



**HAL**  
open science

# Lexicographic Optimal Homologous Chains and Applications to Point Cloud Triangulations

David Cohen-Steiner, André Lieutier, Julien Vuillamy

► **To cite this version:**

David Cohen-Steiner, André Lieutier, Julien Vuillamy. Lexicographic Optimal Homologous Chains and Applications to Point Cloud Triangulations. *Discrete and Computational Geometry*, 2022, 68, 10.1007/s00454-022-00432-6 . hal-03870128

**HAL Id: hal-03870128**

**<https://hal.science/hal-03870128>**

Submitted on 24 Nov 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Lexicographic optimal homologous chains and applications to point cloud triangulations

David Cohen-Steiner · André Lieutier · Julien Vuillamy

Received: date / Accepted: date

**Abstract** This paper considers a particular case of the Optimal Homologous Chain Problem (OHCP) for integer modulo 2 coefficients, where optimality is meant as a minimal lexicographic order on chains induced by a total order on simplices. The matrix reduction algorithm used for persistent homology is used to derive polynomial algorithms solving this problem instance, whereas OHCP is NP-hard in the general case. The complexity is further improved to a quasilinear algorithm by leveraging a dual graph minimum cut formulation when the simplicial complex is a pseudo-manifold. We then show how this particular instance of the problem is relevant, by providing an application in the context of point cloud triangulation.

**Keywords** Simplicial homology, Point cloud triangulation, Delaunay triangulation

**Mathematics Subject Classification (2010)** 68U05

## 1 Introduction

### 1.1 Problem statement

The computation of minimal simplicial homology generators has been a wide subject of interest for its numerous applications related to shape analysis, computer graphics

---

D. Cohen-Steiner  
Université Côte d'Azur, Nice, France  
Inria Sophia Antipolis - Méditerranée, Valbonne, France  
E-mail: david.cohen-steiner@inria.fr

A. Lieutier  
Dassault Systèmes Provence, Aix-en-Provence, France  
E-mail: andre.lieutier@3ds.fr

J. Vuillamy  
Université Côte d'Azur, Nice, France  
Inria Sophia Antipolis - Méditerranée, Valbonne, France  
Dassault Systèmes Provence, Aix-en-Provence, France  
E-mail: julien.vuillamy@3ds.fr

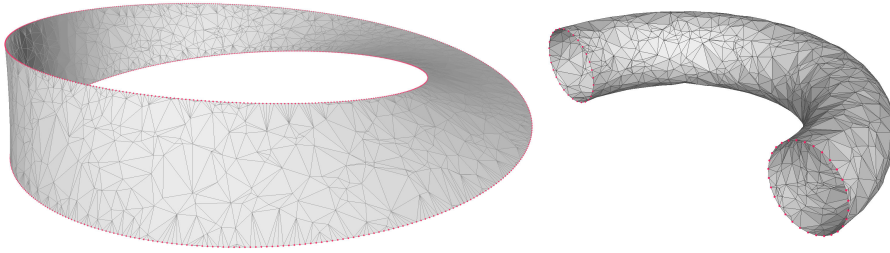


Fig. 1 Open surface triangulations under imposed boundaries (red cycles).

22 or computer-aided design. Coined in [21], we recall the Optimal Homologous Chain  
23 Problem (OHCP):

24 **Problem 1 (OHCP)** Given a  $d$ -chain  $A$  in a simplicial complex  $K$  and a set of weights  
25 given by a diagonal matrix  $W$  of appropriate dimension, find the  $L^1$ -norm minimal  
26 chain  $\Gamma_{\min}$  homologous to  $A$ :

$$27 \quad \Gamma_{\min} = \min_{\Gamma, B} \|W \cdot \Gamma\|_1 \text{ such that } \Gamma = A + \partial_{d+1} B \text{ and } \Gamma \in C_d(K), B \in C_{d+1}(K)$$

28 It has been shown that OHCP is NP-hard in the general case when using coefficients in  
29  $\mathbb{Z}_2$  [9, 15]. However, we consider a specialization of this problem: the Lexicographic  
30 Optimal Homologous Chain Problem (Lex-OHCP). Using coefficients in  $\mathbb{Z}_2$ , mini-  
31 mality is now meant according to a lexicographic order on chains induced by a total  
32 order on simplices. Formulated in the context of OHCP, this would require ordering  
33 the simplices using a total order and taking a weight matrix  $W$  with generic term  
34  $W_{ii} = 2^i$ , where  $i$  is the rank along the total order, allowing the  $L^1$ -norm minimization  
35 to be equivalent to a minimization along the lexicographic order.

## 36 1.2 Contributions

37 After providing some required definitions and notations (Section 2), we show how  
38 an algorithm based on the matrix reduction algorithm used for the computation  
39 of persistent homology [25] allows to solve this particular instance of OHCP in  
40  $O(n^3)$  worst case complexity (Section 3). Using a very similar process, we show that  
41 the problem of finding a minimal  $d$ -chain bounding a given  $(d-1)$ -cycle admits  
42 a similar algorithm with the same algorithmic complexity (Section 4). Section 5  
43 then considers Lex-OHCP in the case where the simplicial complex  $K$  is a strongly  
44 connected  $(d+1)$ -pseudomanifold. By formulating it as a Lexicographic Minimum  
45 Cut (LMC) dual problem, the algorithm can be improved to a quasilinear complexity:  
46 the cost of sorting the dual edges and performing a  $O(E\alpha(E))$  algorithm based on  
47 disjoint-sets, where  $E$  is the number of dual edges and  $\alpha$  is the inverse Ackermann  
48 function. Finally, Section 6 legitimizes this restriction of OHCP by characterizing  
49 the quality of lexicographic optimal homologous chains, namely in the context of  
50 point cloud triangulation. After defining a total order closely related to the Delaunay  
51 triangulation, we provide details on an open surface algorithm given a boundary as

52 well as a watertight surface reconstruction algorithm given an interior and exterior  
53 information.

### 54 1.3 Related work

55 Several authors have studied algorithm complexities for the computation of  $L^1$ -norm  
56 optimal cycles in homology classes [9, 10, 13–15, 21–23, 27, 37]. However, to the  
57 best of our knowledge, considering lexicographic-minimal chains in their homology  
58 classes is a new idea. When minimal cycles are of codimension 1 in a pseudo-  
59 manifold, the idea of considering the minimal cut problem on the dual graph has  
60 been previously studied [35]. In particular, Chambers et al. [9] have considered graph  
61 duality to derive complexity results for the computation of optimal homologous cycles  
62 on 2-manifolds. Chen et al. [15] also use a reduction to a minimum cut problem  
63 on a dual graph to compute minimal non-null homologous cycles on  $d$ -complexes  
64 embedded in  $\mathbb{R}^d$ . Their polynomial algorithm (Theorem 5.2.3 in [15]) for computing  
65 a homology class representative of minimal radius is reminiscent of our algorithm  
66 for computing lexicographic minimal representatives (Algorithm 4). In a recent work  
67 [22], Dey et al. consider the dual graph of pseudo-manifolds in order to obtain  
68 polynomial time algorithms for computing minimal persistent cycles. Since they  
69 consider arbitrary weights, they obtain the  $\mathcal{O}(n^2)$  complexity of best known minimum  
70 cut/maximum flow algorithms [34]. The lexicographic order introduced in our work  
71 can be derived from the idea of a variational formulation of the Delaunay triangulation,  
72 first introduced in [16] and further studied in [1, 17]. Finally, many methods have been  
73 proposed to answer the problem of surface reconstruction in specific acquisition  
74 contexts [30, 31, 33]: [32] classifies a large number of these methods according to the  
75 assumptions and information used in addition to geometry. In the family of purely  
76 geometric reconstruction based on a Delaunay triangulation, one early contribution  
77 is the sculpting algorithm by Boissonnat [6]. The crust algorithm by Amenta et  
78 al. [2, 3] and an algorithm based on natural neighbors [7] were the first algorithms  
79 to guarantee a triangulation of the manifold under sampling conditions. However,  
80 these general approaches usually have difficulties far from these sampling conditions,  
81 in applications where point clouds are noisy or under-sampled. This difficulty can  
82 be circumvented by providing additional information on the nature of the surface  
83 [8, 20, 24]. Our contribution lies in this category of approaches. We provide some  
84 topological information of the surface: a boundary for the open surface reconstruction  
85 and interior and exterior regions for the closed surface reconstruction.

## 2 Definitions

### 2.1 Simplicial complexes

Consider an independent family  $A = (a_0, \dots, a_d)$  of points of  $\mathbb{R}^N$ . We call a  **$d$ -simplex**  $\sigma$  spanned by  $A$  the set of all points:

$$x = \sum_{i=0}^d t_i a_i, \text{ where } \forall i \in \{0..d\}, t_i \geq 0 \text{ and } \sum_{i=0}^d t_i = 1$$

Any simplex spanned by a subset of  $A$  is called a **face** of  $\sigma$ .

A **simplicial complex**  $K$  is a collection of simplices such that every face of a simplex of  $K$  is in  $K$  and the intersection of two simplices of  $K$  is either empty or a common face.

