Vertex-Unfoldings of Simplicial Polyhedra

Erik D. Demaine* David Eppstein† Jeff Erickson‡

George W. Hart§ Joseph O'Rourke¶

July 18, 2001

Abstract

We present two algorithms for unfolding the surface of any polyhedron, all of whose faces are triangles, to a nonoverlapping, connected planar layout. The surface is cut only along polyhedron edges. The layout is connected, but it may have a disconnected interior: the triangles are connected at vertices, but not necessarily joined along edges.

1 Introduction

It is a long-standing open problem to decide whether every convex polyhedron may be cut along edges and unfolded flat in one piece without overlap, i.e., unfolded to a simple polygon. This type of unfolding has been termed *edge-unfolding*; the unfolding consists of the faces of the polyhedron joined along edges. In contrast, unfolding via arbitrary cuts easily leads to nonoverlap. See [O'R00] for history and applications to manufacturing. Recently it was established that not every nonconvex polyhedron may be edge-unfolded, even if the polyhedron is *simplicial*, that is, all of its faces are triangles [BDEK99, BDE+01]. In this paper we loosen the meaning of "in one piece" to permit a nonoverlapping connected region that (in general) does not form a simple polygon, because its interior is disconnected. We call such an unfolding a *vertex-unfolding*: the faces of the polyhedron are joined at vertices (and sometimes edges). With this easier goal we obtain a positive result: the surface of every

*Dept. of Computer Science, Univ. of Waterloo, Waterloo, Ontario N2L 3G1, Canada. eddemaine@uwaterloo.ca.

†Dept. of Information and Computer Science, Univ. of California, Irvine CA 92697-3425, USA. eppstein@ics.uci.edu. Supported by NSF grant CCR-9912338.

‡Dept. of Computer Science, Univ. of Illinois at Urbana-Champaign; http:// www.cs.uiuc.edu/~jeffe; jeffe@cs.uiuc.edu. Partially supported by a Sloan Fellowship and NSF CAREER award CCR-0093348.

§http://www.georgehart.com/; george@georgehart.com.

¶Dept. of Computer Science, Smith College, Northampton, MA 01063, USA. orourke@cs.smith.edu. Supported by NSF grant CCR-9731804.

simplicial polyhedron, convex or nonconvex, of any genus, may be cut along edges and unfolded to a planar, nonoverlapping, connected layout. Our proof relies on the restriction that every face is a triangle. The problem remains open for nonsimplicial polyhedra with simply connected faces (see Section 7).

2 Overview of Algorithm

Let $\mathcal{P}$ be a simplicial polyhedron, and let G be the *lattice graph* of the face lattice of the polyhedron: the nodes of G are the facets (triangles), edges, and vertices of $\mathcal{P}$, with an arc for each incidence.

Define a *facet path* in G to be a path that alternates between vertices and facets, includes each facet exactly once, and never repeats the same vertex twice in a row. In such a path $p = (\ldots, v_1, f, v_2, \ldots)$, v_1 and v_2 are distinct vertices of f.

Our first observation is that if G contains a facet path p, then a vertex-unfolding exists. For the triangle f can be placed inside a vertical strip with v_1 and v_2 on the left and right strip boundaries. Doing this for each triangle in p yields a nonoverlapping unfolding, connected at the strip boundaries.

However, we do not know whether every lattice graph has a facet path. We can prove this only for polyhedra of genus zero (Theorem 3 in Section 6). Instead we establish that that every G (for any simplicial polyhedron, of any

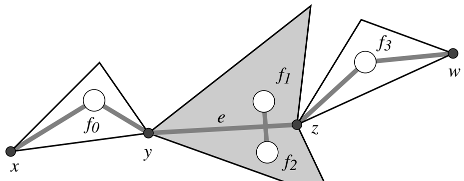

Figure 1: An unfolding path.

genus) has an "unfolding path," which is roughly a facet path that may also include quadrilaterals. More precisely, an *unfolding path* is a path in G that alternates between vertices and nonvertices, covers each facet exactly once, and never repeats the same vertex twice in a row. An edge-node of a path covers the two adjacent facets; otherwise a facet is covered if it is a node of the path. An example is shown in Fig. 1. Here the path is $(x, f_0, y, e, z, f_3, w)$, with e covering f_1 and f_2. Of course every facet path is an unfolding path.

As is evident in the figure, if the quadrilateral $Q = f_1 \cup f_2$ is nonconvex, it is no longer as straightforward to place Q inside a vertical strip. However, we can always choose to open up the end of $e = yz$ at which Q has a convex angle (y in the figure), which then allows Q to be placed in a strip. See Fig. 2.

