Adaptive Simplicial Grids from Cross-sections of Monotone Complexes*

Herbert Edelsbrunner† and Roman Waupotitsch†

Abstract

We study the maintenance of a simplicial grid or complex under changing density requirements. The proposed method works in any fixed dimension and generates grids by projecting cross-sections of a monotone simplicial complex that lives in one dimension higher than the grid. The density of the grid is adapted by locally moving the cross-section up or down along the extra dimension.

Keywords. Geometric and topological algorithms, dynamic data structures, grid generation, multi-grids, simplicial complexes, hierarchies, directed acyclic graphs.

1 Introduction

For many applications of geometric grids, it is important to adapt it to local density requirements. As an example consider finite element analysis for finding an approximate solution to a differential equation. For reasons of efficiency and also accuracy, it is desirable that the density of the grid changes as first approximations to the solution become available, see e.g. [8]. Ideally, the grid generator and the solver should be integrated into a short feed-back cycle. Adaptation requires that in some regions vertices be deleted and in regions of interest vertices be added. During the course of several iterations it can happen that some regions change from coarse to dense and back to coarse. Fluctuations in the desired density occur in particular in grids modeling objects that change over time.

Results. In this paper we describe a hierarchical approach to adaptive grid generation based on simplicial complexes and fast data structures. The method works in any fixed dimension; in this paper we describe it for $(d-1)$-dimensional Euclidean space, R^{d-1}. The main idea is to perform local changes to the $(d-1)$-dimensional grid, which is a simplicial $(d-1)$-complex, and to record these changes in a data structure, which is a directed acyclic graph. A non-geometric interpretation of the graph is a hierarchically accumulated record of the history. The novelty of our approach lies in how we use the history in order to manipulate the presence. Since the past is recorded in an effective data structure, we can roll-back time to re-generate earlier states of the grid. It also offers the possibility to combine states of the grid at different times in different places. This can be understood as a consistent snap-shot in the framework of time defined as a partial order of events. A geometric interpretation of the graph is a simplicial d-complex imbedded in R^d. We can think of a snap-shot as a cross-section within the d-complex; its projection into R^{d-1} is a $(d-1)$-dimensional grid. In order to make these abstract ideas work, we need to understand the topology of the situation, and we need to design and implement efficient data structures and algorithms that provide a responsive environment.

Our specific approach is based on ideas in [3, 4], where a directed acyclic graph, called *history dag*, is used to compute weighted and unweighted Delaunay complexes in R^{d-1}. The simplices of these complexes correspond to the sinks of the dag. We view the complexes as grids that discretize geometric objects in R^{d-1}. The dag is constructed as follows. Initially, it consists of a single node or d-simplex. It is repeatedly extended by attaching nodes or d-simplices to sinks. At any point in time, the dag represents a complex in R^d, and the vertical projection of the lower boundary is a $(d-1)$-dimensional grid. The attachment operation in R^d can be interpreted as flipping a local configuration in R^{d-1}, see [3, 4]. The idea of the history dag can be gener-

*Work of the first author is partially supported by NSF under grant CCR-96-19542 and by ARO under grant DAAG55-98-1-0177. Work of the second author is partially supported through a fellowship by the IBM corporation.

†Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA.

alized to encompass a larger class of d-complexes. In particular, we consider the class of d-complexes satisfying a certain monotonicity property. This paper studies properties of these d-complexes that can be exploited for fast algorithms manipulating the grid and its history. These algorithms will make it possible to apply the data structure to problems where adaptive grids are required to handle large amounts of possibly dynamic data.

Outline. Section 2 presents definitions, including the central notion of a monotone simplicial complex. Section 3 studies cross-sections of monotone d-complexes and their projections into $d-1$ dimensions. Section 4 introduces ancestors and descendants of simplices and discusses their relationship to cross-sections. Section 5 exhibits connectivity properties of monotone 2-complexes that fail in three and higher dimensions. Section 6 proves connectivity results that hold in all dimensions. Section 7 considers the algorithmic problem of combining cross-sections. Section 8 studies the rotation operation for monotone d-complexes. Section 9 concludes the paper.

2 Definitions

We begin by introducing basic concepts and notation. A certain familiarity with simplicial complexes as treated in the algebraic topology literature is useful in dealing with the occasional accumulation of notation. We refer to Munkres [6] for an introductory text in algebraic topology and to Bern and Eppstein [1] for a survey of combinatorial aspects of grid generation.

Simplicial complexes. A *k-simplex*, σ^k, is the convex hull of a set T of $k+1$ affinely independent points; its *dimension* is $\dim \sigma^k = k = \operatorname{card} T - 1$. Special terms are used for small k: a *vertex* is a 0-simplex, an *edge* is a 1-simplex, a *triangle* is a 2-simplex, and a *tetrahedron* is a 3-simplex. A simplex spanned by a subsets $U \subseteq T$ is a *face* of σ^k, or *ℓ-face* if $\ell = \operatorname{card} U - 1$. Examples are the empty set, which is the only (-1)-face of σ^k, and σ^k itself, which is its only k-face. The $(k-1)$-faces are the *facets*, and the $(k-2)$-faces are the *ridges* of σ^k. In R^d, at most $d+1$ points can be affinely independent, so the dimension of simplices can be at most d. A finite collection of simplices, $\mathcal{K}$, is a *simplicial complex* if the faces of every simplex in $\mathcal{K}$ also belong to $\mathcal{K}$, and the intersection of two simplices in $\mathcal{K}$ is either empty or a face of both. The *dimension* of $\mathcal{K}$ is $\dim \mathcal{K} = \max_{\sigma \in \mathcal{K}} \dim \sigma$, and if this dimension is k then we call $\mathcal{K}$ a *k-complex.* A k-complex $\mathcal{K}$ is *pure* if every $\sigma \in \mathcal{K}$ is a face of a k-simplex in $\mathcal{K}$. The *underlying space* of $\mathcal{K}$ is $\|\mathcal{K}\| = \bigcup_{\sigma \in \mathcal{K}} \sigma$. A *subcomplex* of $\mathcal{K}$ is a simplicial complex $\mathcal{L} \subseteq \mathcal{K}$.

Vertical ordering of simplices. A few definitions are needed before the notion of monotonicity of a complex can be introduced. This notion depends on the fact that a $(d-1)$-simplex in R^d can be a facet of at most two d-simplices. Call the direction parallel to the x_d-axis *vertical.* For simplicity, we assume general position with respect to the vertical direction, that is, each σ^k, $k < d$, intersects a vertical line in at most one point. Let σ^{d-1} be a facet of σ^d, and let l be a vertical line that intersects the interior of σ^{d-1}. We call σ^{d-1} a *lower facet* of σ^d if $l \cap \sigma^{d-1}$ is the lowest point of $l \cap \sigma^d$. Symmetrically, we call σ^{d-1} an *upper facet* of σ^d if $l \cap \sigma^{d-1}$ is the highest point of $l \cap \sigma^d$. Two d-simplices, σ_i^d and σ_j^d, are *adjacent* if they share a facet, σ^{d-1}. In this case, σ^{d-1} is a lower facet of one d-simplex, say σ_i^d, and an upper facet of the other, σ_j^d. We say that σ_i^d lies *above* σ_j^d and σ_j^d lies *below* σ_i^d. This relationship between the d-simplices of a simplicial complex, $\mathcal{K}$, in R^d can be expressed by a directed graph, $\mathsf{G} = \mathsf{G}(\mathcal{K})$. The nodes of G are the d-simplices of $\mathcal{K}$, and there is a directed arc (σ_i^d, σ_j^d) in G if the two d-simplices are adjacent and σ_i^d lies above σ_j^d. We call σ_i^d the *predecessor* of σ_j^d, and σ_j^d the *successor* of σ_i^d. Since the arcs of G correspond to facets shared by two d-simplices, every node in G has at most as many predecessors and successors as a d-simplex has facets, namely $d+1$. A node without predecessor is a *source*, and a node without successor is a *sink.* If $\dim \mathcal{K} < d$ then G is empty.

DEFINITION. A simplicial complex, $\mathcal{K}$, in R^d is *monotone* with respect to the vertical direction if

- (M1) its underlying space, $|\mathcal{K}|$, intersects every vertical line in a single and possibly degenerate interval, and
- (M2) the corresponding directed graph, $\mathsf{G} = \mathsf{G}(\mathcal{K})$, is acyclic.

Note, that G may be disconnected. See Figures 2.1 and 2.2 for small examples of monotone and non-monotone complexes.

