mathbf{c}}^{(r)}$ for each cluster it is possible to avoid the exponential "blow-up" in the above bounds.

4.6.5 Proof of Lemma 4.23

Unfortunately, the proof of Lemma 4.23 is quite elaborate. It is based on the special case of three points, which is elementary.

Lemma 4.31 *Consider a triangle with sides a, b, c bounded by $m \le a, b, c \le D$ and angles α, β, γ that are bounded by $\sin\alpha, \sin\beta, \sin\gamma \ge \mu > 0$. If the sides of the triangle are expanded with velocities $a', b', c' \ge 0$, then the change of the angles is bounded by*

$$|\alpha'|, |\beta'|, |\gamma'| \le \frac{2D}{m^2\mu} \cdot (a' + b' + c')$$

Proof: The cosine law gives $-\cos\alpha = -(b^2 + c^2 - a^2)/(2bc) = \frac{1}{2}[a^2/(bc) - b/c - c/b]$. Taking the derivative and omitting negative terms on the right-hand side leads to

$$\alpha' \sin\alpha \le \frac{1}{2}\left(\frac{2aa'}{bc} + \frac{bc'}{c^2} + \frac{cb'}{b^2}\right) \le \frac{1}{2}\left(2a' + c' + b'\right)\frac{D}{m^2} \le \left(a' + c' + b'\right)\frac{D}{m^2}$$

and thus, $\alpha' \leq (a' + b' + c') \cdot D/(m^2\mu)$. We get the same upper bound for β' and γ'. To obtain a lower bound, note that $\alpha + \beta + \gamma = \pi$ and hence $\alpha' = -\beta' - \gamma' \geq -(a' + b' + c') \cdot 2D/(m^2\mu)$. By symmetry, the same bound holds for the derivatives of the other two angles. $\square$

Proof (Lemma 4.23): Let $\mathbf{p}_3$ be the point whose distance h from the segment $\mathbf{p}_1\mathbf{p}_2$ is largest. By assumption, $w/2 \leq h \leq \sqrt{3/4} \cdot D$. So the points are contained in a rectangle K of length D and width $2h$, see Figure 4.15. Without loss of generality, we draw $\mathbf{p}_1\mathbf{p}_2$ horizontal from left to right and $\mathbf{p}_3$ above $\mathbf{p}_1\mathbf{p}_2$.

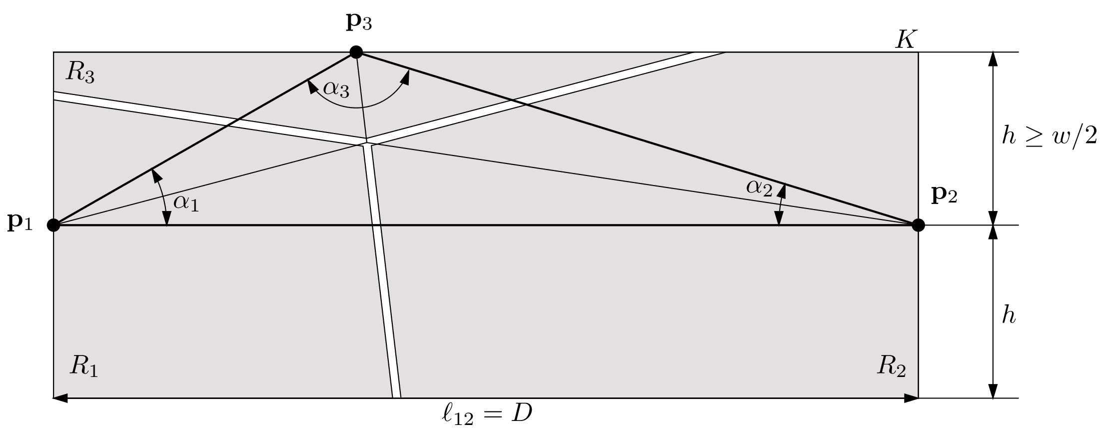


Figure 4.15: Proof of Lemma 4.23.

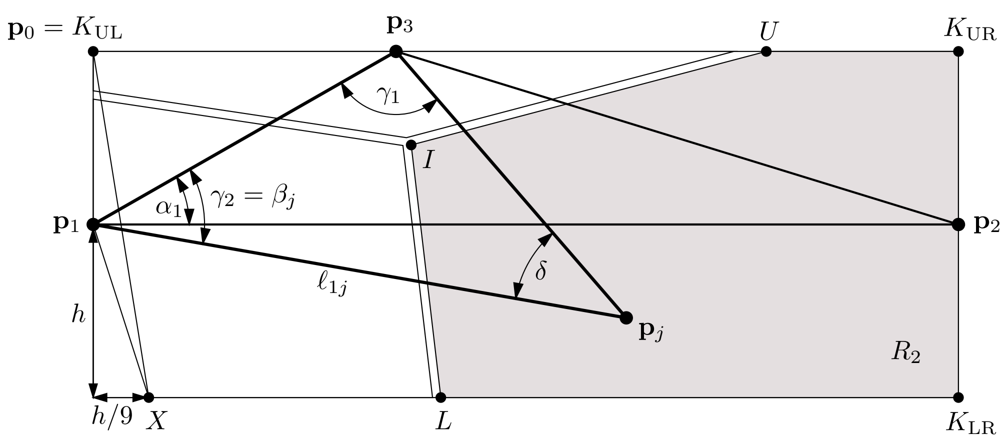


Figure 4.16: The triangle $\mathbf{p}_3\mathbf{p}_1\mathbf{p}_j$ associated with a point $\mathbf{p}_j \in R_2$ in the proof of Lemma 4.32.

We use the triangle $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$ with angles $\alpha_1, \alpha_2, \alpha_3$ as a "reference frame," and we measure the motion of the other points with respect to this frame. In particular, every point will be assigned to one of the sides of the reference triangle: We partition the rectangle K into three regions R_1, R_2, R_3 by extending the three angular bisectors from the incenter into the direction away from the vertices from which they emanate. R_i is the region containing the point $\mathbf{p}_i$, for $i = 1, 2, 3$. Now, every point $\mathbf{p}_j$ is assigned to the triangle side which lies *opposite* to the region in which it lies. For example, a point $\mathbf{p}_j$ in region R_1 is assigned to $\mathbf{p}_2\mathbf{p}_3$. We use this side as a *basis* of the triangle $\mathbf{p}_2\mathbf{p}_3\mathbf{p}_j$ that is associated with $\mathbf{p}_j$, and we select one of the angles at the basis as the *reference angle* β_j for $\mathbf{p}_j$. If $\mathbf{p}_j$ lies in R_1, we select the angle at $\mathbf{p}_2$ in the triangle $\mathbf{p}_2\mathbf{p}_3\mathbf{p}_j$. If $\mathbf{p}_j$ lies in R_2, we take the angle at $\mathbf{p}_1$ in the triangle $\mathbf{p}_3\mathbf{p}_1\mathbf{p}_j$ as the reference angle β_j for $\mathbf{p}_j$, see Figure 4.16; and for $\mathbf{p}_j$ in R_3, β_j is the angle at $\mathbf{p}_1$ in the triangle $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_j$. Note that these assignments are not completely symmetric. The rationale is to select a reference angle incident to $\mathbf{p}_1$ or $\mathbf{p}_2$.

The idea is that the triangle associated with each point cannot have arbitrarily small angles, and hence produces a "firm grip," tying the point to its base: The point cannot move an unlimited distance without increasing the pairwise distances ℓ_{ij} by a noticeable amount.

Lemma 4.32 *The reference triangle $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$ satisfies the assumptions of Lemma 4.31: The sines of the angles are at least $\mu := w/(3D)$ and the sides are at least $m := w/2$.*

The triangle associated with each point satisfies the assumptions of Lemma 4.31: The sines of the angles are at least $\mu := w/(20D)$ and the sides are at least $m := w/4$.

