Efficient open surface reconstruction from lexicographic optimal chains and critical bases

David Cohen-Steiner, André Lieutier, Julien Vuillamy

Efficient open surface reconstruction from lexicographic optimal chains and critical bases

David Cohen-Steiner

Université Côte d'Azur, Sophia Antipolis, France
Inria Sophia Antipolis - Mediterranée, France

André Lieutier

Dassault Systèmes Provence, Aix-en-Provence, France

Julien Vuillamy

Université Côte d'Azur, Sophia Antipolis, France
Inria Sophia Antipolis - Mediterranée, France
Dassault Systèmes Provence, Aix-en-Provence, France

Abstract

Previous works on lexicographic optimal chains have shown that they provide meaningful geometric homology representatives while being easier to compute than their l^1-norm optimal counterparts. This work presents a novel algorithm to efficiently compute lexicographic optimal chains with a given boundary in a triangulation of 3-space, by leveraging Lefschetz duality and an augmented version of the disjoint-set data structure. Furthermore, by observing that lexicographic minimization is a linear operation, we define a canonical basis of lexicographic optimal chains, called *critical basis*, and show how to compute it. In applications, the presented algorithms offer new promising ways of efficiently reconstructing open surfaces in difficult acquisition scenarios.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases simplicial homology, homology generators, surface reconstruction, Delaunay triangulation

Related Version Full version available at `https://hal.archives-ouvertes.fr/hal-03456390`

1 Introduction

1.1 Localization problems

Computing meaningful homology generators has attracted significant interest due to its numerous applications related to shape analysis, computer graphics, computer-aided design, or topological data analysis.

The most studied setting for homology generators associates a weight to each k-simplex and minimizes the l^1-norm over chains, corresponding to the weighted sum of the simplex coefficients in the chain. In particular, we are often interested in finding the optimal k-chain homologous to a given chain or computing the optimal k-chain bounded by a given $(k-1)$-boundary. Whether these problems are tractable depends mainly on the coefficients used. For instance, for integer modulo 2 coefficients ($\mathbb{Z}_2$), both problems are known to be NP-hard for $k > 1$ and even hard to approximate within any constant factor [4, 1]. For integer coefficients, finding optimal cycles in their homology classes can be shown to be tractable in favorable cases, for example when the boundary matrix can be shown to be unimodular, meaning that the solution of the linear relaxation problem has integer coefficients [7].

A few other positive complexity results can be found in particular settings [12, 2, 9]. Most notably, for a d-pseudomanifold, finding the optimal $(d-1)$-cycle in its homology class is equivalent to a minimum-cut problem on the dual graph of the complex [16, 3, 8].

1.2 Lexicographic optimality

Lexicographic optimality offers an alternative way of defining homology representatives, where the order between chains is induced by a total order on simplices. The advantages of this approach are two-fold. Firstly, both previously stated problems are polynomial-time tractable for any coefficient field. Algorithms solving these problems have strong connections with the matrix reduction algorithms used to compute persistent homology and have cubic complexity in the size of the simplicial complex [6]. Secondly, when the total order on simplices is well-chosen, lexicographic optimal chains give geometrically meaningful homology representatives. For instance, Delaunay triangulations, and more generally regular triangulations, can be characterized as lexicographic optimal chains in the full complex built on the vertex set [5].

The particular case of lexicographic optimal cycles in codimension 1 of a pseudomanifold can be computed more efficiently. Whereas the l^1-norm setting had a $\mathcal{O}(n^2)$ time complexity [14], the lexicographic problem is equivalent to a *lexicographic* cut on the dual graph and can be solved in $\mathcal{O}(n \log n)$ time complexity. However, no efficient algorithm was known to compute the lexicographic optimal chain with a given boundary in this favorable setting.

1.3 Contributions

The first contribution of this work is to fill the missing gap by providing an efficient algorithmic solution to compute lexicographic optimal chains from given boundaries in the Delaunay triangulation. In particular, by leveraging Lefschetz duality at the chain level as well as defining an augmentation of the classic disjoint-set data structure, we give an $\mathcal{O}(n \log n)$ algorithm for computing the optimal homologous relative $(d-1)$-chain in a d-pseudomanifold. These results are detailed in Section 3. Note that, aside from its use in this particular context, we believe the described data structure is of independent interest and could be a useful tool in different applications.

We then explore in Section 4 additional properties of lexicographic optimal chains, which we use to define canonical bases for homology generators, called *critical bases*. We then relate the computation of such bases to standard matrix reduction algorithms.

Finally, in Section 5, both results are illustrated in the context of open surface reconstruction. We show that the algorithms developed in this work, both for computing optimal chains with a given boundary and for constructing critical bases, offer interesting possibilities for triangulating open surfaces or even stratified objects in difficult scenarios.

2 Lexicographic optimality on simplicial chains

2.1 Definitions

We expect the reader to be familiar with simplicial homology. We recall in Appendix A all notions required in the context of this work. In all that follows, K will denote a simplicial complex of dimension d. The set of k-simplices of K is denoted by $K^{(k)}$. For an arbitrary field $\mathbb{F}$, $\mathbf{C}_k(K;\mathbb{F})$, or simply $\mathbf{C}_k(K)$, denotes the vector space of absolute k-chains on K over coefficients in $\mathbb{F}$. $\mathbf{Z}_k(K)$ and $\mathbf{B}_k(K)$ denote the vector space of k-cycles and k-boundaries of K. The support $|\Gamma|$ of a chain Γ corresponds to the set of simplices with non-zero coefficients.

We assume a total order on k-simplices is given, which will be denoted by $\leq$. This order allows to derive the following total preorder on k-chains.

▶ **Definition 1** (Lexicographic total preorder). *Given* $\Gamma_1, \Gamma_2 \in \mathbf{C}_k(K)$,

$$\Gamma_1 \sqsubseteq_{lex} \Gamma_2 \underset{def.}{\iff} \begin{cases} |\Gamma_1| = |\Gamma_2| \\ \text{or} \\ \max\left\{\sigma \in |\Gamma_1| \triangle |\Gamma_2|\right\} \in |\Gamma_2| \end{cases}$$

where $\triangle$ *denotes the set symmetric difference.*

A corresponding strict order $\sqsubset_{lex}$ can be defined:

$$\Gamma_1 \sqsubset_{lex} \Gamma_2 \underset{\text{def.}}{\iff} \Big(\Gamma_1 \sqsubseteq_{lex} \Gamma_2\Big) \quad \text{and} \quad \neg\Big(\Gamma_2 \sqsubseteq_{lex} \Gamma_1\Big)$$

▶ **Observation 2.** *Consider* $\Gamma_1, \Gamma_2 \in \mathbf{C}_k(K; \mathbb{F})$.

$$\left(\Gamma_1 \sqsubseteq_{lex} \Gamma_2 \quad \textit{and} \quad \Gamma_2 \sqsubseteq_{lex} \Gamma_1\right) \iff |\Gamma_1| = |\Gamma_2|$$

When $\mathbb{F} = \mathbb{Z}_2$*, the preorder* $\sqsubseteq_{lex}$ *is a total order.*

All previous definitions extend naturally to relative chains for a simplicial pair (K, B).

2.2 Properties of lexicographic optimality

We define the following application $M_{lex}^{\mathbf{B}_k(K)}$, associating to a k-chain its lexicographic minimal representative.

▶ **Definition 3** (Application M_{lex}: ***lex***icographic ***M***inimal representative).

$$\begin{aligned} M_{lex}^{\mathbf{B}_k(K)} : \mathbf{C}_k(K) &\rightarrow \mathbf{C}_k(K) \\ \Gamma &\mapsto \min_{\sqsubseteq_{lex}} \Gamma + \mathbf{B}_k(K) \end{aligned}$$

We omit for simplicity the exponent of $M_{lex}^{\mathbf{B}_k(K)}$. The map M_{lex} is well defined since one has:

▶ **Corollary 4** (see Lemma C.4). *Given* $\Gamma \in \mathbf{C}_k(K)$*, the minimum* $M_{lex}(\Gamma)$ *under the total preorder* $\sqsubseteq_{lex}$ *in Definition 3 exists and is unique.*

A major insight for solving lexicographic optimality problems lies in its linearity.

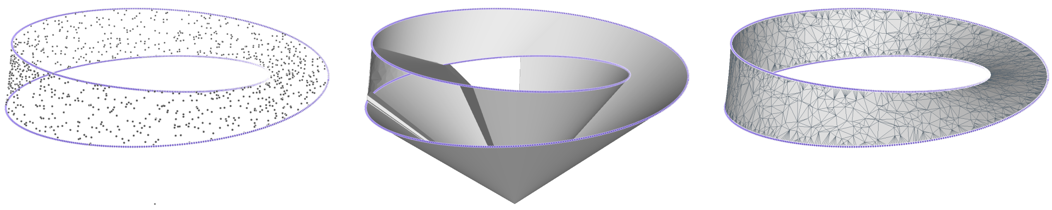
▶ **Corollary 5** (see Lemma C.5). *The application* M_{lex} *is linear. In other words, for* $\Gamma_1, \Gamma_2 \in \mathbf{C}_k(K)$ *and* $\lambda \in \mathbb{F}$,

$$M_{lex}(\Gamma_1 + \lambda\Gamma_2) = M_{lex}(\Gamma_1) + \lambda M_{lex}(\Gamma_2)$$

Similar construction for relative chains leads to the same properties of existence, uniqueness, and linearity. Please see Appendix C for details and proofs of these properties.

3 Lexicographic optimal homologous relative chain in codimension 1

A practical motivation of this work is the design of an efficient algorithm for open surface reconstruction in codimension 1. Indeed, previous work [6] showed that the following problem of finding a lexicographic optimal chain under imposed boundary could be solved in polynomial time for a generic simplicial complex ($\mathcal{O}(n^3)$ for a complex of size n).



(a) Inputs **(b)** Representative chain **(c)** Optimal homologous chain

Figure 1 Illustration of the open surface reconstruction steps. From a set of points and a 1-cycle in the Delaunay 3-complex (a), a representative 2-chain bounding this 1-cycle is computed (b). The lexicographic optimal 2-chain homologous to this representative chain is then computed (c).

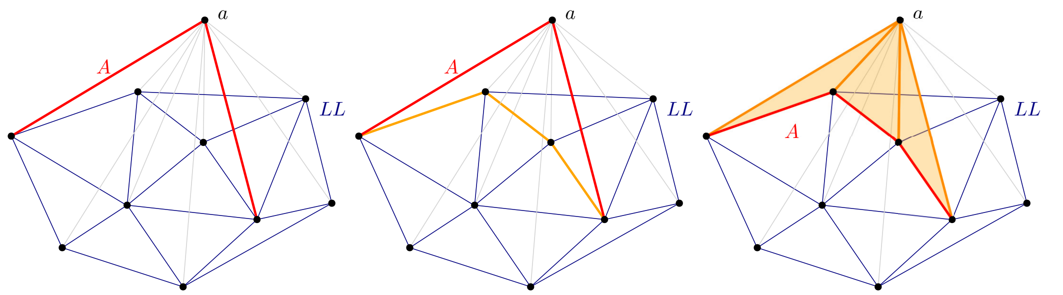


Figure 2 Left: the lower link LL of a with the two adjacent edges of a in A. Center: a path in LL (in orange) between the two adjacent vertices of a in A. Right: triangles created as joins of a. The path in LL between the two adjacent vertices replaces the edges adjacent to a in A (in red).

▶ **Problem 1.** *Given a simplicial complex K with a total order on its k-simplices and a $(k-1)$-boundary A, find the lexicographic optimal chain $\Gamma_{\min}$ bounded by A:*

$$\Gamma_{\min} = \min_{\sqsubseteq_{lex}} \left\{ \Gamma \in \mathbf{C}_k(K) \mid \partial_k \Gamma = A \right\}$$

For the particular case of chains in codimension 1 of a pseudo-manifold, we derive the following two-step algorithm, illustrated in Figure 1. Given a $(d-2)$-boundary A, our approach consists of:

1. constructing a representative $(d-1)$-chain Γ_0 that verifies $\partial\Gamma_0 = A$;
2. finding the lexicographic optimal $(d-1)$-chain homologous to Γ_0.

Note that this differs in general from Problem 1, as being homologous implies having the same boundary but the converse is not always true. The two problems are however equivalent when the complex has trivial homology.

The first part of the approach, finding a representative chain under imposed boundary, can be solved in $\mathcal{O}(n \log n)$ time complexity for the Delaunay complex of size n. The main idea of this algorithm, illustrated in Figure 2, is rather intuitive: roughly speaking, given an arbitrary direction in the Delaunay complex, the algorithm recursively replaces the highest point of the initially given 1-boundary A by a set of edges in the *lower link* of this vertex. By recording the 2-simplices generated as joins of this highest vertex with each added edge of the lower link, we can construct a 2-chain Γ_0 which verifies $\partial\Gamma_0 = A$. For completeness, the proof and complexity of this algorithm is given in Appendix E.

This section focuses instead on solving the problem of finding the lexicographic optimal relative chain homologous to a given chain $\Gamma_0 \in \mathbf{C}_{d-1}(K, B)$.