Interior, closure, and boundary. It is convenient to introduce notions for complexes and their subsets that mimic the common point set topological concepts of interior, closure, and boundary. The notions of

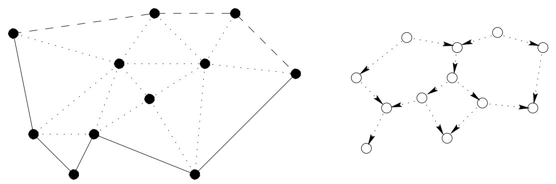

Figure 2.1: Example of a monotone 2-complex and its directed graph. The lower boundary is marked by solid and the upper boundary by dashed edges.

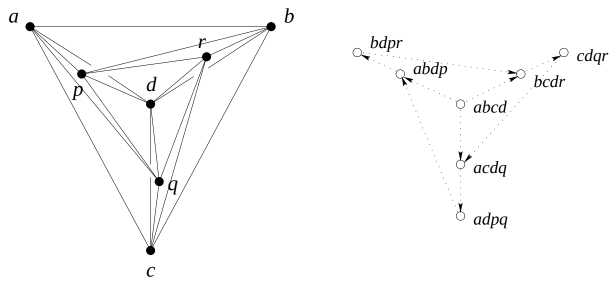

Figure 2.2: Example of a non-monotone 3-complex and its directed graph. Condition (M2) is violated because the digraph of the 3-complex contains a cycle. Here and in later drawings of 3-dimensional examples we look at a complex from below.

boundary and interior are defined relative to the imbedding space, rather than relative to the complex or set of simplices.

Let K be a collection of simplices so that the intersection of any two simplices in K is either empty or a face of both. In other words, K is a subset of a simplicial complex. The *closure* of K is the smallest complex $\mathrm{cl}\, K$ so that $K \subseteq \mathrm{cl}\, K$. If K is a complex, the *boundary* of K is the smallest subcomplex $\mathrm{bd}\, K \subseteq K$ that contains every $\sigma^k \in K$, $k < d$, that is a face of at most one d-simplex. If K is not a complex we define $\mathrm{bd}\, K = \mathrm{bd}\,\mathrm{cl}\, K$. The *interior* of K is $\mathrm{int}\, K = K - \mathrm{bd}\, K$. By first taking the interior and then the closure we can eliminate simplices that are not face of any d-simplex. The result is either empty of a pure d-complex. This sequence of operations will be used repeatedly, so we define $\mathrm{reg}\, K = \mathrm{cl}\,\mathrm{int}\, K$ and call it the *regularization* of K. The corresponding concept for subsets of R^d is indeed common in the solid modeling literature, see e.g. [7]. We list a few properties that are both instructive and useful.

FACT 2.1 Let $\mathcal{K}$ be a pure d-complex, let $K \subseteq \mathcal{K}$, and let $\mathcal{L}_1, \mathcal{L}_2 \subseteq \mathcal{K}$ be pure d-dimensional subcomplexes.

(i) $\mathrm{reg}\, \mathcal{K} = \mathcal{K}$.

(ii) $\mathrm{bd}\, \mathcal{K}$ is a pure $(d-1)$-complex.

(iii) $\mathrm{reg}\, K$ is the largest pure d-complex contained in $\mathrm{cl}\, K$.

(iv) $\mathrm{reg}\,(\mathcal{L}_1 \cup \mathcal{L}_2) = \mathcal{L}_1 \cup \mathcal{L}_2$ is a pure d-dimensional subcomplex of $\mathcal{K}$.

(v) $\mathrm{reg}\,(\mathcal{L}_1 - \mathcal{L}_2) = \mathrm{cl}\,(\mathcal{L}_1 - \mathcal{L}_2)$ is either empty or a pure d-dimensional subcomplex of $\mathcal{K}$.

(vi) $\mathrm{reg}\,(\mathcal{L}_1 \cap \mathcal{L}_2) = \mathrm{cl}\,\mathrm{int}\,(\mathcal{L}_1 \cap \mathcal{L}_2)$ is either empty or a pure d-dimensional subcomplex of $\mathcal{K}$.

3 Cross-sections

The central notion in this section is that of a cross-section of a monotone d-complex. Cross-sections are interesting because their vertical projection into R^{d-1} are grids, and by adjusting the cross-section we can manipulate the grid.

Prefixes and cross-sections. Most complexes considered in this paper are d- or $(d-1)$-dimensional and imbedded in R^d or R^{d-1}. Particularly important are pure monotone d-complexes, and throughout we let $\mathcal{M}$ denote such a complex. Every subcomplex of $\mathcal{M}$ inherits (M2) but not necessarily (M1). We are interested in subcomplexes that also satisfy (M1). These subcomplexes need not be d-dimensional, and if their dimension is less than d then (M2) is vacuous because G is empty. Clearly, the complex obtained by removing a sink from $\mathcal{M}$ satisfies (M1) and (M2). Formally, by *removing* a sink σ^d we mean the operation that deletes σ^d and all faces of σ^d not shared by any other d-simplex. Using the above notation, this is the same as substituting $\mathrm{reg}\,(\mathcal{M} - \{\sigma^d\})$ for $\mathcal{M}$. We call a complex obtained by repeated removal of a sink a *prefix* of $\mathcal{M}$.

FACT 3.1 Every prefix of a pure monotone d-complex is either empty or again a pure monotone d-complex.

By Fact 2.1 (ii), the boundary of $\mathcal{M}$ is a pure $(d-1)$-complex. Each $(d-1)$-simplex $\sigma^{d-1} \in \mathrm{bd}\, \mathcal{M}$ is either a lower or an upper facet of a d-simplex in $\mathcal{M}$. In the more general case of a monotone subcomplex $\mathcal{K} \subseteq \mathcal{M}$, there are also $(d-1)$-simplices that are not a facet of any d-simplex. Let $L, H \subseteq \mathrm{bd}\, \mathcal{K}$ be the set of $(d-1)$-simplices that are lower, upper facet of some d-simplex in $\mathcal{K}$, and let $M \subseteq \mathrm{bd}\, \mathcal{K}$ be the set of remaining $(d-1)$-simplices. The *lower boundary* of $\mathcal{K}$ is $\mathrm{bd}_L \mathcal{K} = \mathrm{cl}\,(L \cup M)$, and the *upper boundary* is $\mathrm{bd}_H \mathcal{K} = \mathrm{cl}\,(H \cup M)$, see Figure 2.1. Note that $\mathrm{bd}\, \mathcal{M} = \mathrm{bd}_L \mathcal{M} \cup \mathrm{bd}_H \mathcal{M}$ and $\mathrm{bd}_L \mathrm{bd}_H \mathcal{M} = \mathrm{bd}_H \mathcal{M}$.

DEFINITION. The *cross-section* defined by a prefix $\mathcal{P}$ of $\mathcal{M}$ is the $(d-1)$-complex $\mathcal{C} = \mathrm{bd}_L(\mathcal{P} \cup \mathrm{bd}_H\mathcal{M})$.

If $\mathcal{P}$ contains the upper boundary of $\mathcal{M}$ then the cross-section it defines is the same as its lower boundary. Otherwise, it is its lower boundary together with some simplices of $\mathrm{bd}_H\mathcal{M}$. In either case, the cross-section is a monotone $(d-1)$-complex.

Projections. Next we consider vertical projections of cross-sections into R^{d-1}. Let $\mathcal{M}$ be a pure monotone d-complex, as usual. We write $\mathrm{proj}\, X$ for the vertical projection of $X \subseteq \mathrm{R}^d$ into R^{d-1}. Thus, $Z = \mathrm{proj}\,\|\mathcal{M}\| = \bigcup_{\sigma\in\mathcal{M}} \mathrm{proj}\,\sigma$ is the vertical projection of $\|\mathcal{M}\|$ into R^{d-1}; it is a polytope in R^{d-1}. Because of condition (M1) and because of the requirement that all σ^k, $k < d$, be non-vertical, the vertical projection of $\mathrm{bd}_L\mathcal{M}$,

$$\mathrm{proj}\,\mathrm{bd}_L\mathcal{M} = \{\mathrm{proj}\,\sigma \mid \sigma \in \mathrm{bd}_L\mathcal{M}\},$$

is a simplicial $(d-1)$-complex in R^{d-1}. Furthermore, $Z = \mathrm{proj}\,\|\mathrm{bd}_L\mathcal{M}\| = \|\mathrm{proj}\,\mathrm{bd}_L\mathcal{M}\|$. We call a simplicial $(d-1)$-complex $\mathcal{L}$ in R^{d-1} a *grid* of Z if $\|\mathcal{L}\| = Z$. For every cross-section $\mathcal{C}$ of $\mathcal{M}$, we can consider its vertical projection, $\mathrm{proj}\,\mathcal{C} = \{\mathrm{proj}\,\sigma \mid \sigma \in \mathcal{C}\}$, which is a pure simplicial $(d-1)$-complex in R^{d-1}.

