Representing Topological Structures Using Cell-Chains⋆

David E. Cardoze[1], Gary L. Miller[2], and Todd Phillips[2]

[1] Tanner Research
cardoze@tanner.com
[2] Computer Science Department
Carnegie Mellon University
glmiller@cs.cmu.edu   tp517@cs.cmu.edu

Abstract. A new topological representation of surfaces in higher dimensions, "cell-chains" is developed. The representation is a generalization of Brisson's cell-tuple data structure. Cell-chains are identical to cell-tuples when there are no degeneracies: cells or simplices with identified vertices. The proof of correctness is based on axioms true for maps, such as those in Brisson's cell-tuple representation. A critical new condition (axiom) is added to those of Lienhardt's n-G-maps to give "cell-maps". We show that cell-maps and cell-chains characterize the same topological representations.

Keywords: computational topology, cell complex, cell tuple, cell chain

1 Introduction

The ability to represent and manipulate topological structures in a computer is central to many areas, such as the finite element method in scientific computation, computational geometry, computer graphics, solid modeling, and scientific visualization. In general, a user has a particular decomposition of a domain into a collection of topological objects and needs a data structure for representing the connectivity information between various simple objects. We present a new data structure that addresses this task more generally than previous data structures, while providing a strong theoretical characterization of those objects that can be represented. This problem has been cited as an important open area of research in computational topology [BEA+99].

A classic example of this task occurs in meshing, wherein a complex geometric objects is decomposed into simple atomic objects. In two dimensions, this is the equivalent to the representation of a graph embedded on a surface. The embedding partitions the surface up into three types of *cells*: vertices, edges, and faces; called cells in general. Another very important example are CAD systems. Here the objects or cells may be much more complicated then simply a collection of simplices or d-dimensional cells, [Far99].

The goal of the topological data structure is to maintain the cells and incidence relationship between these cells in such a way that topological and other information can be stored and retrieved correctly and efficiently. If the full dimensional cells, say d-dimensional cells, are simplices, then one can, in principle, just enumerate the shared $d - 1$-simplices. In general for many applications the d-cells will not be simplices. As an example, for spectral or high-order methods, cube-like elements are used because sparse methods for representing the stiffness matrix are known [DFM02]. Other applications may use a mesh with a mix of element types since, in general, it may be easier to have a mesh with mostly cube elements and only a few tetrahedral elements.

For d-cells that are finite, one can decompose the cells into simplices using the barycentric subdivision. In two dimensions, most, if not all, proposed topological data structures represent the barycentric subdivision, either implicitly or explicitly.

The most popular data structure is the cell-tuple of Brisson [Bri89,Bri93]. In this structure, topological information is stored via **cell-tuples**, which are maximal paths in the **incidence graph** or **incidence poset**
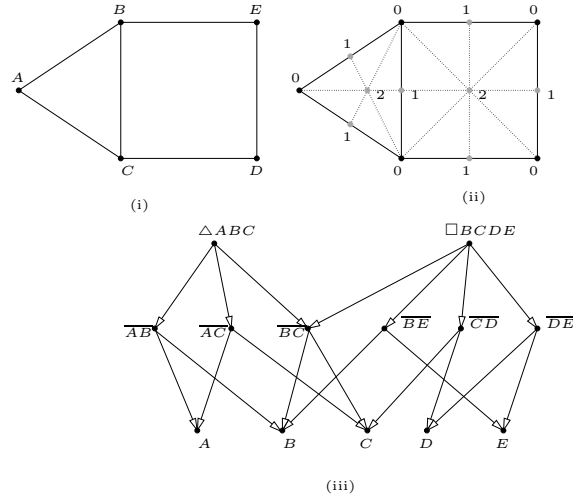
Fig. 1. (i) A simple cell complex made from a square adjacent to a triangle. (ii) The barycentric subdivision of this complex: The barycenter of each cell, vertex, edge, or face, is inserted and labeled with the dimension of the cell. The result is triangulated to form numbered simplices. (iii) The incidence graph (poset) for this complex. The vertices are cells edges connect neighboring cells whose dimension differs by 1.

[Ede87], see Figure 1. This data structure yields extremely efficient and elegant implementations for many operations on topological arrangements, and it is widely used in practice [Hal97].

The two main limitations of Brisson's cell-tuple is that it can only represent a very regular class of structures and that the test for membership in this class is undecidable in higher dimensions. To address the first limitation, we propose an extension to cell-tuples that allows for representation of an **incidence multigraph**, thus allowing various types of degeneracies. We operate using **cell-chains** (Section 4), which are maximal paths in the multigraph. In Section 8 we discuss **clusters** of cell-chains that correspond to a single cell-tuple. We show that in two-dimensions, clusters can be of size at most four, whereas clusters can be arbitrarily large in three or higher dimensions. We discuss implementation details for cell-chains in Section 7.

The ability of the data structure to handle degeneracies is very useful. Several authors have decomposed 2D surfaces up into highly degenerate pieces for efficient processing purposes [Hop96]. We also have used these degeneracies in a biological application to specify cell and artery boundaries [CCM+04] succinctly. The alternate approach for dealing with degeneracies is to refine the cell decomposition to remove these degeneracies. If the software cannot handle these degeneracies then they must be removed manually.

The second limitation of Brisson's cell complexes is that he requires that the cell-tuple representation comes from a manifold, a space locally homeomorphic to $\mathbb{R}^d$, for a d-dimensional structure. But determining if a topological structure is locally homeomorphic to $\mathbb{R}^d$ is undecidable for $d \geq 6$ [Mar58,VKF74], it is open for $d = 5$, and only known to be in NP for $d = 4$ [Sch]. To address the undecidability problem inherent in Brisson's cell complexes, we do not require our representation be locally homeomorphic to $\mathbb{R}^d$. We only require cell complex to satisfy testable conditions that are also known to be true in Brisson's case. These conditions are true for many degenerate structures also. The combinatorial axioms we give can be verified in $O(nd)$ where n is input size and d is dimension. Our axiomatic approach follows those developed for algebraic representations.

One of the early algebraic representations of graph connectivity on surfaces was given by Edmonds and Tutte [Edm60,Tut73]. Other examples of this type of representation include *maps* [Tut84,Vin83a,Vin83b,BS85], *n-G-maps* [Lie91,Lie94,LM], and the *quad-edge* structure [GS85]. Another similar algebraic structure is Tits notion of *buildings* [Tit81].

The benefit of algebraic representations is that they have axiomatic definitions for the combinatorial structure of the topological space that is being represented. The vast majority of algebraic representations (some in hindsight) use permutation groups that act on the simplices of the **barycentric subdivision** or **barycentric complex** (see Figure 1 as well as Section 2.) The barycentric complex is constructed bottom-up by "gluing" together numbered simplices. Axiomatic gluing rules are given so that the complex will be well-formed (in terms of whatever algebraic structure is involved.) Throughout the paper we will call a set of gluing rules a **map**. The most popular data structures of this form are n-G-maps, due to Lienhardt [Lie91,Lie94], which use a overly loose set of axioms to construct a very large class of topological structures.

We propose an additional axiom, to those of Lienhardt, the **Orthogonality Axiom** (Section 3), that restricts the possible combinatorial structures, but still allows a rich class of topological structures.

The central result of this work is Theorem 3 in Section 4, wherein we show that our cell-chain extension to Brisson's cell-tuple is precisely equivalent to our axiomatic restriction of Lienhardt's n-G-maps. This middle ground allows for an intuitive data structure that still represents a large group of topologies, thus combining the benefits of both major approaches to topological data structures. A two-dimensional implementation of cell-chains has been implemented within the TUMBLE software package. [Tum].

