This is a pre-print of an article published in J Math Imaging Vis (2018).
The final authenticated version is available online at:
https://doi.org/10.1007/s10851-018-0843-8

Acyclic Partial Matchings for Multidimensional Persistence: Algorithm and Combinatorial Interpretation

Madjid Allili · Tomasz Kaczynski · Claudia Landi · Filippo Masoni

July 26, 2018

Abstract Given a simplicial complex and a vector-valued function on its vertices, we present an algorithmic construction of an acyclic partial matching on the cells of the complex compatible with the given function. This implies the construction can be used to build a reduced filtered complex with the same multidimensional persistent homology as of the original one filtered by the sublevel sets of the function. The correctness of the algorithm is proved and its complexity is analyzed. A combinatorial interpretation of our algorithm based on the concept of a multidimensional discrete Morse function is introduced for the first time in this paper. Numerical experiments show a substantial rate of reduction in the number of cells achieved by the algorithm.

Keywords Multidimensional persistent homology · Discrete Morse theory · Acyclic partial matchings · Matching algorithm · Reduced complex

The second author was partially supported by NSERC Canada Discovery Grant. The third author was supported by INdAM-GNSAGA.

M, Allili
Department of Computer Science, Bishop's University, Sherbrooke, QC, Canada J1M 1Z7
E-mail: mallili@ubishops.ca

T. Kaczynski
Département de mathématiques, Université de Sherbrooke, Sherbrooke, QC, Canada J1K 2R1
E-mail: tomasz.kaczynski@usherbrooke.ca

Claudia Landi
Dipartimento di Scienze e Metodi dell'Ingegneria, Università di Modena e Reggio Emilia, Reggio Emilia, Italy
E-mail: claudia.landi@unimore.it

Filippo Masoni
Dipartimento di Scienze e Metodi dell'Ingegneria, Università di Modena e Reggio Emilia, Reggio Emilia, Italy
E-mail: fpuro@libero.it

1 Introduction

Persistent homology has been introduced in 2002 by [12]. It extends size functions [15] which record changes in the number of connected components of sublevel sets of measuring functions, thus they can be related to 0-degree persistent homology. Since then, persistent homology has been developed by many authors as an important tool for the topological analysis of discrete data. We refer to surveys [11,6]. However, its effective computation remains a challenge due to the huge size of complexes built from data. Some recent works focussed on algorithms that reduce the original complexes generated from data to much smaller cellular complexes, homotopically equivalent to the initial ones by means of *acyclic partial matchings* of discrete Morse theory.

Although algorithms computing acyclic partial matchings have primarily been used for persistence of one-dimensional filtrations, see e.g. [19,24,21], there is currently a strong interest in combining persistence information coming from multiple functions in multiscale problems, e.g. to study photometric properties of textures in [5] or in biological applications [28], which motivates extensions to generalized types of persistence. The extension of persistent homology to multifiltrations is studied in [7] and is the one of interest in this paper. Other related directions are explored e.g. by the authors of [27] who do statistics on a set of one-dimensional persistence diagrams varied as coordinate system rotates, and in [13], where persistence modules on quiver complexes are studied.

Our attempt parallel to [13] is [2], where an algorithm given by King et al. in [19] is extended to multifiltrations. The algorithm produces a partition of the initial complex into three sets $(\mathsf{A},\mathsf{B},\mathsf{C})$ and an acyclic partial matching $\mathtt{m} : \mathsf{A} \to \mathsf{B}$. Any simplex which is not matched is added to C and defined as critical. The matching algorithm of [2] is used for reducing a simplicial complex to a smaller one by elimination of matched simplices in a way that is guaranteed to preserve multidimensional persistence. Reductions are derived from the works of [18,21,22].

First experiments on the algorithm of [2] with filtrations of triangular meshes show that there is a considerable amount of cells identified by the algorithm as critical but which seem to be spurious, in the sense that they appear in clusters of adjacent critical faces which do not seem to carry significant topological information.

The first contribution of this paper is a new algorithm that aims at improving our previous matching method [2] for optimality, in the sense of reducing the number of spurious critical cells, while still returning an acyclic partial matching $(\mathsf{A},\mathsf{B},\mathsf{C})$ that allows for multidimensional persistent homology preserving reductions. Our new matching algorithm extends the one given in [24] for cubical complexes, which processes lower stars rather than lower links. The major innovation of the matching algorithm presented here with respect to that of [24] emerges from the observation that, in the multidimensional setting, it is not enough to look at lower stars of vertices: one should take into consideration the lower stars of simplices of all dimensions, as there may be vertices of a simplex which are not comparable in the partial order of the multifiltration. Thus, the vector-valued function initially given on vertices of a simplicial complex is first extended to simplices of all dimensions. Then the algorithm processes the lower stars of all simplices, not only the vertices. The resulting acyclic

partial matching can be used as in [2] to construct a reduced filtered Lefschetz complex with the same multidimensional persistent homology as the original simplicial complex filtered by the sublevel sets of the function.