Therefore, once an unfolding path is found for G, a vertex-unfolding can be achieved.

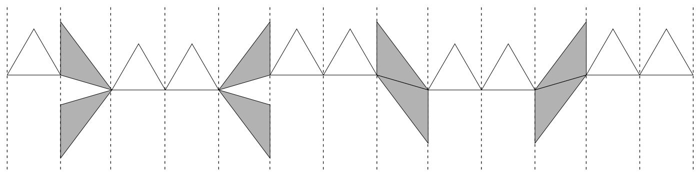

Figure 2: Laying out an unfolding path in vertical strips. The quadrilaterals are shaded. (The example is contrived.)

3 Algorithm for Arbitrary Genus

Our proof that every lattice graph G for a simplicial polyhedron $\mathcal{P}$ has an unfolding path is via an algorithm that finds such a path. The first step is to convert the surface into a simplicial complex D forming a topological disk (henceforth, a *disk*) by cutting a sufficient number of polyhedron edges. For a polyhedron of genus 0, just one edge needs be cut. The algorithm operates on D, finding a path through its triangles, and occasionally employing a quadrilateral when it can no longer extend with a triangle. In particular, the unfolding path starts at a vertex s on the boundary of D, and ends at a boundary vertex $t \neq s$.

Let D be a disk containing at least one triangle. We'll let ∂D represent its boundary. Let s and t be distinct boundary vertices. Vertex s has two distinct neighbors s_1 and s_2 on ∂D, and t has two distinct neighbors t_1 and t_2 on ∂D. Call the triangles incident to ss_i and tt_i, $i = 1, 2$, *s-wings* and *t-wings* respectively. Although there may be as many as four distinct wings, there could be as few as one, because several of the wings might coincide. Say that a triangle T *breaks the disk* D if $D \setminus T$ is not a topological disk. Define an s-wing to be a *good wing* if it is not incident to t and does not break the disk; similarly a t-wing is good if it is not incident to s and does not break the disk. Good wings permit easy advancement of the facet path. Throughout we let $\pi(s, D, t)$ represent an unfolding path from s to t through D, and use $\oplus$ to represent path concatenation: $\pi(a, D_1, b) \oplus \pi(b, D_2, c)$ is the path from a to b in D_1 joined at b to the path from b to c in D_2: $(a, \ldots, b, \ldots, c)$.

1. There is a good wing T. Let $T = \triangle ss_1u$; all other cases are symmetric. We join (s, T, u) to the recursively constructed path in the remainder (Fig. 3(1)):
$$\pi(s, D, t) = (s, T, u) \oplus \pi(u, D \setminus T, t)\ .$$
Note that $u \in D \setminus T$ and $u \neq t$, which justifies the recursion. Whether T is an s- or a t-wing, the structurally similar path construction suffices to reduce to a smaller disk, either from the s- or from the t-end.

2. There is an s-wing T that is not incident to t, or a t-wing that is not incident to s. Again let $T = \triangle ss_1u$; all other cases are symmetric. Because T is not good, it must break the disk, which implies that u is on ∂D. Let D_1 and D_2 be the disks separated by T, with $s \in D_1$. Note that neither s_1 nor u can be t.

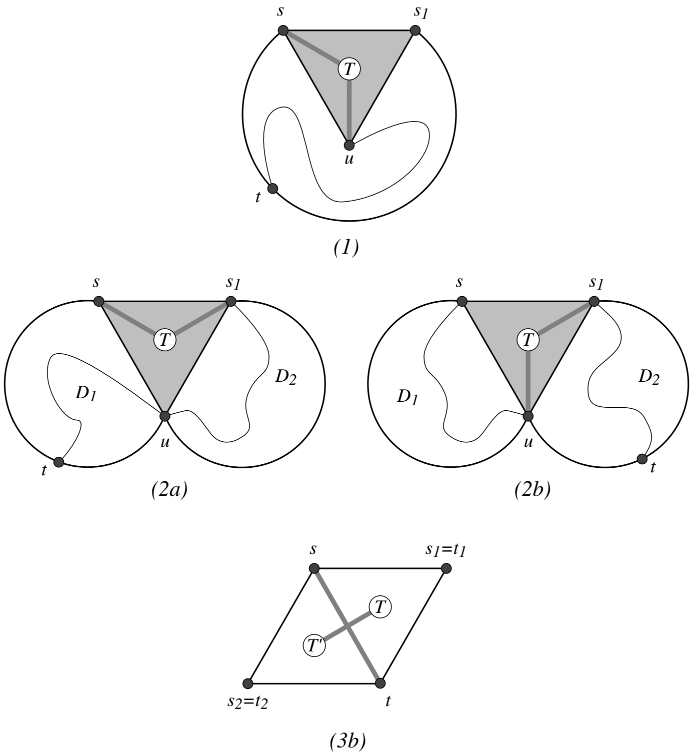

Figure 3: Cases of the algorithm.