▶ **Problem 2** (Lex-OHRCP). *Given a simplicial pair* (K, B)*, and a* $(d-1)$*-chain* $\Gamma_0 \in \mathbf{C}_{d-1}(K, B)$*, find the lexicographic optimal chain* $\Gamma_{\min}$ *homologous to* Γ_0*:*

$$\Gamma_{\min} \underset{def.}{=} M_{lex}^{\mathbf{B}_{d-1}(K,B)}(\Gamma_0) = \min_{\sqsubseteq_{lex}} \left\{ \Gamma_0 + b,\ b \in \mathbf{B}_{d-1}(K, B) \right\}$$

3.1 Dual problem formulation

We start by deriving a dual formulation of Problem 2. For sake of brevity, we only develop the notions of duality required for this formulation: a more elaborate exposition with its connection to Lefschetz duality is given in Appendix B. We consider a simplicial complex K triangulating the d-sphere $\mathbb{S}^d$, and $\widetilde{K}$ its dual cell complex. For a k-simplex σ of K, we denote by $\widetilde{\sigma}$ its dual $(d-k)$-cell in $\widetilde{K}$. We denote by *intersection product* the following bilinear form.

$$\begin{aligned} \otimes : \mathbf{C}_k(K) \times \mathbf{C}_{d-k}(\widetilde{K}) &\to \mathbb{F} \\ (\Gamma, \gamma) \mapsto \Gamma \otimes \gamma &\underset{\text{def.}}{=} \sum_{\sigma \in K^{(k)}} \Gamma(\sigma)\, \gamma(\widetilde{\sigma}) \end{aligned} \tag{1}$$

This intersection product allows characterizing homologous chains by verifying their intersection products with dual cycles.

▶ **Lemma 6.** *Given* $\Gamma, \Gamma_0 \in \mathbf{C}_{d-1}(K, B)$

$$\Gamma \textit{ homologous to } \Gamma_0 \iff \forall \gamma \in \mathbf{Z}_1(\widetilde{K \setminus B}),\ (\Gamma - \Gamma_0) \otimes \gamma = 0$$

Proof. This stems from the algebraic property $\operatorname{Im} A^t = \Big(\operatorname{Ker} A\Big)^{\perp}$, applied to the boundary operator $A = \partial_{\mathbf{C}_1(\widetilde{K})} = \big(\partial_{\mathbf{C}_d(K)}\big)^t$. A complete proof is given in Appendix B.2. ◀

We denote by $G_{K \setminus B} = (\mathcal{V}_{K \setminus B}, \mathcal{E}_{K \setminus B})$ the 1-skeleton of the complex $\widetilde{K \setminus B}$, where $\mathcal{V}_{K \setminus B}$ and $\mathcal{E}_{K \setminus B}$ are respectively the set of vertices and edges of the graph. Note that $\mathbf{Z}_1(G_{K \setminus B}) = \mathbf{Z}_1(\widetilde{K \setminus B})$. We now consider the following graph problem.

▶ **Problem 3** (Dual problem). *Given a subgraph* G *of* $G_{\widetilde{K \setminus B}}$*, with a total order on its edges, and* $\Gamma_0 \in \mathbf{C}_{d-1}(K, B)$*, consider the set:*

$$\triangle_G \underset{def.}{=} \left\{ \Gamma \in \mathbf{C}_{d-1}(K, B) \mid \forall \gamma \in \mathbf{Z}_1(G),\ (\Gamma - \Gamma_0) \otimes \gamma = 0 \right\}$$

and find the lexicographic optimal chain over this set: $\Gamma_{\min} \underset{def.}{=} \min_{\sqsubseteq_{lex}} \triangle_G$.

We have then immediately the following problem equivalency from Lemma 6.

▶ **Lemma 7.** *Consider* $\Gamma_0 \in \mathbf{C}_{d-1}(K, B)$ *and* $(K \simeq \mathbb{S}^d, B)$ *a simplicial pair. Problem 2 for the simplicial pair* (K, B) *and* Γ_0 *is equivalent to Problem 3 for the graph* $G_{K \setminus B}$ *and* Γ_0.

We enumerate the edges of $\mathcal{E}_{K \setminus B}$ in a decreasing order induced by the total order on the primal $(d-1)$-simplices of $K \setminus B$: $e_1 > e_2 > \cdots > e_n$, where n denotes the number of edges of the graph $G_{K \setminus B}$. We then define the following sequence of graphs:

$$G_i \underset{\text{def.}}{=} \Big(\mathcal{V}_{K \setminus B},\ \mathcal{E}_i \underset{\text{def.}}{=} \{ e_j \in \mathcal{E}_{K \setminus B} \mid j \le i \} \Big) \tag{2}$$

In particular, $G_0 = (\mathcal{V}_{K \setminus B}, \varnothing)$ and $G_n = G_{K \setminus B}$. The following observation shows an important property of the solutions of Problem 3 for this increasing sequence of graphs. We naturally extend the dual notation $\widetilde{\cdot}$ to sets of 1-simplices.

▶ **Observation 8.** *Denote by $\triangle_{G_i}$ the set of Problem 3 associated to some graph $G_i = (\mathcal{V}_{K\setminus B}, \mathcal{E}_i)$ defined in Equation (2). For $\Gamma \in \mathbf{C}_{d-1}(K,B)$ and $\gamma \in \mathbf{Z}_1(G_i)$,*

$$|\Gamma - \Gamma_0| \cap \widetilde{|\gamma|} = \emptyset \implies (\Gamma - \Gamma_0) \otimes \gamma = 0$$

The constraints defining the set $\triangle_{G_i}$ rely only on the values of Γ on the $(d-1)$-simplices dual to 1-simplices in $\mathcal{E}_i$. The optimal chain on $\triangle_{G_i}$ therefore verifies:

$$\left|\min_{\sqsubseteq_{lex}} \triangle_{G_i}\right| \subset \widetilde{\mathcal{E}_i}$$

The minimum of Problem 3 for $G_{K\setminus B}$ can be constructed incrementally by considering the increasing sequence of graphs $(G_i)_{i=0,\ldots,n}$. To this end, we borrow terminology from persistent homology [10, Section VII.1] and qualify edges as *positive* if their addition creates new cycles in the graph and as *negative* if their addition merges two connected components.

▶ **Lemma 9.** *Consider $\Gamma_0 \in \mathbf{C}_{d-1}(K,B)$. Denote by $\Gamma^{(i)}$ and $\Gamma^{(i+1)}$ the respective solutions of Problem 3 for the graphs G_i and G_{i+1} and the chain Γ_0. Denote by e_{i+1} the edge added to G_i to form G_{i+1}.*

- *If the edge e_{i+1} is negative:*

$$\Gamma^{(i+1)} = \Gamma^{(i)}$$

- *If the edge e_{i+1} is positive, for any $\gamma_{i+1} \in \mathbf{Z}_1(G_{i+1}) \setminus \mathbf{Z}_1(G_i)$ verifying $\gamma_{i+1}(e_{i+1}) = 1$,*

$$\Gamma^{(i+1)} = \Gamma^{(i)} + \alpha\, \widetilde{e_{i+1}} \text{ with } \alpha = -(\Gamma^{(i)} - \Gamma_0) \otimes \gamma_{i+1}$$

Proof. A negative edge will kill a connected component but no new cycles will be formed by adding this edge and $\mathbf{Z}_1(G_{i+1}) = \mathbf{Z}_1(G_i)$. We have immediately that $\Gamma^{(i)}$ is solution for G_{i+1}.

For a positive edge e_{i+1}, a new graph cycle is formed in G_{i+1}. In terms of chains, this means $\dim \mathbf{Z}_1(G_{i+1}) = \dim \mathbf{Z}_1(G_i)+1$ and we can consider a 1-chain γ_{i+1} in $\mathbf{Z}_1(G_{i+1})\setminus\mathbf{Z}_1(G_i)$ verifying $\gamma_{i+1}(e_{i+1}) = 1$. Let's verify that $\Gamma^{(i)} + \alpha\widetilde{e_{i+1}} \in \triangle_{G_{i+1}}$, with $\alpha = -(\Gamma^{(i)} - \Gamma_0) \otimes \gamma_{i+1}$. Consider any cycle γ in $\mathbf{Z}_1(G_{i+1})$ and $\mu = \gamma(e_{i+1})$ its coefficient for the edge e_{i+1}. The cycle γ can be written as $\gamma' + \mu\gamma_{i+1}$, where $\gamma' = \gamma - \mu\gamma_{i+1}$ is in $\mathbf{Z}_1(G_i)$. The following intersection product along this cycle can then be decomposed as:

$$(\Gamma^{(i)} + \alpha\widetilde{e_{i+1}} - \Gamma_0) \otimes \gamma = (\Gamma^{(i)} - \Gamma_0) \otimes \gamma' + \alpha\widetilde{e_{i+1}} \otimes \gamma' + \mu(\Gamma^{(i)} + \alpha\widetilde{e_{i+1}} - \Gamma_0) \otimes \gamma_{i+1}$$

By definition of $\Gamma^{(i)}$ as solution of Problem 3 for G_i, $(\Gamma^{(i)} - \Gamma_0) \otimes \gamma' = 0$. As $e_{i+1} \notin |\gamma'|$, we also have $\widetilde{e_{i+1}} \otimes \gamma' = 0$. Finally, as $\gamma_{i+1}(e_{i+1}) = 1$ and from the definition of α, the last term of the sum is also zero:

$$(\Gamma^{(i)} + \alpha\widetilde{e_{i+1}} - \Gamma_0) \otimes \gamma_{i+1} = (\Gamma^{(i)} - \Gamma_0) \otimes \gamma_{i+1} + \alpha\, \widetilde{e_{i+1}} \otimes \gamma_{i+1} = 0$$

and we have shown that $\Gamma^{(i)} + \alpha\widetilde{e_{i+1}} \in \triangle_{G_{i+1}}$. As $\Gamma^{(i+1)}$ is defined as the lexicographic minimum of $\triangle_{G_{i+1}}$, $\Gamma^{(i+1)} \sqsubseteq_{lex} \Gamma^{(i)} + \alpha\widetilde{e_{i+1}}$. We have also that $\Gamma^{(i)} \sqsubseteq_{lex} \Gamma^{(i+1)}$ from the inclusion $\triangle_{G_{i+1}} \subseteq \triangle_{G_i}$,. Hence the following bounds for $\Gamma^{(i+1)}$:

$$\Gamma^{(i)} \sqsubseteq_{lex} \Gamma^{(i+1)} \sqsubseteq_{lex} \Gamma^{(i)} + \alpha\widetilde{e_{i+1}} \tag{3}$$

Recall that Observation 8 showed that $\left|\Gamma^{(i+1)}\right| \subset \widetilde{\mathcal{E}_{i+1}}$ which means, together with Equation (3), that the supports of $\Gamma^{(i)}$ and $\Gamma^{(i+1)}$ can only differ on $\widetilde{e_{i+1}}$. When $\alpha = -(\Gamma^{(i)} - \Gamma_0) \otimes \gamma_{i+1}$ is not zero, $\Gamma^{(i)} \notin \triangle_{G_{i+1}}$. Therefore, the support of $\Gamma^{(i+1)}$ needs to contain $\widetilde{e_{i+1}}$ and the unicity of the solution from Corollary 4 implies $\Gamma^{(i+1)} = \Gamma^{(i)} + \alpha\widetilde{e_{i+1}}$. When $\alpha = -(\Gamma^{(i)} - \Gamma_0) \otimes \gamma_{i+1}$ is zero, we can of course also write $\Gamma^{(i+1)} = \Gamma^{(i)} + \alpha\widetilde{e_{i+1}}$. ◀

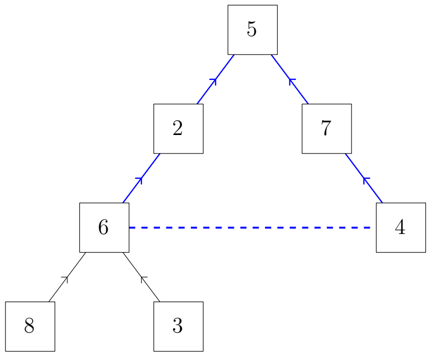


Figure 3 A set of oriented edges obtained by performing FindSet operations for the two nodes 6 and 4 of the same set. The dotted edge, which is not in the disjoint-set structure, illustrates how a cycle can be constructed from the structure.

Lemma 9 shows that two key elements are required to incrementally construct the solution for Problem 3: tracking connected components in the sequence of graphs $(G_i)_{i=0\dots n}$, to distinguish between positive and negative edges, and computing the coefficient α for any new cycle going through a positive edge. The former requires the classic disjoint-set data structure to keep track of incremental connected components. In the next section, we will show how a small augmentation of this data structure allows us to efficiently compute the latter.

3.2 Augmented disjoint-set data structure

Lemma 9 requires, for a *positive* edge, to compute intersection products along a cycle passing through this edge. As illustrated in Figure 3, the tree structure of the classic disjoint-set data structure can be used to define such a cycle which goes through the root element of the set.

We now describe an augmented version of the disjoint-set structure enabling the computation of intersection products and give the corresponding MakeSet*, FindSet* and LinkSet* operations in Algorithm 1. A tree node of this augmented structure consists of a parent pointer to an ancestor and a value in $\mathbb{F}$, representing the intersection product along a directed path in the disjoint-set from this node to the parent (respectively denoted by *parent* and *value* in functions of Algorithm 1).