### 2.2 Simplicial chains

Let  $K$  be a simplicial complex of dimension at least  $d$ . The notion of chains can be defined with coefficients in any ring but we restrict here the definition to coefficients in the field  $\mathbb{Z}_2 = \mathbb{Z}/2\mathbb{Z}$ . A  **$d$ -chain**  $A$  with coefficients in  $\mathbb{Z}_2$  is a formal sum of  $d$ -simplices :

$$A = \sum_i x_i \sigma_i, \text{ with } x_i \in \mathbb{Z}_2 \text{ and } \sigma_i \in K$$

We denote  $\mathbf{C}_d(K)$  the vector space over the field  $\mathbb{Z}_2$  of  $d$ -chains in the complex  $K$ . Interpreting the coefficient  $x_i \in \mathbb{Z}_2 = \{0, 1\}$  in front of simplex  $\sigma_i$  as indicating the existence of  $\sigma_i$  in the chain  $A$ , we can view the  $d$ -chain  $A$  as a set of simplices : for a  $d$ -simplex  $\sigma$  and a  $d$ -chain  $A$ , we write that  $\sigma \in A$  if the coefficient for  $\sigma$  in  $A$  is 1. With this convention, the sum of two chains corresponds to the symmetric difference on their sets. In what follows, a  $d$ -simplex  $\sigma$  can also be interpreted as the  $d$ -chain containing only the  $d$ -simplex  $\sigma$ .

### 2.3 Boundary operator

For a  $d$ -simplex  $\sigma = [a_0, \dots, a_d]$ , the **boundary operator** is defined as the operator:

$$\begin{aligned} \partial_d : \mathbf{C}_d(K) &\rightarrow \mathbf{C}_{d-1}(K) \\ \partial_d \sigma &\stackrel{\text{def.}}{=} \sum_{i=0}^d [a_0, \dots, \widehat{a_i}, \dots, a_d] \end{aligned}$$

where the symbol  $\widehat{a_i}$  means the vertex  $a_i$  is deleted from the array. The kernel of the boundary operator  $Z_d = \text{Ker } \partial_d$  is called the group of  $d$ -cycles and the image of the operator  $B_d = \text{Im } \partial_{d+1}$  is the group of  $d$ -boundaries. We say two  $d$ -chains  $A$  and  $A'$  are **homologous** if  $A - A' = \partial_{d+1} B$ , for some  $(d+1)$ -chain  $B$ .

## 2.4 Lexicographic order

We assume now a total order on the  $d$ -simplices of  $K$ ,  $\sigma_1 < \dots < \sigma_n$ , where  $n = \dim \mathcal{C}_d(K)$ . From this order, we define a lexicographic total order on  $d$ -chains.

**Definition 1 (Lexicographic total order)** For  $C_1, C_2 \in \mathcal{C}_d(K)$ :

$$C_1 \sqsubseteq_{lex} C_2 \stackrel{\text{def.}}{\iff} \begin{cases} C_1 + C_2 = 0 \\ \text{or} \\ \sigma_{\max} = \max \{ \sigma \in C_1 + C_2 \} \in C_2 \end{cases}$$

This total order naturally extends to a strict total order  $\sqsubset_{lex}$  on  $\mathcal{C}_d(K)$ .

## 3 Lexicographic optimal homologous chain

### 3.1 Problem statement

In this section, we define a particular instance of OHCP (Problem 1), the Lexicographic Optimal Homologous Chain Problem (Lex-OHCP):

**Problem 2 (Lex-OHCP)** Given a simplicial complex  $K$  with a total order on the  $d$ -simplices and a  $d$ -chain  $A \in \mathcal{C}_d(K)$ , find the unique chain  $\Gamma_{\min}$  defined by :

$$\Gamma_{\min} = \min_{\sqsubseteq_{lex}} \left\{ \Gamma \in \mathcal{C}_d(K) \mid \exists B \in \mathcal{C}_{d+1}(K), \Gamma - A = \partial_{d+1} B \right\}$$

**Definition 2** A  $d$ -chain  $A \in \mathcal{C}_d(K)$  is said **reducible** if there is a  $d$ -chain  $\Gamma \in \mathcal{C}_d(K)$  (called **reduction**) and a  $(d+1)$ -chain  $B \in \mathcal{C}_{d+1}(K)$  such that:

$$\Gamma \sqsubset_{lex} A \quad \text{and} \quad \Gamma - A = \partial_{d+1} B$$

If this property cannot be verified, the  $d$ -chain  $A$  is said **irreducible**. If  $A$  is reducible, we call **total reduction** of  $A$  the unique irreducible reduction of  $A$ . If  $A$  is irreducible,  $A$  is said to be its own total reduction.

Problem 2 can be reformulated as finding the total reduction of  $A$ .

### 3.2 Boundary matrix reduction

With  $m = \dim \mathcal{C}_d(K)$  and  $n = \dim \mathcal{C}_{d+1}(K)$ , we now consider the  $m$ -by- $n$  boundary matrix  $\partial_{d+1}$  with entries in  $\mathbb{Z}_2$ . We enforce that rows of the matrix are ordered according to a given strict total order on  $d$ -simplices  $\sigma_1 < \dots < \sigma_m$ , where the  $d$ -simplex  $\sigma_i$  is the basis element corresponding to the  $i^{\text{th}}$  row of the boundary matrix. The order of columns, corresponding to an order on  $(d+1)$ -simplices, is not relevant for this section and can be chosen arbitrarily.

For a matrix  $R$ , the index of the lowest (i.e. closest to the bottom) non-zero coefficient of a non-zero column  $R_j$  is denoted by  $\text{low}(j)$ , or sometimes  $\text{low}(R_j)$  when we want to explicit the considered matrix.

143 Algorithm 1 is a slightly modified version of the boundary reduction algorithm  
 144 presented in [25]. Indeed, for our purpose, we do not need the boundary matrix  
 145 storing all the simplices of all dimensions and apply the algorithm to the sub-matrix  
 146  $\partial_{d+1} : \mathbf{C}_{d+1}(K) \rightarrow \mathbf{C}_d(K)$ . One checks easily that Algorithm 1 has  $O(mn^2)$  time  
 complexity. We now introduce a few lemmas useful for solving Problem 2. We allow

---

**Algorithm 1:** Reduction algorithm for the  $\partial_{d+1}$  matrix

---

**Input:** A boundary matrix  $\partial_{d+1}$   
**Output:** A reduced boundary matrix  $R$   
 $R = \partial_{d+1}$   
**for**  $j \leftarrow 1$  **to**  $n$  **do**  
 | **while**  $R_j \neq 0$  **and**  $\exists j_0 < j$  with  $\text{low}(j_0) = \text{low}(j)$  **do**  
 | |  $R_j \leftarrow R_j + R_{j_0}$   
 | **end**  
**end**

---

147 ourselves to consider each column  $R_j$  of the matrix  $R$ , formally an element of  $\mathbb{Z}_2^m$ , as  
 148 the corresponding  $d$ -chain in the basis  $(\sigma_1, \dots, \sigma_m)$ .  
 149

150 **Lemma 1** *A  $d$ -chain  $A$  is reducible if and only if at least one of its  $d$ -simplices is*  
 151 *reducible.*

*Proof* If there is a reducible  $d$ -simplex  $\sigma \in A$ ,  $A$  is reducible by the  $d$ -chain  $A' = A - \sigma + \text{Red}(\sigma)$ , where  $\text{Red}(\sigma)$  is a reduction for  $\sigma$ .

We suppose  $A$  to be reducible. Let  $\Gamma \sqsubseteq_{\text{lex}} A$  be a reduction for  $A$  and  $B$  the  $(d+1)$ -chain such that  $\Gamma - A = \partial B$ . We denote  $\sigma_{\max} = \max \{\sigma \in A - \Gamma\}$ . Note that  $\sigma_{\max}$  is homologous to  $\Gamma - A + \sigma_{\max}$ . The chain  $\Gamma - A + \sigma_{\max}$  only contains simplices smaller than  $\sigma_{\max}$ , by definition of the lexicographic order (Definition 1). We have thus shown that if  $A$  is reducible, it contains at least one simplex that is reducible.  $\square$

152 **Lemma 2** *After matrix reduction (Algorithm 1), a non-zero column  $R_j \neq 0$  can be*  
 153 *described as*

$$R_j = \sigma_{\text{low}(j)} + \Gamma, \text{ where } \Gamma \text{ is a reduction for } \sigma_{\text{low}(j)}.$$

*Proof* As all matrix operations performed on  $R$  by the reduction algorithm consist of sums of columns of  $\partial_{d+1}$ , each non-zero column  $R_j$  of  $R$  is in the image of  $\partial_{d+1}$ . Therefore, there exists a  $(d+1)$ -chain  $B$  such that  $R_j = \sigma_{\text{low}(j)} + \Gamma = \partial_{d+1} B$ , which, is equivalent in  $\mathbb{Z}_2$  to  $\Gamma - \sigma_{\text{low}(j)} = \partial_{d+1} B$ . By definition of the low of a column, we also have immediately:  $\Gamma \sqsubseteq_{\text{lex}} \sigma_{\text{low}(j)}$ . For each non-zero column, the largest simplex is therefore reducible by the other  $d$ -simplices of the column.  $\square$

155 **Lemma 3** *After matrix reduction (Algorithm 1), there is a one-to-one correspondence*  
 156 *between the reducible  $d$ -simplices and non-zero columns of  $R$ :*

$$\sigma_i \in \mathbf{C}_d(K) \text{ is reducible} \iff \exists j \in \{1..n\}, R_j \neq 0 \text{ and } \text{low}(j) = i$$