A plethora of similar data structures and representations exist in the literature. Cell-Chains, n-G-maps, and cell-tuples have much in common with other topological data structures [Bau72,Män88,GS85,Wei85] and [Wei86,DL87,RO89,Ros97]. An excellent review of existing work can be found in [LLLV05].

The rest of this paper is organized as follows. In Sections 2 and 3 respectively, we discuss the concepts of maps from a geometrical and combinatorial point of view. In Section 4 we introduce simplex-chains, cell-chains and prove our main result. In Section 5 we give a formal presentation of cells and show that they are well-defined for our cell-maps. In Section 6 we show that all of our axiom are true for a CW-complex thus justifying our axioms. In Section 7 we discuss some preliminary implementation issues. Finally in the last section we delineate the difference between the complexes we can represent and those of Brisson's cell-tuples.

2 Barycentric Complexes and Maps

In this section we discuss the concept of maps from a geometric point of view. This information should give enough background to introduce a set of axioms and to work in the more formal setting in Section 3. We start by defining some topological notions. For more background on topology refer to [RF90,Kin93,LW69,Moi77,Jan84].

The goal of a topological map is to partition a topological space up into regions that are homeomorphic to open balls. More formally, for $d \geq 1$ define

$$\mathbb{B}^d = \{x \in \mathbb{R}^d : |x| < 1\},$$
$$\overline{\mathbb{B}^d} = \{x \in \mathbb{R}^d : |x| \leq 1\},$$
$$\mathbb{S}^d = \{x \in \mathbb{R}^{d+1} : |x| = 1\}.$$

A space homeomorphic to $\mathbb{B}^d$ is called an **open d-cell**, a space homeomorphic to $\overline{\mathbb{B}^d}$ is called a **closed d-cell**, and a space homeomorphic to $\mathbb{S}^d$ is called an **d-sphere**. By convention we say that single points are both open and closed 0-cells.

A partition of a space into open cells is called a CW-complex. Recall that a d-dimensional **(finite, normal, homogeneous) CW-complex** is a pair (X, C) where X is a Hausdorff space and C is a finite partition of X into open cells such that for every open d-cell, c in C, there is a continuous map $h_c : \overline{\mathbb{B}^d} \to X$ whose restriction to $\mathbb{B}^d$ is a homeomorphism onto c and whose restriction to $\mathbb{S}^{d-1}$ is the union of open cells in C of dimension less than d. In addition it is required that every open cell is either a d-cell or is on the boundary of a d-cell. A **sub-complex** of (X, C) is a CW-complex (Y, D) where $Y \subseteq X$ and $D \subseteq C$. We will usually refer to a CW-complex by its underlying space X alone and do not mention the partition C. A CW-complex is said to be a **pseudo-manifold** if every $(d-1)$-face is adjacent to at most two d-cells and every d-cell can be reached from every other d-cell traveling along $(d-1)$-faces.

Since we are interested in d-dimensional CW-complexes that represent generalized barycentric subdivisions, we add some additional constraints. The idea in representing a CW-complex is to further partition
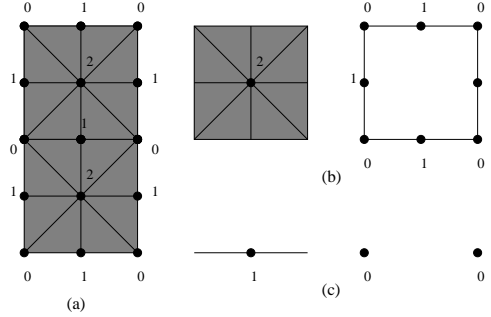
Fig. 2. (a) a barycentric complex with two 2-cells(faces), seven 1-cells(edges) and six 0-cells(vertices). (b) a 2-cell (left) and its boundary (right). (c) a 1-cell (left) and its boundary (right).

each i-cell into i-simplices. We do this by adding a new point interior to each cell. Each point is labeled by the dimension of the cell containing it. Using these points we now form d-simplices out of these points [Bri93] that partition up the d-cells. A **barycentric complex** X is a CW-complex where the sub-complex of every closed cell in X is a combinatorial simplex, and every vertex v in X is assigned a number $\nu(v)$ between 0 and d in such a way that no two vertices on the boundary of any given open cell have the same label, see Figure 2.

Since the closed cells of a barycentric complex are combinatorial simplices, by abuse of the notation we will refer to open cells as open simplices, and closed cells as closed simplices. Sometimes we will drop the open or closed adjective when the meaning is clear from the context or is irrelevant. A vertex whose label is i is said to be an i-vertex.

Instead of taking a manifold or surface and decomposing it into cells, we take numbered d-simplices and prescribe rules for gluing them into surfaces. In particular, a **map** is a set of numbered simplices and gluing rules that hopefully represent a connected barycentric complex that results from gluing numbered closed d-simplices along $(d-1)$-faces whose vertices have the same labels as prescribed by the gluing rules. The gluings must be induced by cell preserving homeomorphisms that respect the labeling. Every $(d-1)$-face can be involved in at most one gluing.

Throughout this paper we will consider only one type of gluing, namely, we will only glue or identify simplices of dimension $d-1$ and only if the set of vertex numbers are the same. Thus if σ_1 and σ_2 are two numbered d-simplices containing, respectively, $d-1$-simplices σ_1' and σ_2'. To identify σ_1' and σ_2' it must be the case that the vertex labels agree and that they contain all the labels from $0 \ldots d$ except one number. If we identify them, then we will identify all simplices in σ_1' with all simplices in σ_2' containing the same vertex numbers.

Hence a map X can be represented by a tuple $(S, \alpha_0, \ldots, \alpha_d)$ where S is a set of numbered closed d-simplices and the α_i's are involutions on S that describe the gluing's. If α_i has no fixed points for $0 \leq i < d$ we say that X has a **proper boundary**. The barycentric complex in Figure 2 is a map while the one in Figure 3(b) is not.

Note that there are maps that are not proper but satisfy the conditions put forward in [Lie94]. See Figure 3(a) for an example.

One possibility for an even stronger condition would be that the cell decomposition itself be a CW-complex, unfortunately, this is undecidable in higher dimension.

3 Formal Presentation of Maps

In this section we present a formal approach to combinatorial maps, as motivated in Section 2. A more formal specification is necessary for at least three reasons. Firstly, a combinatorial structure is closer to what is actually stored and manipulated internally in the machine. Secondly, the proofs for topological data
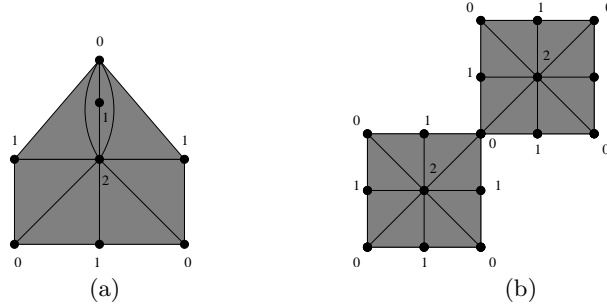
Fig. 3. (a) This map is not proper. The one-cell interior to the two-cell is not an open one-disk. (b) This barycentric complex is not a map. The vertex common to the two two-cells is a face of two triangles which cannot be reached from one another by traversing edges containing it.
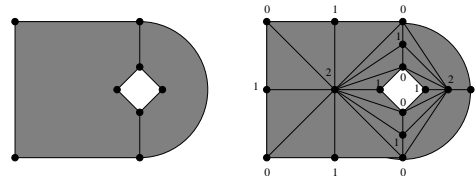
Fig. 4. A CW-complex and its barycentric decomposition.
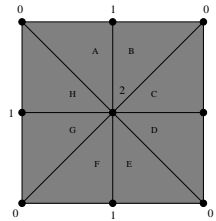
Fig. 5. If the A is glued to F and B is glued to E (both gluings by α_2), then the resulting object is a cylinder. If additionally, C is glued to H and G is glued to D (both gluings by α_2), then the resulting object is a torus.

structures are all based on this formal specification. Third, since it is undecidable to determine if the structure we will consider is a well-formed manifold, we must therefore work in a formal combinatorial model.