FACT 3.2 The vertical projection of any cross-section $\mathcal{C}$ of $\mathcal{M}$ into R^{d-1} is a grid of Z.

By definition, there is a one-to-one correspondence between cross-sections and prefixes. It is important to notice, that the grids of Z corresponding to cross-sections constitute all possible grids of Z that can be obtained by projecting lower boundaries of monotone complexes $\mathcal{K} \subseteq \mathcal{M}$.

Cross-section poset. As we will see shortly, the set of cross-sections of a pure monotone d-complex has itself a nice structure, namely it forms a lattice. Observe that the collection of prefixes of $\mathcal{M}$ together with the containment relation defines a partially ordered set. This set has the structure of a lattice (see below), because intersections and unions of prefixes lead again to prefixes.

LEMMA 3.3 Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be prefixes of $\mathcal{M}$. Then $\mathrm{reg}\,(\mathcal{P}_1 \cap \mathcal{P}_2)$ and $\mathcal{P}_1 \cup \mathcal{P}_2$ are also prefixes of $\mathcal{M}$.

PROOF. Consider intersection first. Note that $\mathcal{P}_1 \cap \mathcal{P}_2 = \mathcal{P}_1 - (\mathcal{P}_1 - \mathcal{P}_2)$. Choose a sink $\sigma^d \in \mathcal{P}_1 - \mathcal{P}_2$. If σ^d does not exist then $\mathcal{P}_1 \cap \mathcal{P}_2 = \mathcal{P}_1$ and we are done. Otherwise, remove σ^d from $\mathcal{P}_1$. By Fact 3.1, the result is again a prefix of $\mathcal{M}$. Iterate this operation until $\mathcal{P}_1 \cap \mathcal{P}_2$ is generated; it is a prefix by induction. Similarly, $\mathcal{P}_1 \cup \mathcal{P}_2 = \mathcal{P}_1 \cup (\mathcal{P}_2 - \mathcal{P}_1)$, and we can construct $\mathcal{P}_1 \cup \mathcal{P}_2$ by repeatedly adding a source of $\mathcal{P}_2 - \mathcal{P}_1$ to $\mathcal{P}_1$. ◻

From the one-to-one correspondence between prefixes and cross-sections we get another partially ordered set for the cross-sections of $\mathcal{M}$. Let $\mathsf{C} = \mathsf{C}_\mathcal{M}$ be the set of cross-sections of $\mathcal{M}$. For $\mathcal{C}_1, \mathcal{C}_2 \in \mathsf{C}$ let $\mathcal{P}_1$ and $\mathcal{P}_2$ be the prefixes so that $\mathcal{C}_1 = \mathrm{bd}_L(\mathcal{P}_1 \cup \mathrm{bd}_H\mathcal{M})$ and $\mathcal{C}_2 = \mathrm{bd}_L(\mathcal{P}_2 \cup \mathrm{bd}_H\mathcal{M})$, and write $\mathcal{C}_1 \preceq \mathcal{C}_2$ if $\mathcal{P}_1 \subseteq \mathcal{P}_2$. The poset $(\mathsf{C}, \preceq)$ has a unique minimum, $\mathrm{bd}_H\mathcal{M}$, and a unique maximum, $\mathrm{bd}_L\mathcal{M}$. A partially ordered set is a *lattice* if for every pair of elements, there is a unique maximal element preceding both and a unique minimal element succeeding both in the order.

THEOREM 3.4 $(\mathsf{C}, \preceq)$ is a lattice.

PROOF. Let $\mathcal{C}_1, \mathcal{C}_2 \in \mathsf{C}$ be cross-sections and let $\mathcal{P}_1, \mathcal{P}_2$ be the corresponding prefixes. Define $\mathcal{P}_H = \mathrm{reg}\,(\mathcal{P}_1 \cap \mathcal{P}_2)$ and $\mathcal{P}_L = \mathcal{P}_1 \cup \mathcal{P}_2$, and let

$$\begin{aligned}\mathcal{C}_H &= \mathrm{bd}_L(\mathcal{P}_H \cup \mathrm{bd}_H\mathcal{M}),\\ \mathcal{C}_L &= \mathrm{bd}_L(\mathcal{P}_L \cup \mathrm{bd}_H\mathcal{M}).\end{aligned}$$

We have $\mathcal{C}_H \preceq \mathcal{C}_1, \mathcal{C}_2 \preceq \mathcal{C}_L$. Furthermore, $\mathcal{C} \preceq \mathcal{C}_H$ for every $\mathcal{C}$ that precedes $\mathcal{C}_1$ and $\mathcal{C}_2$, and $\mathcal{C}_L \preceq \mathcal{C}$ for every $\mathcal{C}$ that succeeds $\mathcal{C}_1$ and $\mathcal{C}_2$. Hence, $\mathcal{C}_H$ is the required unique maximal element that precedes both cross-sections, and $\mathcal{C}_L$ is the unique minimal element that succeeds them. ◻

Changing one cross-section to another is like tracing a path from one element in $(\mathsf{C}, \preceq)$ to another. The fact that $(\mathsf{C}, \preceq)$ is a lattice simplifies navigation significantly. In other words, there are efficient ways to manipulate cross-sections in a predictable manner.

4 Extreme Cross-sections

An important question discussed in Section 7 is how to select and manipulate cross-sections. We introduce terminology that will help us to study this question in detail.

Ancestors and descendants. Let $\mathcal{M}$ be a pure monotone d-complex, as usual, and let $\sigma_i^d \in \mathcal{M}$. Another d-simplex, $\sigma_j^d \in \mathcal{M}$, is an *ancestor* of σ_i^d if $\sigma_j^d = \sigma_i^d$ or σ_j^d is a predecessor of σ_i^d in the transitive closure of $\mathsf{G} = \mathsf{G}(\mathcal{M})$. The *ancestor complex* of σ_i^d is

$$\mathcal{A}(\sigma_i^d) = \mathrm{cl}\,\{\sigma^d \in \mathcal{M} \mid \sigma^d \text{ is ancestor of } \sigma_i^d\}.$$

For a set $M \subseteq \mathcal{M}$ of d-simplices, $\mathcal{A}(M)$ is the union of ancestor complexes of d-simplices in M. Symmetrically, σ_k^d is a *descendant* of σ_i^d if σ_i^d is an ancestor of σ_k^d. The *descendant complex* of σ_i^d is

$$\mathcal{D}(\sigma_i^d) \quad = \quad \text{cl}\,\{\sigma^d \in \mathcal{M} \mid \sigma^d \text{ is descendant of } \sigma_i^d\},$$

and $\mathcal{D}(M)$ is the union of descendant complexes of d-simplices in M. See Figure 4.1 for an example. Observe

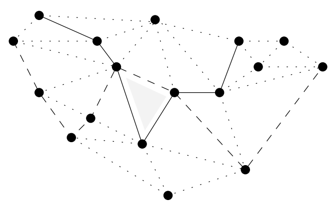

Figure 4.1: The lower boundary of the ancestor complex of the shaded triangle is marked by solid edges. Its projection to the horizontal line is not a grid of $Z = \text{proj}\,|\mathcal{M}|$ because it misses three edges of the upper boundary. The lower boundary of the complex obtained by removing all descendants of the shaded triangle is marked by dashed edges. Its projection is a grid of Z.

that for any set of d-simplices $M \subseteq \mathcal{M}$, it is possible to construct $\mathcal{A}(M)$ and $\text{reg}\,(\mathcal{M} - \mathcal{D}(M))$ by repeatedly removing sinks from $\mathcal{M}$. If not geometrically then this is most easily seen by considering the directed graph $\mathsf{G} = \mathsf{G}(\mathcal{M})$.

FACT 4.1 Let $\mathcal{M}$ be a pure monotone d-complex, and let $M \subseteq \mathcal{M}$ be a set of d-simplices. Then, both $\mathcal{A}(M)$ and $\text{reg}\,(\mathcal{M} - \mathcal{D}(M))$ are prefixes of $\mathcal{M}$.

Next, consider a prefix $\mathcal{P} \subseteq \mathcal{M}$ and let $\sigma^d \in \mathcal{P}$. Observe that $\sigma_j^d \in \mathcal{A}(\sigma^d)$ cannot be a sink of $\mathcal{P}$ unless $\sigma_j^d = \sigma^d$. The following is an extension of this simple observation.