This lemma is not difficult to see but the precise proof is a bit technical and it is given at the end. Lemmas 4.31 and 4.32 allow us to conclude:

$$|\alpha_1'|, |\alpha_3'|, |\alpha_3'| \le \frac{2D}{(w/2)^2 \cdot w/(3D)} \cdot (\ell_{12}' + \ell_{23}' + \ell_{13}') \le \frac{24D^2}{w^3} \cdot L,$$

with $L := \sum_{i<j} \ell_{ij}'$, and for every point $\mathbf{p}_j$:

$$|\beta_j'| \le \frac{2D}{(w/4)^2 \cdot w/(20D)} \cdot L \le \frac{640D^2}{w^3} \cdot L.$$

Now we can bound the motion of $\mathbf{p}_j$ as follows. For $\mathbf{p}_j \in R_3$, the angle β_j between the fixed ray $\mathbf{p}_1\mathbf{p}_2$ and the ray $\mathbf{p}_1\mathbf{p}_j$, and the length ℓ_{1j} determine the position of $\mathbf{p}_j$ like polar coordinates. We therefore have

$$\|\mathbf{p}_j'\| = \sqrt{(\ell_{1j}\beta_j')^2 + (\ell_{1j}')^2} \le \ell_{1j}|\beta_j'| + \ell_{1j}'$$

For $\mathbf{p}_j \in R_2$, the situation is similar, except that the angle between the "x-axis" $\mathbf{p}_1\mathbf{p}_2$ and $\mathbf{p}_1\mathbf{p}_j$ is given by $\alpha_1 - \beta_j$, see Figure 4.16:

$$\|\mathbf{p}_j'\| \le \ell_{1j}(|\alpha_1'| + |\beta_j'|) + \ell_{1j}'$$

For $\mathbf{p}_j \in R_1$, the situation still more complicated, because the "origin" $\mathbf{p}_2$ of the coordinate system moves at a speed of ℓ_{12}'. The angle between the "negative x-axis" $\mathbf{p}_2\mathbf{p}_1$ and $\mathbf{p}_2\mathbf{p}_j$ is given by $\alpha_2 - \beta_j$.

$$\|\mathbf{p}_j'\| \le \|\mathbf{p}_2'\| + \|\mathbf{p}_l' - \mathbf{p}_2'\| \le \ell_{12}' + \ell_{2j}(|\alpha_2'| + |\beta_j'|) + \ell_{2j}'$$

Putting everything together, we get for every j,

$$\|\mathbf{v}_j\| = \|\mathbf{p}'_j\| \le L + D \cdot \left(\frac{24D^2}{w^3} \cdot L + \frac{640D^2}{w^3} \cdot L\right) + L = \left(2 + \frac{664D^3}{w^3}\right) \cdot L \le \frac{666D^3}{w^3} \cdot L,$$

using $D \ge w$. This concludes the proof of Lemma 4.23. □

Finally, we have to prove Lemma 4.32. An elementary geometric fact is needed.

Lemma 4.33 *In a triangle ABC with height h_C through C, the distance of C from the angular bisector g of the angle α at A is at least $h_C/2$.*

Proof: Let C' be the foot of h_C, and let D be the point on g closest to C, see Figure 4.17. Then C' and D lie on the circle over the diameter AC. The chords CD and $C'D$ are equal since their peripheral angles at A are both equal to $\alpha/2$. (If $\alpha > \pi/2$, the peripheral angle over $C'D$ is not $\alpha/2$ but $\pi - \alpha/2$, and for $\alpha = \pi/2$ the equality of the chords follows directly.) From the isosceles triangle $CC'D$ we get $CD \ge h_C/2$. □

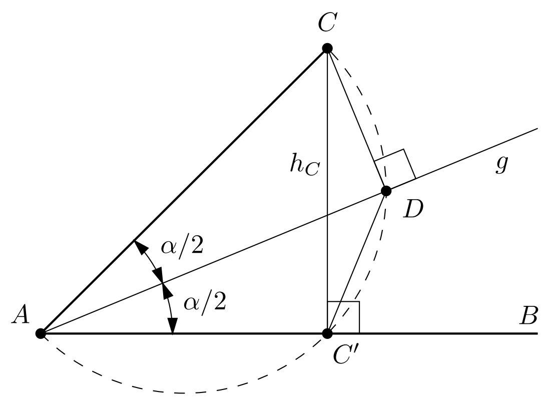


Figure 4.17: Proof of Lemma 4.33.

Proof (Lemma 4.32): We first prove the lower bound on the lengths of the edges. The basis edges $\mathbf{p}_1\mathbf{p}_2$, $\mathbf{p}_2\mathbf{p}_3$, and $\mathbf{p}_3\mathbf{p}_1$ all have lengths at least $h \ge w/2$. By the construction of the three regions, the point $\mathbf{p}_j$ is separated from the point $\mathbf{p}_i$ on its base segment ($i = 1, 2$, or 3) by the angular bisector of the angle at $\mathbf{p}_i$ in the triangle $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$. From Lemma 4.33 it follows that the distance $\mathbf{p}_j\mathbf{p}_i$ is at least 1/2 times one of the three heights of the triangle $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$. The smallest height of this triangle is opposite to the longest edge, which is $\mathbf{p}_1\mathbf{p}_2$. This height is h, and we obtain $\ell_{ij} \ge h/2 \ge w/4$.

Now let us consider the angles. Note that it suffices to check the condition of Lemma 4.31 for the smallest angle: if α is the smallest angle in a triangle with angles α, β, γ, then $\alpha \le \beta, \gamma \le \pi - \alpha$, and hence $\sin\beta, \sin\gamma \ge \sin\alpha$.

The angle α_1 in the triangle $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$ is bigger than the angle between $\mathbf{p}_1\mathbf{p}_2$ and the ray from $\mathbf{p}_1$ to the upper right corner K_{UR} of the rectangle K:

$$\sin\alpha_1 \ge \frac{h}{\sqrt{D^2 + h^2}} \ge \frac{h}{\sqrt{2D^2}} \ge \frac{w/2}{\sqrt{2} \cdot D} \ge \frac{w}{3D}$$

and similarly for α_2. The angle α_3 lies opposite to the longest edge of $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$, and hence it is the largest angle. So we have shown the length and angle bounds for the reference triangle $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$.

Let us now consider the triangle associated with a point $\mathbf{p}_j$. We denote the basis angles of the triangle by γ_1 and γ_2 and the angle at the apex $\mathbf{p}_j$ by δ, see Figure 4.16. By construction of the angular bisectors, an angle γ_j at the base is at least 1/2 times the respective angle α_i in the reference triangle $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$. So we have $\sin\gamma_1, \sin\gamma_2 \geq \sin\frac{\alpha_i}{2} \geq (\sin\alpha_i)/2 \geq w/(6D)$.

So the only angle which remains to be checked is the third angle δ at $\mathbf{p}_j$. If $\mathbf{p}_j$ lies in R_3, δ lies opposite to the longest side of the triangle $\mathbf{p}_1\mathbf{p}_2\mathbf{p}_j$, and we are done.