The reader can verify that the two strategies making the standard disjoint-set structure asymptotically optimal can still be used in this augmented version. During path compression, the *value* associated with a node needs to be updated by summing all coefficients on the compressed path starting from this node. During union by rank, the given value in the LinkSet* operation has to be set to its opposite if the largest set corresponds to the first given representative. The structure makes exactly the same number of addition as the tree height when calling the FindSet* operation. Therefore, the complexity of its augmented version is $c\boldsymbol{\alpha}(n)$, where n is the number of sets, $\boldsymbol{\alpha}$ is the inverse Ackermann function and c denotes the cost of addition in $\mathbb{F}$.

3.3 Algorithmic solution

In this section, from the dual formulation presented in Section 3.1 and with the help of the augmented disjoint-set structure described in Section 3.2, we present an algorithmic solution for Problem 2.

Algorithm 1 Augmented disjoint-set operations

Function `MakeSet*`(*Element* x)**:**
  $x.parent = x$
  $x.value = 0_{\mathbb{F}}$
Function `FindSet*`(*Element* x)**:**
  $g = 0_{\mathbb{F}}$
  while $x \neq x.parent$ **do**
    $g = g + x.value$
    $x = x.parent$
  end
  return (x, g)
Function `LinkSet*`(*Element* r_1, *Element* r_2, $\mathbb{F}$ δ)**:**
  // r_1, r_2 are set representatives
  $r_1.parent = r_2$
  $r_1.value = \delta$

▶ **Lemma 10.** *Taking as inputs $\Gamma_0 \in \mathbf{C}_{d-1}(K, B)$ and the graph $G_{K \setminus B}$ with edges sorted in decreasing order along the total order defined on $(d-1)$-simplices, Algorithm 2 solves Problem 2 in $\mathcal{O}(cn\boldsymbol{\alpha}(n))$ time complexity, where n is the number of edges in the graph $G_{K\setminus B}$, $\boldsymbol{\alpha}$ the inverse Ackermann function and c the cost of addition in $\mathbb{F}$.*

Algorithm 2 Lexicographic optimal homologous relative chain

Inputs : $G = (\mathcal{V}_{K\setminus B}, \mathcal{E}_{K\setminus B})$ with $\mathcal{E}_{K\setminus B} = \{e_i, i = 1, \ldots, n\}$ in decreasing order,
  $\Gamma_0 \in \mathbf{C}_{d-1}(K, B)$.
Output : $\Gamma_{\min} \in \mathbf{C}_{d-1}(K, B)$, lexicographic optimal relative chain homologous to Γ_0.
$\Gamma_{\min} \leftarrow 0$
for $v \in \mathcal{V}_{K\setminus B}$ **do**
  `MakeSet*`(v)
end
for $e \in \mathcal{E}_{K\setminus B}$ *in decreasing order* **do**
  $e = (v_1, v_2) \in \mathcal{V}_{K\setminus B} \times \mathcal{V}_{K\setminus B}$
  $(r_1, g_1) \leftarrow$ `FindSet*`(v_1)
  $(r_2, g_2) \leftarrow$ `FindSet*`(v_2)
  $\alpha = \Gamma_0(\widetilde{e}) + g_2 - g_1$
  if $r_1 = r_2$ **then**
    $\Gamma_{\min} \leftarrow \Gamma_{\min} + \alpha \cdot \widetilde{e}$
  else
    `LinkSet*`(r_1, r_2, α)
end

Proof. Note first that all edges added to the disjoint-set structure are negative and therefore, from Lemma 9, $\Gamma_{\min}(\widetilde{e}) = 0$ for any edge e in the disjoint-set structure. More generally, for any path γ of edges in the disjoint-set structure,

$$\Gamma_{\min} \otimes \gamma = 0 \tag{4}$$

Algorithm 2 is an iterative application of Lemma 9. Every edge of the graph is considered in decreasing order along the total order on 2-simplices, which corresponds to the graph

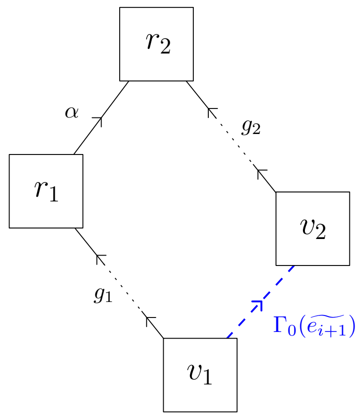


Figure 4 Illustration of the augmented disjoint-set data structure after the addition of a positive edge e_{i+1} between v_1 and v_2. The value g_1 (resp. g_2) corresponds to the sum of the values along the path from the node v_1 (resp. v_2) to the node r_1 (resp. r_2). The blue dashed edge is not present in the disjoint-set structure but illustrates the value that needs to be stored in the disjoint-set structure.

filtration $(G_i)_{i=0,\dots,n}$. At each iteration i, we will show by induction that the following two invariants are verified:

- $\Gamma_{\min}$ is the solution of Problem 3 for G_i,
- the augmentation part of the disjoint-set structure verifies, for any negative edge $e = (v_1, v_2) \in \mathcal{E}_i$, $\Gamma_0 \otimes e = g_2 - g_1$, where g_1, g_2 are the respective values returned by the FindSet* operations on v_1 and v_2.

Sets are created using the MakeSet* operation for each vertex in $\mathcal{V}_{K\setminus B}$. Initially, $\Gamma_{\min} = 0$, which corresponds to the solution of Problem 3 for G_0. As $\mathcal{E}_0 = \varnothing$, the invariant on the augmentation part of the disjoint-set structure is also verified.

At an iteration i, we consider the edge e_{i+1} and suppose $\Gamma_{\min}$ is the solution of Problem 3 for graph G_i. We also assume the invariant on the augmented disjoint-set is verified. Following Lemma 9, we first determine whether the edge e_{i+1} is positive or negative by testing the set representatives of both edge vertices in G_i.

If the edge is negative, i.e. $r_1 \neq r_2$, Lemma 9 shows $\Gamma_{\min}$ is still the solution of Problem 3 for graph G_{i+1}. The LinkSet* operation is then used to update the graph connectivity when adding the edge e_{i+1}. It also updates the augmented part in order to verify the invariant. The updated data structure is illustrated in Figure 4. Indeed, denote by g_1 and g_1' (resp. g_2 and g_2') the values obtained by calling the FindSet* operation on v_1 (resp. v_2) before and after the update of the data structure (i.e. before and after having set α as coefficient for r_1). We now have:

$$g_2' - g_1' = g_2 - (g_1 + \alpha) = g_2 - g_1 - (\Gamma_0(\widetilde{e_{i+1}}) + g_2 - g_1) = \Gamma_0 \otimes e_{i+1}$$

If the edge is positive, i.e. $r_1 = r_2$, the value α of Lemma 9 needs to be computed thanks to the path $\gamma_{i+1} = e_{i+1} + p_2 - p_1$, where p_1 and p_2 are respectively paths from v_1 to r_1 and v_2 to $r_2 = r_1$ inside the disjoint-set structure. Note that the condition $\gamma_{i+1}(e_{i+1}) = 1$ is verified.

$$\alpha = -(\Gamma_{\min} - \Gamma_0) \otimes \gamma_{i+1} \tag{5}$$

From Equation (4) and as $\widetilde{e_{i+1}} \notin |\Gamma_{\min}|$:

$$\alpha = \Gamma_0 \otimes \gamma_{i+1} = \Gamma_0(\widetilde{e_{i+1}}) + \Gamma_0 \otimes p_2 - \Gamma_0 \otimes p_1$$

The invariant on the augmented part of the disjoint-set structure at iteration i means the values g_1 and g_2 obtained by calling FindSet* on v_1 and v_2 verify:

$$g_1 = \Gamma_0 \otimes p_1 \ \text{ and } \ g_2 = \Gamma_0 \otimes p_2$$

and therefore $\alpha = \Gamma_0(\widetilde{e_{i+1}}) + g_2 - g_1$. The chain $\Gamma_{\min}$ is updated in consequence to become the minimum of Problem 3 for graph G_{i+1}.

The output of the algorithm is therefore the solution of Problem 3 for the whole graph G_n. Iterating over all edges of the graph while using disjoint-set operations leads to a $\mathcal{O}(cn\boldsymbol{\alpha}(n))$ complexity. ◀

4 Homology representatives and critical basis

This section highlights deeper connections between lexicographic optimal chains and homology groups. From there, we define a canonical basis of optimal cycles that we call *critical basis* and detail algorithms for its computation.

4.1 Optimal homology representatives

The linearity of M_{lex} (Corollary 5 in Section 2.2) allows to define the following subspace $\mathbf{Z}_k^{\min}(K,B)$ of $\mathbf{Z}_k(K,B)$, made of chains that are minimal among relative homologous chains.

$$\mathbf{Z}_k^{\min}(K,B) \underset{\text{def.}}{=} M_{lex}(\mathbf{Z}_k(K,B))$$

The following diagram commutes and implies an isomorphism from $\mathbf{Z}_k^{\min}(K,B)$ to $\mathcal{H}_k(K,B)$:

$$\begin{array}{ccccc} \mathbf{Z}_k(K,B) & \xrightarrow{M_{lex}} & \mathbf{Z}_k^{\min}(K,B) & \xrightarrow{\subset} & \mathbf{Z}_k(K,B) \\ & \searrow & \downarrow \text{iso.} & & \\ & & \mathcal{H}_k(K,B) & & \end{array}$$

We can now draw a few comparisons between lexicographic and l^1 or l^2 optimalities. On the one hand, we see that the vector space of lexicographic optimal chains is isomorphic to homology groups. This mimics the property of harmonic forms, which can be defined in Hodge theory as l^2 minimal chains in their homology classes. Similar to M_{lex}, the map that associates to a chain its unique homologous harmonic form is linear. On the other hand, l^1 optimal chains in their homology class have been more intensively studied than their l^2 counterpart. This is mainly because l^1 minima are sparse and, as such, are visually meaningful geometric representations of corresponding homology classes.

Lexicographic optimality benefits from both properties: the linearity and therefore isomorphism with homology classes of l^2 minima, as well as the sparsity of support and geometric meaningfulness of l^1 minima.

4.2 Critical basis of cycles

We now construct a canonical basis for the linear space $\mathbf{Z}_k^{\min}(K)$. The definition is only given explicitly for the absolute space $\mathbf{Z}_k^{\min}(K)$, but the same construction applies in the context of relative chains for $\mathbf{Z}_k^{\min}(K,B)$. The name *critical basis* comes from the fact each basis vector will be associated with a critical simplex.

▶ **Definition 11** (Critical simplex). *For $\Gamma \in \mathbf{Z}_k^{\min}(K)$, the critical simplex*

$$\mathrm{crit}(\Gamma) \underset{def.}{=} \max |\Gamma|$$

is the maximal k-simplex in the support $|\Gamma|$ of Γ.

▶ **Definition 12** (Critical basis). *We say that $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_m) \in \left(\mathbf{Z}_k^{\min}(K)\right)^m$ is a critical basis of $\mathbf{Z}_k^{\min}(K)$ if and only if, for $0 \le i \le m-1$:*

$$\mathbf{b}_{i+1} = \min_{\sqsubseteq_{lex}} \left\{ \Gamma \in \mathbf{Z}_k^{\min}(K; \mathbb{F}) \mid \Gamma \notin \mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_i) \quad \textit{and} \quad \Gamma\big(\mathrm{crit}(\Gamma)\big) = 1 \right\}$$

with $\mathrm{span}(\emptyset) = \{0\}$. When $\mathbb{F} = \mathbb{Z}_2$, the condition $\Gamma\big(\mathrm{crit}(\Gamma)\big) = 1$ is unnecessary.

▶ **Lemma 13.** *The critical basis is canonical, in other words there is a unique ordered sequence $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_m)$ satisfying Definition 12.*

Proof. Assume, for $i \le m$, the first $i-1$ critical basis elements have been constructed, and denote $S_{i-1} = \mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})$, with $S_0 = \{0\}$. Since the dimension of $\mathbf{Z}_k^{\min} = \mathbf{Z}_k^{\min}(K)$ is $m > i-1$, there is at least one chain in $\mathbf{Z}_k^{\min} \setminus S_{i-1}$, and the number of possible supports being finite, a minimum among them exists. From Observation 2, two lexicographic minima of $\mathbf{Z}_k^{\min} \setminus S_{i-1}$ must have same support. Consider, for a contradiction, two such minima $\Gamma_1 \neq \Gamma_2$ with $\Gamma_1\big(\mathrm{crit}(\Gamma_1)\big) = \Gamma_2\big(\mathrm{crit}(\Gamma_2)\big) = 1$.