LEMMA 4.2 Let $\mathcal{P}$ be a prefix of $\mathcal{M}$ and $\sigma^d \in \mathcal{M}$.

(i) If $\sigma^d \in \mathcal{P}$ then $\mathcal{A}(\sigma^d) \subseteq \mathcal{P}$.

(ii) If $\sigma^d \notin \mathcal{P}$ then $\text{int}\,\mathcal{D}(\sigma^d) \cap \mathcal{P} = \emptyset$.

To prove Lemma 4.2 one can use the directed graph G of $\mathcal{M}$, which is acyclic by assumption. Let $M \subseteq \mathcal{M}$ be the set of d-simplices in $\mathcal{P}$, and let $\bar{M} \subseteq \mathcal{M}$ contain all other d-simplices. A direct consequence of Lemma 4.2 is $\mathcal{P} = \mathcal{A}(M) = \text{reg}\,(\mathcal{M} - \mathcal{D}(\bar{M}))$. Clearly, M and $\bar{M}$ are not the smallest sets T and S that satisfy this relation.

Highest and lowest cross-sections. A convenient mechanism to select a cross-section is to choose a few simplices of $\mathcal{M}$ and then build a cross-section that contains all these simplices. It can happen that no such cross-section exists, or that there are many such cross-sections. In the latter case, one can ask for the minimal or the maximal such cross-sections. For a subset $M \subseteq \mathcal{M}$, the intersection of all prefixes that contain M defines the minimal prefix

$$\mathcal{P}_H \;=\; \mathcal{P}_H(M) \;=\; \text{reg} \bigcap_{\mathcal{P} \supseteq M} \mathcal{P}.$$

We call $\mathcal{C}_H = \mathcal{C}_H(M) = \text{bd}_L(\mathcal{P}_H \cup \text{bd}_H\mathcal{M})$ the *highest cross-section* of M, see Figure 4.1 for an example. Since $\mathcal{P}_H$ is a prefix of $\mathcal{M}$, it is the ancestor complex of a set of d-simplices. To describe the smallest such set, let $\alpha = \alpha_\sigma$ be the d-simplex in $\mathcal{M}$ so that $\sigma \in \mathcal{M}$ is a face of α, and if $\dim \sigma < d$, α lies vertically above σ. If $\dim \sigma = d$ then $\alpha = \sigma$, and if $\dim \sigma < d$ then α is either undefined (if $\sigma \in \text{bd}_H\mathcal{M}$) or it exists and is unique. Note that all facets of α that contain σ are lower facets of α. Now, $\mathcal{P}_H = \mathcal{A}(\alpha_M)$, where $\alpha_M = \{\alpha_\sigma \mid \sigma \in M\}$.

The notion of a lowest cross-section of M is less natural since $\mathcal{M}$ itself contains M and is, of course, the maximum prefix with this property. Such a notion becomes useful only if we add more requirements. Note that it is not possible in general to require that all $\sigma \in M$ be part of a cross-section. However, if we require that no $\sigma \in M$ be below, then there is a unique maximal subset of M that can be on such a cross-section. This subset is $M_h = M \cap \mathcal{C}_H$. The union of all prefixes with $\sigma \in M_h$ in the corresponding cross-section defines the maximum prefix

$$\mathcal{P}_L \;=\; \mathcal{P}_L(M) \;=\; \bigcup_{\text{bd}_L(\mathcal{P} \cup \text{bd}_H\mathcal{M}) \supseteq M_h} \mathcal{P}.$$

We call $\mathcal{C}_L = \mathcal{C}_L(M) = \text{bd}_L(\mathcal{P}_L \cup \text{bd}_H\mathcal{M})$ the *lowest cross-section* of M.

$\mathcal{C}_H$ and $\mathcal{C}_L$ serve as brackets for the collection of cross-sections that in some sense most accurately represent the chosen set of simplices, M. More specifically, this is the set of cross-sections $\mathcal{C}$ so that $\mathcal{C}_H \preceq \mathcal{C} \preceq \mathcal{C}_L$. We see that Theorem 3.4 has algorithmic consequences. If we have any prefix $\mathcal{P}$, with $M \subseteq \mathcal{P}$, then the highest cross-section of M can be generated by removing sinks that are not α_σ for any $\sigma \in M$. Conversely, we can generate the lowest cross-section of M from the highest cross-section by adding sinks that do not remove any $\sigma \in M$ from the lower boundary. Removing and adding sinks without back-tracking is possible only because $(\mathsf{C}, \preceq)$ is a lattice.

5 Anomalies in 3 Dimensions

Instead through growing and shrinking a prefix one d-simplex at a time, as suggested by Theorem 3.4, we can change cross-sections by direct manipulation of their $(d-1)$-simplices. For example, the cross-section defined by the union of two prefixes can be constructed by choosing the appropriate $(d-1)$-simplices from the two old cross-sections. As it turns out, this approach encounters difficulties that stem from possibly surprising structural irregularities of monotone d-complexes. This section looks at a few such irregularities relevant to the algorithms in Sections 7 and 8. The surprising behavior begins in dimension $d = 3$.

Ancestors without lower boundary facets. Let $\mathcal{M}$ be a pure monotone d-complex in R^d, as usual. For $d = 2$, every d-simplex that has a ridge in $\mathrm{bd}_L\mathcal{M}$ also has a facet in $\mathrm{bd}_L\mathcal{M}$. This is no longer true for $d \geq 3$. Let $\sigma_i^d \in \mathcal{M}$ and let $\mathcal{A} = \mathcal{A}(\sigma_i^d)$ be the corresponding ancestor complex. Let $\sigma^{d-1} \in \mathrm{bd}_L\mathcal{A}$, define $\sigma^d = \alpha_{\sigma^{d-1}}$, and let σ^{d-2} be the ridge common to σ^{d-1} and some upper facet of σ^d. We assume that $\alpha = \alpha_{\sigma^{d-2}}$ exists. By construction we have $\alpha \neq \sigma^d$.

LEMMA 5.1 Let $\mathcal{A}$ and α be as defined above.

(i) If $d = 2$ then α has a lower facet in $\mathrm{bd}_L\mathcal{A}$.

(ii) If $d \geq 3$ then it is possible that α has no lower facet in $\mathrm{bd}_L\mathcal{A}$.

The essential difference between $d = 2$ and $d \geq 3$ that leads to Lemma 5.1 is that in R^2 there are only two sides one can pass by a d-simplex. To verify Lemma 5.1 (i) notice that σ^{d-2} is a vertex and α is a triangle with two lower edges; the two edges meet in σ^{d-2}. If both edges are not in $\mathrm{bd}_L\mathcal{A}$ then both successors of α belong to $\mathcal{A}$. Now, since $\mathcal{A} = \mathcal{A}(\sigma_i^d)$, both successors of α are ancestors of σ_i^d. We thus have two paths from α down to σ_i^d, and they surround the vertex σ^{d-2}. This contradicts that σ^{d-2} belongs to $\mathrm{bd}_L\mathcal{A}$. An example for 5.1 (ii) is given in Figure 5.1.

The positive and negative results in Lemma 5.1 have algorithmic consequences. Because of (i) it is easy in R^2 to construct the lower boundary of an ancestor complex without visiting its interior triangles. Because of (ii) this is not or to a lesser extent possible in R^d, $d \geq 3$.