The second contribution of this paper is a combinatorial interpretation of the critical cells obtained by our new algorithm. As we said, until now, any simplex added to $\mathtt{C}$ by our algorithm has been defined as critical. It was legitimate to do so, because an application-driven extension of the Forman discrete Morse theory [14] to multidimensional functions has not been carried out yet. Hence, there exists no definition of a general combinatorial critical cell in this context. In the conference paper [3], we state this as an open problem and a subject for future work. At this time, we have the first step towards the appropriate extension of the Forman's theory. We propose new definitions of a multidimensional discrete Morse function (for short, `mdm` function), of its gradient field, its regular and critical cells. We next show that the function f used as input for our algorithm gives rise to an `mdm` function g with the same order of sublevel sets and the same partition $(\mathtt{A}, \mathtt{B}, \mathtt{C})$ as the one produced by the algorithm.

As a further contribution of this paper, we present experiments on synthetic data aimed at the geometric interpretation of the critical cells retrieved by our algorithm as Pareto critical points of multiple functions. As such, critical cells in this setting cannot be expected to be isolated but rather to form submanifolds. The fact that critical cells are located around the expected sets of Pareto critical points is an indication of improvement for optimality of our new algorithm. This improvement with respect to our previous algorithm is shown also by tests on real data sets. This is part of the new material added to this paper which is an extension of the work published in [3].

The paper is organized as follows. In Section 2, the preliminaries are introduced. In Section 3, the main Algorithm 2 is presented and its correctness is proved. Next, the complementing reduction method is recalled from [2]. At the section end, complexity of the algorithm is analyzed. In Section 4, we propose the new definition of an `mdm` function and provide combinatorial interpretation of the algorithm. In Section 5, experiments on synthetic and real 3D data are presented. In Section 6, we comment on open questions and prospects for future work.

2 Preliminaries

Let $\mathcal{K}$ be a finite geometric simplicial complex, that is a finite set composed of vertices, edges, triangles, and their q-dimensional counterparts, called simplices. A q-dimensional simplex is the convex hull of affinely independent vertices $v_0, \dots v_q \in \mathbb{R}^n$ and is denoted by $\sigma = [v_0, \dots v_q]$. We will sometimes denote this by $\sigma^{(q)}$ to make the dimension apparent in the notation. The set of q-simplices of $\mathcal{K}$ is denoted by $\mathcal{K}_q$. A *face* of a q-simplex $\sigma \in \mathcal{K}$ is a simplex τ whose vertices constitute a subset of $\{v_0, v_1, \dots, v_q\}$. If $\dim \tau = q - 1$, it is called a *facet* of σ. In this case, σ is called a *cofacet* of τ, and we write $\tau < \sigma$.

A *partial matching* $(\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{m})$ on $\mathcal{K}$ is a partition of $\mathcal{K}$ into three sets $\mathtt{A}, \mathtt{B}, \mathtt{C}$ together with a bijective map $\mathtt{m} : \mathtt{A} \to \mathtt{B}$, also called *discrete vector field*, such that, for each $\tau \in \mathtt{A}$, $\mathtt{m}(\tau)$ is a cofacet of τ. The intuition behind is that projection from τ to the complementing part of the boundary of $\mathtt{m}(\tau)$ induces a ho-

motopy equivalence between $\mathcal{K}$ and a smaller complex. An *$\mathtt{m}$-path* is a sequence $\sigma_0, \tau_0, \sigma_1, \tau_1, \ldots, \sigma_p, \tau_p, \sigma_{p+1}$ such that, for each $i = 0, \ldots, p$, $\sigma_{i+1} \neq \sigma_i$, $\tau_i = \mathtt{m}(\sigma_i)$, and τ_i is a cofacet of σ_{i+1}.

A partial matching $(\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathtt{m})$ on $\mathcal{K}$ is called *acyclic* if there does not exist a closed $\mathtt{m}$-path, that is an $\mathtt{m}$-path such that, $\sigma_{p+1} = \sigma_0$.

The main goal of this paper is to produce an acyclic partial matching which preserves the filtration of a simplicial complex $\mathcal{K}$ by sublevel sets of a vector-valued function $f : \mathcal{K}_0 \to \mathbb{R}^k$ given on the set of vertices of $\mathcal{K}$. We assume that $f : \mathcal{K}_0 \to \mathbb{R}^k$ is a function which is *component-wise injective*, that is, whose components f_i are injective. This assumption is used in Subsection 3.1 for proving correctness of the algorithm.

Given any function $\tilde{f} : \mathcal{K}_0 \to \mathbb{R}^k$, we can obtain a component-wise injective function f which is arbitrarily close to $\tilde{f}$ via the following procedure. Let n denote the cardinality of $\mathcal{K}_0$. For $i = 1, \ldots, k$, let us set $\eta_i = \min\{|\tilde{f}_i(v) - \tilde{f}_i(w)| : v, w \in \mathcal{K}_0 \wedge \tilde{f}_i(v) \neq \tilde{f}_i(w)\}$. For each i with $1 \leq i \leq k$, we can assume that the n vertices in $\mathcal{K}_0$ are indexed by an integer index j, with $1 \leq j \leq n$, increasing with $\tilde{f}_i$. Thus, the function $f_i : \mathcal{K}_0 \to \mathbb{R}$ can be defined by setting $f_i(v_j) = \tilde{f}_i(v_j) + j\eta_i/n^s$, with $s \geq 1$ (the larger s, the closer f to $\tilde{f}$). Finally, it is sufficient to set $f = (f_1, f_2, \ldots, f_k)$. We extend f to a function $f : \mathcal{K} \to \mathbb{R}^k$ as follows:

$$f(\sigma) = (f_1(\sigma), \ldots, f_k(\sigma)) \qquad \text{with} \qquad f_i(\sigma) = \max_{v \in \mathcal{K}_0(\sigma)} f_i(v). \tag{1}$$