(a) $t \in D_1$ (Fig. 3(2a)):

$$\pi(s, D, t) = (s, T, s_1) \oplus \pi(s_1, D_2, u) \oplus \pi(u, D_1, t) \; .$$

(b) $t \in D_2$ (Fig. 3(2b)):

$$\pi(s, D, t) = \pi(s, D_1, u) \oplus (u, T, s_1) \oplus \pi(s_1, D_2, t) \; .$$

3. Every s-wing is incident to t, and every t-wing is incident to s. Then it must be that D is either a single triangle, or a quadrilateral.

   (a) D is a triangle T. Then (s, T, t) is an unfolding path for D.

   (b) D is a quadrilateral $T \cup T'$; Fig. 3(3b). Then (s, st, t) is an unfolding path for D.

It is not difficult to see that the algorithm can be implemented to require time only linear in the number of triangles in the disk D.

4 Algorithm Proof

Theorem 1 *Any triangulated topological disk D has an unfolding path connecting any two distinct boundary vertices s and t.*

Proof: The proof is by induction on the number of triangles n in D. It obviously holds for $n = 1$. The algorithm just described clearly results in an unfolding path for D, by construction. The only issues that remains are verifying that the cases indeed exhaustively cover the possibilities, and that in each case, the conditions of the induction hypothesis hold.

A good wing by definition neither breaks the disk, nor is incident to t (resp. s) for an s- (resp. t-) wing. Case 1 covers good wings, and Cases 2 and 3 cover the two ways to fail being a good wing: Case 2 for wings that do not violate the incidence condition (in which case they must violate the breaking condition), and Case 3 the wings that do violate the incidence condition. Thus the cases are mutually exclusive and comprehensive.

That the induction hypothesis holds in each case is easily seen. We are careful to ensure that the start and end vertices in each recursive application are distinct boundary vertices, and that the subcomplex being traversed is a disk.

The only issue that remains is why Case 3 requires D to be a triangle or a quadrilateral. Suppose there are two s-wings, so that s_1 and s_2 are distinct. Each must be incident to t, so t is neither s_1 nor s_2. Thus the two wings must be $T = \triangle s s_1 t$ and $T' = \triangle s s_2 t$. If $s_1 t$ or $s_2 t$ is not a boundary edge, then some wing of t is not incident to s, which would place us in Case 2. So both are boundary edges, and D is the quadrilateral $T \cup T'$.

Finally, suppose that there is just one s-wing $T = \triangle s s_1 u$. Then both ss_1 and su must be boundary edges, with either $s_1 = t$ or $u = t$. In either case, if $s_1 u$ is not a boundary edge, there would be a t-wing not incident to s, again leading to Case 2. Because we know Case 2 does not hold, $s_1 u$ must be a boundary edge, and $D = T$. $\square$

By our remarks in Section 2, Theorem 1 suffices to establish our main result:

Theorem 2 *The surface of any simplicial polyhedron P (of any genus) may be vertex-unfolded (in linear time): cut along edges and unfolded to a nonoverlapping, connected planar layout. In the layout, adjacent vertical strips each contain one or two triangles of P.*

Note that the resulting unfolding could be viewed as a *hinged dissection* [Fre97] of the surface; see, for example, Fig. 4.

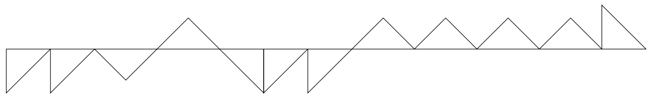

Figure 4: Unfolding of the surface of a triangulated cube.

5 Examples

We have implemented the algorithm of Section 3 and applied it to a number of convex polyhedra. Fig. 5 shows several examples. The polyhedra were generated as convex hulls of randomly generated points. Most unfoldings were in fact facet paths: encountering a quadrilateral (in Case 3b of the algorithm) was somewhat rare. However, the figure illustrates only cases in which one or more quadrilaterals occur.