Absence of connecting directed paths. Let σ_i^d, $\mathcal{A}$, σ^d, σ^{d-1}, σ^{d-2}, and α be as above. By construction, σ^{d-2} is a face of another $(d-1)$-simplex $\tau^{d-1} \neq \sigma^{d-1}$

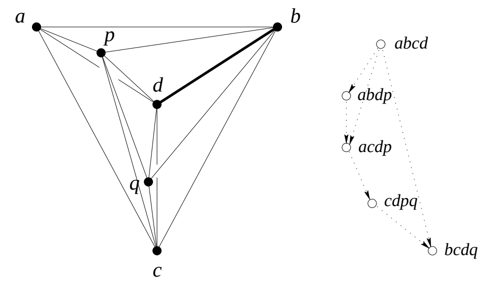

Figure 5.1: A pure monotone 3-complex and its directed graph. Consider the ancestor complex $\mathcal{A} = \mathcal{A}(bcdq)$ and note that no lower facet of $\alpha_{bd} = abcd$ belongs to $\mathrm{bd}_L\mathcal{A}$.

in $\mathrm{bd}_L\mathcal{A}$. Define $\tau^d = \alpha_{\tau^{d-1}}$. Since σ^{d-2} is common to a lower and an upper facet of σ^d, we have $\tau^d \neq \sigma^d$. In $d = 2$ dimensions we have $\tau^d = \alpha$ because, by Lemma 5.1 (i), α has a lower edge in $\mathrm{bd}_L\mathcal{A}$, and this lower edge can only be τ^{d-1}. In $d \geq 3$ dimensions, $\tau^d \neq \alpha$ is possible, as demonstrated in Figure 5.2. Since σ^d and τ^d

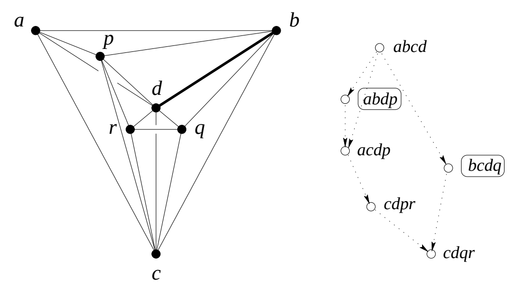

Figure 5.2: A pure monotone 3-complex and its directed graph. Consider $\mathcal{A} = \mathcal{A}(cdqr)$. The tetrahedra $abdp$ and $bcdq$ contain the triangles bdp and bdq of $\mathrm{bd}_L\mathcal{A}$. The two triangles share the ridge bd, but the two tetrahedra do not lie on a common directed path in $\mathsf{G}(\mathcal{A})$.

are both different from α and they are both successors of α, there can be no directed path in $\mathsf{G}(\mathcal{M})$ from σ^d to τ^d or vice versa. We will see some complications in combining cross-sections that arise from the possible absence of a directed path between the two d-simplices.

Non-simply connected ancestor complexes. Another difference between two and higher dimensions occurs for ancestor complexes of a single d-simplex. Let $\mathcal{A} = \mathcal{A}(\sigma_i^d)$ as before. In $d = 2$ dimensions, $\mathrm{bd}_L\mathcal{A}$ and $\mathrm{bd}_H\mathcal{A}$ are two open polygonal curves that share the two endpoints. So $\mathrm{bd}\,\mathcal{A} = \mathrm{bd}_L\mathcal{A} \cup \mathrm{bd}_H\mathcal{A}$ is a closed polygonal curve, and $|\mathcal{A}|$ is a simply connected subset of R^2. As we will see in Section 6, $\|\mathcal{A}\|$ and also $\mathrm{int}\,|\mathcal{A}| = |\mathrm{int}\,\mathcal{A}|$ are connected even for $d \geq 3$. However, already for

$d = 3$ dimensions, it is possible that $|\mathcal{A}|$ is not simply connected. This is demonstrated in Figure 5.3.

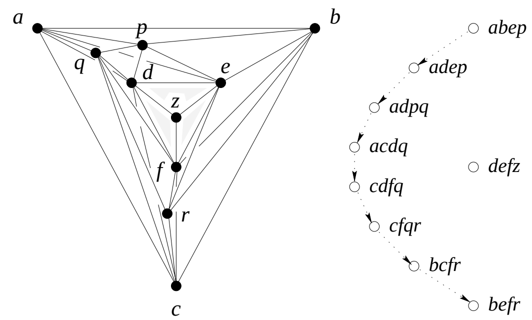

Figure 5.3: A pure monotone 3-simplex and its directed graph. Consider $\mathcal{A} = \mathcal{A}(befr)$. $\mathcal{A}$ contains all tetrahedra except for $defz$ in the middle. Its underlying space is a solid torus pinched along the edge be.

6 General Connectivity Results

In spite of the structural irregularities of monotone d-complexes in $d \geq 3$ dimensions exhibited in Section 5, weaker connectivity requirements hold for all dimensions. These weaker properties are crucial for the efficient manipulation of cross-sections. This section studies what happens when the ancestor complex of a d-simplex is added to a prefix, and it considers the lower boundary of a pure monotone d-complex.

Connectivity in ancestor complexes. We consider connectivity in d-complexes and in $(d-1)$-complexes. Two k-simplices, σ_i^k and σ_j^k, in a k-complex are *adjacent* if they share a $(k-1)$-simplex, and we call them k-*connected* if there is a sequence of k-simplices,

$$\sigma_i^k = \tau_1^k, \tau_2^k, \ldots, \tau_m^k = \sigma_j^k,$$

so that τ_ℓ^k and $\tau_{\ell+1}^k$ are adjacent for all $1 \leq \ell \leq m-1$. A pure k-complex is k-*connected* if every pair of k-simplices is k-connected. Let now $\mathcal{A} = \mathcal{A}(\sigma_i^d)$ be the ancestor complex of $\sigma_i^d \in \mathcal{M}$, and let $\mathcal{P}$ be a prefix of $\mathcal{M}$. As noted in Lemma 4.2 (i), $\mathcal{A} \subseteq \mathcal{P}$ if $\sigma_i^d \in \mathcal{P}$. If $\sigma_i^d \notin \mathcal{P}$ then $\mathcal{B} = \text{reg}\,(\mathcal{A} - \mathcal{P})$ is a non-empty pure d-complex.

LEMMA 6.1 $\mathcal{B} = \text{reg}\,(\mathcal{A} - \mathcal{P})$ is either empty or a d-connected d-complex.

PROOF. If $\sigma_i^d \in \mathcal{P}$ then $\mathcal{A} - \mathcal{P} = \emptyset$, and therefore $\mathcal{B} = \emptyset$. So assume $\sigma_i^d \notin \mathcal{P}$. Every $\sigma^d \in \mathcal{B}$ is ancestor of σ_i^d. It follows that σ_i^d is the only sink in $\mathcal{B}$ and $\mathcal{B}$ is d-connected. ◻

If $\mathcal{B} \neq \emptyset$ then it is a pure d-complex. It follows that its boundary, $\text{bd}\,\mathcal{B}$, is a pure $(d-1)$-complex, and because $\mathcal{B}$ is d-connected, $\text{bd}\,\mathcal{B}$ is $(d-1)$-connected. Monotonicity of $\mathcal{B}$ now implies that $\text{bd}_L\mathcal{B}$ and $\text{bd}_H\mathcal{B}$ are also $(d-1)$-connected. However, as shown in Section 5, the underlying spaces of $\mathcal{B}$, $\text{bd}\,\mathcal{B}$, $\text{bd}_L\mathcal{B}$, and $\text{bd}_H\mathcal{B}$ are not necessarily simply connected.

Paths in lower boundaries. Consider the lower boundary of a pure monotone d-complex $\mathcal{M}$. $(d-1)$-connectivity in $\text{bd}_L\mathcal{M}$ is defined in terms of sequences of adjacent $(d-1)$-simplices. It will be useful to construct such connecting sequences, and we do this indirectly by considering sequences of d-simplices in $\mathcal{M}$. A *path* in $\mathcal{M}$ is a sequence of d-simplices so that any two contiguous d-simplices share a $(d-1)$-simplex; it corresponds to an undirected path in $\mathsf{G} = \mathsf{G}(\mathcal{M})$. A path in $\mathcal{M}$ is *directed* if its corresponding path in G is directed. We think of a path as a subcomplex, $\mathcal{I} \subseteq \mathcal{M}$, namely the closure of the set of d-simplices in the path. The intersection with the lower boundary of $\mathcal{M}$ is a subcomplex of $\text{bd}_L\mathcal{M}$.

LEMMA 6.2 Let $\mathcal{M}$ be a pure monotone d-complex and $\sigma_i^d \in \mathcal{M}$ with a facet in $\text{bd}_L\mathcal{M}$. Then there exists a directed path $\mathcal{I} \subseteq \mathcal{M}$ from σ_i^d to a sink $\sigma_j^d \in \mathcal{M}$ with the property that the facets of σ_i^d and σ_j^d in $\text{bd}_L\mathcal{M}$ are $(d-1)$-connected in $\mathcal{J} = \mathcal{I} \cap \text{bd}_L\mathcal{M}$.