157

158 *Proof* Lemma 2 shows immediately that the simplex corresponding to the lowest  
 159 index of a non-zero column is reducible. Suppose now that a  $d$ -simplex  $\tilde{\sigma}$  is reducible  
 160 and let  $\tilde{\Gamma}$  be a reduction of it:  $\tilde{\sigma} + \tilde{\Gamma} = \partial_{d+1}B$  and  $\tilde{\Gamma} \sqsubset_{lex} \tilde{\sigma}$ . Algorithm 1 realizes  
 161 the matrix factorization  $R = \partial_{d+1} \cdot V$ , where matrix  $V$  is invertible [25]. It follows  
 162 that  $\text{Im } R = \text{Im } \partial_{d+1}$ . Therefore, non-zero columns of  $R$  span  $\text{Im } \partial_{d+1}$  and since  
 163  $\tilde{\sigma} + \tilde{\Gamma} = \partial_{d+1}B \in \text{Im } \partial_{d+1}$ , there is a family  $(R_j)_{j \in J} = (\sigma_{\text{low}(j)}, \Gamma_j)_{j \in J}$  of columns of  
 164  $R$  such that :

$$165 \quad \tilde{\sigma} + \tilde{\Gamma} = \sum_{j \in J} \sigma_{\text{low}(j)} + \Gamma_j$$

166 Every  $\sigma_{\text{low}(j)}$  represents the largest simplex of a column, and  $\Gamma_j$  a reduction chain for  
 167  $\sigma_{\text{low}(j)}$ . As observed in section VII.1 of [25], one can check that the low indexes in  $R$   
 168 are unique after the reduction algorithm. Therefore, there is a  $j_{\text{max}} \in J$  such that for  
 169 all  $j$  in  $J \setminus \{j_{\text{max}}\}$ ,  $\text{low}(j) < \text{low}(j_{\text{max}})$ , which implies:

$$170 \quad \sigma_{j_{\text{max}}} = \max \left\{ \sigma \in \sum_{j \in J} \sigma_{\text{low}(j)} + \Gamma_j \right\} = \max \left\{ \sigma \in \tilde{\sigma} + \tilde{\Gamma} \right\} = \tilde{\sigma}$$

We have shown that for the reducible simplex  $\tilde{\sigma}$ , there is a non-zero column  $R_{j_{\text{max}}}$   
 with  $\tilde{\sigma} = \sigma_{\text{low}(j_{\text{max}})}$  as its largest simplex.  $\square$

### 171 3.3 Total reduction algorithm

172 Combining the three previous lemmas give the intuition on how to construct the total  
 173 reduction solving Problem 2: Lemma 1 allows to consider each simplex individually,  
 174 Lemma 3 characterizes the reducible nature of a simplex using the reduced boundary  
 175 matrix and Lemma 2 describes a column of the reduction boundary matrix as a simplex  
 176 and its reduction. We now present Algorithm 2, referred to as the **total reduction**  
 177 **algorithm**. For a  $d$ -chain  $\Gamma$ ,  $\Gamma[i] \in \mathbb{Z}_2$  denotes the coefficient of the  $i^{\text{th}}$  simplex in  
 the chain  $\Gamma$ .

---

#### Algorithm 2: Total reduction algorithm

---

**Inputs:** A  $d$ -chain  $\Gamma$ , the reduced boundary matrix  $R$   
**Output:** The total reduction of  $\Gamma$   
**for**  $i \leftarrow m$  **to** 1 **do**  
 | **if**  $\Gamma[i] \neq 0$  **and**  $\exists j \in \{1..n\}$  with  $\text{low}(j) = i$  in  $R$  **then**  
 | |  $\Gamma \leftarrow \Gamma + R_j$   
 | **end**  
**end**

---

178

179 **Proposition 1** Algorithm 2 finds the total reduction of a given  $d$ -chain in  $O(m^2)$  time  
 180 complexity.

181 *Proof* In Algorithm 2, let  $\Gamma_{i-1}$  be the value of the variable  $\Gamma$  after iteration  $i$ . Since  
 182 the iteration counter  $i$  decreases from  $m$  to 1, the input and output of the algorithm are



183 respectively  $\Gamma_m$  and  $\Gamma_0$ . At each iteration,  $\Gamma_{i-1}$  are either equal to  $\Gamma_i$  or  $\Gamma_i + R_j$ . Since  
 184  $R_j \in \text{Im } \partial_{d+1}$ ,  $\Gamma_{i-1}$  is in both cases homologous to  $\Gamma_i$ . Therefore,  $\Gamma_0$  is homologous to  
 185  $\Gamma_m$ . We are left to show that  $\Gamma_0$  is irreducible. From Lemma 1, it is enough to check  
 186 that it does not contain any reducible simplex.

187 Let  $\sigma_i$  be a reducible simplex and let us show that  $\sigma_i \notin \Gamma_0$ . Two possibilities may  
 188 occur:

- 189 – if  $\sigma_i \in \Gamma_i$ , then  $\Gamma_{i-1} = \Gamma_i + R_j$ . Since  $\text{low}(j) = i$ , we have  $\sigma_i \in R_j$  and therefore  
 190  $\sigma_i \notin \Gamma_{i-1}$ .
- 191 – if  $\sigma_i \notin \Gamma_i$ , then  $\Gamma_{i-1} = \Gamma_i$  and again  $\sigma_i \notin \Gamma_{i-1}$ .

192 Furthermore, from iterations  $i - 1$  to 1, the added columns  $R_j$  contain only simplices  
 193 smaller than  $\sigma_i$  and therefore  $\sigma_i \notin \Gamma_{i-1} \Rightarrow \sigma_i \notin \Gamma_0$ .

Observe that using an auxiliary array allows to compute the correspondence  
 $\text{low}(j) \rightarrow i$  in time  $\mathcal{O}(1)$ . The column addition nested inside the loop lead to a  $\mathcal{O}(m^2)$   
 time complexity for Algorithm 2.  $\square$

194 It follows that Problem 2 can be solved in  $\mathcal{O}(mn^2)$  time complexity, by applying  
 195 successively Algorithms 1 and 2, or in  $\mathcal{O}(N^3)$  complexity if  $N$  is the size of the  
 196 simplicial complex.

## 197 4 Lexicographic-minimal chain under imposed boundary

### 198 4.1 Problem statement

199 This section will study a variant of Lex-OHCP (Problem 3) in order to solve the  
 200 subsequent problem of finding a minimal  $d$ -chain bounding a given  $(d - 1)$ -cycle  
 201 (Problem 4).

202 **Problem 3** Given a simplicial complex  $K$  with a total order on the  $d$ -simplices and a  
 203  $d$ -chain  $\Gamma_0 \in \mathcal{C}_d(K)$ , find :

$$204 \quad \Gamma_{\min} = \min_{\sqsubseteq_{lex}} \left\{ \Gamma \in \mathcal{C}_d(K) \mid \partial_d \Gamma = \partial_d \Gamma_0 \right\}$$

205 **Problem 4** Given a simplicial complex  $K$  with a total order on the  $d$ -simplices and a  
 206  $(d - 1)$ -cycle  $A$ , check if  $A$  is a boundary:

$$207 \quad B_A \stackrel{\text{def.}}{=} \left\{ \Gamma \in \mathcal{C}_d(K) \mid \partial_d \Gamma = A \right\} \neq \emptyset$$

208 If it is the case, find the minimal  $d$ -chain  $\Gamma$  bounded by  $A$ :

$$209 \quad \Gamma_{min} = \min_{\sqsubseteq_{lex}} B_A$$

210 In Problem 4, finding a representative  $\Gamma_0$  in the set  $B_A \neq \emptyset$  such that  $\partial_d \Gamma_0 = A$  is  
 211 sufficient: we are then taken back to Problem 3 to find the minimal  $d$ -chain  $\Gamma_{\min}$  such  
 212 that  $\partial_d \Gamma_{\min} = \partial_d \Gamma_0 = A$ .



**Algorithm 3:** Total reduction variant

---

**Inputs:** A  $d$ -chain  $\Gamma$  and  $V^{\text{Ker}}$   
**Output:** The minimal chain with the same boundary as  $\Gamma$

```

for  $i \leftarrow m$  to 1 do
  if  $\Gamma[i] \neq 0$  and  $\exists j \in \{1..n^{\text{Ker}}\}$  with  $\text{low}(j) = i$  in  $V^{\text{Ker}}$  then
     $\Gamma \leftarrow \Gamma + V_j^{\text{Ker}}$ 
  end
end
end

```

---

241 *Proof* The proof is similar to the one of Proposition 1.

242 In Algorithm 3, we denote by  $\Gamma_{i-1}$  the value of variable  $\Gamma$  after iteration  $i$ . Since  
 243 iteration counter  $i$  is decreasing from  $m$  to 1, the input and output of the algorithm are  
 244 respectively  $\Gamma_m$  and  $\Gamma_0$ . Since  $V_j^{\text{Ker}} \in \text{Ker } \partial_d$ , at each iteration  $\partial\Gamma_{i-1} = \partial\Gamma_i$  therefore  
 245  $\partial\Gamma_0 = \partial\Gamma_m$ . We are left to show the algorithm's result is the minimal solution.