One has $|\Gamma_1| = |\Gamma_2|$ so that $\mathrm{crit}(\Gamma_1) = \mathrm{crit}(\Gamma_2)$. Take $\sigma \in |\Gamma_1|$ such that $\Gamma_1(\sigma) \neq \Gamma_2(\sigma)$ and construct the k-cycle

$$\Gamma \underset{\text{def.}}{=} \frac{1}{\Gamma_2(\sigma) - \Gamma_1(\sigma)} \Big(\Gamma_2(\sigma)\, \Gamma_1 - \Gamma_1(\sigma)\, \Gamma_2 \Big)$$

Observe that Γ verifies $\Gamma(\mathrm{crit}(\Gamma)) = \Gamma(\mathrm{crit}(\Gamma_1)) = 1$ and $\Gamma(\sigma) = 0$. As $|\Gamma| \subset |\Gamma_1| \setminus \{\sigma\}$, Γ is strictly smaller for the lexicographic order $\sqsubseteq_{lex}$ than Γ_1. If $\Gamma \notin S_{i-1}$, this is an immediate contradiction with the optimality of Γ_1. Otherwise, if $\Gamma \in S_{i-1}$, we can again consider an element $\sigma' \in |\Gamma| \subset |\Gamma_1|$ and construct

$$\Gamma' \underset{\text{def.}}{=} \Gamma_1 - \frac{\Gamma_1(\sigma')}{\Gamma(\sigma')} \Gamma$$

which is smaller than Γ_1 since $|\Gamma'| \subset |\Gamma_1| \setminus \{\sigma'\}$ and $\Gamma' \notin S_{i-1}$, being the sum of $\Gamma_1 \notin S_{i-1}$ and $-\frac{\Gamma_1(\sigma')}{\Gamma(\sigma')}\Gamma \in S_{i-1}$. We have again a contradiction. ◀

We give two essential properties of critical bases. For any $i, j = 1, \ldots, m$:

$$i \le j \implies b_i \sqsubseteq_{lex} b_j \tag{6}$$

$$\mathbf{b}_j(\mathrm{crit}(\mathbf{b}_i)) = \delta_{ij} \tag{7}$$

where δ_{ij} is the Kronecker delta.

Proof. The first equation is an immediate consequence of the definition of the critical basis elements as minima over an inclusion decreasing sequence of sets. One has also immediately that $\mathbf{b}_i(\mathrm{crit}(\mathbf{b}_i)) = 1$. The following property

$$j < i \implies \mathbf{b}_j(\mathrm{crit}(\mathbf{b}_i)) = 0$$

is verified as, if this was not the case, then $\mathbf{b}_j(\text{crit}(\mathbf{b}_i))\,\mathbf{b}_i - \mathbf{b}_j$ would be strictly smaller than $\mathbf{b}_i$ and not in $\text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})$, a contradiction with the minimality of $\mathbf{b}_i$. Symmetrically, a similar argument holds to show that

$$j > i \implies \mathbf{b}_j(\text{crit}(\mathbf{b}_i)) = 0$$

as, if this was not the case, then $\mathbf{b}_j - \mathbf{b}_j(\text{crit}(\mathbf{b}_i))\,\mathbf{b}_i$ would be strictly smaller than $\mathbf{b}_j$ and not in $\text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{j-1})$, contradicting the minimality of $\mathbf{b}_j$. ◀

It follows that, for $i = 1, \ldots, m$, $\Gamma \in \text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}) \implies \Gamma(\text{crit}(\mathbf{b}_i)) = 0$ and therefore:

$$\left\{\Gamma \in \mathbf{Z}_k^{\min}(K),\ \Gamma(\text{crit}(\mathbf{b}_i)) = 1\right\} \subset \mathbf{Z}_k^{\min}(K) \setminus \text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})$$

Since $\mathbf{b}_i$ is in the first set and, by definition, minimum of the second set, one has:

▶ **Observation 14.** *For $i = 1, \ldots, m$:*

$$\mathbf{b}_i = \min_{\sqsubseteq_{lex}} \left\{\Gamma \in \mathbf{Z}_k^{\min}(K),\ \Gamma(\text{crit}(\mathbf{b}_i)) = 1\right\}$$

The definition of critical basis is interesting in the context of open surface reconstruction, as we can expect that basis elements do not have too many overlapping supports between each other. This intuition seems to be verified in experiments.

4.3 Computation of a critical basis

We now bridge the gap between optimal representatives in $\mathbf{Z}_k^{\min}(K)$ and the critical basis of $\mathbf{Z}_k^{\min}(K)$, allowing to construct the critical basis from any basis of $\mathbf{Z}_k^{\min}(K)$.

▶ **Lemma 15.** *Consider a basis $(\Gamma_i)_{i=1,\ldots,m}$ of $\mathbf{Z}_k^{\min}(K)$ that verifies for all $i, j = 1, \ldots, m$*

$$\begin{cases} \Gamma_i(\text{crit}(\Gamma_i)) = 1 \\ i \leq j \implies \Gamma_i \sqsubseteq_{lex} \Gamma_j \\ i \neq j \implies \text{crit}(\Gamma_i) \neq \text{crit}(\Gamma_j) \end{cases} \tag{8}$$

The critical basis of $\mathbf{Z}_k^{\min}(K)$ can be constructed, for all $i = 1, \ldots, m$, as

$$\mathbf{b}_i = \min_{\sqsubseteq_{lex}} \Gamma_i + \text{span}(\Gamma_1, \ldots, \Gamma_{i-1}) \tag{9}$$

Proof. The proof is given in Appendix D. ◀

The rest of this section links the computation of critical bases to standard matrix reduction algorithms. For simplicity, we consider chains with integer modulo 2 coefficients ($\mathbb{F} = \mathbb{Z}_2$), but these results can be extended using equivalent algorithms on matrices in *Smith normal forms* [10, Section IV.2]. Recall that the low index $\text{low}(M_j)$ of a non-zero column M_j corresponds to the index of the lowest non-zero coefficient of the column M_j. Algorithm 3 is a well-known algorithm used for the computation of persistent homology [10, Section VII.1]. We also recall in Algorithm 4 the matrix algorithm presented in [6]. We denote by $(\sigma_i)_{i=1,\ldots,n}$ the set of ordered k-simplices: $\sigma_1 < \cdots < \sigma_n$.

▶ **Observation 16.** *When considering a non-empty k-chain Γ as a vector of $\mathbb{Z}_2^n$ written in the basis of ordered k-simplices of dimension n, the* low *index corresponds to the index of its critical simplex:*

$$\text{crit}(\Gamma) = \sigma_{\text{low}(\Gamma)}$$

The third property of Equation (8) can therefore be understood as verifying that the lows of the set $(\Gamma_i)_{i=1,\ldots,m}$, seen as vectors in $\mathbb{Z}_2^n$, are unique.

Algorithm 3 Matrix reduction algorithm

Input : A n-by-m matrix M
for $j \leftarrow 1$ **to** n **do**
  while $M_j \neq 0$ **and** $\exists i < j$ *with* $\mathrm{low}(M_i) = \mathrm{low}(M_j)$ **do**
    $M_j \leftarrow M_j + M_i$
  end
end

Algorithm 4 Total reduction algorithm

Inputs : A vector C of dimension n and a reduced matrix R
for $i \leftarrow n$ **to** 1 **do**
  if $C[i] \neq 0$ **and** *there exists a column* R_j *that verifies* $\mathrm{low}(R_j) = i$ **then**
    $C \leftarrow \Gamma + R_j$
  end
end

▶ **Observation 17.** *Consider a set* $(\Gamma_i)_{i=1,\ldots,m}$ *basis of* $\mathbf{Z}_k^{\min}(K;\mathbb{Z}_2)$ *and the matrix* M *whose columns are the corresponding vector representations in* $\mathbb{Z}_2^n$ *of* $(\Gamma_i)_{i=1,\ldots,m}$ *in the basis of ordered* k*-simplices of dimension* n*.*

- *The algorithm transforming this set* $(\Gamma_i)_{i=1,\ldots,m}$ *such that it verifies the properties of Equation* (8) *is the matrix reduction algorithm of* M *(Algorithm 3), followed by a sort in increasing order along the lexicographic order on* k*-chains of the columns of* M*, resulting in a matrix* R*.*
- *The construction of each element* $\mathbf{b}_i$ *of the critical basis as described in Lemma 15 corresponds to the total reduction algorithm (Algorithm 4) on the column* R_i *with the submatrix of* R *formed by the first* $(i-1)$ *columns.*

5 Applications

We now explore two efficient methods of reconstruction for open surfaces in 3D using algorithms described in previous sections. The complex K is assumed to be the Delaunay triangulation of the set of input points, completed into a topological 3-sphere. The lexicographic order on 2-chains is induced by the following total order on 2-simplices:

▶ **Definition 18.** *For* $\sigma_1, \sigma_2 \in K^{(2)}$*,*

$$\sigma_1 \leq \sigma_2 \iff \begin{cases} \mathrm{R_B}(\sigma_1) < \mathrm{R_B}(\sigma_2) \\ \text{or} \\ \mathrm{R_B}(\sigma_1) = \mathrm{R_B}(\sigma_2) \quad \text{and}\, \mathrm{R_C}(\sigma_1) \geq \mathrm{R_C}(\sigma_2) \end{cases}$$

where $\mathrm{R_B}(\sigma)$ *and* $\mathrm{R_C}(\sigma)$ *denote respectively the radius of the smallest enclosing ball and the radius of the circumscribed sphere of* σ*.*

This order has been previously motivated by the characterization of Delaunay and regular triangulations [5] and used in point cloud meshing applications [6].

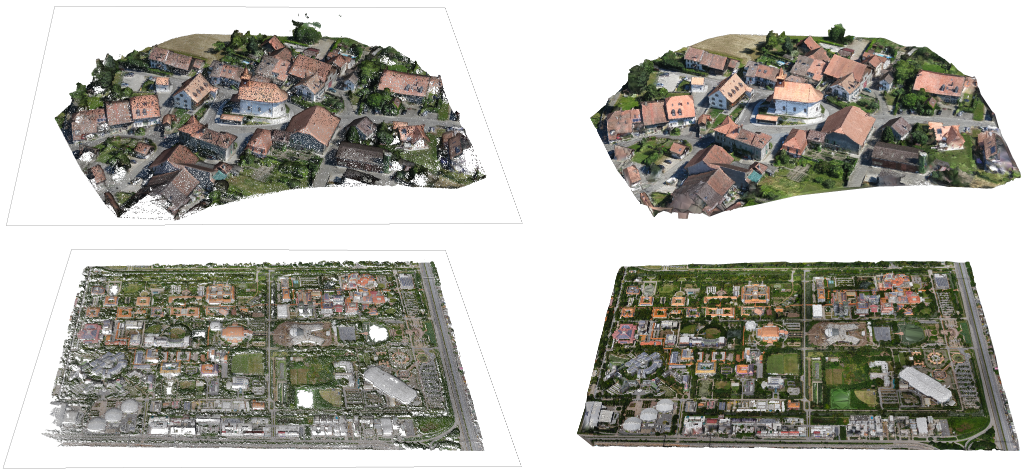


Figure 5 Example of open surface meshing on two models from the senseFly dataset [15]. Computing the optimal chain takes 1.5 seconds (resp. 8.0 seconds) to mesh 1 million (resp. 5 millions) points.

5.1 Open surface reconstruction

We recall the process of open surface reconstruction, illustrated by Figure 1. Assuming a 1-cycle embedded in the Delaunay triangulation K is given, the algorithm presented in Appendix E is used to compute a representative 2-chain having, as its boundary, the provided 1-cycle. Algorithm 2 then computes the optimal absolute chain homologous to the representative chain. As the 3-dimensional Delaunay complex has trivial 2-homology, this is equivalent to the optimal chain bounded by the provided 1-cycle. Different chain coefficients, namely $\mathbb{Z}_2$ and $\mathbb{Q}$, are interesting for open surface reconstruction. Indeed, Figure 1 shows that, when optimizing using $\mathbb{Z}_2$ coefficients, the resulting optimal chain in $\mathbb{Z}_2$ might not be orientable in $\mathbb{Q}$ and therefore the two minimization problems can give different results. In applications, however, the choice of $\mathbb{Q}$ is more appealing to obtain an oriented result. We conjecture, thanks to the unimodularity property of the boundary operator for oriented pseudomanifold [7], that optimality problems with coefficients in $\mathbb{Z}$ can be relaxed to coefficients in $\mathbb{Q}$. Both Algorithm 2 and Algorithm 5 of Appendix E can therefore be extended to chains with coefficients in $\mathbb{Z}$. Figure 5 illustrates use cases of open surface reconstruction for terrains, where the boundary can be defined outside the set of points and the Delaunay triangulation is constructed on both the set of points and the points of this boundary. The observed performance for the reconstruction correlates with the quasi-linear theoretical complexity.

5.2 Boundary detection and critical basis

Based on the construction of a critical basis , the second process intends to automatically reconstruct open surfaces when defining the boundary, as in the previous section, would be too time-consuming. To this end, we first use a curvature estimator on the initial set of points, for instance, the Voronoi Covariance Measure (VCM) [13]. By thresholding this estimator, we can extract a subset $\mathcal{S}$ of vertices and construct a 2-subcomplex B of the Delaunay triangulation K, parameterized by a distance ϵ, as the union of all simplices up to dimension 2 that lie inside the ϵ-offset of $\mathcal{S}$.