PROOF. If σ_i^d itself is a sink of $\mathcal{M}$ we are done. So assume that σ_i^d is not a sink. Then it has at least two lower facets, at least one in $\text{bd}_L\mathcal{M}$ and at least one other not in $\text{bd}_L\mathcal{M}$, which it shares with a successor d-simplex. Let σ^{d-2} be common to two lower facets of σ_i^d, one in $\text{bd}_L\mathcal{M}$ and one not in $\text{bd}_L\mathcal{M}$. There exists a d-simplex $\sigma_k^d \neq \sigma_i^d$ with a facet in $\text{bd}_L\mathcal{M}$ that contains σ^{d-2}. Consider a maximal sequence $\sigma_i^d = \tau_1^d, \sigma_k^d = \tau_2^d, \ldots, \tau_m^d$, where $\tau_{\ell+1}^d$, for $1 \leq \ell \leq m-1$, is obtained from τ_ℓ^d in the same manner as σ_k^d is obtained from σ_i^d. By maximality of the sequence, $\tau_m^d = \sigma_j^d$ is a sink. For every $1 \leq \ell \leq m-1$, there is a directed a path in $\mathsf{G}(\mathcal{M})$ from τ_ℓ^d to $\tau_{\ell+1}^d$. By acyclicity of $\mathsf{G}(\mathcal{M})$, these paths and their concatenation into a single path, $\mathcal{I}$, are finite. By construction, the $(d-1)$-simplices in $\mathcal{I} \cap \text{bd}_L\mathcal{M}$ define a subcomplex of $\text{bd}_L\mathcal{M}$ in which the lower facets of σ_i^d and σ_j^d are $(d-1)$-connected. The assertion follows because this subcomplex of $\text{bd}_L\mathcal{M}$ is also a subcomplex of $\mathcal{J}$. ◻

7 Combining Cross-sections

Cross-sections can be combined using set operations on the corresponding prefixes. In this section we follow a more direct approach that avoids an exhaustive search through the d-simplices of a prefix. We represent a cross-section as a collection of $(d-1)$-simplices identified by marks within an otherwise unmarked representation of $\mathcal{M}$. The few details about the data structure necessary for our discussion are reviewed after characterizing the resulting cross-section combinatorially. We focus on adding the ancestors of a d-simplex to a given prefix as a representative example. In terms of cross-sections, this operation corresponds to constructing the point-wise minimum of two cross-sections interpreted as continuous maps from R^{d-1} to R.

Characterization of lower boundary. Let $\mathcal{M}$ be a pure monotone d-complex, as usual, and let $\mathcal{P}_1$ and $\mathcal{P}_2 = \mathcal{A}(\sigma_0^d)$ be two prefixes. The corresponding cross-sections are $\mathcal{C}_1 = \mathrm{bd}_L(\mathcal{P}_1 \cup \mathrm{bd}_H\mathcal{M})$, $\mathcal{C}_2 = \mathrm{bd}_L(\mathcal{P}_2 \cup \mathrm{bd}_H\mathcal{M})$, and $\mathcal{C}_L = \mathrm{bd}_L(\mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathrm{bd}_H\mathcal{M})$. A simplex $\sigma \in \mathcal{M}$ belongs to $\mathcal{C}_L$ iff

(1) $\sigma \in \mathcal{C}_1$ and $\sigma \notin \mathcal{P}_2$,

(2) $\sigma \in \mathcal{C}_2$ and $\sigma \notin \mathcal{P}_1$, or

(3) $\sigma \in \mathcal{C}_1$ and $\sigma \in \mathcal{C}_2$.

It is algorithmically more convenient to combine cases (1) and (3) and to write the new cross-section as the union of two sets: $\mathcal{C}_L = (\mathcal{C}_1 - \mathrm{int}\,\mathcal{P}_2) \cup (\mathcal{C}_2 - \mathcal{P}_1)$. The only shared simplices in the two sets belong to $\mathrm{bd}_H\mathcal{M}$, so we can write $\mathcal{C}_L$ as the disjoint union of two sets if we subtract $\mathrm{bd}_H\mathcal{M}$ from the second set. This leads to the formula used for the algorithm.

FACT 7.1 $\mathcal{C}_L = (\mathcal{C}_1 - \mathrm{int}\,\mathcal{P}_2) \cup (\mathrm{bd}_L\mathcal{P}_2 - \mathcal{P}_1)$.

Data structure details. We assume a representation of $\mathcal{M}$ based on $(d-1)$-simplices linked together at common $(d-2)$-simplices. For brevity we refer to $(d-1)$-simplices as facets and to $(d-2)$-simplices as ridges. A ridge, σ^{d-2}, belongs to an ordered cycle of facets, and we store the cycle as two sorted lists, one for each side of the vertical hyperplane passing through σ^{d-2}. Let σ^{d-1} and τ^{d-1} be two facets that both contain σ^{d-2}. They are *comparable* if they lie on the same side of of the hyperplane, and *incomparable* otherwise. In the former case, σ^{d-1} lies *above* τ^{d-1}, and τ^{d-1} lies *below* σ^{d-1}, if $\alpha_{\sigma^{d-1}}$ is an ancestor of $\alpha_{\tau^{d-1}}$ in $\mathsf{G}(\mathcal{M})$. The *topmost* facet is the one above all other comparable facets.

The vertical direction is used to distinguish between different positions of a ridge σ^{d-2} relative to a facet σ^{d-1} that contains it. Let $\alpha = \alpha_{\sigma^{d-1}}$ be the d-simplex above σ^{d-1}. σ^{d-2} is a face of two facets of α, one being σ^{d-1}, and we call σ^{d-2} a *rim* if the other is an upper facet of α. For a rim σ^{d-2} of σ^{d-1} the data structure provides direct access to the topmost facet, τ^{d-1}, that shares σ^{d-2} with σ^{d-1} and is incomparable with σ^{d-1}. We call τ^{d-1} the *steepest extension* of σ^{d-1} *across* σ^{d-2}, see Figure 7.1.

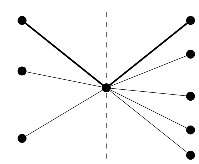

Figure 7.1: The edges to the left of the dashed vertical line are pairwise comparable and so are the edges to the right of that line. Each top edge is the steepest extension of every edge on the other side of the vertical line.

Algorithm. Initially, all facets in $\mathcal{C}_1$ are marked, no other simplices in $\mathcal{M}$ are marked, and $\sigma_0^d \notin \mathcal{P}_1$ is given. The basic idea in constructing $\mathcal{C}_L$ is to extend the lower facets of σ_0^d in the steepest possible way across their rims until facets of $\mathcal{C}_1$ are encountered. This works fine in R^2, but for reasons discussed in Lemma 5.1 (ii), there is trouble in three and higher dimensions. We account for the indicated difficulties by extending the lower facets of σ_0^d in two steps: the first step distributes temporary marks, and the second step converts appropriate temporary marks into permanent ones. Together, the two steps compute the second set on the right side of the formula in Fact 7.1. A third step handles the first set by visiting and unmarking all facets of $\mathcal{C}_1$ that lie in the interior of $\mathcal{P}_2$. We see that the work focuses on the changes in the cross-section and avoids visiting the remaining facets in $\mathcal{C}_1 \cap \mathcal{C}_L$. However, because of the mentioned complications we cannot substantiate the claim that the running time of the algorithm is at most proportional to the number of facets added to or removed from $\mathcal{C}_1$.

Each step is a search through a subset of the facets in $\mathcal{M}$; breadth-first and depth-first search [9] are possible implementations. A search is completely determined by specifying which facets adjacent to the current facet are visited next. If no adjacent facet is visited then the search backtracks and possibly continues elsewhere.

STEP 1. Start a search at every lower facet of σ_0^d. Let σ^{d-1} be the currently visited facet, and attach a temporary mark to σ^{d-1}. For each rim σ^{d-2} of σ^{d-1} that does not belong to $\mathcal{C}_1$ let τ^{d-1} be the steepest extension across σ^{d-2}. Visit τ^{d-1} recursively, unless τ^{d-1} already has a temporary mark or there is another facet below τ^{d-1} that shares σ^{d-2} and has a temporary mark.

Next, temporary marks are converted to permanent marks, which are of the same type as the marks used for facets of $\mathcal{C}_1$.

STEP 2. Start a search at every lower facet of σ_0^d. Let σ^{d-1} again denote the currently visited facet, and change its temporary mark to a permanent mark. For each rim σ^{d-2} of σ^{d-1} that does not belong to $\mathcal{C}_1$ and is not yet a face of two permanently marked facets, let τ^{d-1} be the lowest temporarily marked facet that shares σ^{d-2} and is incomparable to σ^{d-1}. Visit τ^{d-1} recursively.