Any function $f : \mathcal{K} \to \mathbb{R}^k$ that is an extension of a component-wise injective function $f : \mathcal{K}_0 \to \mathbb{R}^k$ defined on the vertices of the complex $\mathcal{K}$ in such a way that f satisfies equation (1) will be called *admissible*. In $\mathbb{R}^k$ we consider the following partial order. Given two values $a = (a_i), b = (b_i) \in \mathbb{R}^k$ we set $a \preceq b$ if and only if $a_i \leq b_i$ for every i with $1 \leq i \leq k$. Moreover we write $a \precneqq b$ whenever $a \preceq b$ and $a \neq b$. The *sublevel set filtration* of $\mathcal{K}$ induced by an admissible function f is the family $\{\mathcal{K}^a\}_{a \in \mathbb{R}^k}$ of subsets of $\mathcal{K}$ defined as follows:

$$\mathcal{K}^a = \{\sigma = [v_0, v_1, \ldots, v_q] \in \mathcal{K} \mid f(v_i) \preceq a,\ i = 0, \ldots, q\}.$$

It is clear that, for any parameter value $a \in \mathbb{R}^k$ and any simplex $\sigma \in \mathcal{K}^a$, all faces of σ are also in $\mathcal{K}^a$. Thus $\mathcal{K}^a$ is a simplical subcomplex of $\mathcal{K}$ for each a. The changes of topology of $\mathcal{K}^a$ as we change the multiparameter a permit recognizing some features of the shape of $|\mathcal{K}|$ if f is appropriately chosen. For this reason, the function f is called in the literature a *measuring function* or, more specifically, a *multidimensional measuring function* [4]. The *lower star* of a simplex is the set

$$L(\sigma) = \{\alpha \in \mathcal{K} \mid \sigma \text{ face of } \alpha \quad \text{and} \quad f(\alpha) \preceq f(\sigma)\},$$

and the *strict lower star* is the set $L_*(\sigma) = L(\sigma) \setminus \{\sigma\}$.

2.1 Indexing Map

An *indexing map* on the simplices of the complex $\mathcal{K}$ of cardinality N, compatible with an admissible function f, is a bijective map $I : \mathcal{K} \to \{1, 2, \ldots, N\}$ such that, for each $\sigma, \tau \in \mathcal{K}$ with $\sigma \neq \tau$, if σ is a face of τ or $f(\sigma) \precneqq f(\tau)$ then $I(\sigma) < I(\tau)$.

To build an indexing map I on the simplices of the complex $\mathcal{K}$, we will revisit the algorithm introduced in [2] that uses the topological sorting of a Directed Acyclic Graph (DAG) to build an indexing for vertices of a complex that is compatible with the ordering of values of a given function defined on the vertices. We will extend the algorithm to build an indexing for all cells of a complex that is compatible with both the ordering of values of a given admissible function defined on the cells and the ordering of the dimensions of the cells.

We recall that a topological sorting of a directed graph is a linear ordering of its nodes such that for every directed edge (u, v) from node u to node v, u precedes v in the ordering. This ordering is possible if and only if the graph has no directed cycles, that is, if it is a DAG. A simple well known algorithm (see [25]) for this task consists of successively finding nodes of the DAG that have no incoming edges and placing them in a list for the final sorting. Note that at least one such node must exist in a DAG, otherwise the graph must have at least one directed cycle.

Algorithm 1 Topological sorting

```
1: Input: A DAG whose list of nodes with no incoming edges is I
2: Output: The list L containing the sorted nodes
3: while there are nodes remaining in I do
4:     remove a node u from I
5:     add u to L
6:     for each node v with an edge e from u to v  do
7:         remove edge e from the DAG
8:         if v has no other incoming edges then
9:             insert v into I
10:        end if
11:    end for
12: end while
```

When the graph is a DAG, there exists at least one solution for the sorting problem, which is not necessarily unique. We can easily see that each node and each edge of the DAG is visited once by the algorithm, therefore its running time is linear in the number of nodes plus the number of edges in the DAG.

We can prove the following lemma which builds an indexing map by means of a topological sorting on a DAG given by the Hasse diagram of a suitable partial ordering on the simplicial complex $\mathcal{K}$.

Lemma 1 *Let $f : \mathcal{K} \to \mathbb{R}^k$ be an admissible function. There exists an injective function $I : \mathcal{K} \to \mathbb{N}$ such that, for each $\sigma, \tau \in \mathcal{K}$ with $\sigma \neq \tau$, if σ is a face of τ or $f(\sigma) \preceq f(\tau)$ then $I(\sigma) < I(\tau)$.*

Proof The set $\mathcal{K}$ is partially ordered by the following relation: $\sigma \sqsubseteq \tau$ if and only if either $\sigma = \tau$ or $\sigma \neq \tau$ and, in the latter case, σ is a face of τ or $f(\sigma) \preceq f(\tau)$. Indeed, it can be straightforwardly checked that this relation is reflexive, antisymmetric and transitive. Hence $(\mathcal{K}, \sqsubseteq)$ can be represented in a directed graph by its Hasse diagram that is acyclic.