By keeping the size of the subcomplex B, denoted by p, relatively small compared to the size n of the Delaunay 3-complex, we can use traditional matrix reduction algorithm [10] to compute the reduced boundary matrix of the complex B in $\mathcal{O}(p^3)$ time complexity. After extracting a basis of $\mathbf{Z}_1(B)$ by exploring the 1-skeleton of the subcomplex, we can compute the lexicographic optimal homologous cycles for all 1-cycles of B, and by filtering out all zero chains, we get a set of cycles $(\gamma_i)_{i=1,\ldots,\beta} \in \mathbf{Z}_1^{\min}(B)$ whose homology classes form a basis of $\mathcal{H}_1(B)$, with $\beta = \dim \mathcal{H}_1(B)$. As the Delaunay complex K triangulates the 3-sphere $\mathbb{S}^3$, its 1- and 2-homologies are trivial. With $\rho^\star$ and $\iota^\star$ the homology morphisms induced respectively by the inclusion map $\rho : \mathbf{C}_2(K) \to \mathbf{C}_2(K,B)$ and the inclusion map $\iota : \mathbf{C}_1(B) \to \mathbf{C}_1(K)$, the following long exact sequence of relative homology

$$\{0\} = \mathcal{H}_2(K) \xrightarrow{\rho^\star} \mathcal{H}_2(K,B) \xrightarrow{\partial} \mathcal{H}_1(B) \xrightarrow{\iota^\star} \mathcal{H}_1(K) = \{0\}$$

implies that $\mathcal{H}_2(K,B) \xrightarrow{\partial} \mathcal{H}_1(B)$ is an isomorphism. By finding representative chains in $\mathbf{Z}_2(K,B)$ (using the algorithm presented in Appendix E) for each element of $(\gamma_i)_{i=1,\ldots,\beta}$, we get a set of cycles whose homology classes form a basis of $\mathcal{H}_2(K,B)$. Using Algorithm 2, we then find the lexicographic optimal relative cycle homologous to each representative chain and construct a basis $(\Gamma_i)_{i=1,\ldots,\beta}$ of $\mathbf{Z}_2^{\min}(K,B)$. Finally, following Observation 17, the critical basis can be obtained from any basis of $\mathbf{Z}_2^{\min}(K,B)$ using matrix reduction algorithms. This construction has a $\mathcal{O}(\beta^2 n)$ time complexity.

The whole process, excluding the complexity of computing the Delaunay triangulation and calling the feature estimator, has a $\mathcal{O}(p^3 + \beta n \log n + \beta^2 n)$ time complexity, where n is the size of the complex K, p the size of the subcomplex B, β the dimension of 1-homology of B (or equivalently the 2-homology of (K,B)). A reconstruction result is presented in Figure 6, where the incomplete nature and presence of occlusions in the point cloud make it a difficult reconstruction problem. However, from the VCM estimation, a reasonable set of cycles can be extracted. The critical basis associated with these cycles creates a complete reconstruction of the scene as well as an interesting geometric decomposition. This allows to either fill or keep the holes corresponding to occluded regions of the input set of points, by removing some of the largest elements of this critical basis.

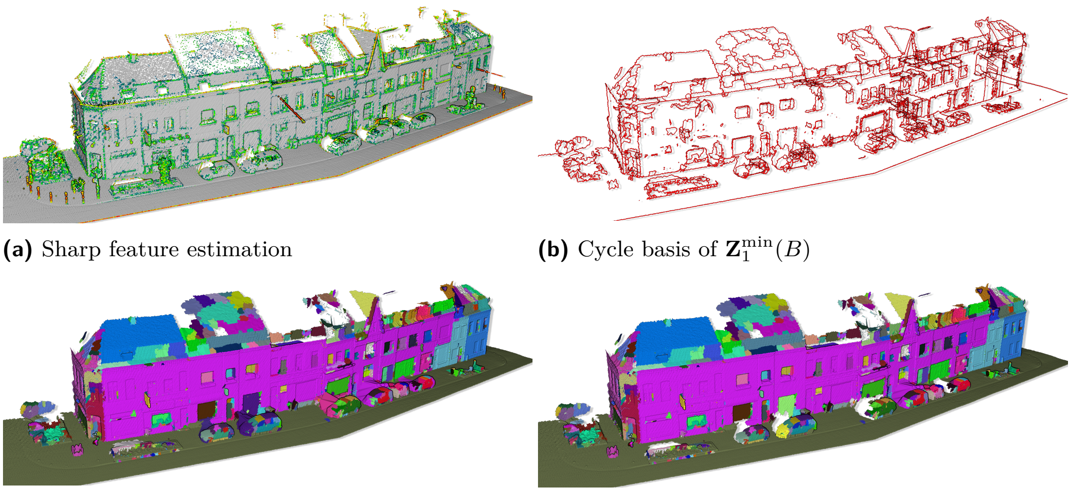
We believe that many other interesting applications can be further developed from the algorithms presented in this work. For instance, the ability to enforce a given boundary can be useful for meshing large sets of points, by implementing a tiling strategy and guaranteeing that each tile mesh coincides with other neighboring tiles. Also, the process described in this section using critical bases could allow to reconstruct stratified objects, given a subcomplex B that correctly captures the singular edges of the object.

References

1 Glencora Borradaile, William Maxwell, and Amir Nayyeri. Minimum Bounded Chains and Minimum Homologous Chains in Embedded Simplicial Complexes. In Sergio Cabello and Danny Z. Chen, editors, *36th International Symposium on Computational Geometry (SoCG 2020)*, volume 164 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:15, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.SoCG.2020.21`.

2 Oleksiy Busaryev, Sergio Cabello, Chao Chen, Tamal K. Dey, and Yusu Wang. Annotating Simplices with a Homology Basis and Its Applications. In Fedor V. Fomin and Petteri Kaski, editors, *Algorithm Theory – SWAT 2012*, pages 189–200, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

3 Erin W. Chambers, Jeff Erickson, and Amir Nayyeri. Minimum cuts and shortest homologous cycles. In *Proceedings of the 25th Annual Symposium on Computational Geometry - SCG '09*, page 377, Aarhus, Denmark, 2009. ACM Press. `doi:10.1145/1542362.1542426`.



(a) Sharp feature estimation

(b) Cycle basis of $\mathbf{Z}_1^{\min}(B)$

(c) Critical basis of $\mathbf{Z}_2^{\min}(K, B)$

(d) Removing the first few chains

Figure 6 After choosing a threshold for the sharp feature estimation (a) and constructing an offset complex B, the basis of $\mathbf{Z}_1^{\min}(B)$ forms a "wireframe" representation of the scene (b). The critical basis correctly reconstructs most parts of the scene (c). In this basis, the largest elements often correspond to the perceived holes of the original set of points (d).

4 Chao Chen and Daniel Freedman. Hardness Results for Homology Localization. *Discrete & Computational Geometry*, 45(3):425–448, April 2011. `doi:10.1007/s00454-010-9322-8`.

5 David Cohen-Steiner, André Lieutier, and Julien Vuillamy. Regular triangulations as lexicographic optimal chains. working paper or preprint, December 2019. URL: `https://hal.archives-ouvertes.fr/hal-02391285`.

6 David Cohen-Steiner, André Lieutier, and Julien Vuillamy. Lexicographic Optimal Homologous Chains and Applications to Point Cloud Triangulations. In Sergio Cabello and Danny Z. Chen, editors, *36th International Symposium on Computational Geometry (SoCG 2020)*, volume 164 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32:1–32:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.SoCG.2020.32`.

7 Tamal K. Dey, Anil N. Hirani, and Bala Krishnamoorthy. Optimal Homologous Cycles, Total Unimodularity, and Linear Programming. *SIAM Journal on Computing*, 40(4):1026–1044, January 2011. `doi:10.1137/100800245`.

8 Tamal K. Dey, Tao Hou, and Sayan Mandal. Computing minimal persistent cycles: Polynomial and hard cases. In *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '20, pages 2587–2606, USA, January 2020. Society for Industrial and Applied Mathematics.

9 Tamal K. Dey, Tianqi Li, and Yusu Wang. Efficient algorithms for computing a minimal homology basis. *arXiv:1801.06759 [cs, math]*, January 2018. `arXiv:1801.06759`.

10 Herbert Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. American Mathematical Society, Providence, R.I, 2010.

11 Herbert Edelsbrunner and Ernst Peter Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Transactions on Graphics*, 9(1):66–104, January 1990. `doi:10.1145/77635.77639`.

12 Jeff Erickson and Kim Whittlesey. Greedy optimal homotopy and homology generators. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 1038–1046, USA, January 2005. Society for Industrial and Applied Mathematics.

13 Quentin Mérigot, M Ovsjanikov, and L J Guibas. Voronoi-Based Curvature and Feature Estimation from Point Clouds. *IEEE Transactions on Visualization and Computer Graphics*, 17(6):743–756, June 2011. `doi:10.1109/TVCG.2010.261`.

14 James B. Orlin. Max flows in O(nm) time, or better. In *Proceedings of the 45th Annual ACM Symposium on Symposium on Theory of Computing - STOC '13*, page 765, Palo Alto, California, USA, 2013. ACM Press. `doi:10.1145/2488608.2488705`.

15 Sensefly. Example dataset for mapping. https://www.sensefly.com/education/datasets/, 2011.

16 John Matthew Sullivan. *A Crystalline Approximation Theorem for Hypersurfaces*. PhD Thesis, Princeton Univ., October 1990.

A Crash course on simplicial homology

A.1 Simplices and complexes

An **(abstract) simplicial complex** K is a collection of finite non-empty sets that is closed under taking subsets, i.e. every subset of an element of K is also in K. A ***k*-simplex** of K is an element of K of size $k+1$. We denote by $K^{(k)}$ the set of k-simplices in K.

A simplex τ subset of σ is called a **face** of σ, and is denoted as $\tau \preceq \sigma$. Reversely, we say σ a **coface** of τ.

A **subcomplex** B of a simplicial complex K is a simplicial complex whose simplices are included in K. In particular, the ***k*-skeleton** of K, containing all simplices in K of dimension at most k, is a subcomplex of K.

A.2 Absolute homology

Let K be a simplicial complex of dimension at least k. While the notions of chains and homology can be defined with coefficients in any ring, we consider here coefficients in a field $\mathbb{F}$, where we have in mind, in particular, the cases of the field of integers modulo 2 ($\mathbb{F} = \mathbb{Z}_2$) and the field of rationals ($\mathbb{F} = \mathbb{Q}$). A ***k*-chain** A with coefficients in $\mathbb{F}$ is a formal sum of k-simplices:

$$A = \sum_i x_i \sigma_i, \text{ with } x_i \in \mathbb{F} \text{ and } \sigma_i \in K^{(k)}$$

We denote by $\mathbf{C}_k(K)$, or $\mathbf{C}_k(K;\mathbb{F})$ when we want to emphasize the chain coefficient, the vector space of k-chains in the complex K.

For a chain $\Gamma \in \mathbf{C}_k(K)$, its **support**, denoted $|\Gamma|$, is the set of simplices for which the coefficient in Γ is not zero:

$$|\Gamma| \underset{\text{def.}}{=} \left\{\sigma \in K^{(k)}, \Gamma(\sigma) \neq 0\right\}$$

When $\mathbb{F} = \mathbb{Z}_2$, the chain coefficient in front of any simplex can be interpreted as indicating the existence of this simplex in the chain. We can view k-chains as sets of k-simplices: for a k-simplex σ and a k-chain A, we then write that $\sigma \in A$ if the coefficient for σ in A is 1. With this convention, the sum (or difference) of two chains corresponds to the symmetric difference on their sets. In what follows, a k-simplex σ can also be interpreted as the k-chain containing only the k-simplex σ.

The **boundary operator** $\partial_k : \mathbf{C}_k(K) \to \mathbf{C}_{k-1}(K)$ is the linear map defined for any k-simplex $\sigma = [a_0, \ldots, a_k]$ as:

$$\partial_k \sigma \underset{\text{def.}}{=} \sum_{i=0}^{k} (-1)^i [a_0, \ldots, \widehat{a_i}, \ldots, a_k]$$

where the symbol $\widehat{a_i}$ means the vertex a_i is deleted from the array. The kernels and images of the boundary operator form respectively the vector space of **cycles** and **boundaries**:

$$\mathbf{Z}_k(K) \underset{\text{def.}}{=} \operatorname{Ker} \partial_k = \left\{\Gamma \in \mathbf{C}_k(K), \partial_k \Gamma = 0\right\}$$

$$\mathbf{B}_k(K) \underset{\text{def.}}{=} \operatorname{Im} \partial_{k+1} = \left\{\Gamma \in \mathbf{C}_k(K), \exists A \in \mathbf{C}_{k+1}(K) \mid \Gamma = \partial_{k+1} A\right\}$$

The following fundamental property of the boundary operator $\partial_k \partial_{k+1} = 0$ implies that $\operatorname{Im} \partial_{k+1} \subset \operatorname{Ker} \partial_k$, or equivalently $\mathbf{B}_k(K) \subset \mathbf{Z}_k(K)$.

Homology groups – forming vector spaces over the field $\mathbb{F}$ – are defined as quotient spaces of cycles over boundaries:

$$\mathcal{H}_k(K) \underset{\text{def.}}{=} \frac{\mathbf{Z}_k(K)}{\mathbf{B}_k(K)}$$

Two k-cycles $A, A' \in \mathbf{Z}_k(K)$ are then said to be **homologous** if they belong to the same homology class in $\mathcal{H}_k(K)$, or equivalently if $A - A' = \partial_{k+1} B$ for some $(k+1)$-chain B. By extension, we also say two k-chains $\Gamma, \Gamma' \in \mathbf{C}_k(K)$ are homologous if their difference is a boundary, i.e. there exists a $(k+1)$-chain B such that $\Gamma - \Gamma' = \partial B$.