In the remainder of the proof we show rather tediously that $\sin\delta \geq w/(20D)$ when $\mathbf{p}_j \in R_1$ or $\mathbf{p}_j \in R_2$. Let us consider a triangle $\mathbf{p}_3\mathbf{p}_1\mathbf{p}_j$ with $\mathbf{p}_j \in R_2$, see Figure 4.16. (The case $\mathbf{p}_j \in R_1$ is symmetric.) Keeping $\mathbf{p}_j$ and $\mathbf{p}_1$ fixed, δ is smallest when $\mathbf{p}_3$ moves on the upper side of the rectangle K to the upper left corner $\mathbf{p}_0 = K_{\mathrm{UL}}$. (This position of $\mathbf{p}_3$ is hypothetical because ℓ_{23} would be larger than D.) Keeping $\mathbf{p}_1$ and $\mathbf{p}_0$ fixed, the angle $\delta = \angle\mathbf{p}_1\mathbf{p}\mathbf{p}_0$ is a quasi-concave function of $\mathbf{p} = \mathbf{p}_j$ as $\mathbf{p}$ moves inside K. (This means that the region in K where δ is bigger than some given threshold is alway convex.) Hence the minimum value of δ on any line segment occurs at one of its endpoints, and it suffices to check the lower bound for δ for the corners of the region R_2. More specifically, it is sufficient to check it for three points: (i) $\mathbf{p} = K_{\mathrm{UR}}$, the upper right corner of the rectangle K; (ii) $\mathbf{p} = K_{\mathrm{LR}}$, the lower right corner of K; and (iii) $\mathbf{p} = X$, a point on the lower edge of K at distance $h/9$ from the lower left corner.

The other corners of R_2 can be treated as follows: If $\mathbf{p}$ is the intersection U of the bisector of α_1 with the upper edge of K, then δ decreases as $\mathbf{p}$ moves right towards K_{UR}. If the bisector of α_1 intersects the boundary of K in the right edge, then the intersection point lies on the segment between K_{UR} and K_{LR}, and hence it is dominated by one of these points. (This situation occurs for region R_1 in Figure 4.16.) The incenter I can be treated in the same way as U, by moving $\mathbf{p}$ horizontally to the right until it hits the right edge.

Another corner is the intersection L of the bisector of α_3 with the lower edge of K. We cover this case by showing that the point X lies in region R_1. Therefore L lies on the segment between X and K_{LR}, and hence it is dominated by these two points. If the bisector of α_3 intersects the boundary of K in the right edge, it is dominated by K_{UR} and K_{LR}. The bisector cannot intersect the boundary of K in the left edge because of the point X.

Let us now consider the points $\mathbf{p} = K_{\mathrm{UR}}$ and $\mathbf{p} = K_{\mathrm{LR}}$. By symmetry, $\mathbf{p} = K_{\mathrm{UR}}$ yields the same angle $\delta = \angle\mathbf{p}_0\mathbf{p}\mathbf{p}_1$ as $\mathbf{p} = \mathbf{p}_2$ and that cannot be the minimum because it lies between K_{UR} and K_{LR}. For $\mathbf{p} = K_{\mathrm{LR}}$, the sine law for the triangle $\mathbf{p}_0\mathbf{p}\mathbf{p}_1$ with the angle $\delta = \angle\mathbf{p}_0\mathbf{p}\mathbf{p}_1$ gives

$$\frac{\sin\delta}{h} = \frac{\sin\angle\mathbf{p}_0\mathbf{p}_1\mathbf{p}}{\mathbf{p}_0\mathbf{p}} = \frac{D/\sqrt{D^2+h^2}}{\sqrt{D^2+(2h)^2}} \geq \frac{D}{\sqrt{2D^2}\cdot\sqrt{5D^2}},$$

using $h \leq D$. So we get

$$\sin\delta \geq \frac{1}{\sqrt{10}}\cdot\frac{h}{D} \geq \frac{1}{2\sqrt{10}}\cdot\frac{w}{D},$$

and the lemma holds in this case.

Let us finally consider the point X. Since $h \leq \sqrt{3/4}\cdot D$, the point X lies on the lower edge of K. We show that the distance of X from the ray $\mathbf{p}_3\mathbf{p}_1$ is smaller than the distance from the ray $\mathbf{p}_3\mathbf{p}_2$, and therefore X lies in R_1. The distance of X from the ray $\mathbf{p}_3\mathbf{p}_1$ is at most the distance $X\mathbf{p}_1 = \sqrt{1+1/81}\cdot h < 1.0062\,h$. The point on the ray $\mathbf{p}_3\mathbf{p}_2$ which is closest to X lies either on the segment $\mathbf{p}_3\mathbf{p}_2$ or in the region to the right of the rectangle. In the latter case, the distance is at least $D - h/9 \geq (\sqrt{4/3} - 1/9)h > 1.043\,h$, which is larger than $X\mathbf{p}_1$. In the former case, the distance between X and the segment $\mathbf{p}_3\mathbf{p}_2$ decreases if $\mathbf{p}_3$ or $\mathbf{p}_2$ are moved horizontally to the left. In the extreme cases, when $\mathbf{p}_3 = K_{\mathrm{UL}} = \mathbf{p}_0$ and $\ell_{12} = D = \sqrt{4/3} \cdot h$, this distance can be calculated as $> 1.439\,h$, which is also larger than $X\mathbf{p}_1$. A final calculation show that the angle $\delta = \angle \mathbf{p}_0 X \mathbf{p}_1$ satisfies $\sin\delta > 0.055 > 1/20 \geq \frac{w}{20D}$, since $w \leq D$, and the proof is complete. $\square$

Chapter 5

Conclusion

One contribution of this thesis has been to survey the results in the newly developing area of folding and unfolding, which offers many beautiful mathematical and computational problems. Much progress has been made recently in this area, but many important problems remain open. For example, most aspects of unfolding polyhedra remain unsolved, and we highlight two key problems in this context. The classic problem in the area is whether every convex polyhedron can be cut along its edges and unfolded into a simple polygon. An intriguing related problem is whether every nonconvex polyhedron, in which every edge is incident to exactly two faces, can be cut along its surface and unfolded into a simple polygon.

The other contribution of this thesis is to prove that a planar linkage formed by a collection of polygonal arcs and polygonal cycles of bars, none nested inside each other, can be folded into any desired configuration while preserving the bar lengths and never crossing bars. This proof consisted of two parts: (1) that a polygonal arc can be straightened and a polygonal cycle can be convexified, and (2) that once a polygonal cycle is convex it can be folded into any other convex configuration. Below we pose several open problems related to these results.

Algorithms for Unfolding Planar Linkages. The main result of this thesis is that every linkage consisting of polygonal arcs and polygonal cycles can be folded into a configuration in which every outermost arc is straight and every outermost cycle is convex. Our proof is *constructive* in terms of the derivative of the motion, in the sense that the derivative of the motion is given by a convex optimization problem for which there are reasonably efficient algorithms. This property is better than an *existential* proof, but not as good as an *algorithmic* proof. We are not ready to make claims about the existence of finite algorithms because of the difficulty in proving theorems about solving such ordinary differential equations.

In particular, it remains open whether there is a finite algorithm to animate the “movie” associated with our motion. More precisely, a finite *animation algorithm* should take a linkage whose vertices are at rational coordinates, and compute the snapshot of the linkage at a given rational time t during the motion such that each computed coordinate is within a given tolerance ε of exact. The running time of the algorithm would of course depend on the encoding size of the input linkage and t, and on ε, but should be finite in all cases.

We conjecture that such an animation algorithm exists for our motion, based on the practical approach used to generate the animation in Figure 2.2. The difficulty is in approximating solutions to the differential equation defined implicitly by the optimization problem (4.7–4.9), with guaranteed error bounds.

As described in Section 2.1.2.2, our theorem implies the existence of a finite sequence of simple motions, each changing at most four vertex angles in an easy-to-compute fashion. Although this sequence of motions is not strictly expansive, it is an even stronger notion of finite algorithm, because it provides a finite, simple, and easy-to-compute description of the entire motion. In principle, by avoiding complex algebraic barriers in the motion, this description can avoid all ε tolerances, and compute all positions exactly on a machine supporting basic arithmetic and roots.