246 Suppose there is  $\Gamma^*$  such that  $\partial_d\Gamma^* = \partial\Gamma$  and  $\Gamma^* \sqsubset_{lex} \Gamma_0$ . Let's consider the  
 247 difference  $\Gamma_0 - \Gamma^*$ , and its largest element index  $\text{low}(\Gamma_0 - \Gamma^*) = i$ , with  $\sigma_i \in \Gamma_0$   
 248 and  $\sigma_i \notin \Gamma^*$  by Definition 1 of the lexicographic order. As  $\Gamma_0 - \Gamma^* \in \text{Ker } \partial_d \setminus \{0\}$ ,  
 249 there has to be a column  $V_j^{\text{Ker}}$  in  $V^{\text{Ker}}$  where  $\text{low}(V_j^{\text{Ker}}) = i$ , from Lemma 4. Two  
 250 possibilities may occur at iteration  $i$ :

- 251 – if  $\sigma_i \in \Gamma_i$ , then  $\Gamma_{i-1} = \Gamma_i + V_j^{\text{Ker}}$ . Since  $i = \text{low}(j)$ , we have  $\sigma_i \in V_j^{\text{Ker}}$  and  
 252 therefore  $\sigma_i \notin \Gamma_{i-1}$ .
- 253 – if  $\sigma_i \notin \Gamma_i$ , then  $\Gamma_{i-1} = \Gamma_i$  and again  $\sigma_i \notin \Gamma_{i-1}$ .

However, from iterations  $i-1$  to 1, the added columns  $V_j^{\text{Ker}}$  contains only simplices with  
 indices smaller than  $i$  and therefore we obtain  $\sigma_i \notin \Gamma_{i-1} \Rightarrow \sigma_i \notin \Gamma_0$ , a contradiction  
 to the definition of  $\sigma_i$  as the largest element of  $\Gamma^* - \Gamma_0$ .  $\square$

254 4.4 Finding a representative of  $\mathbf{B}_A$ 

255 As previously mentioned, solving Problem 4 requires deciding if the set  $B_A$  is empty  
 256 and in case it is not empty, finding an element of the set  $B_A$ . Algorithm 3 can then be  
 257 used to minimize this element under imposed boundary. In the following algorithm,  
 258  $m = \dim C_{d-1}(K)$  and  $n = \dim C_d(K)$ .

259 **Proposition 3** Algorithm 4 decides if the set  $B_A$  is non-empty, and in that case, finds  
 260 a representative  $\Gamma_0$  such that  $\partial\Gamma_0 = A$  in  $O(m^2)$  time complexity.

261 *Proof* We start by two trivial observations from the definition of a reduction. First,  $A$   
 262 is a boundary if and only if its total reduction is the null chain. Second, if a non-null  
 263 chain is a boundary, then its greatest simplex is reducible.

264 If, at iteration  $i$ ,  $A_0[i] \neq 0$ , then  $\sigma_i$  is the greatest simplex in  $A_0$ . In the case  $R$  has  
 265 no column  $R_j$  such that  $\text{low}(j) = i$ ,  $\sigma_i$  is not reducible by Lemma 3 and therefore  $A_0$   
 266 is not a boundary. Since  $A$  and  $A_0$  differ by a boundary (added columns of  $R$ ),  $A$  is  
 267 not a boundary either. This means the set  $B_A$  is empty.

268 The main difference with the previous chain reduction is we keep track of the column

**Algorithm 4:** Finding a representative

---

**Inputs:** A  $(d - 1)$ -chain  $A$ , a boundary matrix  $R$  reduced by  $V$   
**Output:** A  $d$ -chain  $\Gamma_0$  bounded by  $A$  if  $A$  is a boundary.  
 $\Gamma_0 \leftarrow \emptyset$   
 $A_0 \leftarrow A$   
**for**  $i \leftarrow m$  **to** 1 **do**  
  **if**  $A_0[i] \neq 0$  **then**  
    **if**  $\exists j \in \{1..n\}$  with  $\text{low}(j) = i$  **in**  $R$  **then**  
       $A_0 \leftarrow A_0 + R_j$   
       $\Gamma_0 \leftarrow \Gamma_0 + V_j$   
    **else**  
      The set  $B_A$  is empty.  
    **end**  
  **end**  
**end**

---

269 operations in  $\Gamma_0$ . If the total reduction of  $A$  is null, we have found a linear combination  
270  $(R_j)_{j \in J}$  such that  $A = \sum_{j \in J} R_j$ . We have also computed  $\Gamma_0$  as the sum of the  
271 corresponding columns in  $V$ :  $\Gamma_0 = \sum_{j \in J} V_j$ . As  $R = \partial_d \cdot V$ , we can now verify:

$$272 \quad \partial_d \Gamma_0 = \partial_d \left( \sum_{j \in J} V_j \right) = \sum_{j \in J} R_j = A$$

□

273 **5 Efficient algorithm for codimension 1 (dual graph)**

274 In this section we focus on Problem 5, a specialization of Problem 2, namely when  $K$   
275 is a subcomplex of a  $(d + 1)$ -pseudomanifold.

276 **5.1 Problem statement**

277 Recall that a  $d$ -dimensional simplicial complex is said **pure** if it is of dimension  $d$   
278 and any simplex has at least one coface of dimension  $d$ .

279 **Definition 3** A  $d$ -**pseudomanifold** is a pure  $d$ -dimensional simplicial complex for  
280 which each  $(d - 1)$ -face has exactly two  $d$ -dimensional cofaces.

281 **Definition 4** The **dual graph** of a  $d$ -pseudomanifold  $\mathcal{M}$  is the graph whose vertices  
282 are in one-to-one correspondence with the  $d$ -simplices of  $\mathcal{M}$  and whose edges are in  
283 one-to-one correspondence with  $(d - 1)$ -simplices of  $\mathcal{M}$ : an edge  $e$  connects two  
284 vertices  $v_1$  and  $v_2$  of the graph if and only if  $e$  corresponds to the  $(d - 1)$ -face with  
285 cofaces corresponding to  $v_1$  and  $v_2$ .

286 **Definition 5** A **strongly connected**  $d$ -pseudomanifold is a  $d$ -pseudomanifold whose  
287 dual graph is connected.

288 Given a strongly connected  $(d + 1)$ -pseudomanifold  $\mathcal{M}$  and  $\tau_1 \neq \tau_2$  two  $(d + 1)$ -  
 289 simplices of  $\mathcal{M}$ , we consider a special case of Problem 2 where  $K = \mathcal{M} \setminus \{\tau_1, \tau_2\}$  and  
 290  $A = \partial\tau_1$ :

291 **Problem 5** Given a strongly connected  $(d + 1)$ -pseudomanifold  $\mathcal{M}$  with a total order  
 292 on the  $d$ -simplices and two distinct  $(d + 1)$ -simplices  $(\tau_1, \tau_2)$  of  $\mathcal{M}$ , find:

$$293 \quad \Gamma_{\min} = \min_{\sqsubseteq_{lex}} \left\{ \Gamma \in \mathbf{C}_d(\mathcal{M}) \mid \exists B \in \mathbf{C}_{d+1}(\mathcal{M} \setminus \{\tau_1, \tau_2\}), \Gamma - \partial\tau_1 = \partial B \right\}$$

294 **Definition 6** Seeing a graph  $G$  as a 1-dimensional simplicial complex, we define  
 295 the **coboundary operator**  $\partial^0 : \mathbf{C}_0(G) \rightarrow \mathbf{C}_1(G)$  as the linear operator defined by  
 296 the transpose of the matrix of the boundary operator  $\partial_1 : \mathbf{C}_1(G) \rightarrow \mathbf{C}_0(G)$  in the  
 297 canonical basis of simplices.<sup>1</sup>

298 For a given graph  $G = (\mathcal{V}, \mathcal{E})$ ,  $\mathcal{V}$  and  $\mathcal{E}$  respectively denote its vertex and edge sets.  
 299 For a  $d$ -chain  $\alpha \in \mathbf{C}_d(\mathcal{M})$  and a  $(d + 1)$ -chain  $\beta \in \mathbf{C}_{d+1}(\mathcal{M})$ ,  $\tilde{\alpha}$  and  $\tilde{\beta}$  denote their  
 300 respective dual 1-chain and dual 0-chain in the dual graph  $G(\mathcal{M})$  of  $\mathcal{M}$ . We easily  
 301 see that:

302 *Remark 1* For a set of vertices  $\mathcal{V}_0 \subset \mathcal{V}$ ,  $\partial^0\mathcal{V}_0$  is exactly the set of edges in  $G = (\mathcal{V}, \mathcal{E})$   
 303 that connect vertices in  $\mathcal{V}_0$  with vertices in  $\mathcal{V} \setminus \mathcal{V}_0$ .

304 *Remark 2* Let  $\mathcal{M}$  be a  $(d + 1)$ -pseudomanifold. If  $\alpha \in \mathbf{C}_d(\mathcal{M})$  and  $\beta \in \mathbf{C}_{d+1}(\mathcal{M})$ ,  
 305 then  $\tilde{\alpha} = \partial^0\tilde{\beta} \iff \alpha = \partial_{d+1}\beta$ .

## 306 5.2 Codimension 1 and Lexicographic Min Cut (LMC)

307 The order on  $d$ -simplices of a  $(d + 1)$ -pseudomanifold  $\mathcal{M}$  naturally defines a corre-  
 308 sponding order on the edges of the dual graph:  $\tau_1 < \tau_2 \iff \tilde{\tau}_1 < \tilde{\tau}_2$ . This order  
 309 extends similarly to a lexicographic order  $\sqsubseteq_{lex}$  on sets of edges (or, equivalently,  
 310 1-chains) in the graph.