A.3 Relative homology

Roughly speaking, the relative homology of a simplicial pair (K, B) is constructed similarly to absolute homology but "ignores" all the part of K inside of B. Given a simplicial subcomplex B of K, relative k-chains are defined as classes in the following quotient space:

$$\mathbf{C}_k(K, B) \underset{\text{def.}}{=} \frac{\mathbf{C}_k(K)}{\mathbf{C}_k(B)}$$

Denote by $\partial_k^{(r)}$ the relative k-boundary operator in (K, B) and by $\partial_k^{(a)}$ the absolute k-boundary operator in K. For any absolute k-chain $\Gamma \in \mathbf{C}_k(K)$,

$$\partial_k^{(r)}(\Gamma + \mathbf{C}_k(B)) \underset{\text{def.}}{=} \partial_k^{(a)}(\Gamma) + \mathbf{C}_{k-1}(B)$$

This is well-defined as $\partial_k^{(a)}(\mathbf{C}_k(B)) \subseteq \mathbf{C}_{k-1}(B)$. Relative cycles and boundaries, are still defined as respective kernel and image of the relative boundary operator:

$$\mathbf{Z}_k(K, B) \underset{\text{def.}}{=} \operatorname{Ker} \partial_k^{(r)}$$

$$\mathbf{B}_k(K, B) \underset{\text{def.}}{=} \operatorname{Im} \partial_{k+1}^{(r)}$$

Translated in term of absolute chains, for $\Gamma \in \mathbf{C}_k(K)$,

$$\Gamma + \mathbf{C}_k(B) \in \mathbf{Z}_k(K, B) \iff |\partial_k^{(a)} \Gamma| \subset B$$

$$\Gamma + \mathbf{C}_k(B) \in \mathbf{B}_k(K, B) \iff \exists A \in \mathbf{C}_{k+1}(K), |\Gamma - \partial_{k+1}^{(a)} A| \subset B$$

Relative homology is again defined as the quotient space of relative cycles over relative boundaries:

$$\mathcal{H}_k(K, B) \underset{\text{def.}}{=} \frac{\mathbf{Z}_k(K, B)}{\mathbf{B}_k(K, B)}$$

While, strictly speaking, a relative k-chain $\Gamma \in \mathbf{C}_k(K, B)$ is a class, we allow ourselves to write, for simplicity,

$$\Gamma = \sum_{\sigma_i \in K \setminus B} x_i \sigma_i$$

where the simplex $\sigma_i \in K \setminus B$ stands in this case for $\sigma_i + \mathbf{C}_k(B)$. For all that follows, the distinction between absolute and relative boundary operators will be omitted and the notation ∂_k will denote either the absolute and relative boundary operator depending on the context.

B Simplicial and Lefschetz duality

B.1 Simplicial duality

Any triangulated manifold K admits a dual polyhedral decomposition. We denote $\widetilde{K}$ the cell complex dual to K, where k-cells are in bijective correspondence with the $(d-k)$-simplices of K. A prime example of this duality is Voronoi diagrams, dual to Delaunay triangulations.

For $0 \leq k \leq d$, denote by $K^{(k)}$ and $\widetilde{K}^{(k)}$ the set of k-simplices of K and the set of k-cells of $\widetilde{K}$. For $\sigma, \tau \in K$ with $\dim(\sigma) = \dim(\tau) - 1$ one write $\sigma \preceq_s \tau$, where $s \in \{-1, 1\}$, when σ is a face of τ and $s \in \{-1, 1\}$ is the sign of σ as it appears in $\partial\tau$ [1]. The duality between K and $\widetilde{K}$ means precisely that:

- for $0 \leq k \leq d$, there is a bijection between $K^{(k)}$ and $\widetilde{K}^{(d-k)}$, where the cell dual of the k-simplex $\sigma \in K^{(k)}$ is denoted $\tilde{\sigma} \in \widetilde{K}^{(d-k)}$,
- for $\sigma, \tau \in K$ such that $\dim(\sigma) = \dim(\tau) - 1$ then:

$$\sigma \preceq_s \tau \iff \tilde{\tau} \preceq_s \tilde{\sigma} \tag{10}$$

Equation (10) can be equivalently formulated as saying the matrices of the respective boundary operators $\partial_{\mathbf{C}_{k+1}(K)} : \mathbf{C}_{k+1}(K) \to \mathbf{C}_k(K)$ and $\partial_{\mathbf{C}_{d-k}(\widetilde{K})} : \mathbf{C}_{d-k}(\widetilde{K}) \to \mathbf{C}_{d-k-1}(\widetilde{K})$, expressed in the simplices (or cells) bases, are transpose of each other:

$$\partial_{\mathbf{C}_{d-k}(\widetilde{K})} = \left(\partial_{\mathbf{C}_{k+1}(K)}\right)^t$$

B.2 Leftchetz duality

For this section, we consider a simplicial complex K triangulating the d-sphere $\mathbb{S}^d$, and $\widetilde{K}$ its dual cell complex. By considering k-chains (respectively $(d-k)$-chains) as vectors expressed in their simplex (resp. cell) bases, we denote by *intersection product* the bilinear form defined by:

$$\begin{aligned} \otimes : \mathbf{C}_k(K) \times \mathbf{C}_{d-k}(\widetilde{K}) &\to \mathbb{F} \\ (\Gamma, \gamma) &\mapsto \Gamma \otimes \gamma \underset{\text{def.}}{=} \sum_{\sigma \in K^{(k)}} \Gamma(\sigma)\, \gamma(\widetilde{\sigma}) = \Gamma^t\, \gamma \end{aligned} \tag{11}$$

As shown in Appendix B.1, the duality between K and $\widetilde{K}$ can be expressed in terms of boundary operators.

$$\partial_{\mathbf{C}_{d-k}(\widetilde{K})} = \left(\partial_{\mathbf{C}_{k+1}(K)}\right)^t$$

This translates to the following commutativity property for the intersection product.

$$\forall \Gamma \in \mathbf{C}_{k+1}(K), \forall \gamma \in \mathbf{C}_{d-k}(\widetilde{K}), \quad \partial\Gamma \otimes \gamma = (\partial\Gamma)^t \gamma = \Gamma^t(\partial^t \gamma) = \Gamma \otimes \partial\gamma \tag{12}$$

While $K \setminus B$ is not a simplicial complex in general, because some simplices in $K \setminus B$ may have faces in B, by contrast, the set of its dual cells, $\widetilde{K \setminus B}$, is a cellular complex. Observe that the set of k-simplices in $K \setminus B$, in bijection with their dual $(d-k)$-cells in $\widetilde{K \setminus B}$, defines a canonical basis for both $\mathbf{C}_k(K, B)$ and $\mathbf{C}_{d-k}(\widetilde{K \setminus B})$ in which the intersection product $\otimes$ corresponds to the canonical dot product between respective coordinates in $\mathbb{F}^n$, where n is the cardinal of $(\widetilde{K \setminus B})^{(d-k)}$. Identifying a chain $\Gamma \in \mathbf{C}_{d-1}(K, B)$ with the dual element $\Gamma^* : \gamma \mapsto \Gamma \otimes \gamma$, the orthogonal complement can be used to describe duality properties.

[1] The sign s can be ignored if $\mathbb{F} = \mathbb{Z}_2$

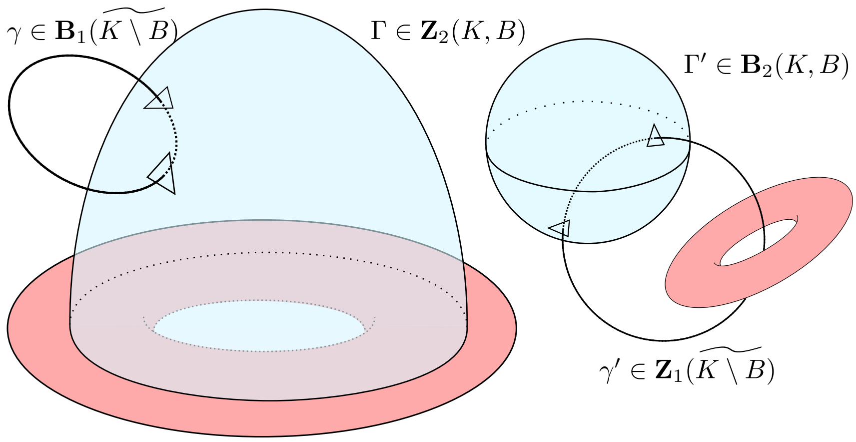


Figure 7 Illustration of the intersection product and Lemma B.2 in dimension 3. The pink region corresponds to subcomplex B of the ambient complex K. The intersection product between $\Gamma \in \mathbf{Z}_2(K, B)$ and any $\gamma \in B_1(\widetilde{K \setminus B})$ verifies $\Gamma \otimes \gamma = 0$. The intersection product between $\Gamma' \in \mathbf{B}_2(K, B)$ and any $\gamma' \in \mathbf{Z}_1(\widetilde{K \setminus B})$ also verifies $\Gamma' \otimes \gamma' = 0$.

▶ **Definition 1.** *For a vector subspace V of $C_1(\widetilde{K \setminus B})$,*

$$V^{\perp} \underset{def.}{=} \left\{ \Gamma \in \mathbf{C}_{d-1}(K, B) \mid \forall \gamma \in V,\ \Gamma \otimes \gamma = 0 \right\}$$

Properties of duality between relative 2-chains of (K, B) and 1-chains of the dual $\widetilde{K \setminus B}$, illustrated in dimension 3 by Figure 7, can be elegantly summarized by the following lemma.

▶ **Lemma B.2.** *Let $K \simeq \mathbb{S}^d$ and B a subcomplex of K.*

$$\mathbf{Z}_{d-1}(K, B) = \mathbf{B}_1(\widetilde{K \setminus B})^{\perp} \quad \textit{and} \quad \mathbf{B}_{d-1}(K, B) = \mathbf{Z}_1(\widetilde{K \setminus B})^{\perp}$$

Proof. This is obtained from an elementary property of linear algebra. For a matrix A:

$$\text{Ker } A = \left(\text{Im } A^t \right)^{\perp} \quad \text{and} \quad \text{Im } A^t = \left(\text{Ker } A \right)^{\perp}$$

Thanks to Equation (12), Lemma B.2 is derived from previous equations by considering respectively the matrices $\partial_{\mathbf{C}_{d-1}(K,B)}$ and $\partial_{\mathbf{C}_2(\widetilde{K \setminus B})}$. ◀

The intersection product of previous section extends to homology classes. Indeed, for two homologous relative $(d-1)$-cycles $\Gamma, \Gamma' \in \mathbf{Z}_{d-1}(K, B)$, the fact that $\Gamma' - \Gamma$ belongs to $\mathbf{B}_{d-1}(K, B)$ implies, thanks to Lemma B.2:

$$\forall \gamma \in \mathbf{Z}_1(\widetilde{K \setminus B}), \quad \Gamma' \otimes \gamma = \Gamma \otimes \gamma$$

Similarly, for two homologous 1-cycles $\gamma, \gamma' \in \mathbf{Z}_1(\widetilde{K \setminus B})$:

$$\forall \Gamma \in \mathbf{Z}_{d-1}(K, B), \quad \Gamma \otimes \gamma' = \Gamma \otimes \gamma$$

The intersection product is therefore independent of the chosen representative of the homology class and extends to a bilinear form on homology groups:

$$\otimes : \mathcal{H}_{d-1}(K, B) \times \mathcal{H}_1(\widetilde{K \setminus B}) \to \mathbb{F}$$

▶ **Observation 3.** *The pairing $\otimes : \mathcal{H}_{d-1}(K, B) \times \mathcal{H}_1(\widetilde{K \setminus B}) \to \mathbb{F}$ is perfect, which means it induces an isomorphism between $\mathcal{H}_1(\widetilde{K \setminus B})$ and the dual of $\mathcal{H}_{d-1}(K, B)$. This can be seen as a particular case of Lefschetz duality.*

C Properties of M_{lex}

We characterize the existence, uniqueness, and linearity of lexicographic optimal chains. These properties of lexicographic minima can be best understood in the abstract context of quotient vector spaces. Consider a pair $A \subset E$ of finite-dimensional vector spaces, with a given ordered basis:

$$e_1 < \ldots < e_n$$

of E inducing a lexicographic preorder $\sqsubseteq_{lex}$ on E. Consider the application

$$\begin{aligned} M_{lex}^A : E &\rightarrow E \\ x &\mapsto \min_{\sqsubseteq_{lex}} x + A \end{aligned}$$

The absolute setting presented in Section 2.2 corresponds to $E = \mathbf{C}_k(K)$ and $A = \mathbf{B}_k(K)$. The relative setting corresponds to $E = \mathbf{C}_k(K, B)$ and $A = \mathbf{B}_k(K, B)$.