The open question is whether such a sequence of simple motions can be computed in finite time. Again we conjecture that a finite algorithm exists. Further, we suspect that the running time (and output size) is pseudopolynomial in the encoding size of the input, that is, polynomial in the unary encoding of the input. Note that the analogous problems for folding between two convex configurations are solved in Chapter 3: we presented a linear-time algorithm on a real RAM to compute a sequence of four-joint moves.

A related result by Streinu [Str00] is that a different expansive unfolding of arcs and cycles can be achieved by a sequence of $O(n^2)$ *algebraic* motions obtained by adding bars to the framework until only one degree of freedom remains. Algebraic curves can be approximated in finite time, so in particular this result leads to a finite algorithm for the animation problem for Streinu's motion. Algebraic motions are not as simple as the ideal four-joint motions, but still provide a relatively succinct description of the entire motion, and might also lead more easily to a sequence of four-joint motions.

Tight Linkages. We have assumed the strict sense of simplicity, requiring two bars to intersect only at common endpoints. Because of expansiveness, this property is preserved throughout the motion. A natural problem is to consider what happens when the arc-and-cycle set is allowed to touch but not properly cross itself. We conjecture that even in this case the arc-and-cycle set can be brought to an outer-convex configuration.

Our methods do not directly apply in this case, because the obvious limiting process leads to more than two bars in the arc-and-cycle set effectively meeting at a point. If we could prove the existence of a strict infinitesimal motion from a self-touching position, however, we should immediately fall into the normal scenario of an arc-and-cycle set that does not touch itself, and the rest of the proof would go through.

Decision Problems. It is known that not every polygonal tree in 2D can be flattened, and not every polygonal arc in 3D can be straightened. What is the complexity of deciding when such unfoldings are possible? The only known algorithm is the general roadmap technique for exploring connectivity of algebraic spaces [Can87, Can88] which uses polynomial space and exponential time.

Via linear programming, our techniques allow detection of expansive motions. However, there are 2D trees that can be flattened and 3D arcs that can be straightened but not by expansive motions. The basic idea in constructing examples of such 2D trees and 3D arcs is to take the locked examples in Figures 2.3 and 2.9 and change the parameters so that they are no longer locked, but only barely so.

Equilateral 3D Linkages and Protein Folding. There are several connections between linkage folding and protein folding that remain to be explored. Two attractive linkage problems arising in this context were raised in Section 2.1.6.1. Can every equilateral polygonal arc in 3D be straightened? More generally, can every equilateral polygonal tree in 3D be flattened?

It seems difficult to attack these problems, because we have no footholds for a positive solution, and manual attempts at constructing counterexamples have so far failed. This situation is similar to the former state of the planar arc-straightening and cycle-convexification problems, before the ideas of expansiveness and tools from rigidity theory came into place. For 3D equilateral linkages, however, our techniques of expansive motions do not seem to be of use: there is an equilateral polygonal arc in 3D that has no expansive motion but can be straightened. The example is essentially Figure 2.9 with the first and last bars trimmed to unit length. The lack of an expansive motion can be verified by linear programming, and the existence of a straightening motion has been established for all equilateral arcs with five bars [CJ98].

Bibliography

[AAOS97] Pankaj K. Agarwal, Boris Aronov, Joseph O'Rourke, and Catherine A. Schevon. Star unfolding of a polytope with applications. *SIAM Journal on Computing*, 26(6):1689–1713, December 1997.

[ABB+01] Greg Aloupis, Prosenjit Bose, David Bremner, Erik Demaine, Vida Dujmović, Jeff Erickson, Danny Krizanc, Strefan Langerman, Henk Meijer, Pat Morin, Mark Overmars, Suneeta Ramaswami, Michael Soss, and Godfried Toussaint. Convexifying 3D polygons with convex projection using simpler moves. Manuscript in preparation, 2001.

[ABC+00] Hee-Kap Ahn, Prosenjit Bose, Jurek Czyzowicz, Nicolas Hanusse, Evangelos Kranakis, and Pat Morin. Flipping your lid. *Geombinatorics*, 10(2):57–63, 2000.

[ABD+01a] Oswin Aichholzer, David Bremner, Erik D. Demaine, Henk Meijer, Vera Sacristán, and Michael Soss. Long proteins with unique optimal foldings in the h-p model. In *Proceedings of the 17th European Workshop on Computational Geometry*, Berlin, Germany, March 2001.

[ABD+01b] Esther M. Arkin, Michael A. Bender, Erik D. Demaine, Martin L. Demaine, Joseph S. B. Mitchell, Saurabh Sethia, and Steven S. Skiena. When can you fold a map? In *Proceedings of the 7th International Workshop on Algorithms and Data Structures*, Providence, Rhode Island, August 2001. Full version available as `http://www.arXiv.org/abs/cs.CG/0011026`.

[ACD+00] Oswin Aichholzer, Carmen Cortés, Erik D. Demaine, Vida Dujmović, Jeff Erickson, Henk Meijer, Mark Overmars, Belén Palop, Suneeta Ramaswami, and Godfried T. Toussaint. Flipturning polygons. In *Proceedings of the Japan Conference on Discrete and Computational Geometry*, Lecture Notes in Computer Science, Tokyo, Japan, November 2000. To appear in *Discrete & Computational Geometry*.

[ADE+00] Oswin Aichholzer, Erik D. Demaine, Jeff Erickson, Ferran Hurtado, Mark Overmars, Michael A. Soss, and Godfried T. Toussaint. Reconfiguring convex polygons. In *Proceedings of the 12th Annual Canadian Conference on Computational Geometry*, pages 17–20, Fredericton, Canada, August 2000. `http://www.cs.unb.ca/conf/cccg/eProceedings/42.ps.gz`.

[ADE+01] 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), October 2001.

[AGP99] Boris Aronov, Jacob E. Goodman, and Richard Pollack. Convexification of planar polygons in $\mathbb{R}^3$. Manuscript, October 1999. `http://www.math.nyu.edu/faculty/pollack/convexifyingapolygon10-27-99.ps`.

[Ale50] A. D. Aleksandrov. *Convex Polyhedra* (in Russian). State Press of Technical and Theoretical Literature, Moscow, 1950. See also the German translation, *Konvexe Polyeder*, Akademie Verlag, 1958.

[AO92] Boris Aronov and Joseph O'Rourke. Nonoverlap of the star unfolding. *Discrete & Computational Geometry*, 8(3):219–250, 1992.

[AR78] L. Asimow and B. Roth. The rigidity of graphs. *Transactions of the American Mathematical Society*, 245:279–289, 1978.

[AR79] L. Asimow and B. Roth. The rigidity of graphs. II. *Journal of Mathematical Analysis and Applications*, 68(1):171–190, 1979.

[Bar94] Paulo Taborda Barreto. Lines meeting on a surface: The "Mars" paperfolding. In Koryo Miura, editor, *Proceedings of the 2nd International Meeting of Origami Science and Scientific Origami*, pages 323–331, Otsu, Japan, November–December 1994.

[Bat01] Alex Bateman. Computational tools for origami tessellations. In *Proceedings of the 3rd International Meeting of Origami Science, Math, and Education*, Monterey, California, March 2001.

[BC99] Károly Bezdek and Robert Connelly. Two-distance preserving functions from Euclidean space. *Periodica Mathematica Hungarica*, 39(1–3):185–200, 1999.

[BD01] Therese C. Biedl and Erik D. Demaine. Flipping nonsimple polygons. Manuscript in preparation, 2001.

[BDD+98] Therese Biedl, Erik Demaine, Martin Demaine, Anna Lubiw, Mark Overmars, Joseph O'Rourke, Steve Robbins, and Sue Whitesides. Unfolding some classes of orthogonal polyhedra. In *Proceedings of the 10th Canadian Conference on Computational Geometry*, Montréal, Canada, August 1998. `http://cgm.cs.mcgill.ca/cccg98/proceedings/cccg98-biedl-unfolding.ps.gz`.