The topological sorting Algorithm 1 allows us to sort and store the simplices in $\mathcal{K}$ in an array L of size N, with indexes that can be chosen from 1 to N. It follows that

the map $I : \mathcal{K} \to \{1, 2, \ldots, N\}$ that associates to every node its index in the array L is bijective. Moreover, and due to the topological sorting, I satisfies the constraint that for $\sigma, \tau \in \mathcal{K}$ with $\sigma \neq \tau$, if σ is a face of τ or $f(\sigma) \precneqq f(\tau)$, then $I(\sigma) < I(\tau)$. □

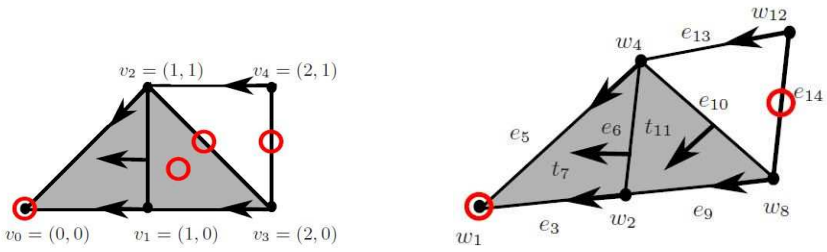
3 Matching Algorithm

The first contribution of this paper is the Matching Algorithm 2.

Algorithm 2 Matching

Input: A finite simplicial complex $\mathcal{K}$ with an admissible function $f : \mathcal{K} \to \mathbb{R}^k$ and an indexing map $I : \mathcal{K} \to \{1, 2, \ldots, N\}$ on its simplices compatible with f.
Output: Three lists A, B, C of simplices of $\mathcal{K}$, and a function m : A → B.
for $i = 1$ to N **do**
  $\sigma := I^{-1}(i)$
  if classified(σ)=**false then**
    if $L_*(\sigma)$ contains no cells **then**
      add σ to C, classified(σ)=**true**
    else
      $\delta :=$ the cofacet in $L_*(\sigma)$ of minimal index $I(\delta)$
      add σ to A and δ to B and define m(σ) = δ, classified(σ)=**true**, classified(δ)=**true**
      add all $\alpha \in L_*(\sigma) - \{\delta\}$ with $\text{num\_unclass\_facets}_\sigma(\alpha) = 0$ to PQzero
      add all $\alpha \in L_*(\sigma)$ with $\text{num\_unclass\_facets}_\sigma(\alpha) = 1$ and $\alpha > \delta$ to PQone
      while PQone $\neq \emptyset$ or PQzero $\neq \emptyset$ **do**
        while PQone $\neq \emptyset$ **do**
          $\alpha :=$ PQone.pop_front
          if $\text{num\_unclass\_facets}_\sigma(\alpha) = 0$ **then**
            add α to PQzero
          else
            add $\lambda \in \text{unclass\_facets}_\sigma(\alpha)$ to A, add α to B and define m(λ) = α, classified(α)=**true**, classified(λ)=**true**
            remove λ from PQzero
            add all $\beta \in L_*(\sigma)$ with $\text{num\_unclass\_facets}_\sigma(\beta) = 1$ and either $\beta > \alpha$ or $\beta > \lambda$ to PQone
          end if
        end while
        if PQzero $\neq \emptyset$ **then**
          $\gamma :=$ PQzero.pop_front
          add γ to C, classified(γ)=**true**
          add all $\tau \in L_*(\sigma)$ with $\text{num\_unclass\_facets}_\sigma(\tau) = 1$ and $\tau > \gamma$ to PQone
        end if
      end while
    end if
  end if
end for

It uses as input a finite simplicial complex $\mathcal{K}$ of cardinality N, an admissible function $f : \mathcal{K} \to \mathbb{R}^k$ built from a component-wise injective function $f : \mathcal{K}_0 \to \mathbb{R}^k$ using the extension formula given in equation (1), and an indexing map I compatible with f. It can be precomputed using the topological sorting in Algorithm 1 and explained



(a) (b)

Fig. 1 In (a), the complex and output of Algorithm 3.3 of [2] are displayed. Gray-shaded triangles are those which are present in the simplicial complex. Critical simplices are marked by red circles and the matched simplices are marked by arrows. In (b), the complex is modified so to satisfy the coordinate-wise injectivity assumption. Labeling of all simplices by the indexing function and the output of Algorithm 2 are displayed.

in the proof of Lemma 1. Given a simplex σ, we use $\texttt{unclass\_facets}_\sigma(\alpha)$ to denote the set of facets of a simplex α that are in $L(\sigma)$ and have not been classified yet, that is, not inserted in either A, B, or C, and $\texttt{num\_unclass\_facets}_\sigma(\alpha)$ to denote the cardinality of $\texttt{unclass\_facets}_\sigma(\alpha)$. We initialize $\texttt{classified}(\sigma)$=**false** for every $\sigma \in \mathcal{K}$. We use priority queues PQzero and PQone which store candidates for pairings with zero and one unclassified facets respectively in the order given by I. We initialize both as empty sets. The algorithm processes cells in the increasing order of their indexes. Each cell σ can be set to the states of $\texttt{classified}(\sigma)$=**true** or $\texttt{classified}(\sigma)$=**false** so that if it is processed as part of a lower star of another cell it is not processed again by the algorithm. The algorithm makes use of extra routines to calculate the cells in the lower star $L(\sigma)$ and the set of unclassified facets $\texttt{unclass\_facets}_\sigma(\alpha)$ of α in $L_*(\sigma)$ for each cell $\sigma \in \mathcal{K}$ and each cell $\alpha \in L_*(\sigma)$.