A k-**simplex** σ is a set of $k + 1$ atomic objects and all subsets of these objects. Each subset and all the subsets containing it is also a simplex (a subsimplex). We call an atomic object a **vertex** or **0-simplex**. Throughout this paper we deal only with numbered simplices. A **numbered** k-**simplex** σ is a k-simplex where each vertex is assigned a distinct integer label. A **consecutively numbered** k-**simplex** is a numbered simplex with numbers from the range $\{i, \ldots, i + k\}$. We call a consecutively numbered 1-simplex an **edge-simplex**. The **span** of a numbered simplex is the set of its integer labels.

Throughout this paper we consider only one type of gluing of two simplices, namely, we only glue or identify two subsimplices of dimension $d - 1$ of two distinct d-simplices and only if the set of vertex numbers are the same. That is, if σ_1 and σ_2 are two numbered d-simplices containing, respectively, $d - 1$-simplices σ_1' and σ_2' then to identify σ_1' and σ_2' it must be the case that the vertex labels agree and must contain all numbers but say i. If we identify them then we also identify all subsimplices in σ_1' with all subsimplices in σ_2' with the same vertex labels.

In the definition to follow one can think of the set Σ as n numbered d-simplices, but formally it is just a finite set of size n. By the discussion in the last paragraph to describe the $d - 1$ simplices that are to be identified it will suffice to list pairs of d-simplices and, for each pair, the vertex that will not be identified. As

has been the tradition since Tutte we do this in a group theoretic way [Tut73,Lie91]. There is yet another important way to view the gluing of the d-simplices described above. Here we think of going from σ_1 to σ_2 by reflecting σ_1 about the $d-1$-simplex common to σ_1 and σ_2. Thus the $\alpha_{i's}$ in Definition 1 are some kind of reflection and we are interested in the group generated by these "reflection".

We let $\langle a_1, \ldots, a_k \rangle$ denote the subgroup generate by the permutations $a_1, \ldots, a_k$. We say the permutation group G **acts transitively** on Σ if for every pair of elements $\sigma_1, \sigma_2 \in \Sigma$ there is a permutation $\alpha \in G$ such that $\alpha(\sigma_1) = \sigma_2$.

Definition 1. *A **map** $\mathcal{M}$ is a tuple $(\Sigma, \alpha_0, \ldots, \alpha_d)$ where Σ is a finite set of size n and each α_i is a permutation of Σ for $0 \leq i \leq d$ such that*

1. *$\alpha_i^2 = ident$ for $0 \leq i \leq d$.*
2. *α_i is fix-point-free (fpf) for $0 \leq i < d$.*

*We say that $\mathcal{M}$ is **connected** if $\langle \alpha_0, \ldots, \alpha_d \rangle$ acts transitively.*

If we view Σ as a disjoint set of numbered d-simplices and the $\alpha's$ as gluing rules as defined above then $\mathcal{M}$ gives a simplicial complex, denoted by $complex(\mathcal{M})$. One interprets the fixed points of α_d as determining the $d-1$-dimensional boundary faces of the complex.

We view the permutations $\alpha_0, \ldots, \alpha_d$ as acting on the d-simplices of $\mathcal{M}$. In general the permutations do not act in a well defined way on the i-simplices for $0 \leq i < d$. The first restriction that Lienhardt and others require is that many of the α_i commute. We shall call a map with the commutivity property a commuting-map. For complexes of commuting-maps, there will be a natural and well-defined action of permutations on simplices.

Definition 2 (Commuting-Map). *A **commuting-map** $\mathcal{M} = (\Sigma, \alpha_0, \ldots, \alpha_d)$ is a map with the further property:*

   Commutivity: *$\alpha_i \alpha_j = \alpha_j \alpha_i$ whenever $2 \leq i + 2 \leq j \leq d$.*

The commuting condition is very natural one. In 2D it just says that exactly four numbered triangles contain an edge unless the edge is on the boundary in which case there are two. It is implicit in Tutte's axioms [Tut73]. A commuting-map is what Lienhardt calls an n-bG-map where his n is our d, [Lie94]. Lienhardt replaces our commutivity condition with the condition that $\alpha_i \alpha_j$ is an involution whenever $2 \leq i + 2 \leq j \leq d$ which can easily be seen to be equivalent. He has also proposed that $\alpha_i \alpha_j$ is fpf whenever $2 \leq i + 2 \leq j \leq d$. The orthogonality condition is a stronger condition and we will show that it is necessary and sufficient to prove Theorem 2. Again, in Section 6 a formal justification will be given for these axioms.

Denote the group $\langle \alpha_j, \ldots, \alpha_i \rangle$ for $0 \leq j \leq i \leq d$ by G_j^i. Given a numbered simplex λ we denote by G_λ the subgroup $\langle \alpha_i \mid i \notin span(\lambda) \rangle$.

Definition 3 (Cell-Map). *A **cell-map** $\mathcal{M} = (\Sigma, \alpha_0, \ldots, \alpha_d)$ is a commuting-map with the further property:*

   Orthogonality: *For every $0 < i < d$, $\sigma \in \Sigma$, $\alpha \in G_0^{i-1}$, and $\beta \in G_{i+1}^d$ if $\alpha\beta(\sigma) = \sigma$ then $\alpha(\sigma) = \beta(\sigma) = \sigma$.*

Observe that for a cell-map the groups G_0^{i-1} and G_{i+1}^d commute and their intersection is the identity permutation. Thus the group generated by G_0^{i-1} and G_{i+1}^d is isomorphic to their direct product. The main goal of this section is to show that their action on the cell-chains(to be defined) is the natural direct product action.

In what follows we will first prove properties of maps. Using these properties prove properties of commuting-maps. Finally we consider properties of cell-maps. An important property held by any map is local strong connectivity.

Definition 4 (Local Strong Connectivity[3]). *A d-simplicial complex $\mathcal{C}$ is locally strongly connected or simply locally connected if whenever τ and γ are two d-simplices that share a nonempty simplex λ, then there exists a sequence $\sigma_1, \ldots, \sigma_t$ of d-simplices such that $\sigma_0 = \tau$, $\sigma_t = \gamma$, and σ_i and σ_{i+1} have a common $(d-1)$-simplex containing λ for $0 \le i < t$.*

As is well known, the simplicial complex of a map is locally connected. In fact, the simplicial complex of a map is a pseudo-manifold.

Lemma 1. *If $\mathcal{C}$ is the simplicial complex of a map $\mathcal{M}$ then $\mathcal{C}$ is locally strongly connected.*

We prove something slightly stronger. Namely, if we glue d-simplices by identifying $d-1$-faces then the complex is locally connected. The proof is by induction on the number of identifications or gluings. Initially we have n disjoint d-simplices. In this base case we are done since no two distinct d-simplices can have a nontrivial intersection. Thus we may assume that after k identification the complex is locally connected. Suppose by induction that τ and γ are distinct d-simplices that share a simplex λ after $k+1$ identifications. There are two cases depending on whether or not $\lambda \subseteq \tau \cap \gamma$ before the $k+1$ first identification. In the former case we are done since we get a sequence by induction.
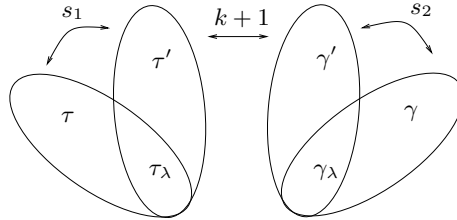
Fig. 6. Illustration of the proof of Lemma 1. s_1 and s_2 represent the inductively generated sequences. The $k+1$ gluing connects the sequence.