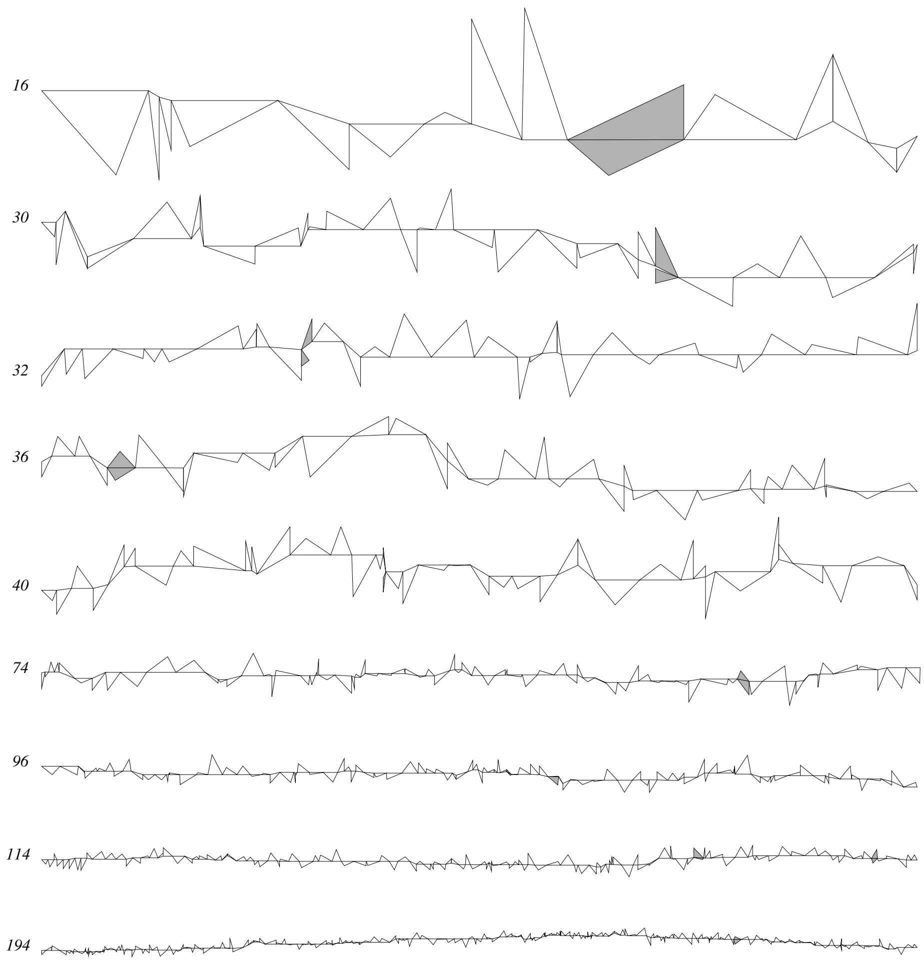

Figure 5: Unfoldings of random convex polyhedra (generated by code from [O'R98]). The number of triangles is indicated to the left of each unfolding. Quadrilaterals are shaded.

The next section shows that for polyhedra of genus zero, quadrilaterals can be avoided entirely.

6 Genus-Zero Facet Cycles

Theorem 3 *The lattice graph of any simplicial polyhedron P of genus zero contains a facet cycle $C(P)$.*

A *facet cycle* is a facet path (cf. Sec. 2) that is also a cycle.

Proof: Note that any facet path in which each vertex is incident to an even number of path edges is a facet cycle, because it supports an Eulerian tour.

The proof is by induction on the number of polyhedron edges. If P is a tetrahedron, then there is a facet cycle, as shown in Fig. 6.

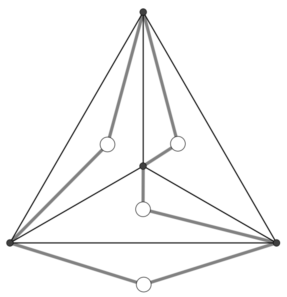

Figure 6: The lattice graph of a tetrahedron contains a facet cycle.

Otherwise, let xy be any edge of P, and contract it to form the lattice graph of a polyhedron Q. (In a triangulated planar graph, at least one edge incident to any vertex can be contracted, unless the graph is K_4, which is the induction base.) By induction, there is a facet cycle $C(Q)$ in the graph for Q. See Fig. 7(a-c).