The goal of the process is to build a partition of $\mathcal{K}$ into three lists A, B, and C where C is the list of critical cells and in which each cell in A is paired in a one-to-one manner with a cell in B which defines a bijective map $\texttt{m} : \texttt{A} \to \texttt{B}$. When a cell σ is considered, each cell in its lower star $L(\sigma)$ is processed exactly once. The cell σ is inserted into the list of critical cells C if $L_*(\sigma) = \emptyset$. Otherwise, σ is paired with the cofacet $\delta \in L_*(\sigma)$ that has minimal index value $I(\delta)$. The algorithm makes additional pairings which can be interpreted topologically as the process of constructing $L_*(\sigma)$ with simple homotopy expansions or the process of reducing $L_*(\sigma)$ with simple homotopy contractions. When no pairing is possible a cell is classified as critical and the process is continued from that cell. A cell α is candidate for a pairing when $\texttt{unclass\_facets}_\sigma(\alpha)$ contains exactly one element λ that belongs to PQzero. For this purpose, the priority queues PQzero and PQone which store cells with zero and one available unclassified faces respectively are created. As long as PQone is not empty, its front is popped and either inserted into PQzero or paired with its single available unclassified face. When PQone becomes empty, the front cell of PQzero is declared as critical and inserted in C.

We illustrate the algorithm by a simple example. We use the simplicial complex S from our first paper [2, Figure 2] to compare the outputs of the previous matching algorithm and the new one. Figure 1(a) displays S and the output of [2, Algorithm 3.3]. The coordinates of vertices are the values of the function considered in [2]. Since that function is not component-wise injective, we denote it by $\tilde{f}$ and we start from constructing a component-wise injective approximation f discussed at the beginning of Section 2. If we interpret the passage from $\tilde{f}$ to f as a displacement of the coordinates of vertices, the new complex $\mathcal{K}$ is illustrated by Figure 1(b). The partial order relation is preserved when passing from $\tilde{f}$ to f, and the indexing of vertices in [2, Figure 2] may be kept for f. Hence, it is easy to see that [2, Algorithm 3.3] applied to $\mathcal{K}$ gives the same result as that displayed in Figure 1(a). In order to apply our new Algorithm 2, we need to index all 14 simplices of $\mathcal{K}$. For convenience of presentation, we label the vertices w_i, edges e_i, and triangles t_i by the index values $i = 1, 2, ..., 14$. The result is displayed in Figure 1(b). The sequence of vertices $(v_0, v_1, v_2, v_3, v_4)$ is replaced by $(w_1, w_2, w_4, w_8, w_{12})$.

Here are the main steps of the algorithm:

$i = 1$	$L_*(w_1) = \emptyset$, $w_1 \in$ C
$i = 2$	$L_*(w_2) = \{e_3\}$, $\mathtt{m}(w_2) = e_3$.
$i = 3$	e_3 classified.
$i = 4$	$L_*(w_4) = \{e_5, e_6, t_7\}$, $\mathtt{m}(w_4) = e_5$, $e_6 \in$ PQzero, $t_7 \in$ PQone, line 15: $\alpha = t_7$ leaves PQone, line 19: $\lambda = e_6$, $\mathtt{m}(e_6) = t_7$, e_6 leaves PQzero.
$i = 5, 6, 7$	e_5, e_6, t_7 classified.
$i = 8$	$L_*(w_8) = \{e_9\}$, $\mathtt{m}(w_8) = e_9$.
$i = 9$	e_9 classified.
$i = 10$	$L_*(e_{10}) = \{t_{11}\}$, $\mathtt{m}(e_{10}) = t_{11}$.
$i = 11$	t_{11} classified.
$i = 12$	$L_*(w_{12}) = \{e_{13}, e_{14}\}$, $\mathtt{m}(w_{12}) = e_{13}$, $e_{14} \in$ PQzero, PQone $= \emptyset$, line 25: $\gamma = e_{14} \in$ C.
$i = 13, 14$	e_{13}, e_{14} classified.

The output is displayed in Figure 1(b). As it can be noticed in the example of Figure 1, Algorithm 2 processes all the simplices, and permits to perform all the expected matchings, since it is based on lower stars of simplices. In contrast, Algorithm 3.3 of [2], being vertex-based, does not process simplices e_{10} and t_{11} as they do not belong to the lower star of any vertex. Thus, e_{10} and t_{11} are residually classified by Algorithm 3.3 of [2] as critical at the end of the whole process. This phenomenon is to be expected any time a simplex does not belong to the lower star of any vertex. The more numerous are the non-comparable values of the function on the vertices, the more numerous simplices are residually classified as critical by Algorithm 3.3 of [2], while they have the chance of being matched by our present Algorithm 2. That this is really the case in practical situations can be checked in Table 1.