311 In what follows, we say a set of edges  $\tilde{\Gamma}$  **disconnects**  $\tilde{\tau}_1$  and  $\tilde{\tau}_2$  in the graph  $(\mathcal{V}, \mathcal{E})$   
 312 if  $\tilde{\tau}_1$  and  $\tilde{\tau}_2$  are not in the same connected component of the graph  $(\mathcal{V}, \mathcal{E} \setminus \tilde{\Gamma})$ .

313 Given a graph with weighted edges and two vertices, the min-cut/max-flow prob-  
 314 lem [26, 34] consists in finding the minimum cut (i.e. set of edges) disconnecting the  
 315 two vertices, where minimum is meant as minimal sum of weights of cut edges. We  
 316 consider a similar problem where the minimum is meant in term of a lexicographic  
 317 order: the Lexicographic Min Cut (LMC).

318 **Problem 6 (LMC)** Given a connected graph  $G = (\mathcal{V}, \mathcal{E})$  with a total order on  $\mathcal{E}$  and  
 319 two vertices  $\tilde{\tau}_1, \tilde{\tau}_2 \in \mathcal{V}$ , find the set  $\tilde{\Gamma}_{\text{LMC}} \subset \mathcal{E}$  minimal for the lexicographic order  
 320  $\sqsubseteq_{lex}$ , that disconnects  $\tilde{\tau}_1$  and  $\tilde{\tau}_2$  in  $G$ .

<sup>1</sup> In order to avoid to introduce non essential formal definitions, the coboundary operator is defined over chains since, for finite simplicial complexes, the canonical inner product defines a natural bijection between chains and cochains.

321 **Proposition 4**  $\Gamma_{\min}$  is solution of Problem 5 if and only if its dual 1-chain  $\tilde{\Gamma}_{\min}$  is  
 322 solution of Problem 6 on the dual graph  $G(\mathcal{M})$  of  $\mathcal{M}$  where  $\tilde{\tau}_1$  and  $\tilde{\tau}_2$  are respective  
 323 dual vertices of  $\tau_1$  and  $\tau_2$ .

324 *Proof* Problem 5 can be equivalently formulated as:

$$325 \quad \Gamma_{\min} = \min_{\sqsubseteq_{lex}} \left\{ \partial_{d+1}(\tau_1 + B) \mid B \in \mathcal{C}_{d+1}(\mathcal{M} \setminus \{\tau_1, \tau_2\}) \right\} \quad (1)$$

326 Using Observation 2, we see that  $\Gamma_{\min}$  is the minimum in Equation (1) if and only if  
 327 its dual 1-chain  $\tilde{\Gamma}_{\min}$  satisfies:

$$328 \quad \tilde{\Gamma}_{\min} = \min_{\sqsubseteq_{lex}} \left\{ \partial^0(\tilde{\tau}_1 + \tilde{B}) \mid \tilde{B} \subset \mathcal{V} \setminus \{\tilde{\tau}_1, \tilde{\tau}_2\} \right\} \quad (2)$$

329 Denoting  $\tilde{\Gamma}_{\text{LMC}}$  the minimum of Problem 6, we need to show that  $\tilde{\Gamma}_{\text{LMC}} = \tilde{\Gamma}_{\min}$ .  
 330 As  $\tilde{\Gamma}_{\text{LMC}}$  disconnects  $\tilde{\tau}_1$  and  $\tilde{\tau}_2$  in  $G = (\mathcal{V}, \mathcal{E})$ ,  $\tilde{\tau}_2$  is not in the connected component of  
 331  $\tilde{\tau}_1$  in  $(\mathcal{V}, \mathcal{E} \setminus \tilde{\Gamma}_{\text{LMC}})$ . We define  $\tilde{B}$  as the connected component of  $\tilde{\tau}_1$  in  $(\mathcal{V}, \mathcal{E} \setminus \tilde{\Gamma}_{\text{LMC}})$   
 332 minus  $\tilde{\tau}_1$ . We have that  $\tilde{B} \subset \mathcal{V} \setminus \{\tilde{\tau}_1, \tilde{\tau}_2\}$ . Consider an edge  $e \in \partial^0(\tilde{\tau}_1 + \tilde{B})$ . From  
 333 Observation 1,  $e$  connects a vertex  $v_a \in \{\tilde{\tau}_1\} \cup \tilde{B}$  with a vertex  $v_b \notin \{\tilde{\tau}_1\} \cup \tilde{B}$ . From  
 334 the definition of  $\tilde{B}$ ,  $\tilde{\Gamma}_{\text{LMC}}$  disconnects  $v_a$  and  $v_b$  in  $G$ , which in turn implies  $e \in \tilde{\Gamma}_{\text{LMC}}$ .  
 335 We have therefore shown that  $\partial^0(\tilde{\tau}_1 + \tilde{B}) \subset \tilde{\Gamma}_{\text{LMC}}$ . Using Equation (2), we get:

$$336 \quad \tilde{\Gamma}_{\min} \sqsubseteq_{lex} \partial^0(\tilde{\tau}_1 + \tilde{B}) \sqsubseteq_{lex} \tilde{\Gamma}_{\text{LMC}} \quad (3)$$

Now we claim that if there is a  $\tilde{C} \subset \mathcal{V} \setminus \{\tilde{\tau}_1, \tilde{\tau}_2\}$  with  $\tilde{\Gamma} = \partial^0(\tilde{\tau}_1 + \tilde{C})$ , then  $\tilde{\Gamma}$   
 disconnects  $\tilde{\tau}_1$  and  $\tilde{\tau}_2$  in  $(\mathcal{V}, \mathcal{E})$ . Consider a path in  $G$  from  $\tilde{\tau}_1$  to  $\tilde{\tau}_2$ . Let  $v_a$  be the last  
 vertex of the path that belongs to  $\{\tilde{\tau}_1\} \cup \tilde{C}$  and  $v_b$  the next vertex on the path (which  
 exists since  $\tilde{\tau}_2$  is not in  $\{\tilde{\tau}_1\} \cup \tilde{C}$ ). From Observation 1, we see that the edge  $(v_a, v_b)$   
 must belong to  $\tilde{\Gamma} = \partial^0(\tilde{\tau}_1 + \tilde{C})$ . We have shown that any path in  $G$  connecting  $\tilde{\tau}_1$  and  
 $\tilde{\tau}_2$  has to contain an edge in  $\tilde{\Gamma}$  and the claim is proved.

In particular, the minimum  $\tilde{\Gamma}_{\min}$  disconnects  $\tilde{\tau}_1$  and  $\tilde{\tau}_2$  in  $(\mathcal{V}, \mathcal{E})$ . As  $\tilde{\Gamma}_{\text{LMC}}$  denotes  
 the minimum of Problem 6,  $\tilde{\Gamma}_{\text{LMC}} \sqsubseteq_{lex} \tilde{\Gamma}_{\min}$  which, together with Equation (3), gives  
 us  $\tilde{\Gamma}_{\text{LMC}} = \tilde{\Gamma}_{\min}$ . We have therefore shown the minimum defined by Equation (2)  
 coincides with the minimum defined in Problem 6.  $\square$

### 337 5.3 Algorithm for Lexicographic Min Cut

338 In light of the new problem equivalency, we will study an algorithm solving Problem  
 339 6. As we will only consider the dual graph for this section, we leave behind the dual  
 340 chain notation: vertices  $\tilde{\tau}_1$  and  $\tilde{\tau}_2$  are replaced by  $\alpha_1$  and  $\alpha_2$ , and the solution to the  
 341 problem is simply noted  $\Gamma_{\text{LMC}}$ . The following lemma exposes a constructive property  
 342 of the solution on subgraphs.

343 **Lemma 5** Consider  $\Gamma_{\text{LMC}}$  solution of Problem 6 for the graph  $G = (\mathcal{V}, \mathcal{E})$  and  
 344  $\alpha_1, \alpha_2 \in \mathcal{V}$ . Let  $e_0$  be an edge in  $\mathcal{V} \times \mathcal{V}$  such that  $e_0 < \min\{e \in \mathcal{E}\}$ . Then:

345 – The solution for  $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$  is either  $\Gamma_{\text{LMC}}$  or  $\Gamma_{\text{LMC}} \cup \{e_0\}$ .

346 –  $\Gamma_{\text{LMC}} \cup \{e_0\}$  is solution for  $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$  if and only if  $\alpha_1$  and  $\alpha_2$  are connected  
 347 in  $(\mathcal{V}, \mathcal{E} \cup \{e_0\} \setminus \Gamma_{\text{LMC}})$ .

348 *Proof* Let's call  $\Gamma'_{\text{LMC}}$  the solution for  $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$ . Since  $\Gamma'_{\text{LMC}} \cap \mathcal{E}$  disconnects  
 349  $\alpha_1$  and  $\alpha_2$  in  $(\mathcal{V}, \mathcal{E})$ , one has  $\Gamma_{\text{LMC}} \sqsubseteq_{\text{lex}} \Gamma'_{\text{LMC}}$ . Since  $\Gamma_{\text{LMC}} \cup \{e_0\}$  disconnects  
 350  $\alpha_1$  and  $\alpha_2$  in  $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$ , we also have  $\Gamma'_{\text{LMC}} \sqsubseteq_{\text{lex}} \Gamma_{\text{LMC}} \cup \{e_0\}$ . Therefore,  
 351  $\Gamma_{\text{LMC}} \sqsubseteq_{\text{lex}} \Gamma'_{\text{LMC}} \sqsubseteq_{\text{lex}} \Gamma_{\text{LMC}} \cup \{e_0\}$ .