Let L be the link of x on P: the edges opposite x of all triangles incident to x. L forms a cycle on P, and on Q. Now, remove the portion of $C(Q)$ that is inside L on Q, as in Fig. 7(d); let H be the resulting subgraph of $C(Q)$. The task now is to augment H on P so that it is connected, its vertices are even, and it covers all the triangles inside L on P.

Let u be a vertex of L. Label u *odd* or *even* if the number of edges of H incident to u is odd or even respectively (see Fig. 7(d)). Let $T(u)$ be the triangle $\triangle uxv$ of P, where u, x, and v are in counterclockwise order. We use the following rule to augment H. For each $u \in L$, if u is odd, add $(v, T(u), x)$; if u is even, add $(u, T(u), v)$. See Fig. 7(e). Call the augmented graph $C(P)$.

By construction, each triangle inside L is covered by $C(P)$. Each vertex $u \in L$ becomes incident to an added edge from the clockwise previous u', regardless of whether u' is odd or even. If u is odd, then u only receives one new edge from the clockwise previous vertex; if u is even, it in addition adds an edge to cover $T(u)$. Consequently, every vertex on L has even degree in $C(P)$. The vertex x is even because it receives an edge from every odd vertex on L, and there are an even number of odd vertices in H (as in any graph), all of which are on L.

Next we argue that the vertices on the link L are connected. If there is an odd vertex u, then the added edges form a collection of "arms" extending

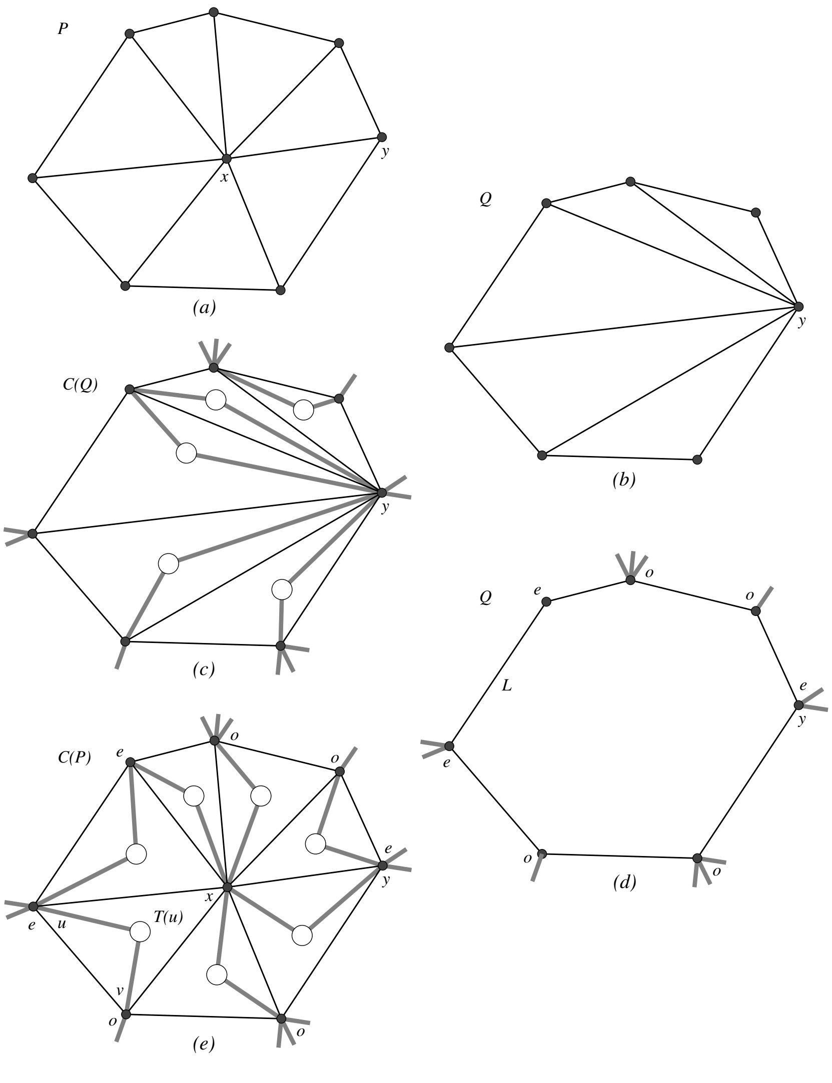

Figure 7: (a) P; (b) Q after contracting xy; (c) a facet cycle on Q; (d) after removing edges inside the contracted neighborhood; e=even, o=odd; (d) a facet cycle on P.

from each odd vertex, clockwise through consecutive even vertices, and then connecting through x. If all vertices are even, then the added vertices form a cycle excluding x. In either case, the vertices of L become connected in $C(P)$.