▶ **Lemma C.4.** *Given $x \in E$, the minimum $M_{lex}^A(x)$ under the total preorder $\sqsubseteq_{lex}$ in Definition 3 exists and is unique.*

Proof. We call the support $|x| \subset \{e_1, \ldots, e_n\}$ of $x \in E$ the set of basis elements for which the corresponding coordinates in x are not zero. Since $x \in x + A$, the set $x + A$ is not empty. Since there is a finite number of supports on the space $x + A$, there exists a minimal support, and therefore at least one minimal vector. Assume, for a contradiction, that there exists $x_1, x_2 \in x + A$, $x_1 \neq x_2$, minimum under the preorder $\sqsubseteq_{lex}$. Consider a basis element $\sigma \in |x_2 - x_1|$. From Observation 2, we have that $|x_1| = |x_2|$, therefore $|x_2 - x_1| \subset |x_1|$ and σ is in $|x_1|$. We now consider the following vector:

$$x_3 \underset{\text{def.}}{=} x_1 + \frac{x_1(\sigma)}{(x_2 - x_1)(\sigma)}(x_2 - x_1)$$

As $x_1 \in x + A$ and $x_2 - x_1 \in A$, we have that x_3 belongs to $x + A$. By construction, it is also smaller than x_1 for the preorder $\sqsubseteq_{lex}$, as $|x_3| \subset |x_1| \setminus \{\sigma\}$. This is a clear contradiction with the definition of x_1 as a minimum of $x + A$. ◀

▶ **Lemma C.5.** *The application M_{lex}^A is linear. In other words, for $x_1, x_2 \in E$ and $\lambda \in \mathbb{F}$,*

$$M_{lex}(x_1 + \lambda x_2) = M_{lex}(x_1) + \lambda M_{lex}(x_2)$$

Proof. Recall that $\forall x \in E, M_{lex}^A(x) - x \in A$, so that one can define $a \in A$ as:

$$a \underset{\text{def.}}{=} M_{lex}(x_1 + \lambda x_2) - M_{lex}(x_1) - \lambda M_{lex}(x_2) \tag{13}$$

Assume for a contradiction that $a \neq 0$ and take $\sigma_{\max} = \max |a|$. The basis element $\sigma_{\max}$ belongs to the support of (at least) one of the three terms of Equation (13). We assume first that $\sigma_{\max} \in |M_{lex}(x_1)|$. We now have the following chain:

$$x \underset{\text{def.}}{=} M_{lex}(x_1) - \frac{M_{lex}(x_1)(\sigma_{\max})}{a(\sigma_{\max})} a$$

that verifies by design $\sigma_{\max} \notin |x|$, so that $x \sqsubset_{lex} M_{lex}(x_1)$. Since $x \in x_1 + A$, this is a clear contradiction with the definition of $M_{lex}(x_1)$: we conclude that $\sigma_{\max} \notin |M_{lex}(x_1)|$. A similar construction assuming $\sigma_{\max} \in |M_{lex}(x_1 + \lambda x_2)|$ or $\sigma_{\max} \in |M_{lex}(x_2)|$ leads to the same contradiction. We conclude that $a = 0$ and M_{lex} is linear. ◀

D Computation of critical basis from lexicographic optimal chains

▶ **Lemma 15.** *Consider a basis $(\Gamma_i)_{i=1,\dots,m}$ of $\mathbf{Z}_k^{\min}(K)$ that verifies for all $i,j=1,\dots,m$*

$$\begin{cases} \Gamma_i(\mathrm{crit}(\Gamma_i)) = 1 \\ i \leq j \implies \Gamma_i \sqsubseteq_{lex} \Gamma_j \\ i \neq j \implies \mathrm{crit}(\Gamma_i) \neq \mathrm{crit}(\Gamma_j) \end{cases} \tag{8}$$

The critical basis of $\mathbf{Z}_k^{\min}(K)$ can be constructed, for all $i=1,\dots,m$, as

$$\mathbf{b}_i = \min_{\sqsubseteq_{lex}} \Gamma_i + \mathrm{span}(\Gamma_1,\dots,\Gamma_{i-1}) \tag{9}$$

Proof. By writing an element $\mathbf{b}_i$ in the basis $(\Gamma_j)_{j=1,\dots,m}$, the third property of Equation (8) requires that the critical simplex of $\mathbf{b}_i$ corresponds to a critical simplex of one element of the basis $(\Gamma_j)_{j=1,\dots,m}$. Therefore, the critical simplices of $(\mathbf{b}_i)_{i=1,\dots,m}$ form a subset of the critical simplices of $(\Gamma_i)_{i=1,\dots,m}$, with same cardinality. This implies that the set of critical simplices is the same between $(\mathbf{b}_i)_{i=1,\dots,m}$ and $(\Gamma_i)_{i=1,\dots,m}$, and in the same order thanks to the second property of Equation (8) and Section 4.2. For $i=1,\dots,m$, we have:

$$\mathrm{crit}(\Gamma_i) = \mathrm{crit}(\mathbf{b}_i)$$

We now show Equation (9). Denote by

$$\Gamma_i^{\min} \underset{\text{def.}}{=} \min_{\sqsubseteq_{lex}} \Gamma_i + \mathrm{span}(\Gamma_1,\dots,\Gamma_{i-1})$$

Any element Γ of the set $\Gamma_i + \mathrm{span}(\Gamma_1,\dots,\Gamma_{i-1})$ verifies $\Gamma(\mathrm{crit}(\mathbf{b}_i)) = \Gamma(\mathrm{crit}(\Gamma_i)) = 1$ as $\Gamma_i(\mathrm{crit}(\Gamma_i)) = 1$ (first condition of Equation (8)) and, from the two last conditions of Equation (8), for any $j < i$, $\mathrm{crit}(\Gamma_j) < \mathrm{crit}(\Gamma_i)$ therefore $\Gamma_j(\mathrm{crit}(\Gamma_i)) = 0$. We have thus that:

$$\Gamma_i + \mathrm{span}(\Gamma_1,\dots,\Gamma_{i-1}) \subset \left\{\Gamma \in \mathbf{Z}_k^{\min}(K),\ \Gamma(\mathrm{crit}(\mathbf{b}_i)) = 1\right\}$$

and from Observation 14,

$$\mathbf{b}_i \sqsubseteq_{lex} \Gamma_i^{\min}$$

Next, we write $\mathbf{b}_i$ in the basis $(\Gamma_j)_{j=1,\dots,m}$ of $\mathbf{Z}_k^{\min}(K)$:

$$\mathbf{b}_i = \sum_{j=1}^{m} \lambda_j \Gamma_j$$

From the third condition of Equation (8), we have that:

$$\mathrm{crit}(\mathbf{b}_i) = \max\left\{\mathrm{crit}(\Gamma_j), \forall 1 \leq j \leq m \mid \ \lambda_j \neq 0\right\}$$

As $\mathrm{crit}(\mathbf{b}_i) = \mathrm{crit}(\Gamma_i)$, this implies that $\lambda_j = 0$ for any $j > i$ and thus $\mathbf{b}_i \in \Gamma_i + \mathrm{span}(\Gamma_1,\dots,\Gamma_{i-1})$ resulting in:

$$\Gamma_i^{\min} \sqsubseteq_{lex} \mathbf{b}_i$$

We have therefore that $|\Gamma_i^{\min}| = |\mathbf{b}_i|$ from Observation 2. Suppose for a contradiction that $\Gamma_i^{\min} \neq \mathbf{b}_i$ and consider $\sigma \in |\Gamma_i^{\min} - \mathbf{b}_i|$. The following chain

$$\Gamma \underset{\text{def.}}{=} \Gamma_i^{\min} - \frac{\Gamma_i^{\min}(\sigma)}{\Gamma_i^{\min}(\sigma) - \mathbf{b}_i(\sigma)} \left(\Gamma_i^{\min} - \mathbf{b}_i\right)$$

verifies $|\Gamma| \subset |\Gamma_i^{\min}| \setminus \{\sigma\}$ and thus $\Gamma \sqsubset_{lex} \Gamma_i^{\min}$. Also as $\mathbf{b}_i \in \Gamma_i + \mathrm{span}(\Gamma_1,\dots,\Gamma_{i-1})$, $\Gamma_i^{\min} - \mathbf{b}_i \in \mathrm{span}(\Gamma_1,\dots,\Gamma_{i-1})$ and $\Gamma \in \Gamma_i + \mathrm{span}(\Gamma_1,\dots,\Gamma_{i-1})$. We have a contradiction with $\Gamma_i^{\min}$ as minimum of the set $\Gamma_i + \mathrm{span}(\Gamma_1,\dots,\Gamma_{i-1})$. We conclude that $\Gamma_i^{\min} = \mathbf{b}_i$. ◀

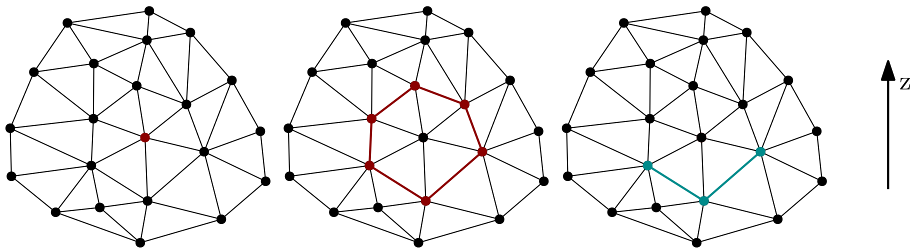


Figure 8 A vertex in a two-dimensional Delaunay complex (left), its link (center) and lower link (right).

E Representative chain in the Delaunay 3-complex

As detailed in the introduction of Section 3, computing a lexicographic optimal 2-chain bounded by a given 1-boundary requires first to compute a "representative" 2-chain bounded by the given boundary. In this section, we describe Algorithm 5 that does exactly that in the Delaunay 3-complex. For simplicity, the algorithm assumes the chain coefficient is $\mathbb{F} = \mathbb{Z}_2$, but the approach extends similarly to an arbitrary coefficient field. We consider chains as sets of simplices.

▶ **Observation 6.** *The* 1*-homology of the Delaunay* 3*-complex is trivial, therefore all cycles are boundaries, which implies the existence of a representative chain bounding any cycle in the Delaunay complex.*

E.1 Lower link of a vertex in the 3D Delaunay complex

In all that follows, $\mathbf{P}$ denotes a set of points in $\mathbb{R}^3$ verifying the following generic condition. In practice, non-generic configurations can be solved by simulation of simplicity [11].

▶ **Condition 1.** *No pair of vertices in* $\mathbf{P}$ *have the same* z*-coordinate.*

We recall the notions of link and lower link.

▶ **Definition 7** (Link of a simplex). *The link* $\mathrm{lk}_K(\tau)$ *of a simplex* τ *in a simplicial complex* K *is the simplicial complex made of all simplices* $\sigma \in K$ *such that* $\tau \cup \sigma \in K$ *and* $\tau \cap \sigma = \emptyset$.

▶ **Definition 8** (Lower link of a vertex). *The lower link* $\mathrm{llk}_K(a)$ *of a vertex* $a \in \mathbf{P}$ *is the simplicial complex made of all simplices in the link* $\mathrm{lk}_K(a)$ *of* a *whose vertices have all their* z*-coordinates smaller than the* z*-coordinate of* a.

Figure 8 illustrates the definitions of link and lower link in a Delaunay 2-complex. The lower link in the Delaunay 3-complex is also illustrated in Figure 2.

Denote by $\mathcal{D}el(\mathbf{P})$ the Delaunay 3-complex of the set of points $\mathbf{P}$. The link $\mathrm{lk}_{\mathcal{D}el(\mathbf{P})}(a)$ and lower link $\mathrm{llk}_{\mathcal{D}el(\mathbf{P})}(a)$ of vertex a are two-dimensional simplicial complexes: each tetrahedron of $\mathcal{D}el(\mathbf{P})$ containing a gives rise to a triangle in $\mathrm{lk}_{\mathcal{D}el(\mathbf{P})}(a)$, each triangle of $\mathcal{D}el(\mathbf{P})$ containing a gives rise to an edge in $\mathrm{lk}_{\mathcal{D}el(\mathbf{P})}(a)$ and each edge of $\mathcal{D}el(\mathbf{P})$ containing a gives rise to a vertex in $\mathrm{lk}_{\mathcal{D}el(\mathbf{P})}(a)$. These triangles or edges belong to $\mathrm{llk}_{\mathcal{D}el(\mathbf{P})}(a)$ if and only if all their vertices have z-coordinates smaller than the z-coordinate a_z of a (see Figure 2, left).

Recall that a topological space and in particular a simplicial complex is said **contractible** if it has the homotopy type of a point. In particular, a contractible simplicial complex K is connected, which means that any pair of vertices in K can be connected by a path of edges in K.

Algorithm 5 relies on the fact that, in a Delaunay complex, the lower link of a vertex is either empty or contractible.

▶ **Lemma E.9.** *Let $\mathcal{D}el(\mathbf{P})$ be the Delaunay complex of a set of vertices $\mathbf{P}$ in $\mathbb{R}^3$ verifying Condition 1 and $a \in \mathbf{P}$ a vertex of the complex.*

- *the lower link $\mathrm{llk}_{\mathcal{D}el(\mathbf{P})}(a)$ is empty if and only if vertex a has minimal z-coordinate in $\mathbf{P}$,*
- *if the z-coordinate of a is not minimal in $\mathbf{P}$, $\mathrm{llk}_{\mathcal{D}el(\mathbf{P})}(a)$ is contractible.*

Proof. Denote by a_z the z-coordinate of a. By definition of the Delaunay triangulation and of $\mathrm{lk}_{\mathcal{D}el(\mathbf{P})}(a)$, there is a one-to-one correspondence between the vertices in $\mathrm{lk}_{\mathcal{D}el(\mathbf{P})}(a)$ and the set of (possibly unbounded) facets contributing to the boundary of the Voronoi cell of a. By definition, a vertex v with z-coordinate v_z is in the lower link of a if and only if its Voronoi cell has a common boundary with the Voronoi cell of a and $v_z < a_z$. It follows that the Voronoi cell of a is contained in the half-space containing a and bounded by the plane bisector of a and v. Such a vertex exists if and only if the vertical half-line starting at a and pointing toward negative z is not contained in the Voronoi cell of a. One can see that the vertical half-line starting at a and pointing toward negative z is contained in the Voronoi cell of a if and only if a has minimal z in V and the first statement is proven.