[BDD+01a] 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 three dimensions. *Discrete & Computational Geometry*, 26(3):283–287, October 2001. The full version is Technical Report 060, Smith College, 1999, and arXiv:cs.CG/9910009, `http://www.arXiv.org/abs/cs.CG/9910009`.

[BDD+01b] Therese Biedl, Erik Demaine, Martin Demaine, Sylvain Lazard, Anna Lubiw, Joseph O'Rourke, Steve Robbins, Ileana Streinu, Godfried Toussaint, and Sue Whitesides. A note on reconfiguring tree linkages: Trees can lock. *Discrete Applied Mathematics*, 2001. To appear. The full version is Technical Report SOCS-00.7, School of Computer Science, McGill University, September 2000, and Computing Research Repository paper cs.CG/9910024. http://www.arXiv.org/abs/cs.CG/9910024.

[BDE+01] Marshall Bern, Erik D. Demaine, David Eppstein, Eric Kuo, Andrea Mantler, and Jack Snoeyink. Ununfoldable polyhedra with convex faces. *Computational Geometry: Theory and Applications*, 2001. To appear.

[BDEH01] Marshall Bern, Erik Demaine, David Eppstein, and Barry Hayes. A disk-packing algorithm for an origami magic trick. In *Proceedings of the 3rd International Meeting of Origami Science, Math, and Education*, Monterey, California, March 2001. Improvement of version appearing in *Proceedings of the International Conference on Fun with Algorithms*, Isola d'Elba, Italy, June 1998, pages 32–42.

[BDL+99] Therese C. Biedl, Erik D. Demaine, Sylvain Lazard, Steven M. Robbins, and Michael A. Soss. Convexifying monotone polygons. In *Proceedings of the 10th Annual International Symposium on Algorithms and Computation*, volume 1741 of *Lecture Notes in Computer Science*, pages 415–424, Chennai, India, December 1999. An expanded version is Technical Report CS-99-03, Department of Computer Science, University of Waterloo, 1999.

[BGR88] Brenda S. Baker, Eric Grosse, and Conor S. Rafferty. Nonobtuse triangulation of polygons. *Discrete & Computational Geometry*, 3(2):147–168, 1988.

[BGRT99] Prosenjit Bose, Francisco Gomez, Pedro Ramos, and Godfried Toussaint. Drawing nice projections of objects in space. *Journal of Visual Communication and Image Representation*, 10(2):155–172, 1999.

[BH96] Marshall Bern and Barry Hayes. The complexity of flat origami. In *Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 175–183, Atlanta, January 1996. Reprinted in *Proceedings of the 3rd International Meeting of Origami Science, Math, and Education*, 2001.

[Bie00] Therese Biedl. Polygons needing many flipturns. Technical Report CS-2000-04, Department of Computer Science, University of Waterloo, 2000. ftp://cs-archive.uwaterloo.ca/cs-archive/CS-2000-04/.

[BL98] Bonnie Berger and Tom Leighton. Protein folding in the hydrophobic-hydrophilic (*HP*) model is NP-complete. *Journal of Computational Biology*, 5(1):27–40, 1998.

[BLS99] Therese Biedl, Anna Lubiw, and Julie Sun. When can a net fold to a polyhedron? In *Proceedings of the 11th Canadian Conference on Computational Geometry*, Vancouver, Canada, August 1999. http://www.cs.ubc.ca/conferences/CCCG/elec_proc/fp16.ps.gz.

[BMR95] Marshall Bern, Scott Mitchell, and Jim Ruppert. Linear-size nonobtuse triangulation of polygons. *Discrete & Computational Geometry*, 14(4):411–428, 1995.

[BS74] James H. Bigelow and Norman Z. Shapiro. Implicit function theorems for mathematical programming and for systems of inequalities. *Mathematical Programming*, 6:141–156, 1974.

[Can87] John Canny. A new algebraic method for robot motion planning and real geometry. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, pages 39–48, Los Angeles, California, October 1987.

[Can88] John Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 460–469, Chicago, Illinois, May 1988.

[Cau13] A. L. Cauchy. Deuxième mémoire sur les polygones et polyèdres (Second memoire on polygons and polyhedra). *Journal de l'École Polytechnique*, 9:87–98, May 1813.

[CD90] Hue Sun Chan and Ken A. Dill. Origins of structure in globular proteins. *Proceedings of the National Academy of Sciences USA*, 87:6388–6392, August 1990.

[CD91] H. S. Chan and K. A. Dill. Polymer principles in protein structure and stability. *Annual Reviews of Biophysics and Biophysical Chemistry*, 20:447–490, 1991.

[CD93] Hue Sun Chan and Ken A. Dill. The protein folding problem. *Physics Today*, pages 24–32, February 1993.

[CDR00] Robert Connelly, Erik D. Demaine, and Günter Rote. Straightening polygonal arcs and convexifying polygonal cycles. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 432–442, Redondo Beach, California, November 2000.

[CDR01] Robert Connelly, Erik D. Demaine, and Günter Rote. Infinitesimally locked linkages with applications to locked trees. Manuscript in preparation, 2001.

[CGP+98] Pierluigi Crescenzi, Deborah Goldman, Christos Papadimitriou, Antonio Piccolboni, and Mihalis Yannakakis. On the complexity of protein folding. *Journal of Computational Biology*, 5(3), 1998.

[CH96] Jindong Chen and Yijie Han. Shortest paths on a polyhedron. *International Journal of Computational Geometry and Applications*, 6(2):127–144, 1996.

[CJ98] Jason Cantarella and Heather Johnston. Nontrivial embeddings of polygonal intervals and unknots in 3-space. *Journal of Knot Theory and Its Ramifications*, 7(8):1027–1039, 1998.

[CKM+01] Jorge Alberto Calvo, Danny Krizanc, Pat Morin, Michael Soss, and Godfried Toussaint. Convexifying polygons with simple projections. *Information Processing Letters*, 80(2):81–86, 2001.

[CO99] Roxana Cocan and Joseph O'Rourke. Polygonal chains cannot lock in 4D. In *Proceedings of the 11th Canadian Conference on Computational Geometry*, Vancouver, Canada, August 1999. `http://www.cs.ubc.ca/conferences/CCCG/elec_proc/c17.ps.gz`.

[CO01] Roxana Cocan and Joseph O'Rourke. Polygonal chains cannot lock in 4D. Technical Report 063, Smith College, February 2001. `http://www.arXiv.org/abs/cs.CG/9908005`.

[Con80] Robert Connelly. The rigidity of certain cabled frameworks and the second-order rigidity of arbitrarily triangulated convex surfaces. *Advances in Mathematics*, 37(3):272–299, 1980.

[Con82] Robert Connelly. Rigidity and energy. *Inventiones Mathematicae*, 66(1):11–33, 1982.

[Con93] Robert Connelly. Rigidity. In *Handbook of Convex Geometry*, volume A, pages 223–271. North-Holland, Amsterdam, 1993.

[Cro97] Peter R. Cromwell. Equality, rigidity, and flexibility. In *Polyhedra*, chapter 6, pages 219–247. Cambridge University Press, 1997.

[CSW97] R. Connelly, I. Sabitov, and A. Walz. The bellows conjecture. *Beiträge zur Algebra und Geometrie (Contributions to Algebra and Geometry)*, 38(1):1–10, 1997.

[CW82] Henry Crapo and Walter Whiteley. Statics of frameworks and motions of panel structures, a projective geometric introduction (with a French translation). *Structural Topology*, 6:43–82, 1982.

[CW93] Henry Crapo and Walter Whiteley. Autocontraintes planes et polyèdres projetés. I. Le motif de base [Plane self stresses and projected polyhedra. I. The basic pattern]. *Structural Topology*, 20:55–78, 1993.