In this latter case, the common simplex λ, must have been formed by identifying two simplices during the $k+1$ gluing. Thus there must have been $d-1$-simplices τ_λ and γ_λ, contained in the simplices τ and γ, respectively, that were identified to form λ at the $k+1$ gluing. It is critical to the proof that a single identification must have formed this common simplex λ. Before this identification τ and γ would not contain this common simplex but after this single identification they do. Thus, it must be that case that at gluing $k+1$, τ_λ and γ_λ are identified. There must exist two d-simplices, τ' and γ' containing τ_λ and γ_λ, respectively, that are identified along a $d-1$-simplex λ' and the simplex λ' contains λ. Here we use the fact that we do not ever identify d-simplices. By induction there a sequences of d-simplices from τ to τ' and one from γ' to γ. By combining these two sequences with the $k+1$st gluing we get the prescribed sequence from τ to γ. □

Observe that the $k+1$ gluing in the proof above is obtained by applying the permutations α_i, where $i \notin G_\lambda$, such that $\alpha_i(\tau') = \gamma'$. Therefore by induction all the gluings on the path obtained from permutations in G_λ.

It is not hard to see that the sequence of d-simplices can be obtain one from another by applying permutations α_i, where $i \notin G_\lambda$, where λ is the intersection. Thus, we reformulate Lemma 1 group theoretically. This formulation of the locally connectivity lemma is very simple group theoretically and will be used throughout the paper.

Lemma 2 (Local Connectivity Lemma). *If τ and γ are d-simplices in the complex of a map that share a nonempty simplex λ then there exists a permutation $\alpha \in G_\lambda$ such that $\alpha(\tau) = \gamma$.*

---

[3] *Locally Strongly Connected* has appeared in other topological literature in a different context.

It follows by induction for a complex of a map that every d-simplex has a unique subsimplex with a given subset of labels from the set $0, \ldots, d$. Using this fact we define a natural **boundary map**. If σ is a simplex and S is a set of labels then $[\sigma]_S$ denotes the sub-simplex of σ with label in S. When there is no confusion we let $[\sigma]_i$ denote the subsimplex $[\sigma]_{\{0,\ldots,i\}}$.

We say the permutation $\alpha \in \langle \alpha_0, \ldots, \alpha_d \rangle$ **acts in a well defined way** on a simplex σ if for all d-simplices τ and γ containing σ

$$[\alpha(\gamma)]_S = [\alpha(\tau)]_S$$

where S is the set of labels of σ

Lemma 3. *If $\mathcal{C}$ is the simplicial complex of a map $\mathcal{M}$, σ is a simplex of $\mathcal{C}$, and $\alpha \in G_\sigma$ then α acts in a well defined way on σ and $\alpha(\sigma) = \sigma$.*

Proof. Suppose that σ is an i-simplex with span S. It will suffice to prove the lemma for any permutation α_j for $j \notin S$. If $i = d$ the lemma is vacuously true. Suppose that $i < d$ and let γ be any d-simplex that contains σ. There must exist a $d - 1$-simplex τ with span $S' = \{0, \ldots, d\} - \{j\}$ such that $\sigma \subseteq \tau \subset \gamma$. If $j = d$ and γ is a fixed point then so it τ. If not then there a a unique γ' such that $\alpha_j(\gamma) = \gamma'$ and γ' also contains τ. Thus in either case α_j fixes τ. That is $[\alpha_j(\gamma)]_{S'} = \tau$. This gives the following sequence of equalities:

$$[\alpha_j(\gamma)]_S = [[\alpha_j(\gamma)]_{S'}]_S = [\tau]_S = \sigma$$

$\square$

Additionally, we need one more technical lemma that will be used to prove our main theorem, Theorem 3.

Lemma 4. *If $\mathcal{C}$ is a numbered simplicial-complex of a commuting-map and γ, γ', and λ be three simplices such that $span(\gamma) = span(\gamma') \subset \{0, \ldots, i\}$, $\lambda \subseteq \gamma \cap \gamma'$, and $\{j, \ldots, i\} = span(\lambda)$ for $j \leq i$ then there exits an $\alpha_L \in G_0^{j-1}$ such that $\alpha_L(\gamma) = \gamma'$.*

Proof. In general $\alpha(\gamma)$ may not be well-defined. Here all we mean is that there exists a d-simplex $\gamma \subset \sigma$ such that $\gamma' \subset \alpha(\sigma)$. By Lemma 1 there exists an $\alpha \in G_\lambda$ such that $\alpha(\gamma) = \gamma'$. By the Commutivity condition $\alpha = \alpha_L \alpha_H$ where $\alpha_L \in G_0^{j-1}$ and $\alpha_H \in G_{i+1}^d$. Thus, $\alpha_L(\gamma) = \alpha_H^{-1}(\gamma')$. Since α_H^{-1} fixes γ' by Lemma 3 we get that $\alpha_L(\gamma) = \gamma'$. $\square$

4 Simplex-Chains and Cell-Chains

In this section we introduce the notation of simplex-chains. Simplex-chains are a natural generalization of Brisson's cell-tuples. Let $\mathcal{C}$ be a numbered simplicial complex.

Definition 5 (Simplex-Chain). *A sequence $(X_{L_0}, \gamma_1, X_{L_1}, \ldots, \gamma_k, X_{L_k})$ is a length k **simplex-chain** of $\mathcal{C}$ if:*

1. *Each γ_i is a consecutively numbered simplex of $\mathcal{C}$ of dimension $d_i \geq 1$ for $1 \leq i \leq k$.*
2. *Each X_{L_i} is 0-simplex with label L_i for $0 \leq i \leq k$*
3. *$X_{L_i} = \gamma_i \cap \gamma_{i+1}$ for $0 < i < k$.*
4. *X_{L_0} (X_{L_k}) is the 0-simplex in γ_1 (γ_k) with minimum (maximum) label.*

*A simplex-chain is an **cell-chain** if every $d_i = 1$. The **dimension** of the chain is $d_1 + \cdots + d_k$. In general we may drop some or all of the 0-simplices when they are not being considered since they are determined by the simplices containing them.*

Lemma 5. *If $\mathcal{C}$ is a numbered simplicial-complex of a commuting-map then every simplex-chain of length 2 and dimension d' is contained in a d'-simplex and thus a d-simplex.*

Proof. Let $(\lambda_L, X_i, \lambda_H)$ be a simplex-chain of length 2. There must exist d-simplices σ_L and σ_H containing λ_L and λ_H, respectively. Therefore $X_i \subset \sigma_L \cap \sigma_H$.

By the local connectivity property there exists a permutation $\alpha \in G_{X_i}$ such that $\alpha(\sigma_L) = \sigma_H$ By the commutivity property for cell-maps we know that $\alpha = \alpha_L \alpha_H$ where $\alpha_L \in G_0^{i-1}$ and $\alpha_H \in G_{i+1}^d$. Thus $\alpha_H(\sigma_L) = \alpha_L^{-1}(\sigma_H)$. We claim that $\alpha_H(\sigma_L)$ contains both λ_L and λ_H since $\alpha_H(\sigma_L)$ contains λ_L and $\alpha_L^{-1}(\sigma_H)$ contains λ_H by Lemma 3.

In a d-simplex every subset is a simplex thus there must exist a simplex of dimension that of (λ_L, λ_H) containing λ_L and λ_H. $\square$

To prove that every simplex-chain is contained in a simplex we then use induction on the length of the chain. This gives the following Theorem:

Theorem 1. *If $\mathcal{C}$ is a numbered simplicial-complex of a commuting-map and SC is a simplex-chain of dimension d' then SC is contained in a d'-simplex.*

We now show that the simplex of minimum dimension containing a chain is unique. This will require that the map is a cell-map. We start by considering chains of length two.