3.1 Correctness

We now prove that our algorithm is correct in the sense that it always terminates, and produces an acyclic partial matching compatible with the filtration by sublevel sets induced by the input function.

Recall that $f = (f_1, \ldots, f_k) : \mathcal{K}_0 \to \mathbb{R}^k$ is a function whose components f_i are injective on the vertices of $\mathcal{K}$; moreover, f is extended to $f = (f_1, \ldots, f_k) : \mathcal{K} \to \mathbb{R}^k$ defined on cells σ of any dimension by using formula (1). The assumption that f is component-wise injective on the vertices is not sufficient to obtain disjoint lower stars, but when two lower stars meet, then they get classified at the same time. This is expressed by the following statements.

Lemma 2 *The following statements hold:*

(1) If $\tau \in L(\sigma)$, then $f(\tau) = f(\sigma)$.
(2) If $\tau \in L_*(\sigma)$, then $I(\sigma) < I(\tau)$.
(3) If $f(\sigma) = f(\tau)$ then there exists α face of $\sigma \cap \tau$ with $f(\alpha) = f(\sigma) = f(\tau)$.
(4) Assume that σ_1 and σ_2 are two simplices of $\mathcal{K}$ such that $L(\sigma_1) \cap L(\sigma_2) \neq \emptyset$. Then, there exists a simplex $\beta \in \mathcal{K}$ such that $L(\sigma_1) \cup L(\sigma_2) \subseteq L(\beta)$ and $I(\beta) \leq \min\{I(\sigma_1), I(\sigma_2)\}$.

Proof (1) If $\tau \in L(\sigma)$, then $f(\tau) \preceq f(\sigma)$ by definition of lower star. On the other hand, since σ is a face of τ, by definition of f, $f(\sigma) \preceq f(\tau)$. Thus $f(\sigma) = f(\tau)$.
(2) If $\tau \in L_*(\sigma)$, then σ is a face of τ and the conclusion follows from the definition of the indexing map.
(3) If $f(\sigma) = f(\tau)$, then, for every i, $\max_{v \in \mathcal{K}_0(\sigma)} f_i(v) = \max_{v \in \mathcal{K}_0(\tau)} f_i(v)$. By the injectivity of f_i, the two maxima must be attained at the same vertex. Therefore σ and τ have a common face.
(4) If there exists a simplex $\gamma \in L(\sigma_1) \cap L(\sigma_2)$, then we get $f(\gamma) = f(\sigma_1) = f(\sigma_2)$ from (1). By (3), there exists a simplex β face of $\sigma_1 \cap \sigma_2$ such that $f(\beta) = f(\sigma_1) = f(\sigma_2)$. It is now clear that for any $\delta \in L(\sigma_1) \cup L(\sigma_2)$, β face of δ and $f(\delta) = f(\sigma_1) = f(\sigma_2) = f(\beta)$, thus $\delta \in L(\beta)$. By (2), $I(\beta) \leq \min\{I(\sigma_1), I(\sigma_2)\}$. □

In the next two lemmas we show that, if a cell σ is unclassified when Algorithm 2 reaches line 5, then either $L_*(\sigma)$ is empty for which case σ is classified as critical, or $L_*(\sigma)$ contains at least one cofacet of σ that has σ as a unique facet in $L(\sigma)$. Then σ is paired with the cofacet with minimal index and the remainder of its cofacets in $L_*(\sigma)$ have no unclassified facets in $L_*(\sigma)$ and hence they must enter `PQzero` at line 11 of Algorithm 2. Moreover, if for every cell α with $I(\alpha) < I(\sigma)$, $L(\alpha)$ consists only of classified cells, then all cells in $L(\sigma)$ are also unclassified.

Lemma 3 *Assume that σ is a cell in $\mathcal{K}$. If $\alpha \in L_*(\sigma)$ is a cofacet of σ then, at any stage of the algorithm, $\texttt{num\_unclass\_facets}_\sigma(\alpha) \leq 1$, and it is equal to 1 if and only if σ is still unclassified. In this case, the unclassified face of α is exactly σ.*

Proof Let us assume that $\texttt{num\_unclass\_facets}_\sigma(\alpha) \geq 1$. For any unclassified face γ of α such that $\gamma \in L(\sigma)$, it holds that $\dim \gamma = \dim \sigma$. Indeed, $\dim \gamma < \dim \alpha = \dim \sigma + 1$, and $\dim \gamma \geq \dim \sigma$ because $\gamma \in L(\sigma)$. Thus, if $\gamma \neq \sigma$,

the assumption $\gamma \in L(\sigma)$ is contradicted. As a consequence, if $\gamma \neq \sigma$ for all γ, $\texttt{num\_unclass\_facets}_\sigma(\alpha) = 0$; if $\gamma = \sigma$, then $\texttt{num\_unclass\_facets}_\sigma(\alpha) = 1$. □

Lemma 4 *Assume that $\sigma \in \mathcal{K}$ is unclassified when Algorithm 2 reaches line 5 for $i = I(\sigma)$, and that* `classified`(γ)=**true** *for all simplexes $\beta \in \mathcal{K}$ with $I(\beta) < I(\sigma)$ and all cells $\gamma \in L(\beta)$. Then the following statements hold true:*

(i) All simplexes in $L(\sigma)$ are also unclassified at step 5 when $i = I(\sigma)$;
(ii) If Algorithm 2 gets to line 9, then there exists at least one cofacet of σ. Moreover, the one with minimal index, say δ, has exactly σ as unclassified facet, and it is still unclassified. Thus σ and δ get classified at line 10.
(iii) If $\alpha \in L_*(\sigma)$ and $\texttt{num\_unclass\_facets}_\sigma(\alpha) = 0$ at line 11 of Algorithm 2, then α is a facet of σ.