[CW94] Henry Crapo and Walter Whiteley. Spaces of stresses, projections and parallel drawings for spherical polyhedra. *Beiträge zur Algebra und Geometrie*, 35(2):259–281, 1994.

[CW96] Robert Connelly and Walter Whiteley. Second-order rigidity and prestress tensegrity frameworks. *SIAM Journal on Discrete Mathematics*, 9(3):453–491, 1996.

[DD97] Erik D. Demaine and Martin L. Demaine. Computing extreme origami bases. Technical Report CS-97-22, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, May 1997.

[DD01] Erik D. Demaine and Martin L. Demaine. Recent results in computational origami. In *Proceedings of the 3rd International Meeting of Origami Science, Math, and Education*, Monterey, California, March 2001. To appear.

[DDL98] Erik D. Demaine, Martin L. Demaine, and Anna Lubiw. Folding and cutting paper. In J. Akiyama, M. Kano, and M. Urabe, editors, *Revised Papers from the Japan Conference on Discrete and Computational Geometry*, volume 1763 of *Lecture Notes in Computer Science*, pages 104–117, Tokyo, Japan, December 1998.

[DDL+99a] Erik Demaine, Martin Demaine, Anna Lubiw, Joseph O'Rourke, and Irena Pashchenko. Metamorphosis of the cube. In *8th Annual Video Review of Computational Geometry, Proceedings of the 15th Annual ACM Symposium on Computational Geometry*, pages 409–410, Miami Beach, Florida, June 1999. `http://db.uwaterloo.ca/~eddemain/metamorphosis/`.

[DDL99b] Erik D. Demaine, Martin L. Demaine, and Anna Lubiw. Folding and one straight cut suffice. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 891–892, Baltimore, MD, January 1999.

[DDL00] Erik D. Demaine, Martin L. Demaine, and Anna Lubiw. Flattening polyhedra. Manuscript, 2000.

[DDLO] Erik D. Demaine, Martin L. Demaine, Anna Lubiw, and Joseph O'Rourke. Enumerating foldings and unfoldings between polygons and polytopes. *Graphs and Combinatorics*. To appear. `http://www.arXiv.org/abs/cs.CG/0107024`.

[DDLO00a] Erik Demaine, Martin Demaine, Anna Lubiw, and Joseph O'Rourke. The 85 foldings of the latin cross, 2000. `http://db.uwaterloo.ca/~eddemain/aleksandrov/cross/`.

[DDLO00b] Erik Demaine, Martin Demaine, Anna Lubiw, and Joseph O'Rourke. Examples, counterexamples, and enumeration results for foldings and unfoldings between polygons and polytopes. Technical Report 069, Smith College, Northampton, MA, July 2000. `http://www.arXiv.org/abs/cs.CG/0007019`.

[DDM99] Erik D. Demaine, Martin L. Demaine, and Joseph S. B. Mitchell. Folding flat silhouettes and wrapping polyhedral packages: New results in computational origami. In *Proceedings of the 15th Annual ACM Symposium on Computational Geometry*, Miami Beach, Florida, June 1999.

[DDM00] Erik D. Demaine, Martin L. Demaine, and Joseph S. B. Mitchell. Folding flat silhouettes and wrapping polyhedral packages: New results in computational origami. *Computational Geometry: Theory and Applications*, 16(1):3–21, 2000.

[DEE+01] Erik D. Demaine, David Eppstein, Jeff Erickson, George W. Hart, and Joseph O'Rourke. Vertex-unfoldings of simplicial manifolds. arXiv:cs.CG/0110054, October 2001. `http://www.arXiv.org/abs/cs.CG/0110054`.

[Dem00] Erik D. Demaine. Folding and unfolding linkages, paper, and polyhedra. In *Proceedings of the Japan Conference on Discrete and Computational Geometry*, Lecture Notes in Computer Science, Tokyo, Japan, November 2000.

[DGK63] Ludwig Danzer, Branko Grünbaum, and Victor Klee. Helly's theorem and its relatives. In *Proc. Sympos. Pure Math.*, volume VII, pages 101–180. American Mathematical Society, 1963.

[Dil90] Ken A. Dill. Dominant forces in protein folding. *Biochemistry*, 29(31):7133–7155, August 1990.

[DLO01] Erik D. Demaine, Stefan Langerman, and Joseph O'Rourke. Short interlocked linkages. In *Proceedings of the 13th Canadian Conference on Computational Geometry*, Waterloo, Canada, August 2001.

[DM01] Erik D. Demaine and Joseph S. B. Mitchell. Reaching folded states of a rectangular piece of paper. In *Proceedings of the 13th Canadian Conference on Computational Geometry*, Waterloo, Canada, August 2001.

[DO99] Erik D. Demaine and Joseph O'Rourke. Open problems from CCCG'99. In *Proceedings of the 11th Canadian Conference on Computational Geometry*, Vancouver, Canada, August 1999. `http://www.cs.ubc.ca/conferences/CCCG/elec_proc/open.ps.gz`.

[Dür77] Albrecht Dürer. *The Painter's Manual: A Manual of Measurement of Lines, Areas, and Solids by Means of Compass and Ruler Assembled by Albrecht Dürer for the Use of All Lovers of Art with Appropriate Illustrations Arranged to be Printed in the Year MDXXV*. Abaris Books, Inc., New York, 1977. English translation of *Unterweysung der Messung mit dem Zirkel un Richtscheyt in Linien Ebnen und Gantzen Corporen*, 1525.

[ELR+98] H. Everett, S. Lazard, S. Robbins, H. Schröder, and S. Whitesides. Convexifying star-shaped polygons. In *Proceedings of the 10th Canadian Conference on Computational Geometry*, Montréal, Canada, August 1998. `http://cgm.cs.mcgill.ca/cccg98/proceedings/cccg98-everett-convexifying.ps.gz`.

[Epp97] David Eppstein. Faster circle packing with application to nonobtuse triangulation. *International Journal of Computational Geometry and Applications*, 7(5):485–491, 1997.

[Erd35] Paul Erdős. Problem 3763. *American Mathematical Monthly*, 42:627, 1935.

[Eve93] Brian S. Everitt. *Cluster Analysis*. Edward Arnold (London) and Halsted Press (New York), third edition, 1993.

[FHM+01] Thomas Fevens, Antonio Hernandez, Antonio Mesa, Michael Soss, and Godfried Toussaint. Simple polygons that cannot be deflated. *Beiträge zur Algebra und Geometrie (Contributions to Algebra and Geometry)*, 2001. To appear.

[Fia76] Anthony V. Fiacco. Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical Programming*, 10:287–311, 1976.

[Fuj82] S. Fujimoto. *Sojo suru origami asobi no shotai (Invitation to creative origami playing)*. Asahi Culture Center, 1982.

[Gar60] Martin Gardner. Mathematical games. *Scientific American*, 202, 1960. The problem is posed in the February issue (number 202) on page 150. A dissection into seven parts appears in the March issue (number 3), pages 177–178.

[Gar95a] Martin Gardner. *New Mathematical Diversions*, pages 34 and 39–42. Mathematical Association of America, Washington, D.C., 1995.

[Gar95b] Martin Gardner. Paper cutting. In *New Mathematical Diversions (Revised Edition)*, Spectrum Series, chapter 5, pages 58–69. The Mathematical Association of America, Washington, D.C., 1995.

[GCK91] U. Grenander, Y. Chow, and D. M. Keenan. *Hands: A Pattern Theoretic Study of Biological Shapes*. Springer-Verlag, 1991.

[Glu74] Herman Gluck. Almost all simply connected closed surfaces are rigid. In *Proceedings of the Geometric Topology Conference*, volume 438 of *Lecture Notes in Mathematics*, pages 225–239, Park City, Utah, February 1974.