Lemma 6. *If $\mathcal{C}$ is a numbered simplicial-complex of a cell-map and SC is a length two simplex-chain of dimension d' then SC is contained in a unique d'-simplex.*

Proof. Let $(X_i, \gamma_1, X_j, \gamma_2, X_k)$ be a simplex-chain of dimension d' and suppose by way of a contradiction that it is contained in two distinct d'-simplices λ and λ'. Let σ be some d-simplex that contains λ. Using the fact that $\gamma_2 \subset \lambda \cap \lambda'$ and Lemma 4 there exists $\alpha_L \in G_0^{j-1}$ such that $\lambda' \subset \alpha_L(\sigma) \neq \sigma$. Letting $\sigma' = \alpha_L(\sigma)$ and using the fact that $\gamma_1 \subset \lambda \cap \lambda'$ and the dual to Lemma 4 there exists $\alpha_H \in G_{j+1}^d$ such that $\alpha_H(\sigma') = \sigma$.

We have a contradiction since $\alpha_L \alpha_H(\sigma) = \sigma$ but $\alpha_L(\sigma) \neq \sigma$. $\square$

Following by induction on the length of simplex-chains, we get the following theorem:

Theorem 2. *If $\mathcal{C}$ is a numbered simplicial-complex of a cell-map and SC is a d'-dimensional simplex-chain then SC is contained in a unique d'-simplex.*

We will use the following simple corollary:

Corollary 1. *For a complex of a cell-map there is a one-to-one correspondence between the cell-chains and d-simplices.*

We next show that the orthogonality condition is not only sufficient but necessary to get the uniqueness condition of Corollary 1.

Theorem 3. *If $\mathcal{C}$ is the simplicial complex of a commuting-map $\mathcal{M}$, then $\mathcal{M}$ is a cell-map if and only if there is a one-to-one correspondence between the cell-chains and d-simplices.*

Proof. The one-to-one correspondence is necessary by Corollary 1. We claim that it is also sufficient to make $\mathcal{M}$ a cell-map.

We must show orthogonality. To this end, let a d-simplex σ be given, and let $\alpha_L \in G_0^{j-1}$ and $\alpha_H \in G_{j+1}^d$ be given for some j, such that $\alpha_L \alpha_H(\sigma) = \sigma$.

Define $\sigma' = \alpha_H(\sigma)$. It then suffices to show that $\sigma = \sigma'$.

By assumption, we have a one-to-one correspondence between edge chains and d-simplices, so let us define X and X' as the unique edge chains associated with σ and σ' respectively. Let γ_i and γ_i' be the edges of X and X'.

Since $\sigma' = \alpha_H(\sigma)$, and since $\alpha_H \in G_{j+1}^d$, we have

$$[\sigma']_j = [\alpha_H(\sigma)]_j = [\sigma]_j$$

So that σ and σ' must have the same j-cell. Therefore $\gamma_i = \gamma_i'$ for $i < j$.

By a dual argument for the high-dimensional edges (using $\sigma = \alpha_L(\sigma')$), it holds that $\gamma_i = \gamma_i'$ for $i \geq j$. Hence $X = X'$, so by the assumed correspondence, $\sigma = \sigma'$, so orthogonality must hold. $\square$

We can use this theorem to give an algorithm for testing if a map is a cell-map. Given a map $\mathcal{M} = (\Sigma, \alpha_0, \ldots, \alpha_d)$, we can first verify that it is a commuting-map. If $|\Sigma| = n$, then this can be done naively in $O(nd^2)$, since there are $O(d^2)$ permutations that must commute for each of the n elements.

We can then use a commuting-map to build the incidence multigraph as follows. Begin with a set of n disconnected cell-chains of length d, one for each element. Then for each α_i, for each element σ, we join the two chains associated with σ and $\sigma' = \alpha_i(\sigma)$. Joining these two chains consists of identifying all of their edges except γ_i and γ_{i+1}, this makes $O(d)$ edges that must be joined for each σ and for each α_i. Naively this takes $O(nd^2)$, but a simple divide-and-conquer approach can yield $O(nd \log d)$.

Once the incidence multigraph is constructed, we need simply count the number of maximal paths (number of cell-chains). Denote this as t. This path-counting can be accomplished by simple dynamic programming.

Since we have a commuting-map, by Theorem 1, each cell-chain is contained in some σ. Since each σ can contain at most one cell-chain, it follows by counting that the cell-chains and Σ are in one-to-one correspondence iff the $t = n$. Hence by Theorem 3, we have a cell-map iff $t = n$.

5 Cells

In this section we consider cells of a numbered simplicial complex of commuting-maps and cell-maps. We consider the natural decomposition of the complex into cells, Definition 6.

The main goal of this section is to show that for cell-maps each cell, in a natural way, can also be viewed as a complex of a cell-map. This justifies the definition of a **cell-map** (Definition 3). A CW-complex is a homotopy-theoretic generalization of the notion of a simplicial complex. A CW-complex is a space X which is partitioned in a collection of cells each homeomorphic to an open ball of some dimension. Under certain conditions a CW-complex can be partitioned into a simplicial complex using a barycentric decomposition. We are interested in simplicial complexes which admit a natural decomposition into cells. We will show that cell-maps do this for us.

We first show that many of the permutation that we have defined to act on the d-simplices actually act in a well defined way on subsimplices.

In this section we give a standard formal definition of a cell in a barycentric-complex. We then show that each cell can be viewed as barycentric-complex and if the original complex comes from a cell-map then the cells also can be viewed as coming from a cell-map.

Let $\mathcal{C}$ be a numbered simplicial-complex derived from a cell-map $\mathcal{M} = (\Sigma, \alpha_0, \ldots, \alpha_d)$.

Definition 6. *Let X_i be a 0-simplex of $\mathcal{C}$ with label i. The cell or i-cell C of X_i is the set of all simplices of $\mathcal{C}$ containing X_i with labels $\leq i$.*

Except for 0-cells a cell is not simplicial-complex (They are not closed under subsimplices). There are two natural closures, namely, the closure in $\mathcal{C}$ and the free-closure. The closure in $\mathcal{C}$ is simply the smallest simplicial-complex in $\mathcal{C}$ containing C. To obtain the **free-closure** we add new simplices to C to close it under subsimplices.

We first show that the elements of G_0^{i-1} act in a well defined way on the i-simplices of an i-cell C.

Lemma 7. *If $\mathcal{C}$ is a numbered simplicial-complex of a commuting-map and τ and γ are d-simplices that contain an i-simplex λ of an i-cell, and $\alpha \in G_0^{i-1}$ then $[\alpha(\tau)]_i = [\alpha(\gamma)]_i$*

Proof. It will suffice to prove the lemma for each α_j where $0 \leq j < i$. Let τ, γ, and λ be as in the lemma hypothesis. By the Local Connectivity Lemma, Lemma 2, there exists a permutation $\alpha \in G_{i+1}^d$ such that $\alpha(\tau) = \gamma$. This gives the following sequence of equalities:

$$[\alpha_j(\gamma)]_i = [\alpha_j(\alpha(\tau))]_i = [\alpha(\alpha_j(\tau))]_i = [\alpha_j(\tau)]_i$$

□

By the last lemma we see that we can define a natural action of $\alpha \in G_0^{i-1}$ on an i-simplex λ of an i-cell C. Namely, $\alpha(\lambda)$ equals the i-simplex λ' of C that is obtained by taking any d-simplex containing λ, applying

α, and then restricting the image to an i-cell of C. The lemma says that it does not matter which d-simplex we pick.