As  $e_0 < \min\{e \in \mathcal{E}\}$ , there is no set in  $\mathcal{E} \cup \{e_0\}$  strictly between  $\Gamma_{\text{LMC}}$  and  $\Gamma_{\text{LMC}} \cup \{e_0\}$  for the lexicographic order. It follows that  $\Gamma'_{\text{LMC}}$  is either equal to  $\Gamma_{\text{LMC}}$  or  $\Gamma_{\text{LMC}} \cup \{e_0\}$ . The choice for  $\Gamma'_{\text{LMC}}$  is therefore ruled by the property that it should disconnect  $\alpha_1$  and  $\alpha_2$ : if  $\alpha_1$  and  $\alpha_2$  are connected in  $(\mathcal{V}, \mathcal{E} \cup \{e_0\} \setminus \Gamma_{\text{LMC}})$ ,  $\Gamma_{\text{LMC}}$  does not disconnect  $\alpha_1$  and  $\alpha_2$  in  $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$  and  $\Gamma_{\text{LMC}} \cup \{e_0\}$  has to be the solution for  $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$ . On the other hand, if  $\alpha_1$  and  $\alpha_2$  are not connected in  $(\mathcal{V}, \mathcal{E} \cup \{e_0\} \setminus \Gamma_{\text{LMC}})$ , then both  $\Gamma_{\text{LMC}}$  and  $\Gamma_{\text{LMC}} \cup \{e_0\}$  disconnect  $\alpha_1$  and  $\alpha_2$  in  $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$ , but as  $\Gamma_{\text{LMC}} \sqsubset_{\text{lex}} \Gamma_{\text{LMC}} \cup \{e_0\}$ ,  $\Gamma_{\text{LMC}} \cup \{e_0\}$  is not the solution for  $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$ .  $\square$

352 Building an algorithm from Lemma 5 suggests a data structure able to check  
 353 if vertices  $\alpha_1$  and  $\alpha_2$  are connected in the graph: the disjoint-set data structure,  
 354 introduced for finding connected components [28], does exactly that. In this structure,  
 355 each set of elements has a different root value, called representative. Calling the  
 356 operation **MakeSet** on an element creates a new set containing this element. The  
 357 **FindSet** operation, given an element of a set, returns the representative of the set. For  
 358 all elements of the same set, **FindSet** will of course return the same representative.  
 359 Finally, the structure allows merging two sets, by using the **UnionSet** operation. After  
 360 this operation, elements of both sets will have the same representative.

361 We now describe Algorithm 5. The algorithm expects a set of edges sorted in  
 362 decreasing order according to the lexicographic order.

---

#### Algorithm 5: Lexicographic Min Cut

---

**Inputs:**  $G = (\mathcal{V}, \mathcal{E})$  and  $\alpha_1, \alpha_2 \in \mathcal{V}$ , with  $\mathcal{E} = \{e_i, i = 1, \dots, n\}$  in decreasing order  
**Output:**  $\Gamma_{\text{LMC}}$   
 $\Gamma_{\text{LMC}} \leftarrow \emptyset$   
**for**  $v \in \mathcal{V}$  **do**  
  | **MakeSet**( $v$ )  
**end**  
**for**  $e \in \mathcal{E}$  in decreasing order **do**  
  |  $e = (v_1, v_2) \in \mathcal{V} \times \mathcal{V}$   
  |  $r_1 \leftarrow \mathbf{FindSet}(v_1), r_2 \leftarrow \mathbf{FindSet}(v_2)$   
  |  $c_1 \leftarrow \mathbf{FindSet}(\alpha_1), c_2 \leftarrow \mathbf{FindSet}(\alpha_2)$   
  | **if**  $\{r_1, r_2\} = \{c_1, c_2\}$  **then**  
  |  |  $\Gamma_{\text{LMC}} \leftarrow \Gamma_{\text{LMC}} \cup e$   
  | **else**  
  |  | **UnionSet**( $r_1, r_2$ )  
  | **end**  
**end**

---

363 **Proposition 5** *Algorithm 5 computes the solution of Problem 6 for a given graph*  
 364  *$(\mathcal{V}, \mathcal{E})$  and two vertices  $\alpha_1, \alpha_2 \in \mathcal{V}$ . Assuming the input set of edges  $\mathcal{E}$  are sorted,*  
 365 *the algorithm has  $O(n\alpha(n))$  time complexity, where  $n$  is the cardinal of  $\mathcal{E}$  and  $\alpha$  the*  
 366 *inverse Ackermann function.*

367 *Proof* We denote by  $e_i$  the  $i^{\text{th}}$  edge along the decreasing order and  $\Gamma_{\text{LMC}}^i$  the value  
 368 of the variable  $\Gamma_{\text{LMC}}$  of the algorithm after iteration  $i$ . The algorithm works by  
 369 incrementally adding edges in decreasing order and tracking the growing connected  
 370 components of the set associated with  $\alpha_1$  and  $\alpha_2$  in  $(\mathcal{V}, \{e \in \mathcal{E}, e \geq e_i\} \setminus \Gamma_{\text{LMC}}^i)$ , for  
 371  $i = 1, \dots, n$ .

372 At the beginning, no edges are inserted, and  $\Gamma_{\text{LMC}}^0 = \emptyset$  is indeed solution for  
 373  $(\mathcal{V}, \emptyset)$ . With Lemma 5, we are guaranteed at each iteration  $i$  to find the solution for  
 374  $(\mathcal{V}, \{e \in \mathcal{E}, e \geq e_i\})$  by only adding to  $\Gamma_{\text{LMC}}^{i-1}$  the current edge  $e_i$  if  $\alpha_1$  and  $\alpha_2$  are  
 375 connected in  $\{e \in \mathcal{E}, e \geq e_i\} \setminus \Gamma_{\text{LMC}}^{i-1}$ , which is done in the if-statement. If the edge is  
 376 not added, we update the connectivity of the graph  $(\mathcal{V}, \{e \in \mathcal{E}, e \geq e_i\} \setminus \Gamma_{\text{LMC}}^i)$  by  
 377 merging the two sets represented by  $r_1$  and  $r_2$ . After each iteration,  $\Gamma_{\text{LMC}}^i$  is solution  
 378 for  $(\mathcal{V}, \{e \in \mathcal{E}, e \geq e_i\})$  and when all edges are processed,  $\Gamma_{\text{LMC}}^n$  is solution for  
 379  $(\mathcal{V}, \mathcal{E})$ .

The complexity of the **MakeSet**, **FindSet** and **UnionSet** operations have been  
 shown to be respectively  $O(1)$ ,  $O(\alpha(v))$  and  $O(\alpha(v))$ , where  $\alpha(v)$  is the inverse  
 Ackermann function on the cardinal of the vertex set [36]. Assuming sorted edges  
 as input of the algorithm – which is performed in  $O(n \ln(n))$ , the algorithm runs in  
 $O(n\alpha(n))$  time complexity.  $\square$

380 The similarity of Algorithm 5 with Kruskal’s algorithm for minimum spanning-  
 381 tree suggests an even better theoretical time complexity, by using Chazelle’s algorithm  
 382 [12] for minimum spanning-tree, running in  $O(n\alpha(n))$  complexity without requiring  
 383 sorted edges.

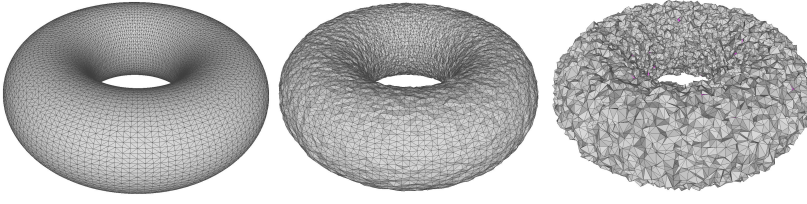
## 384 6 Application to point cloud triangulation

385 In all that precedes, the order on simplices was not specified and one can wonder if  
 386 choosing such an ordering makes the specialization of OHCP too restrictive for it to  
 387 be useful. In this section, we give a concrete example where this restriction makes  
 388 sense and provides a simple and elegant application to the problem of point cloud  
 389 triangulation. Whereas all that preceded dealt with an abstract simplicial complex,  
 390 we now consider a bijection between vertices and a set of points in Euclidean space,  
 391 allowing to compute geometric quantities on simplices.

### 392 6.1 Simplicial ordering

393 Recent works have studied a characterization of the 2D Delaunay triangulation as a  
 394 lexicographic minimum over 2-chains. Denote by  $R_{\text{B}}(\sigma)$  the radius of the smallest  
 395 enclosing ball and  $R_{\text{C}}(\sigma)$  the radius of the circumcircle of a 2-simplex  $\sigma$ . For an





**Fig. 2** Watertight reconstructions under different perturbations. Under small perturbations (first two images from the left), the reconstruction is a triangulation of the sampled manifold. A few non-manifold configurations appear however under larger perturbations (Rightmost image).

396 acute triangle  $\sigma$ ,  $R_B(\sigma) = R_C(\sigma)$ . Based on [18, 19], we define the total order on  
 397 2-simplices:

$$398 \quad \sigma_1 \leq \sigma_2 \iff \begin{cases} R_B(\sigma_1) < R_B(\sigma_2) \\ \text{or} \\ R_B(\sigma_1) = R_B(\sigma_2) \quad \text{and} \quad R_C(\sigma_1) \geq R_C(\sigma_2) \end{cases} \quad (4)$$

399 Under generic condition on the position of points, we can show this order is total. In  
 400 what follows, the lexicographic order  $\sqsubseteq_{lex}$  is induced by this order on simplices. The  
 401 following proposition from [19] shows a strong link between the simplex ordering  
 402 and the 2D Delaunay triangulation.