Finally we show that $C(P)$ is connected. Let $u \in L$ and let z be a vertex of $C(P)$ outside of L. Because $C(Q)$ is connected, there must be some path p in $C(Q)$ from z to a vertex on L; let $w \in L$ be the first such. (Note that L contains at least one triangle, and $C(Q)$ visits two of its vertices, at least one of which is on L.) The portion of the path p from z to w is unchanged on P. Because the vertices on L are connected in $C(P)$, there is a connection from z to w to u in $C(P)$. Thus $C(P)$ is connected.

We have established that $C(P)$ covers the facets, is connected, and has even degree at each vertex. Therefore it is a facet cycle. $\square$

Following this proof, and taking care to contract an edge incident to a low-degree vertex [CE91], leads to a linear-time algorithm. Through the vertical-strip layout, this theorem permits the surface of any triangulated genus-zero polyhedron to be unfolded to a string of triangles joined at vertices:

Corollary 4 *The surface of any simplicial polyhedron P of genus zero may be vertex-unfolded (in linear time) into parallel strips each containing one triangle of P.*

Theorem 3 also yields an "ideal rendering" of any such surface on a computer graphics system with a 1-vertex cache: each triangle shares one vertex with the previous triangle in the graphics pipeline. It is known that sharing two vertices is not always achievable: some triangulations do not admit a "sequential triangulation," that is, an ordering of the triangles corresponding to a Hamiltonian path in the dual graph [AHMS96].

The restriction to simplicial polyhedra in Theorem 3 (and indeed in Theorems 1-2 as well) is necessary, for the truncated cube has no facet path: no pair of its eight triangles can be adjacent in a path, but the six octagons are not enough to separate the triangles.

7 Discussion

Our work raises three new open problems:

1. Does every lattice graph of a simplicial polyhedron of genus more than zero have a facet path, i.e., can Theorem 3 be extended to higher-genus polyhedra?

2. Does every polyhedron with faces homeomorphic to a disk have a nonoverlapping vertex-unfolding? The strip construction fails for faces with more than three sides. If faces are permitted to have holes, then there are examples that cannot be vertex-unfolded, e.g., a box-on-top-of-a-box (cf. Fig. 7 of [BDD+98]).

3. Does every 4-polytope, or more generally, every polyhedral complex in $\mathbb{R}^4$, have a vertex-unfolding? It was this question that prompted our investigation.

Acknowledgments We thank Anna Lubiw for a clarifying discussion, and Allison Baird, Dessislava Michaylova, and Amanda Toop for assisting with the implementation.

References

[AHMS96] E. M. Arkin, M. Held, J. S. B. Mitchell, and S. S. Skiena. Hamiltonian triangulations for fast rendering. *Visual Comput.*, 12(9):429–444, 1996.

[BDD+98] T. Biedl, E. Demaine, M. Demaine, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, and S. Whitesides. Unfolding some classes of orthogonal polyhedra. In *Proc. 10th Canad. Conf. Comput. Geom.*, pages 70–71, 1998. Fuller version in Elec. Proc. http://cgm.cs.mcgill.ca/cccg98/proceedings/welcome.hml.

[BDE+01] M. Bern, E. D. Demaine, D. Eppstein, E. Kuo, A. Mantler, and J. Snoeyink. Ununfoldable polyhedra with convex faces. *Comput. Geom. Theory Appl.*, 2001. To appear.

[BDEK99] M. Bern, E. D. Demaine, D. Eppstein, and E. Kuo. Ununfoldable polyhedra. In *Proc. 11th Canad. Conf. Comput. Geom.*, pages 13–16, 1999. Full version: LANL archive paper number cs.CG/9908003.

[CE91] M. Chrobak and D. Eppstein. Planar orientations with low outdegree and compaction of adjacency matrices. *Theoretical Computer Science*, 86(2):243–266, September 1991.

[Fre97] G. Frederickson. *Dissections: Plane and Fancy.* Cambridge University Press, 1997.

[O'R98] J. O'Rourke. *Computational Geometry in C (Second Edition).* Cambridge University Press, 1998.

[O'R00] J. O'Rourke. Folding and unfolding in computational geometry. In *Proc. Japan Conf. Discrete Comput. Geom. '98*, volume 1763 of *Lecture Notes Comput. Sci.*, pages 258–266. Springer-Verlag, 2000.