Thus, if $\bar{C}$ is the free closure of an i-cell in $\mathcal{C}$ with Σ' being the i-simplices of $\bar{C}$ then $(\Sigma', \alpha'_0, \ldots, \alpha'_i)$ is a map where α'_i is the identity permutation and α'_j for $0 \leq j < i$ is the permutation given by Lemma 7 as described above. If we want $(\Sigma', \alpha'_0, \ldots, \alpha'_i)$ to be a map then we must insure that α'_j is fixed-point-free for $0 \leq j < i$. This requires that $(\Sigma, \alpha_0, \ldots, \alpha_d)$ be a cell-map. Its will give us the nice property that $(\Sigma', \alpha'_0, \ldots, \alpha'_i)$ is also a cell-map.

Lemma 8. *If $(\Sigma, \alpha_0, \ldots, \alpha_d)$ is a cell-map then $(\Sigma', \alpha'_0, \ldots, \alpha'_i)$ as defined above is a cell-map.*

Proof. We first will show that $(\Sigma', \alpha'_0, \ldots, \alpha'_i)$ is map. It will suffice to show that α'_j for $0 \leq j < i$ is fpf. Suppose by way of contradiction that γ is a fix point of α'_j and σ is a d-simplex containing γ. By the local connectivity lemma there exists an $\alpha \in G^d_{i+1}$ such that $\alpha\alpha_j(\sigma) = \sigma$. Therefore by the orthogonality condition $\alpha_j(\sigma) = \sigma$, a contradiction. Thus $\alpha_j(\gamma) \neq \gamma$.

It should be clear that Σ' will inherit the commutivity property. We next show that it has the orthonality property.

Suppose that $\alpha \in G^{j-1}_0$ and $\alpha' \in G^i_{j+1}$ and $\alpha\alpha'(\gamma) = \gamma$ for a i-simplex γ in the i-cell. Let σ be a d-simplex containing γ. We claim that $\alpha\alpha'$ fixes σ. For suppose that $\alpha\alpha'(\sigma) = \sigma'$. By the local connectivity lemma we know that there exists a $\alpha'' \in G^d_{i+1}$ such that $\alpha''(\sigma') = \sigma$. Thus $\alpha''\alpha\alpha'(\sigma) = \sigma$. By the orthogonality property $\alpha\alpha'(\sigma) = \sigma$. Again by the orthogonality property $\alpha(\sigma) = \alpha'(\sigma) = \sigma$. Therefore $\alpha(\gamma) = \alpha'(\gamma) = \gamma$.

Thus $(\Sigma', \alpha'_0, \ldots, \alpha'_i)$ has all the properties of a cell-map. $\square$

Thus the free-closure of our cells are also cells-maps and their permutations are just the natural ones inherited from the full cell-map. We think of the permutation as also acting on the subsimplices when it is well-defined.

6 Axiom Justification

In this section we justify our axioms of a cell-map. In particular, using the work of Brisson [Bri93] we show that our combinatorial axioms hold for the generalized barycentric subdivision of a subdivided d-manifold (also known as a finite regular CW-complex). Thus the topological class of structures able to be represented by cell-complexes can be characterized combinatorially as cell-maps.

Throughout this section we will assume the manifold and subdivision has no boundary.

Brisson's notion of a cell-tuple can be defined it in the following terms, where $\mathcal{C}$ is a numbered d-simplicial complex.

Definition 7. *A sequence $(X_0, \ldots, X_d)$ is a **cell-tuple** if*

1. *Each X_i is a vertex with label i.*
2. *Each pair X_i and X_{i+1} is contained in a edge-simplex.*

A **cell-triple** is just a consecutive sub-tuple of length three.

Since each X_i is interior to a unique cell we could as Brisson does replace each vertex with its cell.

Brisson does not actually prove the following lemma but does state it to be true, page 404, line 8, [Bri93].

Lemma 9. *Every cell-tuple of $\mathcal{C}$ is contained in a unique d-simplex of $\mathcal{C}$.*

Following this lemma for each $0 \leq i \leq d$, there is a unique fixed-point free involution that Brisson calls $switch_i$, which takes a cell-tuple $(X_0, \ldots, X_i, \ldots, X_d)$ to another cell-tuple $(X_0, \ldots, \bar{X}_i, \ldots, X_d)$ where $X_i \neq \bar{X}_i$.

The main fact we need from Brisson is Corollary 1 from [Bri93]:

Lemma 10. *The action of $switch_i$ is well-defined on the cell-triple (X_{i-1}, X_i, X_{i+1}).*

Here well defined means that the image of X_i under $switch_i$ is only dependent on X_{i-1} and X_{i+1}.

We next claim that the switch acting on the cell-tuple is a cell-map.

Lemma 11. *Let Σ be the cell-tuples of a generalized barycentric subdivision of a subdivided d-manifold and $switch_i$'s be the switch operators acting on Σ then $(\Sigma, switch_0, \ldots, switch_d)$ is a cell map.*

Proof. The switch operators are fixed-point free involutions by Lemma 9. The commutivity and orthogonality conditions follow from Lemma 10. □

7 Implementation Issues

As we stated in the introduction, the use of Brisson's cell-tuples has the nice property that the cells are explicitly presented. This is why we propose using cell-chains as defined in Section 4. The reason is that we would expect most edges (in the incidence multigraph) of a cell-chain to be determined by their vertices (cells). In particular if we label the edges between two consecutive vertices with integers starting from zero we would expect that all but a small number would be labeled with zero.

One very important feature of Brisson's representation for non-degenerate structures is that the permutation $switch_i$ is well defined and is determined solely by its action on cell triples of the form (X_{i-1}, X_i, X_{i+1}). For degenerate structures this is not the case. For cell-maps, by a simple extension of Lemma 7, the permutation α_i is well defined and determined by its action on consecutively numbered 2-simplices with labels $\{i-1, i, i+1\}$. By Theorem 3 there is a one-to-one correspondence between every cell-chain of length two and every consecutively numbered 2-simplex. Thus we have:

Lemma 12. *For cell-maps for each $0 \leq i \leq d$ the permutation α_i is well defined and determined by its action on cell-chains of the form $(X_{i-1}, \gamma_1, X_i, \gamma_2, X_{i+1})$*

Thus we need only store the action of each α_i on cell-chains $(X_{i-1}, \gamma_1, X_i, \gamma_2, X_{i+1})$. Since most cell-chains will be determined by their cells we propose to have two data structures for storing the value of length two cell-chains. One for the case when the cell-chain is determined by its cell-tuple and one for the others. In the prior case, we use a regular cell-tuples structure, and in the latter case, we use complete length two cell-chains structure. If the image of a cell-tuple is an cell-chain then we store the image in the cell-chain data structure costing two lookups.

The ability to store the action on triples instead of full chains leads to an important improvement in space efficiency for high dimensions. The number of cell-chains within a given cell is nominally $O(d!)$, but by only storing triples we see a vast decrease in complexity, allowing the switch structure to be stored in space smaller than that of the entire incidence multi-graph.

8 Two and Three Dimensional Clusters

In this section we show that 2D clusters have size at most 4. In 3D we give an example of a cell-map with unbounded size cluster. Recall a cluster is a set of d-simplices all containing the same cell-tuple. We say the cluster is nontrivial if it has size two or more.

Theorem 4. *A nontrivial cluster of a 2-dimensional simplicial complex of a cell-map has size either two or four. If the cluster has size four then the surface of the complex must be non-planar.*

Proof. : Let (X_0, X_1, X_2) be a cell-tuple of the simplicial complex. We know by Theorem 2 that there is a one-to-one correspondence between the simplices containing the cell-tuple and cell-chains containing the cell-tuple. Thus it will suffice to count the containing cell-chains. Since a 1-cell contains exactly two vertices there can be at most two edges from X_0 to X_1. By duality there are at most two edges from X_1 to X_2. Thus there are at most four cell-chains containing (X_0, X_1, X_2) and the number can either be one, two, or four.