403 **Proposition 6 (Proposition 7.9 in [19])** *Let  $\mathbf{P} = \{P_1, \dots, P_N\} \subset \mathbb{R}^2$  with  $N \geq 3$  be*  
 404 *in general position and let  $K_{\mathbf{P}}$  be any 2-dimensional complex containing the Delaunay*  
 405 *triangulation of  $\mathbf{P}$ . Denote by  $\beta_{\mathbf{P}} \in C_1(K_{\mathbf{P}})$  the 1-chain made of edges belonging to*  
 406 *the boundary of convex hull  $CH(\mathbf{P})$ . If  $\Gamma_{\min} = \min_{\sqsubseteq_{lex}} \{\Gamma \in C_2(K_{\mathbf{P}}), \partial\Gamma = \beta_{\mathbf{P}}\}$ , the*  
 407 *simplicial complex  $|\Gamma_{\min}|$  support of  $\Gamma_{\min}$  is the Delaunay triangulation of  $\mathbf{P}$ .*

408 As the 2D Delaunay triangulation has some well-known optimality properties, such  
 409 as maximizing the minimal angle, we can hope that using the same order to minimize  
 410 2-chains in dimension 3 will keep some of those properties. In fact, it has been shown  
 411 that for a Čech or Vietoris-Rips complex, under strict conditions linking the point set  
 412 sampling, the parameter of the complex and the reach of the underlying manifold of  
 413 Euclidean space, the minimal lexicographic chain using the described simplex order  
 414 is a triangulation of the sampled manifold [18]. Experimental results (Figure 2) show  
 415 that this property remains true relatively far from these theoretical conditions.

## 416 6.2 Open surface triangulation

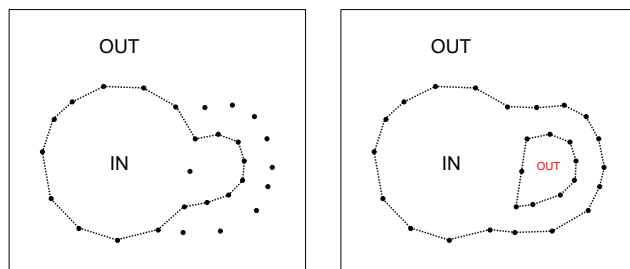
417 Given a point cloud sampling an open surface and a 1-cycle sampling the boundary of  
 418 the surface, we generate a Čech complex of the point cloud using the Phat library [5].  
 419 The parameter of the complex should be sufficient to capture the homotopy type  
 420 of the surface to reconstruct and should contain the provided boundary cycle. After  
 421 constructing the 2-boundary matrix, we apply the boundary reduction algorithm,  
 422 slightly modified to store the transformation matrix  $V$  (Section 4.2). We then apply

423 Algorithm 4 to find out if a 2-chain bounded by the cycle exists in the current Čech  
 424 complex. In this case, we get a chain bounded by the provided cycle and apply  
 425 Algorithm 3 to minimize the chain under imposed boundary. Otherwise, we might  
 426 have to increase the Čech parameter to capture the homotopy type of the surface to  
 427 reconstruct [4, 11]. Figure 1 shows results of this method.

### 428 6.3 Closed surface triangulation

429 Using Algorithm 5 requires a strongly connected 3-pseudomanifold: we therefore use  
 430 a 3D Delaunay triangulation, for its efficiency and non-parametric nature, using the  
 431 CGAL library [29], and complete it into a topological 3-sphere by connecting, for any  
 432 triangle on the convex hull of the Delaunay triangulation, its dual edge to an "infinite"  
 433 dual vertex.

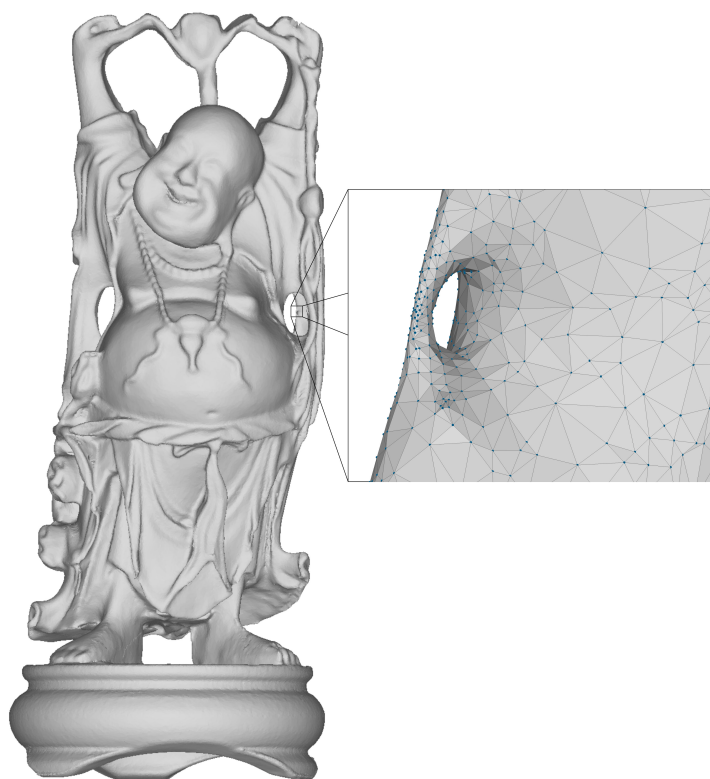
434 Experimentally, sorting triangles does not require exact predicates: the  $R_B$  and  
 435  $R_C$  quantities can simply be calculated in fixed precision. The quasilinear complexity  
 436 of Algorithm 5 makes it competitive in large point cloud applications. Outliers are  
 437 naturally ignored and, being parameter free, the algorithm adapts to non uniform point  
 438 densities, as seen in the closeup of Figure 4.



**Fig. 3** Providing additional topological information can improve the result of the reconstruction. Here, the lexicographic order on 1-chains is induced by edge length comparison.

439 The choice of  $\alpha_1$  and  $\alpha_2$  defines the location of the closed separating surface and  
 440 are chosen interactively. Although we could devise an algorithm where these inputs  
 441 are not required – the algorithm would simply merge regions until only two connected  
 442 components remain – this would only work for uniform and non-noisy point clouds  
 443 but not make for a robust algorithm. On the contrary, adding multiple interior and  
 444 exterior regions can guide the algorithm by providing better topological constraints,  
 445 as depicted in Figure 3.

446 Algorithm 5 requires to be slightly modified to take as input multiple  $\alpha_1, \alpha_2$ : after  
 447 creating all sets with MakeSet, we need to combine all  $\alpha_1$  sets together, and all  $\alpha_2$   
 448 sets together. The algorithm remains unchanged for the rest.



**Fig. 4** Closed surface triangulation of 440K points in 7.33 seconds. Beside the point cloud, the only user input is one inner tetrahedron. The closeup shows that small features are correctly recovered.

## References

- 450 1. Alliez, P., Cohen-Steiner, D., Yvinec, M., Desbrun, M.: Variational tetrahedral meshing. *ACM Trans. Graph.* **24**(3), 617–625 (2005). DOI 10.1145/1073204.1073238. URL <https://doi.org/10.1145/1073204.1073238>
- 451 2. Amenta, N., Bern, M.W., Kamvysselis, M.: A new voronoi-based surface reconstruction algorithm. In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1998*, Orlando, FL, USA, July 19-24, 1998, pp. 415–421 (1998). DOI 10.1145/280814.280947. URL <https://doi.org/10.1145/280814.280947>
- 452 3. Amenta, N., Choi, S., Dey, T.K., Leekha, N.: A simple algorithm for homeomorphic surface reconstruction. *Int. J. Comput. Geometry Appl.* **12**(1-2), 125–141 (2002). DOI 10.1142/S0218195902000773. URL <https://doi.org/10.1142/S0218195902000773>
- 453 4. Attali, D., Lieutier, A., Salinas, D.: Vietoris-rips complexes also provide topologically correct reconstructions of sampled shapes. *Comput. Geom.* **46**(4), 448–465 (2013). DOI 10.1016/j.comgeo.2012.02.009. URL <https://doi.org/10.1016/j.comgeo.2012.02.009>
- 454 5. Bauer, U., Kerber, M., Reininghaus, J., Wagner, H.: Phat – persistent homology algorithms toolbox. *Journal of Symbolic Computation* **78**, 76 – 90 (2017). DOI 10.1016/j.jsc.2016.03.008. *Algorithms and Software for Computational Topology*
- 455 6. Boissonnat, J.: Geometric structures for three-dimensional shape representation. *ACM Trans. Graph.* **3**(4), 266–286 (1984). DOI 10.1145/357346.357349. URL <https://doi.org/10.1145/357346.357349>
- 456
- 457
- 458
- 459
- 460
- 461
- 462
- 463
- 464
- 465
- 466
- 467
- 468