The final step unmarks facets of $\mathcal{C}_1$ that no longer belong to $\mathcal{C}_L$. By Lemma 6.1, these facets are $(d-1)$-connected and can thus be unmarked in a single search. An arbitrary first such facet can be identified by remembering one facet, τ^{d-1}, marked in Step 2, that has a rim, τ^{d-2}, in $\mathcal{C}_1$. The marked facet, σ_0^{d-1}, that shares τ^{d-2} and lies above τ^{d-1} belongs to $\mathcal{C}_1$ but not to $\mathcal{C}_L$.

STEP 3. Start a search at σ_0^{d-1}. Let σ^{d-1} be the currently visited facet, and unmark it. Consider each ridge σ^{d-2} of σ^{d-1} not contained in a facet marked in Step 2. Recursively visit the other marked facet, τ^{d-1}, that contains σ^{d-2}. If τ^{d-1} exists then it is incomparable to σ^{d-1}.

After executing the three steps we need to remove all remaining temporary marks. This can be done by repeating a search as in Step 1.

8 Rotations

As described in [3, 4], the history dag $\mathcal{M}$ of a $(d-1)$-dimensional Delaunay complex can be constructed incrementally by adding one point at a time. Provided the points are added in a random order, $\mathcal{M}$ is only a small constant times the size of its lower boundary, which is also the size of the Delaunay complex. Since we incorporate update operations issued by the user, we cannot assume randomness and a bias in the order is indeed likely to exist. For this and other reason, we introduce the rotation operation, whose sole purpose is the restructuring of the dag, without adding or removing points. As a side effect, a rotation changes the number of simplices in $\mathcal{M}$ and can thus be used to reduce the memory requirements of the hierarchy. Owing its name to the more familiar rotations for balanced binary trees [9], this operation has been studied in the general context of dynamically maintaining randomized data structures by Mulmuley [5].

We begin with the case where $\mathcal{M}$ is a pure monotone d-complex constructed with the incremental algorithm in [3]. Let $p_0, \ldots, p_i, p_{i+1}, \ldots, p_\ell$ be the points added to the Delaunay complex in this sequence. Different permutations result in different monotone complexes, but the lower boundaries are all the same. Although we assume a particular ordering of the points, we do not actually have to know that ordering. Let $\mathcal{M}'$ denote the pure monotone complex that results by adding the points in an order that differs from the above permutation by exchanging p_i and p_{i+1}, for some $0 \le i < \ell$. We define the operation $\texttt{rotation}(\mathcal{M}, i)$ for $\mathcal{M}$, p_i, and p_{i+1} as the transformation that takes $\mathcal{M}$ to $\mathcal{M}'$. Figure 8.1 shows a rotation for a complex in R^2. Define the

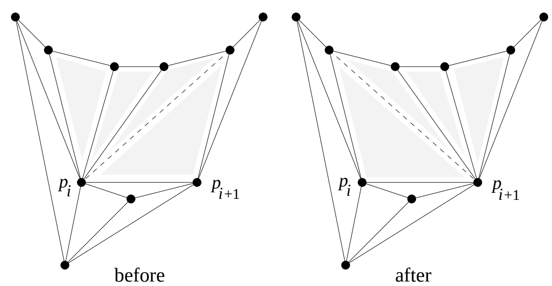

Figure 8.1: Example of a pure monotone 2-complex before and after the rotation that exchanges the order of p_i and p_{i+1}. The triangles in the cups of p_i and p_{i+1} are separated by the dashed edges. The triangles in the symmetric difference of the two complexes are shaded.

cup of a point p_j as the closure of the set of d-simplices added by the algorithm beginning with α_{p_j} and ending before $\alpha_{p_{j+1}}$. The cup of p_j in $\mathcal{M}$ is denoted as U_j and in $\mathcal{M}'$ it is denoted as U_j'. The cups of points p_0 through p_{i-1} are unaffected by later insertions, so they are the same in $\mathcal{M}$ and in $\mathcal{M}'$. Similarly, the cups of points p_{i+2} through p_ℓ are the same in both complexes. Furthermore, the cups of p_i and p_{i+1} together cover the same subset of R^d in both complexes. We can thus define the rotation operation more formally:

$$\texttt{rotation}(\mathcal{M}, i) \; = \; \mathcal{M} - U_i - U_{i+1} \cup U_{i+1}' \cup U_i'.$$

This expression suggests a somewhat different view of a rotation. Instead of explicitly changing the insertion order of the points, we can view a rotation as changing the dependency among the cups.

There are different ways to implement a rotation. For example in $d = 2$ dimensions, a rotation can be performed using edge flips. In general, a good implementation will avoid removing and reinserting d-simplices that belong to $U_i \cup U_{i+1}$ as well as $U'_i \cup U'_{i+1}$. The correctness of a rotation operation depends on convexity conditions for certain cross-sections of the complex. These conditions are satisfied by the monotone d-complexes generated by incremental Delaunay complex algorithms, but they may fail for general monotone complexes. In other words, rotations as described may generally not exist.

9 Discussion

A few years ago, Guibas, Knuth and Sharir [4] introduced the history dag as a data structure that supports efficient point location necessary in the incremental construction of Delaunay complexes in $d - 1 = 2$ dimensions. The history dag has been generalized to weighted Delaunay complexes and to dimensions $d - 1 \geq 3$ in [3]. In this paper we promote the notion that this dag is a versatile representation of simplicial grids. There are several advantages the dag offers over more traditional representations of grids:

(i) it is *hierarchical* and combines fine and crude simplicial decompositions into a single data structure,

(ii) it is *adaptive* so that the grid can be changed locally by moving the active cross-section within the dag,

(iii) it is *dynamic* and admits operations such as adding, moving, and removing a point.

While the dag is more flexible and more efficient in terms of speed, it suffers from somewhat higher memory requirements. Experimental results obtained with the second author's implementation of the dag for dimension $d - 1 = 2$ and 3 show that the required memory for the dag is only a small constant times the memory required by the most refined grid. In exchange the dag offers improved access efficiency. For example for $d - 1 = 3$ the selection of grids with various densities in a precomputed dag is approximately one thousand times faster than the computation of that grid, see [10].

We conclude this paper by mentioning a few experiments related to possible applications of the dag. First we illustrate the construction of grids needed in the solution cycle of the multigrid method [2]. Figure 9.1 depicts cross-sections defined by initial percentages of the set of points whose Delaunay complex is shown in the first of the six pictures. Second, we demonstrate the general efficiency of the dag achieved through hierarchical organization. Figure 9.2 shows results of an iso-surface visualization tool that allows the user to change the iso-value and get feedback in terms of an updated picture in real time.

Third, we examine the use of the dag in adapting grids to local density requirements. The need for dynamic grid adaptation arises in finite element analysis and in fly-through scenarios. We illustrate the gradual change in density enforced by the combinatorial constraint of keeping cross-sections $(d-1)$-connected. Redundancy in explaining the experiment is avoided by focusing on the 2-dimensional case. Choose a large number of points uniformly distributed in the unit-square $[0,1]^2$. From the history dag of these points choose the cross-section that contains *all* points in the central square, defined as $Q_0 = [\frac{1}{4}, \frac{3}{4}]^2$, and as few points outside the central square as possible. Figure 9.3 (a) plots the fraction of points in a narrow square annulus that moves out from the boundary of Q_0 to the boundary of $[0,1]^2$. Specifically, define $Q_x = [\frac{1}{4} - xa, \frac{3}{4} + xa]^2$, where $x \geq 0$ and a^2 is the area of Q_0 divided by the number of points in Q_0. Partition $[0,1]^2$ into the center square Q_0 and a sequence of square annuli $Q_{x+\varepsilon} - Q_x$. The graph in Figure 9.3 (a) shows the fraction of the points in $Q_{x+\varepsilon} - Q_x$ that are used in the cross-section. Both in 2 and in 3 dimensions the fraction quickly decreases with growing distance from the center.

References

[1] M. Bern and D. Eppstein. Mesh generation and optimal triangulations. In *Computing in Euclidean Geometry, vol. 1*, D.-Z. Du and F. K. Hwang (eds.), World Scientific, Singapore, 1992, 23–90.

[2] W. L. Briggs. *A Multigrid Tutorial.* Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1987.

[3] H. Edelsbrunner and N. R. Shah. Incremental topological flipping works for regular triangulations. *Algorithmica* **15** (1996), 223–241.

[4] L. J. Guibas, D. E. Knuth and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica* **7** (1992), 381–413.

[5] K. Mulmuley. Randomized multidimensional search trees: lazy balancing and dynamic shuffling. *In* "Proc. 32nd Ann. IEEE Sympos. Found. Comput. Sci., 1991", 180–196.