Proof (i) If $L(\sigma) = \{\sigma\}$ the claim is true by assumption. Let us assume that σ has at least one coface $\alpha \in L_*(\sigma)$. If α is classified it belongs to the lower star of another cell β different from σ with $I(\beta) < I(\sigma)$. By Lemma 2(4), also σ belongs to $L(\beta)$ and, therefore, σ is already classified by assumption. This gives a contradiction. Hence α is not classified.

(ii) If σ had no cofaces, then $L_*(\sigma)$ would be empty. Therefore line 9 would not be reached, contradicting the hypothesis. So σ has at least one coface $\alpha \in L_*(\sigma)$. By Lemma 2(1), $f(\alpha) = f(\sigma)$. Assuming $\dim \sigma = p$ and $\dim \alpha = p + r$, there exists a sequence of simplices $\alpha_1, \ldots \alpha_{r-1}$ of dimensions $p + 1, \ldots, p + r - 1$ such that

$$\sigma < \alpha_1 < \alpha_2 < \ldots < \alpha_{r-1} < \alpha.$$

By definition of f, $f(\sigma) \preceq f(\alpha_h) \preceq f(\alpha)$ for $h = 1, \ldots r - 1$. Recalling that $f(\alpha) = f(\sigma)$ we see that $f(\alpha_1) = \ldots = f(\alpha_{r-1}) = f(\sigma)$. Thus $\alpha_h \in L_*(\sigma)$ for $h = 1, \ldots r - 1$. In particular, α_1 is a cofacet of σ that belongs to $L_*(\sigma)$. Every cofacet of σ in $L_*(\sigma)$ has only σ as unclassified facet in $L(\sigma)$ by Lemma 3. Let δ be the cofacet of σ with minimal index. Statement *(i)* implies that δ is still unclassified.

(iii) Let $\dim \sigma = p$ and $\dim \alpha = p + r$. If $r > 1$ then there are at least two sequences $\sigma < \alpha_1 < \ldots < \alpha_{r-1} < \alpha$ and $\sigma < \alpha'_1 < \ldots < \alpha'_{r-1} < \alpha$ of cells belonging to $L(\sigma)$ with $\alpha_{r-1} \neq \alpha'_{r-1}$. These cells α_{r-1} and α'_{r-1} need to be already classified at line 11 because of the assumption $\texttt{num\_unclass\_facets}_\sigma(\alpha) = 0$. By *(i)*, they had not been classified when $i < I(\sigma)$. Since we are at line 11, it has necessarily occurred when $i = I(\sigma)$ at line 9. But the coface δ of σ with minimal index is unique so only one between α_{r-1} and α'_{r-1} has been classified at line 9, giving a contradiction. Thus, $r = 1$.

□

We shall prove by induction on the index of the cells that every cell is classified in a unique fashion by the algorithm. The proof is simple when the index takes values 1 and 2 since the cells can be only vertices or edges:

Lemma 5 *Let $\sigma \in \mathcal{K}$ such that $I(\sigma) = i$ with $i \leq 2$. Then, σ is a vertex. Moreover,*

1. *if $i = 1$, then Algorithm 2 classifies σ as critical at line 7;*
2. *if $i = 2$, then either $L(\sigma) = \{\sigma\}$ or $L(\sigma) = \{\sigma, \delta\}$ where δ is an edge whose vertices are the cells with indexes 1 and 2. Moreover, if $L(\sigma) = \{\sigma\}$, then σ is classified as critical at line 7; if $L(\sigma) = \{\sigma, \delta\}$, then δ is an edge and σ is paired with δ at line 9.*

Proof If $I(\sigma) \leq 2$, then σ needs to be a vertex. Indeed, if we assume $\dim \sigma \geq 1$, then it can be written as $\sigma = \langle v_0, v_1, \ldots, v_k \rangle$ with $k \geq 1$. It follows that σ has at least two faces of lower dimension and lower value by f which should also have lower indexes than that of σ. This contradicts the fact that $I(\sigma) \leq 2$.

Let us now prove separately statements 1 and 2.

1. We note that `classified`(σ)=**false** at line 5 because of the initialization. Moreover, $L_*(\sigma)$ is empty. To see this, let us observe that, for any coface γ of σ, it must hold that $f_i(\sigma) < f_i(\gamma)$ for at least one index $i = 1, \ldots, k$. Indeed, σ is a vertex of γ and at any other vertex of γ the value of f_i must be greater than $f_i(\sigma)$ because f_i is injective and σ has minimal index. Hence σ gets classified at line 7 and there is no other cell in $L(\sigma)$ to classify.