- 469 7. Boissonnat, J., Cazals, F.: Smooth surface reconstruction via natural neighbour interpolation of distance  
470 functions. In: Proceedings of the Sixteenth Annual Symposium on Computational Geometry, Clear  
471 Water Bay, Hong Kong, China, June 12-14, 2000, pp. 223–232 (2000). DOI 10.1145/336154.336208.  
472 URL <https://doi.org/10.1145/336154.336208>
- 473 8. Carvalho, J.F., Vejdemo-Johansson, M., Kragic, D., Pokorny, F.T.: An algorithm for calculating top-  
474 dimensional bounding chains. *PeerJ Computer Science* **4**, e153 (2018)
- 475 9. Chambers, E.W., Erickson, J., Nayyeri, A.: Minimum cuts and shortest homologous cycles. In:  
476 Proceedings of the 25th ACM Symposium on Computational Geometry, Aarhus, Denmark, June 8-10,  
477 2009, pp. 377–385 (2009). DOI 10.1145/1542362.1542426. URL [https://doi.org/10.1145/](https://doi.org/10.1145/1542362.1542426)  
478 [1542362.1542426](https://doi.org/10.1145/1542362.1542426)
- 479 10. Chambers, E.W., Vejdemo-Johansson, M.: Computing minimum area homologies. In: Computer  
480 graphics forum, vol. 34, pp. 13–21. Wiley Online Library (2015)
- 481 11. Chazal, F., Cohen-Steiner, D., Lieutier, A.: A sampling theory for compact sets in euclidean space.  
482 *Discrete & Computational Geometry* **41**(3), 461–479 (2009). DOI 10.1007/s00454-009-9144-8. URL  
483 <https://doi.org/10.1007/s00454-009-9144-8>
- 484 12. Chazelle, B.: A minimum spanning tree algorithm with inverse-ackermann type complexity. *Journal*  
485 *of the ACM (JACM)* **47**(6), 1028–1047 (2000)
- 486 13. Chen, C., Freedman, D.: Quantifying homology classes. *CoRR* **abs/0802.2865** (2008)
- 487 14. Chen, C., Freedman, D.: Measuring and computing natural generators for homology groups. *Comput.*  
488 *Geom.* **43**(2), 169–181 (2010). DOI 10.1016/j.comgeo.2009.06.004. URL [https://doi.org/10.](https://doi.org/10.1016/j.comgeo.2009.06.004)  
489 [1016/j.comgeo.2009.06.004](https://doi.org/10.1016/j.comgeo.2009.06.004)
- 490 15. Chen, C., Freedman, D.: Hardness results for homology localization. *Discrete & Computational*  
491 *Geometry* **45**(3), 425–448 (2011). DOI 10.1007/s00454-010-9322-8. URL [https://doi.org/10.](https://doi.org/10.1007/s00454-010-9322-8)  
492 [1007/s00454-010-9322-8](https://doi.org/10.1007/s00454-010-9322-8)
- 493 16. Chen, L.: Mesh smoothing schemes based on optimal delaunay triangulations. In: Proceedings of the  
494 13th International Meshing Roundtable, IMR 2004, Williamsburg, Virginia, USA, September 19-22,  
495 2004, pp. 109–120 (2004). URL <http://imr.sandia.gov/papers/abstracts/Ch317.html>
- 496 17. Chen, L., Holst, M.: Efficient mesh optimization schemes based on optimal delaunay triangulations.  
497 *Computer Methods in Applied Mechanics and Engineering* **200**(9), 967–984 (2011). DOI <https://doi.org/10.1016/j.cma.2010.11.007>
- 498 18. Cohen-Steiner, D., Lieutier, A., Vuillamy, J.: Lexicographic optimal chains and manifold triangulations.  
499 Research Report HAL: hal-02391190 (2019)
- 500 19. Cohen-Steiner, D., Lieutier, A., Vuillamy, J.: Regular triangulations as lexicographic optimal chains.  
501 Preprint HAL: hal-02391285 (2019)
- 502 20. Dey, T.K., Goswami, S.: Tight cocone: A water-tight surface reconstructor. *J. Comput. Inf. Sci. Eng.*  
503 **3**(4), 302–307 (2003)
- 504 21. Dey, T.K., Hirani, A.N., Krishnamoorthy, B.: Optimal homologous cycles, total unimodularity, and  
505 linear programming. *SIAM J. Comput.* **40**(4), 1026–1044 (2011). DOI 10.1137/100800245. URL  
506 <https://doi.org/10.1137/100800245>
- 507 22. Dey, T.K., Hou, T., Mandal, S.: Computing minimal persistent cycles: Polynomial and hard cases.  
508 *CoRR* **abs/1907.04889** (2019). URL <http://arxiv.org/abs/1907.04889>
- 509 23. Dey, T.K., Sun, J., Wang, Y.: Approximating loops in a shortest homology basis from point data. In:  
510 D.G. Kirkpatrick, J.S.B. Mitchell (eds.) Proceedings of the 26th ACM Symposium on Computational  
511 Geometry, Snowbird, Utah, USA, June 13-16, 2010, pp. 166–175. ACM (2010). DOI 10.1145/  
512 1810959.1810989. URL <https://doi.org/10.1145/1810959.1810989>
- 513 24. Edelsbrunner, H.: Surface Reconstruction by Wrapping Finite Sets in Space, pp. 379–404. Springer  
514 Berlin Heidelberg (2003). DOI 10.1007/978-3-642-55566-4\_17
- 515 25. Edelsbrunner, H., Harer, J.: Computational Topology - an Introduction. American Mathematical  
516 Society (2010). URL <http://www.ams.org/bookstore-getitem/item=MBK-69>
- 517 26. Edmonds, J., Karp, R.M.: Theoretical improvements in algorithmic efficiency for network flow prob-  
518 lems. *J. ACM* **19**(2), 248–264 (1972). DOI 10.1145/321694.321699. URL [https://doi.org/10.](https://doi.org/10.1145/321694.321699)  
519 [1145/321694.321699](https://doi.org/10.1145/321694.321699)
- 520 27. Erickson, J., Whittlesey, K.: Greedy optimal homotopy and homology generators. In: Proceedings  
521 of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver,  
522 British Columbia, Canada, January 23-25, 2005, pp. 1038–1046 (2005). URL [http://dl.acm.org/](http://dl.acm.org/citation.cfm?id=1070432.1070581)  
523 [citation.cfm?id=1070432.1070581](http://dl.acm.org/citation.cfm?id=1070432.1070581)
- 524 28. Galler, B.A., Fischer, M.J.: An improved equivalence algorithm. *Commun. ACM* **7**(5), 301–303  
525 (1964). DOI 10.1145/364099.364331. URL <https://doi.org/10.1145/364099.364331>  
526

- 527 29. Jamin, C., Pion, S., Teillaud, M.: 3D triangulations. In: CGAL User and Reference Manual, 5.0 edn.  
528 CGAL Editorial Board (2019). URL [https://doc.cgal.org/5.0/Manual/packages.html#](https://doc.cgal.org/5.0/Manual/packages.html#PkgTriangulation3)  
529 [PkgTriangulation3](https://doc.cgal.org/5.0/Manual/packages.html#PkgTriangulation3)
- 530 30. Kazhdan, M.M., Hoppe, H.: Screened poisson surface reconstruction. *ACM Trans. Graph.* **32**(3),  
531 29:1–29:13 (2013). DOI 10.1145/2487228.2487237. URL [https://doi.org/10.1145/2487228.](https://doi.org/10.1145/2487228.2487237)  
532 [2487237](https://doi.org/10.1145/2487228.2487237)
- 533 31. Labatut, P., Pons, J., Keriven, R.: Robust and efficient surface reconstruction from range data. *Comput.*  
534 *Graph. Forum* **28**(8), 2275–2290 (2009). DOI 10.1111/j.1467-8659.2009.01530.x
- 535 32. Lefebvre, S., Spagnuolo, M. (eds.): Eurographics 2014 - State of the Art Reports, Strasbourg, France,  
536 April 7-11, 2014. Eurographics Association (2014). URL [https://diglib.eg.org/handle/10.](https://diglib.eg.org/handle/10.2312/7707)  
537 [2312/7707](https://diglib.eg.org/handle/10.2312/7707)
- 538 33. Li, Y., Wu, X., Chrysanthou, Y., Sharf, A., Cohen-Or, D., Mitra, N.J.: Globfit: consistently fitting  
539 primitives by discovering global relations. *ACM Trans. Graph.* **30**(4), 52 (2011). DOI 10.1145/  
540 2010324.1964947. URL <https://doi.org/10.1145/2010324.1964947>
- 541 34. Orlin, J.B.: Max flows in  $o(nm)$  time, or better. In: Symposium on Theory of Computing Conference,  
542 STOC'13, Palo Alto, CA, USA, June 1-4, 2013, pp. 765–774 (2013). DOI 10.1145/2488608.2488705.  
543 URL <https://doi.org/10.1145/2488608.2488705>
- 544 35. Sullivan, J.M.: A crystalline approximation theorem for hypersurfaces. Ph.D. thesis, Princeton Univ.  
545 (1990). URL <http://torus.math.uiuc.edu/jms/Papers/thesis/>
- 546 36. Tarjan, R.E., van Leeuwen, J.: Worst-case analysis of set union algorithms. *J. ACM* **31**(2), 245–281  
547 (1984). DOI 10.1145/62.2160. URL <https://doi.org/10.1145/62.2160>
- 548 37. Wu, P., Chen, C., Wang, Y., Zhang, S., Yuan, C., Qian, Z., Metaxas, D., Axel, L.: Optimal topolog-  
549 ical cycles and their application in cardiac trabeculae restoration. In: International Conference on  
550 Information Processing in Medical Imaging, pp. 80–92. Springer (2017)