[Gre87] Ulf Grenander. Can one understand shapes in nature? Seminar talk, Indiana University, March 27 1987.

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

[GS87] Branko Grünbaum and G. C. Shephard. *Tilings and patterns*. W. H. Freeman, 1987.

[GSS93] Jack Graver, Brigitte Servatius, and Herman Servatius. *Combinatorial rigidity*. American Mathematical Society, Providence, 1993.

[GZ98] Branko Grünbaum and Joseph Zaks. Convexification of polygons by flips and by flipturns. Technical Report 6/4/98, Department of Mathematics, University of Washington, Seattle, 1998.

[Har73] National standards and emblems. *Harper's New Monthly Magazine*, 47(278):171–181, July 1873.

[Hay98] Brian Hayes. Prototeins. *American Scientist*, 86:216–221, 1998.

[Hea56] Sir Thomas Little Heath. *The thirteen books of Euclid's Elements translated from the text of Heiberg with introduction and commentary*. Dover Publications, New York, 1956.

[HI95] William E. Hart and Sorin Istrail. Fast protein folding in the hydrophobic-hydrophilic model within three-eights of optimal. In *Proceedings of the 27th Annual ACM Symposium on the Theory of Computing*, pages 157–168, Las Vegas, Nevada, May–June 1995.

[Hir00] Koichi Hirata. Find all polytope, 2000. `http://weyl.ed.ehime-u.ac.jp/cgi-bin/WebObjects/Polytope2`.

[HJ97] Pierre Hansen and Brigitte Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79B:191–215, 1997.

[HJW84] John Hopcroft, Deborah Joseph, and Sue Whitesides. Movement problems for 2-dimensional linkages. *SIAM Journal on Computing*, 13(3):610–629, August 1984.

[HK92] John E. Hopcroft and Peter J. Kahn. A paradigm for robust geometric algorithms. *Algorithmica*, 7(4):339–380, 1992.

[Hou22] Harry Houdini. *Paper Magic*, pages 176–177. E. P. Dutton & Company, 1922. Reprinted by Magico Magazine.

[Hul94] Thomas Hull. On the mathematics of flat origamis. *Congressus Numerantium*, 100:215–224, 1994.

[Hul01] Thomas Hull. Counting mountain-valley assignments for flat folds. *Ars Combinatoria*, 2001. To appear.

[JS99] D. Jordan and M. Steiner. Configuration spaces of mechanical linkages. *Discrete & Computational Geometry*, 22:297–315, 1999.

[Jus94] Jacques Justin. Towards a mathematical theory of origami. In Koryo Miura, editor, *Proceedings of the 2nd International Meeting of Origami Science and Scientific Origami*, pages 15–29, Otsu, Japan, November–December 1994.

[JW90] E. J. Janse van Rensburg and S. G. Whittington. The pivot algorithm and polygons: results on the FCC lattice. *Journal of Physics A: Mathematical and General Physics*, 23:1589–1612, 1990.

[Kap99] Sanjiv Kapoor. Efficient computation of geodesic shortest paths. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 770–779, Atlanta, Georgia, May 1999.

[Kaw89] Toshikazu Kawasaki. On the relation between mountain-creases and valley-creases of a flat origami. In H. Huzita, editor, *Proceedings of the 1st International Meeting of Origami Science and Technology*, pages 229–237, Ferrara, Italy, December 1989. An unabridged Japanese version appeared in *Sasebo College of Technology Report*, 27:153–157, 1990.

[Kem76] A. B. Kempe. On a general method of describing plane curves of the n^{th} degree by linkwork. *Proceedings of the London Mathematical Society*, 7:213–216, 1876.

[Kem77] A. B. Kempe. *How to Draw a Straight Line: A Lecture on Linkages*. Macmillan and co., London, 1877.

[Kir97] Rob Kirby. Problems in low-dimensional topology. In *Geometric Topology: Proceedings of the 1993 Georgia International Topology Conference*, volume 2.2 of *AMS/IP Studies in Advanced Mathematics*, pages 35–473. American Mathematical Society and International Press, 1997. `http://www.math.berkeley.edu/~kirby/problems.ps.gz`.

[KM95] Michael Kapovich and John Millson. On the moduli space of polygons in the Euclidean plane. *Journal of Differential Geometry*, 42(1):133–164, 1995.

[KO00] Biliana Kaneva and Joseph O'Rourke. An implementation of Chen & Han's shortest paths algorithm. In *Proceedings of the 12th Canadian Conference on Computational Geometry*, pages 139–146, Fredericton, Canada, August 2000. `http://www.cs.unb.ca/conf/cccg/eProceedings/8.ps.gz`. See also `http://www.cs.smith.edu/~orourke/ShortestPaths/` for a gallery of examples.

[KY89] Toshikazu Kawasaki and Masaaki Yoshida. Crystallographic flat origamis. In H. Huzita, editor, *Proceedings of the 1st International Meeting of Origami Science and Technology*, pages 223–227, Ferrara, Italy, December 1989. An unabridged Japanese version appeared in *Sasebo College of Technology Report*, 24:101–109, 1987.

[Lan94a] Robert J. Lang. Mathematical algorithms for origami design. *Symmetry: Culture and Science*, 5(2):115–152, 1994.

[Lan94b] Robert J. Lang. The tree method of origami design. In Koryo Miura, editor, *Proceedings of the 2nd International Meeting of Origami Science and Scientific Origami*, pages 73–82, Otsu, Japan, November–December 1994.

[Lan96] Robert J. Lang. A computational algorithm for origami design. In *Proceedings of the 12th Annual ACM Symposium on Computational Geometry*, pages 98–105, Philadelphia, PA, May 1996.

[Lan98] Robert J. Lang. *TreeMaker 4.0: A Program for Origami Design*, 1998. `http://origami.kvi.nl/programs/TreeMaker/trmkr40.pdf`.

[Lan01] Robert J. Lang. Trees and circles: an efficient algorithm for origami design. In *Proceedings of the 3rd International Meeting of Origami Science, Math, and Education*, Monterey, California, March 2001.

[LD89] K. F. Lau and K. A. Dill. A lattice statistical mechanics model of the conformation and sequence spaces of proteins. *Macromolecules*, 22:3986–3997, 1989.

[LD90] K. F. Lau and K. A. Dill. Theory for protein mutability and biogenesis. *Proceedings of the National Academy of Sciences USA*, 87:638–642, January 1990.

[LO96] Anna Lubiw and Joseph O'Rourke. When can a polygon fold to a polytope? Technical Report 048, Smith College, June 1996.

[Lun68] W. F. Lunnon. A map-folding problem. *Mathematics of Computation*, 22(101):193–199, 1968.

[Lun71] W. F. Lunnon. Multi-dimensional map-folding. *The Computer Journal*, 14(1):75–80, February 1971.

[LW91a] William J. Lenhart and Sue H. Whitesides. Turning a polygon inside-out. In *Proceedings of the 3rd Canadian Conference on Computational Geometry*, pages 66–69, Vancouver, Canada, August 1991.

[LW91b] David J. Lipman and W. John Wilber. Modelling neutral and selective evolution of protein folding. *Proceedings of The Royal Society of London, Series B*, 245(1312):7–11, 1991.

[LW93] William J. Lenhart and Sue H. Whitesides. Reconfiguring simple polygons. Technical Report SOCS-93.3, School of Computer Science, McGill University, March 1993.

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

[Man60] Wallace Manheimer. Solution to problem E 1406: Dissecting an obtuse triangle into acute triangles. *American Mathematical Monthly*, 67:923, November 1960.

[Mil94] Kenneth C. Millett. Knotting of regular polygons in 3-space. *Journal of Knot Theory and its Ramifications*, 3(3):263–278, 1994.