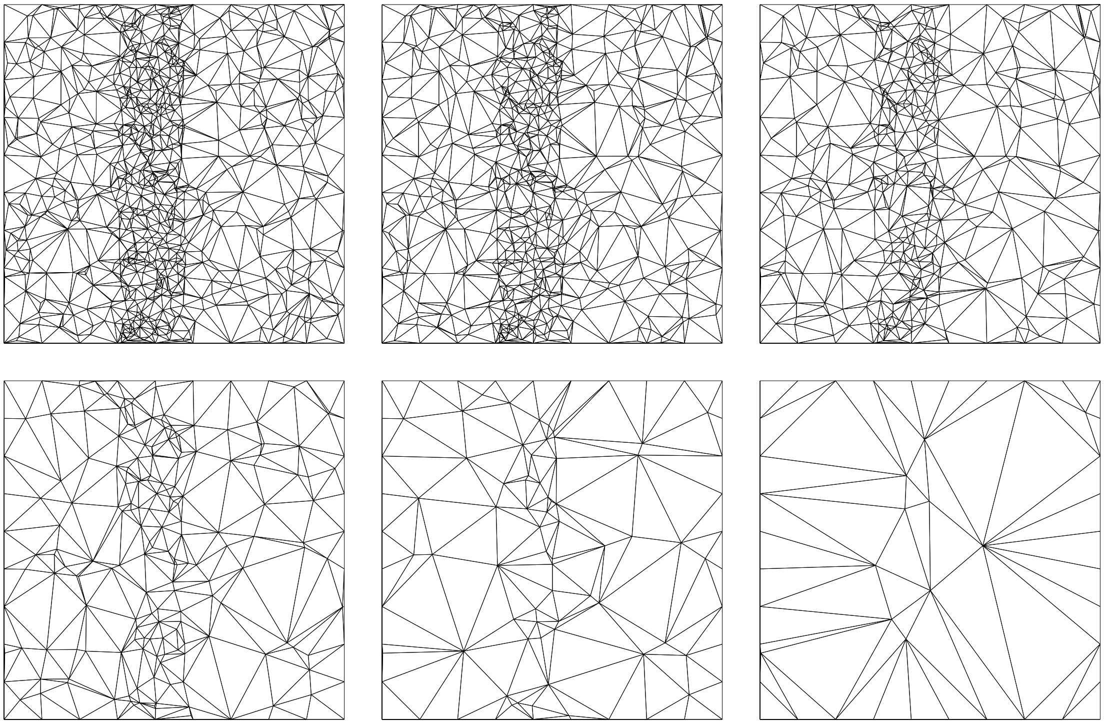

Figure 9.1: Cross-sections defined by initial 100%, 75%, 50%, 25%, 12%, 6% of the points used for the finest grid in the first picture. Each cross-section projects to the Delaunay complex of its points.

[6] J. R. MUNKRES. *Elements of Algebraic Topology.* Addison-Wesley, Redwood City, California, 1984.

[7] A. A. G. REQUICHA. Representations of solid objects – theory, methods, and systems. *ACM Computing Surveys* **12** (1980), 437–464.

[8] B. SONI. Grid quality control in CFD. Paper presented at the 3rd SIAM Conf. Geometric Design, Tempe, Arizona, 1993.

[9] R. E. TARJAN. *Data Structures and Network Algorithms.* Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1983.

[10] R. WAUPOTITSCH. *Simplifying and Deforming Hierarchies of Simplicial Grids.* Ph.D. Thesis, Tech. Report 1559, Dept. Comput. Sci., Univ. Illinois, Urbana, 1996.

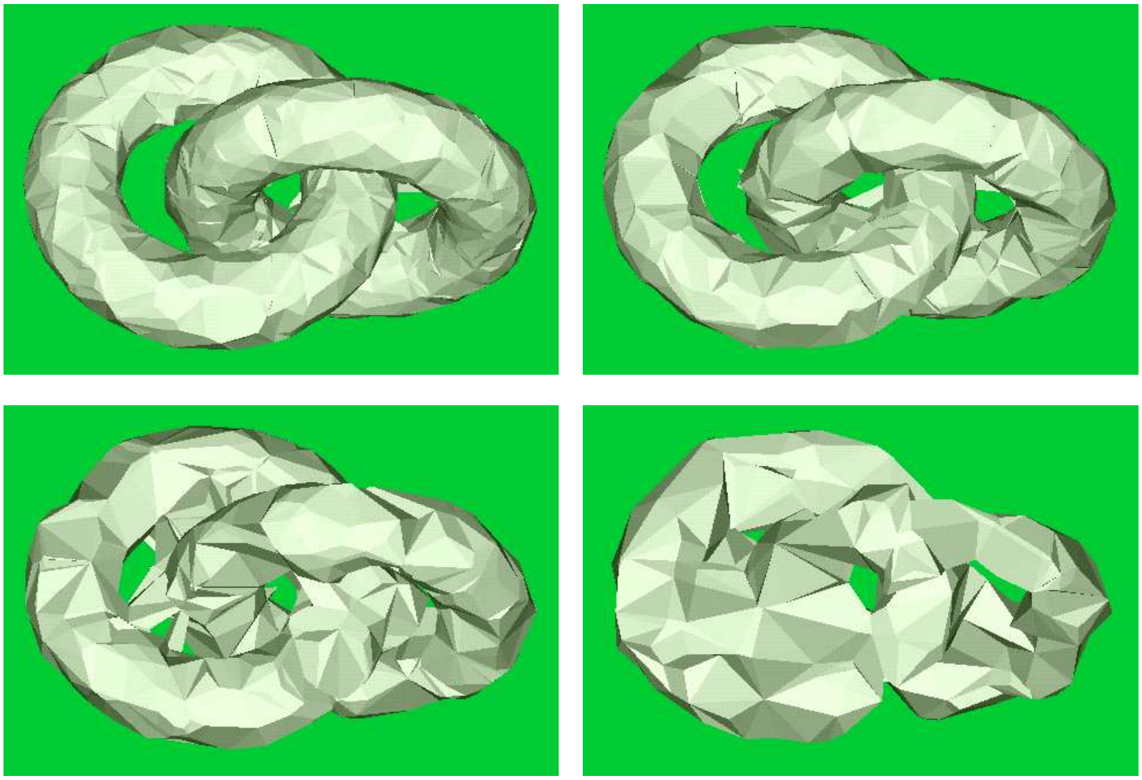

Figure 9.2: Iso-surfaces constructed from 3-dimensional grids obtained for the initial 100%, 50%, 25%, 10% of the points used for the finest grid in the first picture. Each grid is the projection of a cross-section in the 4-dimensional history dag.

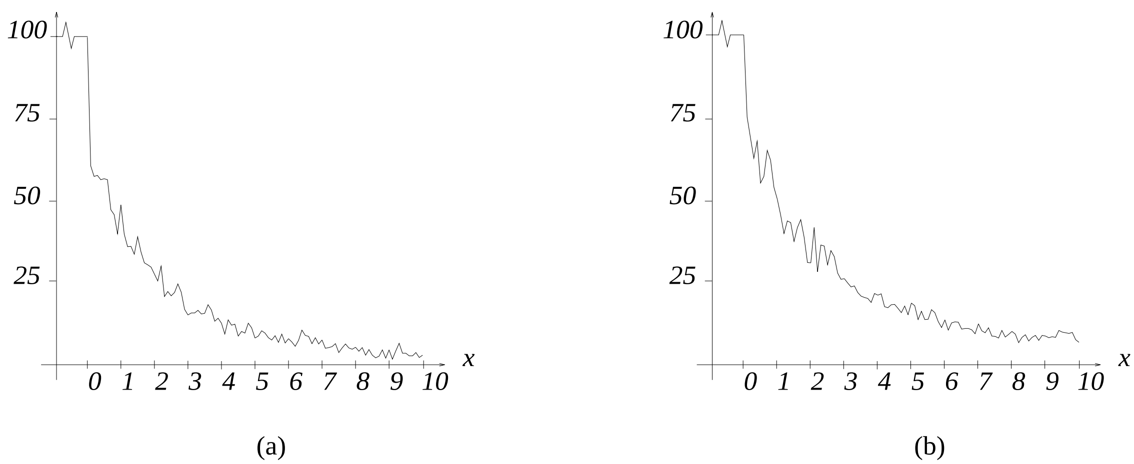

Figure 9.3: The decay of the function relates to the steepness of the ancestor complex defined by the points near the center of $[0,1]^{d-1}$. We have $d-1=2$ in (a) and $d-1=3$ in (b). By comparing the functions we see that the decay is faster in two than in three dimensions.