Consider the case when there are four cell-chains containing a cell-tuple. The two edges containing (X_0, X_1) form one cycle and the two edges containing (X_1, X_2) form another one. These two cycles cross only at X_1 and in a fundamental way. Thus the surface must not be planar. $\square$

In the case of 3D clusters we need to understand the cell-chains containing a cell-tuple (X_0, X_1, X_2, X_3). Based on the discussion in the proof of Theorem 4 there can be at most two edges containing (X_0, X_1) and at most two containing (X_2, X_3). We show by an example that there can be an unbounded number of edges containing (X_1, X_2).
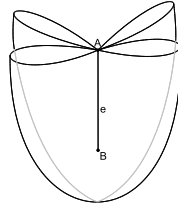
Fig. 7. The napkin example with four folds.

Consider the example of the so-called "napkin" surface in 3D. This surface is formed by taking a flat sheet and folding together several corners, then pinching the surface where the corners meet. In Figure 7, we show a napkin with four corners, though in general the number may be increased without bound. The surface in Figure 7 has two vertices (A and B), and five edges. The vertical edge is denoted e, and the four loops around the top are denoted e_1, e_2, e_3, and e_4.

The surface is shown unfolded in Figure 8(a), where we see that this surface is homeomorphic to an open disk. It's barycentric decomposition is seen in Figure 8(b). Note that there are 8 simplices defined by C, e, B, and in general the size of this cluster is twice the number of leaves chosen. The use of cell-chains allows us to distinguish within this cluster by numbering the different spokes connecting C and e in Figure 8(b).
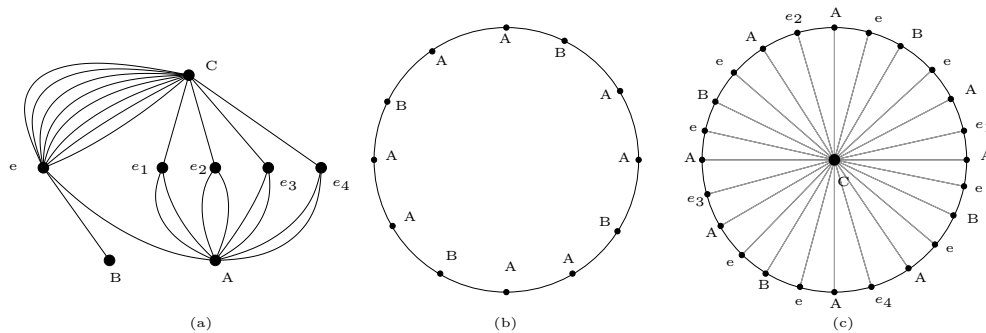
Fig. 8. (a) The incidence multigraph of the napkin. (b) The napkin surface and its boundary. (c) Barycentric decomposition of the napkin surface.

9 Cell Automorphisms and Cell-Chains

In many, if not most, applications of meshes each cell has data associated with it. In general, this data will not be invariant under automorphisms of the cell. Thus the data must be permuted or changed depending on the orientation of the cell. The simplest example is a data on a triangle. For a triangle, any permutation of the vertices is an automorphism. If the data of a triangle is initially given with respect to a fixed cell-chain, then for each of the six cell-chains we must permute the data accordingly. In practice, this is often done in a fairly ad-hoc fashion.

Throughout, let $\mathcal{C} = (\Sigma, \alpha_0, \ldots, \alpha_d)$ be a cell-map of a d-dimensional cell where α_d is the identity permutation. We let $G = \langle \alpha_0, \ldots, \alpha_d \rangle$ be the group called the group of switches and let B be the group of automorphisms of $\mathcal{C}$. Since $\mathcal{C}$ is connected G acts transitively on Σ. It is not hard to see that only the identity permutation in B has fixed points.

The interaction between these to groups is the focus of this section. Recall, that an automorphism of $\mathcal{C}$ is by definition a permutation β of Σ such that β commutes with each α_i for $0 \leq i \leq d$. That is $\alpha_i(\beta(\sigma)) = \beta(\alpha_i(\sigma))$ for any $\sigma \in \Sigma$. We begin with a few simple properties of these groups.

Lemma 13. *If $\alpha \in G$, $\sigma, \sigma' \in \Sigma$, $\beta \in B$ such that $\alpha(\sigma) = \sigma$, and $\beta(\sigma) = \sigma'$ then $\alpha(\sigma') = \sigma'$.*

Proof. By the definition of a automorphism we know that α and β commute. Thus $\alpha = \beta\alpha\beta^{-1}$. This gives $\alpha(\sigma') = \beta\alpha\beta^{-1}(\sigma') = \sigma'$ $\square$

A simple corollary to Lemma 13 is the following:

Corollary 2. *If $\mathcal{O} \subseteq \Sigma$ is an orbit of the group B of automorphisms of $\mathcal{C}$ then a permutation $\alpha \in G$ must fix all the elements of $\mathcal{O}$ or none. In the special case when $\mathcal{O} = \Sigma$ it follows that $|G| = |\Sigma| = |B|$.*

We first consider the case of cells that have a transitive automorphism group such as the case of a cell being a simplex and the case of it being a hypercube. At first hand one might think that the groups G and B are similar. In fact, the only elements they have in common is the identity.

The goal is to take a canonical simplex $\sigma_0 \in \Sigma$ and any other simplex σ and find an automorphism β such that $\beta(\sigma_0) = \sigma$. To this end suppose we have found an $\alpha \in G$ such that $\alpha(\sigma_0) = \sigma$. In particular, suppose that $\alpha = \alpha_{i_0} \cdots \alpha_{i_k}$ where $0 \leq i_t \leq d$ for $0 \leq t \leq k$.

Let β_i be an automorphism such that $\beta_i(\sigma_0) = \alpha_i(\sigma_0)$. Assuming that B acts transitively, there will always be such a β_i.

Lemma 14. $\alpha(\sigma_0) = \beta_{i_k} \cdots \beta_{i_0}(\sigma_0)$

Proof. We use the fact that the two groups G and B commute. $\square$

Thus by precomputing the β_i's we can find our needed automorphism if we can find the permutation in G taking σ_0 to σ

We next discuss two special but important cases. Namely when the d-cell is a simplex and when it is a hypercube. We start with the simplex case first.

In this case the automorphism are precisely all permutations of the 0-simplices of the d-cell. The construction will be in two parts. In the first we will describe the additional information that we need to maintain besides the switch operators. The issue is given a d-cell $\mathcal{C}$, its canonical edge chain $Chain$, and a arbitrary chain $Chain'$ find a sequence of permutation W from $\alpha_0, \ldots, \alpha_{d-1}$ such that $W(Chain) = Chain'$.

We give two types of procedures: One where no extra data is stored with each cell and a second where we store what we call a down-map. In both cases, we will find a word of length $O(d^2)$, that is of optimal length up to constant factors. In the first case we will use $O(d^3)$ switch operators to find the word while only $O(d^2)$ in the later case. These bounds will apply to both the case of a simplex cell as well as a hypercube cell even though the algorithms are slightly different.

Throughout the discussion, let $Chain = X_0, e_1, X_1, \ldots, e_d, X_d$ and $Chain' = X_0', e_1', X_1', \ldots, e_d', X_d'$. Further more, let i be the index of the first edge for which they differ starting from the largest dimension. That is, $e_i \neq e_i'$ and for $i < j \leq d$ $e_j \neq e_j'$.

The subgoal is to find a word W such that $W(Chain)$ and $Chain'$ agree through i. In the first algorithm we simply observe the we can explicitly construct at most $i + 1$ words each of length at most i such that one of these words "works". Thus there exists words $W_0, \ldots, W_i$ in $G_{i-1} = \langle \alpha_0, \ldots, \alpha_{i-1} \rangle$ each of length $i - 1$ such that $\alpha_i W_0(Chain), \ldots, \alpha_i W_i(Chain)$ is a sequence of chains containing each of the i-cells contained in the $i + 1$-cell of $Chain$. Applying this idea recursively we get the bounds as stated. The fact that the words W_j exist will follow from the construction below.