The lower envelope of the Voronoi cell of a is the union of the facets dual to edges connecting a and a vertex in the lower link of a. When this lower envelope is not empty, its projection on the horizontal plane is a homeomorphism with a convex two-dimensional polytope. This lower envelope is therefore contractible. Since the lower link of a is the nerve of the set of facets in the lower envelope of the Voronoi cell, the second statement follows from the nerve theorem. ◀

E.2 Algorithmic description

Algorithm 5 Finding a set of triangles for a given boundary

Inputs : $\mathcal{D}el(\mathbf{P})$ a Delaunay complex and a 1-cycle $A_0 \in \mathbf{Z}_1(\mathcal{D}el(\mathbf{P}); \mathbb{Z}_2)$
Output : A 2-chain $\Gamma_0 \in \mathbf{C}_2(\mathcal{D}el(\mathbf{P}); \mathbb{Z}_2)$ verifying $\partial\Gamma_0 = A_0$

```
Γ0 ← 0
A ← A0
while A ≠ 0 do
    a ← GetHighestVertex(A)
    Va ← GetAdjacentVertices(a, A)
    LL ← GetLowerLink(a, Del(P))
    E ← GetEdgesConnecting(Va,LL)
    for v ∈ Va do
        A ← A − a ∨ v
    end
    for e ∈ E do
        A ← A + e
        Γ0 ← Γ0 + a ∨ e
    end
end
```

The discussion of Algorithm 5 is divided in two: the first lemma is given to prove the correctness of the algorithm and describes succinctly each subroutine. A better description of these subroutines is then given to analyze the algorithm's complexity. We denote by $\vee$ the join operator on simplices, corresponding to the union for disjoint abstract simplices. In particular, for two 0-simplices a, v, $a \vee v$ denotes the 1-simplex $[a, v]$. For a 0-simplex a and a 1-simplex $e = [e_1, e_2]$, $a \vee e$ denotes the 2-simplex $[a, e_1, e_2]$.

▶ **Lemma E.10.** *Given a Delaunay complex $K = \mathcal{D}el(\mathbf{P})$ of a set of points $\mathbf{P} \subset \mathbb{R}^3$ verifying Condition 1 and an 1-cycle $A_0 \in \mathbf{Z}_1(K; \mathbb{Z}_2)$, Algorithm 5 computes a 2-chain $\Gamma_0 \in \mathbf{C}_2(K; \mathbb{Z}_2)$ such that $\partial \Gamma_0 = A_0$.*

Proof. We verify immediately that with the algorithm's initialization of Γ_0, if A_0 is empty, the algorithm returns, and the property $\partial \Gamma_0 = A_0$ is verified.

The following invariants are shown at each iteration of the while loop:

- $\partial \Gamma_0 = A_0 + A$
- A is a cycle and the z-coordinate of its highest vertex decreases,

The first invariant will imply that, if the algorithm terminates with $A = 0$, we have that $\partial \Gamma_0 = A_0$. The second invariant implies that the algorithm must terminate.

We now describe the operations performed inside the while loop. Figure 2 illustrates one iteration of this loop. The procedure **GetHighestVertex**(A) returns a, the vertex in A with the maximal z-coordinate. Since A is a cycle, an even number of edges of A connects a with a set V_a of vertices in the link $\mathrm{lk}_{\mathcal{D}el(\mathbf{P})}(a)$ of a. This set is returned by the procedure **GetAdjacentVertices**(a, A). The fact a is the highest vertex in A implies the set V_a is a subset of the vertices of lower link $\mathrm{llk}_{\mathcal{D}el(\mathbf{P})}(a)$. This set is not empty and has again an even cardinality. As A is not zero, it must contain at least 2 distinct vertices and therefore, its highest vertex a cannot be the lowest vertex in $\mathcal{D}el(\mathbf{P})$. Lemma E.9 asserts then that the lower link of a is non empty and contractible, proving as a consequence the existence of a 1-chain E bounded by the 0-chain formed by the even set of vertices in V_a. The procedure **GetLowerLink**($a, \mathcal{D}el(\mathbf{P})$) returns the 1-skeleton LL of the lower link of a in $\mathcal{D}el(\mathbf{P})$. **GetEdgesConnecting**(V_a, LL) uses this 1-skeleton LL to construct a 1-chain $E \in \mathbf{C}_1(LL; \mathbb{Z}_2)$ that verifies $\partial E = V_a$.

The main step of the algorithm consists in replacing the edges connecting a in A by the edges on this chain E and adding, for each edge e of E, the corresponding triangle $a \vee e$ in Γ_0. The following 2-chain is added to Γ_0:

$$\sum_{e \in E} a \vee e \tag{14}$$

and its boundary can be evaluated as:

$$\partial\Big(\sum_{e \in E} a \vee e\Big) = \sum_{e \in E} (e - a \vee \partial e) = E - a \vee \partial E = E - a \vee V_a \tag{15}$$

From Equation (15), the boundary of the 2-chain added to Γ_0 corresponds to the 1-chain added to A. This means if the invariant $\partial \Gamma_0 = A_0 + A$ was verified at the previous iteration, it remains true after additions to Γ_0 and A: the first invariant is therefore shown. As A is initialized to A_0 and is updated by adding a boundary, A remains a cycle at each step. Also, all edges connecting a are removed from A and the added edges are in the lower link $\mathrm{llk}_{\mathcal{D}el(\mathbf{P})}(a)$ of a. This shows the second invariant. ◀

We now give more details on each subroutine used in Algorithm 5 to derive its complexity.

▶ **Lemma E.11.** *Under Condition 1, Algorithm 5 in the Delaunay complex $\mathcal{D}el(\mathbf{P})$ can be implemented in $\mathcal{O}(n \log n)$ time complexity, where n corresponds to the size of the complex $\mathcal{D}el(\mathbf{P})$.*

Proof. We first describe a preprocessing step to query the Delaunay triangulation, especially for the **GetLowerLink** operation. Under Condition 1, by writing each edge in $\mathcal{D}el(\mathbf{P})$ as a pair of vertices (v_1, v_2) such that $z(v_1) > z(v_2)$, one can define the following total order on the edges of $\mathcal{D}el(\mathbf{P})$:

$$(v_1, v_2) \leq (v_1', v_2') \quad \underset{\text{def.}}{\iff} \quad \begin{array}{l} z(v_1) > z(v_1') \\ \text{or} \quad v_1 = v_1' \text{ and } z(v_2) \geq z(v_2') \end{array}$$

The set of vertices of the lower link $\text{llk}_{\mathcal{D}el(\mathbf{P})}(a)$ of a vertex a is in one-to-one correspondence with the set of ordered pairs whose first vertex is a. These pairs in the form $(a, \cdot)$ are contiguous in the set of all edges sorted according to this total order.

Similarly, each triangle in $\mathcal{D}el(\mathbf{P})$ can be represented by the ordered triple of vertex (v_1, v_2, v_3) such that $z(v_1) > z(v_2) > z(v_3)$ and one can define the following total order on the triangles of $\mathcal{D}el(\mathbf{P})$:

$$(v_1, v_2, v_3) \leq (v_1', v_2', v_3') \quad \underset{\text{def.}}{\iff} \quad \begin{cases} z(v_1) > z(v_1') \\ \text{or} \quad v_1 = v_1' \text{ and } z(v_2) > z(v_2') \\ \text{or} \quad v_1 = v_1' \text{ and } v_2 = v_2' \text{ and } z(v_3) \geq z(v_3') \end{cases}$$

As previously, the set of edges of the lower link $\text{llk}_{\mathcal{D}el(\mathbf{P})}(a)$ of a vertex a is in one-to-one correspondence with the set of ordered triples whose first vertex is a. These triples in the form $(a, \cdot, \cdot)$ are again contiguous in the set of all triples ordered according to this total order.

Creating these representations (sorted edges and triangles) for the whole Delaunay complex $\mathcal{D}el(\mathbf{P})$ costs $O(n \log n)$, where n is the size of $\mathcal{D}el(\mathbf{P})$.

Procedure **GetLowerLink**$(a, \mathcal{D}el(\mathbf{P}))$, which returns the 1-skeleton LL of the lower link of a in $\mathcal{D}el(\mathbf{P})$, now has a $\log(n)$ time complexity to find the contiguous entries in the ordered sets of edges and triangles of the respective forms $(a, \cdot)$ and $(a, \cdot, \cdot)$.

The cycle A of the algorithm is also represented as an ordered set along the described total order on edges. Getting the largest element of A in the subroutine **GetHighestVertex**(A) and the edges associated with this vertex with the subroutine **GetAdjacentVertices**(a, A) can be performed in a $\mathcal{O}(1)$ time complexity in this ordered representation. Finally, each update of the cycle A (insertion or deletion) can be computed in $\mathcal{O}(\log n)$ time complexity.

The subroutine **GetEdgesConnecting**(V_a, LL) requires a bit more attention to derive the correct complexity. Denote by m the size of the lower link LL of a vertex a. The most obvious way of implementing this subroutine is by creating a spanning tree T of the lower link graph (using a breadth-first search for instance), which has a $\mathcal{O}(m)$ time complexity. From this spanning tree, we could partition the even set of vertices V_a into pairs and construct a path in T for each of these pairs. However, if V_a contains $2k = \mathcal{O}(m)$ vertices, this leads to computing k paths in the spanning tree, which has a $\mathcal{O}(m^2)$ time complexity, followed by adding up to km edges in A, which has an $\mathcal{O}(m^2 \log n)$ time complexity. Instead, we describe Algorithm 6. When creating this spanning tree T of the 1-skeleton LL of the lower link of a vertex a, we also assign an integer rank to each vertex such that the root has rank 0 and each non-root vertex has a rank higher than its parent (this is a linear-time operation and can be seen as a topological sort of the spanning tree).

Note the resemblance of Algorithm 6 to Algorithm 5, but with one dimension less: given a 0-cycle V_a, Algorithm 6 finds a 1-chain E such that $\partial E = V_a$. In fact, Lemma E.9 still

Algorithm 6 GetEdgesConnecting

Inputs : An even set of vertices V_a in the lower link of a vertex a and a spanning tree T of the 1-skeleton LL of this lower link
Output : A set of edges E in T such that $\partial E = V_a$

```
E ← 0
V ← V_a
for rank ← m′ − 1, 1 do
    v ← Vertex(rank)
    if v ∈ V then
        p ← GetParentInTree(v, T)
        V ← V − v + p
        E ← E + v ∨ p
    end
end
```

applies, where the height z is replaced by the vertex rank. Indeed, each vertex which is not the root has as lower link in T: a single vertex, representing its unique parent node in T. This lower link is then contractible. The lower link of the root is empty.

The procedure **GetHighestVertex**(V) of Algorithm 5 is replaced by an iteration on all vertices of T, from the vertex of highest rank $m-1$ to the vertex with rank 1, and the test verifying that v belongs to V. This iteration ends at rank 1, because, since at each step V is a 0-boundary and therefore contains an even number of vertices, the highest (minimal rank) vertex on V cannot be **Vertex**(0). If an array of size m contains all vertices indexed by their rank, each call to the procedure **Vertex**$(rank)$ costs $O(1)$. If the evolving set of vertex V is represented by an array of booleans of size m, where entry k indicates the membership to V of the vertex with rank k, the membership predicate $v \in V$ costs $O(1)$. When V is updated, updating this membership array can be done also in time $O(1)$. It follows that each line in the algorithm costs $O(1)$. For this reason, the cost of Algorithm 6 is $O(m)$. The proof of correctness is similar to the proof of correctness of Algorithm 5 and is based on preserving the following property along the algorithm:

$$\partial E = V_a + V \tag{16}$$

We now summarize the discussion of complexity. After an initial $\mathcal{O}(n \log n)$ time complexity one-time preprocessing of edges and triangles of the Delaunay triangulation, the complexity of each operation **GetLowerLink**, **GetHighestVertex** and **GetAdjacentVertices** can be upper bounded by $\mathcal{O}(\log n)$. With m the size of the lower link of a vertex, the subroutine **GetEdgesConnecting** can be performed in $\mathcal{O}(m)$ time complexity and outputs a set of edges E of size $\mathcal{O}(m)$. Keeping the ordered representation of A at each iteration (i.e. insertion and deletion into an ordered map) leads to a $\mathcal{O}(m \log n)$ time complexity. Finally, each vertex of $\mathbf{P}$ is visited at most once as the highest vertex of A. Note also that the set of edges of all lower links are in 1-to-1 correspondence with the triangles of $\mathcal{D}el(\mathbf{P})$, which means the sum of the sizes of all 1-skeletons of lower links is upper bounded by the size n of the complex. We can therefore conclude the global complexity of Algorithm 5 is $\mathcal{O}(n \log n)$. ◀