[ML94] Kenneth M. Merz and Scott M. Le Grand, editors. *The Protein Folding Problem and Tertiary Structure Prediction*. Birkhäuser, Boston, 1994.

[MMP87] Joseph S. B. Mitchell, David M. Mount, and Christos H. Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, August 1987.

[Mon91] John Montroll. *African Animals in Origami*. Dover Publications, 1991.

[Mon93] John Montroll. *Origami Inside-Out*. Dover Publications, 1993.

[MS88] Neal Madras and Alan D. Sokal. The pivot algorithm: A highly efficient Monte Carlo method for the self-avoiding walk. *Journal of Statistical Physics*, 50:109–186, 1988.

[MS93] Neal Madras and Gordon Slade. *The Self-Avoiding Walk*. Birkhäuser, Boston, 1993.

[Nag39] Béla de Sz. Nagy. Solution to problem 3763. *American Mathematical Monthly*, 46:176–177, March 1939.

[O'R98] Joseph O'Rourke. Folding and unfolding in computational geometry. In *Revised Papers from the Japan Conference on Discrete and Computational Geometry*, volume 1763 of *Lecture Notes in Computer Science*, pages 258–266, Tokyo, Japan, December 1998.

[Pal] Chris K. Palmer. *Bay Area Rapid Folders Newsletter*. Issues January 1994, March 1994, Winter 1995, Winter 1996, Spring 1996, Summer 1996, and Spring 1997. The last article is joint with Paulo Barreto.

[Pal94] Chris K. Palmer. Extruding and tesselating polygons from a plane. In Koryo Miura, editor, *Proceedings of the 2nd International Meeting of Origami Science and Scientific Origami*, pages 323–331, Otsu, Japan, November–December 1994.

[RW81] B. Roth and W. Whiteley. Tensegrity frameworks. *Transactions of the American Mathematical Society*, 265(2):419–446, 1981.

[Sab00] Idzhad Kh. Sabitov. Some applications of the volume formula for polyhedra. Presentation at the Discrete Geometry Seminar, Mathematisches Forschungsinstitut Oberwolfach, March 2000. See also `http://www.cms.math.ca/CMS/Events/winter98/w98-abs/node46.html`.

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

[Sch87] Catherine Schevon. Unfolding polyhedra. `sci.math` Usenet article, February 1987. See `http://www.ics.uci.edu/~eppstein/gina/unfold.html`.

[Sch89] Catherine Schevon. *Algorithms for Geodesics on Polytopes.* PhD thesis, Johns Hopkins University, 1989.

[Sen21] Kan Chu Sen. *Wakoku Chiyekurabe (Mathematical Contests).* 1721.

[SEO01] Michael Soss, Jeff Erickson, and Mark Overmars. On the difficulty of preprocessing for dihedral rotation queries. Manuscript in preparation, 2001.

[She75] G. C. Shephard. Convex polytopes with convex nets. *Mathematical Proceedings of the Cambridge Philosophical Society*, 78:389–403, 1975.

[Smi76] John S. Smith. Origami profiles. *British Origami*, 58, June 1976.

[Smi93] John S. Smith. *Pureland Origami 1, 2, and 3.* British Origami Society. Booklets 14, 29, and 43, 1980, 1988, and 1993.

[Sos01] Michael Soss. *Geometric and Computational Aspects of Molecular Reconfiguration.* PhD thesis, School of Computer Science, McGill University, 2001.

[SR34] Ernst Steinitz and Hans Rademacher. *Vorlesungen über die Theorie der Polyeder.* Springer-Verlag, Berlin, 1934. Reprinted 1976.

[ST00] Michael Soss and Godfried T. Toussaint. Geometric and computational aspects of polymer reconfiguration. *Journal of Mathematical Chemistry*, 27(4):303–318, 2000.

[Str00] Ileana Streinu. A combinatorial approach to planar non-colliding robot arm motion planning. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 443–453, Redondo Beach, California, November 2000.

[Sun99] Julie Sun. Folding orthogonal polyhedra. Master's thesis, Department of Computer Science, University of Waterloo, 1999.

[SZ67] I. J. Schoenberg and S. K. Zaremba. Cauchy's lemma concerning convex polygons. *Canadian Journal of Mathematics*, 19(4):1062–1071, 1967.

[Tou99a] Godfried Toussaint. Computational polygonal entanglement theory. In *Proceedings of the VIII Encuentros de Geometria Computacional*, Castellon, Spain, July 1999.

[Tou99b] Godfried Toussaint. The Erdős-Nagy theorem and its ramifications. In *Proceedings of the 11th Canadian Conference on Computational Geometry*, Vancouver, Canada, August 1999. `http://www.cs.ubc.ca/conferences/CCCG/elec_proc/fp19.ps.gz`.

[Tou01] Godfried Toussaint. A new class of stuck unknots in pol_6. *Beiträge zur Algebra und Geometrie (Contributions to Algebra and Geometry)*, 2001. To appear.

[UM93a] Ron Unger and John Moult. Finding the lowest free energy conformation of a protein is a NP-hard problem: Proof and implications. *Bulletin of Mathematical Biology*, 55(6):1183–1198, 1993.

[UM93b] Ron Unger and John Moult. A genetic algorithm for 3D protein folding simulations. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 581–588, San Mateo, CA, 1993.

[UM93c] Ron Unger and John Moult. Genetic algorithms for protein folding simulations. *Journal of Molecular Biology*, 231(1):75–81, 1993.

[Ver98] Helena Verrill. Origami tessellations. In R. Sarhangi, editor, *Conference Proceedings of Bridges: Mathematical Connections in Art, Music, and Science*, pages 55–68, 1998.

[Wal96] Wolfgang Walter. *Gewöhnliche Differentialgleichungen*. Springer-Verlag, Berlin, 6th edition, 1996. English translation: Ordinary differential equations, Springer, New York, 1998.

[Wan97] Cheng-Hua Wang. *Manufacturability-driven decomposition of sheet metal*. PhD thesis, Carnegie Mellon University, Robotics Institute, September 1997. Technical report CMU-RI-TR-97-35.

[Weg93] Bernd Wegner. Partial inflation of closed polygons in the plane. *Beiträge zur Algebra und Geometrie*, 34(1):77–85, 1993.

[Wel91] David Wells. Hinged tessellations. In *The Penguin Dictionary of Curious and Interesting Geometry*, pages 101–103. Penguin Books, 1991.

[Whi82] Walter Whiteley. Motions and stresses of projected polyhedra (with a French translation). *Structural Topology*, 7:13–38, 1982.

[Whi84a] Walter Whiteley. Infinitesimally rigid polyhedra. I. statics of frameworks. *Transactions of the American Mathematical Society*, 285(2):431–465, 1984.

[Whi84b] Walter Whiteley. The projective geometry of rigid frameworks. In *Finite geometries*, volume 103 of *Lecture Notes in Pure and Applied Mathematics*, pages 353–370, Winnipeg, Manitoba, 1984.

[Whi87] Walter Whiteley. Rigidity and polarity. I. Statics of sheet structures. *Geometriae Dedicata*, 22(3):329–362, 1987.

[Whi88] Walter Whiteley. Infinitesimally rigid polyhedra. II. Modified spherical frameworks. *Transactions of the American Mathematical Society*, 306(1):115–139, 1988.

[Whi92a] Walter Whiteley. Matroids and rigid structures. In Neil White, editor, *Matroid applications*, volume 40 of *Encyclopedia of Mathematics and its Applications*, pages 1–52. Cambridge University Press, Cambridge, 1992.

[Whi92b] Sue Whitesides. Algorithmic issues in the geometry of planar linkage movement. *Australian Computer Journal*, 24(2):42–50, May 1992.