To get the more efficient algorithm we will add a new data structure to store more information about the complex. In the case of a cell that is a simplex we use the fact that simplices are closed under intersection.

That is given an two edge-chains X_i, e, X_{i+1} and X_i', e', X_{i+1}. there exist a unique edge-simplices e'' and e''' and vertex-simplex X_{i-1} such that $\alpha_i(X_{i-1}, e'', X_i, e, X_{i+1}) = X_{i-1}, e''', X_i', e', X_{i+1}$

Thus we shall store the pair X_{i-1}, e'' for every 5-tuple $X_i, e, X_{i+1}, e', X_i'$. We call the **down-map**. This should require about as much space the storing the switch operators.

The algorithm is straightforward. In order to find the word W such that $Chain$ and $Chain'$ agree up to e_i we compute the down-map e_i'' for the tuple $X_i, e_i, X_{i+1}, e_i', X_i'$ and recursively find a word W' such the $W'(Chain)$ contains the edge e''. Thus in $O(i)$ work we have found a chain which agree with $Chain'$ up to the ith edge.

For the case of the hypercube the down-map may not exists. In which case we will simply store NULL. In this case either the down-map exists or the down-map of $\alpha_i(Chain)$ and $Chain'$ exists. Thus the only change from the simplex case is when the down-map is NULL. In which case we apple α_i where e_i is the first edge where the chains differ and do a down-map lookup.

References

[Bau72]   B. Baumgardt. Winged-edge polyhedron representation. Technical Report CS-320, Stanford University, 1972.

[BEA+99] Marshall Bern, David Eppstein, Pankaj K Agarwal, Nina Amenta, Paul Chew, Tamal Dey, David P Dobkin, Herbert Edelsbrunner, Cindy Grimm, Leonidas J Guibas, John Harer, Joel Hass, Andrew Hicks, Carroll K Johnson, Gilad Lerman, David Letscher, Paul Plassmann, Eric Sedgwick, Jack Snoeyink, Jeff Weeks, Chee Yap, and Denis Zorin. Emerging challenges in computational topology, 1999.

[Bri89]   E. Brisson. Representing geometric structures in d dimensions: Topology and order. In *Symposium on Computational Geometry*, pages 218–227, 1989.

[Bri93]   E. Brisson. Representing geometric structures in d dimension: Topology and order. *Discrete and Computational Geometry*, 9:387–426, 1993.

[BS85]   R. Bryant and D. Singerman. Foundations of the theory of maps on surfaces with boundaries. *Quarterly Journal of Mathematics*, 2(36):17–41, 1985.

[CCM+04] D. Cardoze, A. Cunha, G. Miller, T. Phillips, and N. Walkington. A Bezier-Based Approach to Unstructured Moving Meshes. In *20th Symposium on Computational Geometry*, 2004.

[DFM02]   M.O. Deville, P.F. Fischer, and E.H. Mund. *High-Order Methods for Incompressible Fluid Flow*. Cambridge University Press, 2002.

[DL87]   D. Dobkin and M. Laszlo. Primitives for the manipulation of three-dimensional subdivisions. In *Third ACM Symosium on Computational Geometry*, pages 86–99, 1987.

[Ede87]   H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, New York, 1987.

[Edm60]   J.R. Edmonds. A combinatorial representation for polyhedral surfaces. *Notices Amer. Mac. Soc.*, 7, 1960.

[Far99]   Rida T. Farouki. Closing the gap between cad model and downstream application. *SIAM News*, 32(5), June 1999.

[GS85]   L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions an the computation of voronoi diagrams. *ACM Transactions on Graphics*, 4(2):74–123, 1985.

[Hal97]   Halperin. Arrangements. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*. CRC Press, 1997.

[Hop96]   H. Hoppe. Progressive meshes. In *ACM SIGGRAPH*, pages 99–108, 1996.

[Jan84]   K. Janich. *Topology*. Springer-Verlag, 1984.

[Kin93]   L.C. Kinsey. *Topology of Surfaces*. Springer-Verlag, 1993.

[Lie91]   P. Lienhardt. Topological models for boundary representation: a comparison with n-dimensional generalized maps. *Computer-Aided Design*, 23:59–82, 1991.

[Lie94]   P. Lienhardt. N-dimensional generalized combinatorial maps and cellular quasi-manifolds. *International Journal of Computational Geometry & Applications*, 4:275–324, 1994.

[LLLV05] Marcos Lage, Thomas Lewiner, Hlio Lopes, and Luiz Velho. Chf: A scalable topological data structure for tetrahedral meshes. In *In: Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing*, pages 349–356. IEEE Press, 2005.

[LM]   Bruno Lèvy and Jean-Laurent Mallet. Cellular modeling in arbitrary dimension using generalized maps.

[LW69]   A. Lundell and S. Weingram. *The Topology of CW Complexes*. Van Nostrand Reinhold, 1969.

[Män88]   Mäntylä. *An Introduction to Solid Modeling*. Computer Science Press, Inc., 1988.

[Mar58]   A.A. Markov. Insolubility of the problem of homeomorphy. In *Proceedings of the International Congress of Mathematics*, pages 300–306, 1958. Engish Translation by Afra Zomorodian.

[Moi77]   E. Moise. *Geometric Topology in Dimensions* 2 *and* 3. Springer-Verlag, 1977.

[RF90]   R. Piccinini R. Fritsch. *Cellular Structures in Topology*. Cambridge, 1990.

[RO89]   J. Rossignac and M. O'Conner. Sgc: A dimension independent model for pointers with internal structures and incomplete boundaries. In *IFIP/NSF Workshop on Geometric Modeling*, pages 145–180, 1989.

[Ros97]   J. Rossignac. Structured topological complexes: A featured-based api for non-manifold topologies. In *ACM Symposium on Solid Modeling and Applications*, pages 1–9, 1997.

[San]   The Sangria Project. `http://cs.cmu.edu/~sangria`. Supported by NSF ITR ACI-0086093.

[Sch]   Saul Schleimer. Sphere recognition lies in NP. `http://front.math.ucdavis.edu/math.GT/0407047`.

[Tit81]   J. Tits. A local approach to buildings. In C. Davis, B. Grümbaum, and F.A. Sherk, editors, *The Geometric Vein*, pages 519–547. Springer-Verlag, 1981.

[Tum]   The tumble software package. `http://rioja.sangria.cs.cmu.edu/`. Supported by NSF ITR ACI-0086093.

[Tut73]   W.T. Tutte. What is a map? In *New Directions in the Theory of Graphs*, pages 309–325. Acad. Press, 1973.

[Tut84]   W.T. Tutte. *Graph Theory*. Cambridge University Press, 1984.

[Vin83a]   A. Vince. Combinatorial maps. *Journal of Combinatorial Theory B*, 34:1–21, 1983.

[Vin83b]   A. Vince. Regular combinatorial maps. *Journal of Combinatorial Theory B*, 34:256–277, 1983.

[VKF74]   I.A. Volodin, V.E. Kuznetsov, and A.T. Fomenko. The problem of discriminating the standard sphere. *Russian Mathematical Surveys*, 29(5):71–172, 1974.

[Wei85]   K. Weiler. Edge-based data structures for solid modeling in curve-surface environment. *IEEE Computer Graphics and Applictions*, 5(1):21–40, 1985.

[Wei86]   K. Weiler. *Topological Structures for Geometric Modeling*. PhD thesis, Rensselaer Polytechnic Institute, Troy. N.Y., 1986.