2. If $\alpha \in L_*(\sigma)$, then all the vertices in α other than σ should have f values lower than σ. They should therefore have lower indexes too. The only possibility left is to have $\alpha = \langle v, w \rangle$ where $I(v) = 1$ and $I(w) = 2$. □

For the general index, we first prove the following property:

Lemma 6 *Let $\alpha \in L_*(\sigma)$ be such that when it is popped from `PQone` at line 15 of Algorithm 2, `unclass_facets`$_\sigma(\alpha)$ is a singleton $\{\lambda\}$. Then λ belongs to `PQzero`. Therefore all cells α popped out from `PQone` at line 15 of Algorithm 2 for which this condition holds get paired at line 19.*

Proof We reason by induction on $r \geq 2$ where $\dim \alpha = p + r$. Note that for $r = 1$, α is a cofacet of σ with 0 unclassified faces in $L(\sigma)$ after step 9 is executed. Therefore α cannot enter `PQone`. For $r = 2$, λ is a primary facet of σ with 0 unclassified faces in $L(\sigma)$ after step 9 is executed. Therefore $\lambda \in$ `PQzero`. Assume by induction that for each natural number j from 2 up to value $r - 1$, when α with $\dim \alpha = p + j$ is popped from `PQone` with `num_unclass_facets`$_\sigma(\alpha)$ = 1, its unique unclassified face λ belongs to `PQzero` and therefore α and λ get paired at line 19. Let now $j = r$, and $\dim \alpha = p + j$. Let us assume that λ is not in `PQzero`. Then there are two cases. If λ has entered `PQone`, then it has been processed before α. Since λ is not in `PQzero`, by the induction hypothesis, it must have been paired with some cell in `PQzero`. This is a contradiction to the statement `num_unclass_facets`$_\sigma(\alpha)$ = 1. If λ did not enter `PQone`, then the number of unclassified faces of λ in $L(\sigma)$ is greater than or equal to 1. Thus, since λ is of dimension $p + r - 1$, there must exist a face $\tau^{(p+r-2)}$ of λ of dimension $p + r - 2$ in $L_*(\sigma)$ that is not paired and not added to `C`. This process can be carried out until we get a $(p + 1)$-cell $\tau^{(p+1)}$ in $L_*(\sigma)$ that is not classified by the algorithm. In general, we get sequences in $L(\sigma)$ such that

$$\sigma^{(p)} < \tau^{(p+1)} < \tau^{(p+2)} < \ldots < \tau^{(p+r-2)} < \lambda^{(p+r-1)} < \alpha^{(p+r+2)}$$

with $\tau^{(p+1)}$ not classified by the algorithm. By Lemma 3 the number of unclassified faces of $\tau^{(p+1)}$ is 0, implying that $\tau^{(p+1)}$ has entered `PQzero`. Let us fix the

sequence for which $\tau^{(p+2)}$ is of minimal index and has only one unclassified face, hence it has entered PQone before α. It exists because δ has been classified and $\mathcal{K}$ is a simplicial complex. We deduce that $\tau^{(p+1)}$ and $\tau^{(p+2)}$ have been paired, contradicting the assumption that $\tau^{(p+1)}$ has not been classified by the algorithm. Hence, λ should belong to PQzero, which completes the proof. □

We next state that each cell that is still unclassified after line 10 of the algorithm ultimately enters PQone or PQzero and gets classified. This requires an argument based on the dimension of the cell and the number of its unclassified faces in the considered lower star. Moreover, we prove that if a cell is already classified, it cannot be considered again for classification or enter the priority queues PQone or PQzero.

Lemma 7 *Let $\sigma \in \mathcal{K}$. Each cell in $L(\sigma)$ is processed exactly once by the algorithm and it is paired with some other cell or classified as critical. Hence Algorithm 2 classifies all cells of $\mathcal{K}$ and always terminates.*

Proof We break the conclusion into three statements:

(a) Each cell in the lower star eventually enters PQone or PQzero.
(b) Each cell that has entered PQone or PQzero is eventually classified.
(c) A cell that has already been classified cannot enter PQone or PQzero again.

We simultaneously prove the three statements by induction on $i = I(\sigma)$. For $i = 1, 2$ the claim is proved by Lemma 5. Let us now assume by induction that the claim is true from 2 up to $i - 1$. Let $I(\sigma) = i$. If classified(σ)=**true**, then σ has already been classified as part of $L(\beta)$ for some cell β that is a face of σ. Thus, $I(\beta) < I(\sigma) = i$ and $L(\sigma) \subset L(\beta)$. By induction hypothesis, every cell of $L(\sigma)$ is processed once by the algorithm and it is paired with some other cell or classified as critical. If classified(σ)=**false**, σ is either declared critical at line 7 or, by Lemma 4*(ii)*, paired with some other cell δ in $L_*(\sigma)$ at line 10. The cells σ and δ are no further processed.

Let γ be a cell left in $L_*(\sigma)$, if any. Suppose that

$$\texttt{num\_unclass\_facets}_\sigma(\gamma) \leq 1. \tag{2}$$

Then γ is either added to PQzero or to PQone and it is ultimately either paired or classified as critical. More precisely, if γ is added to PQone, then it is either moved to PQzero at line 17, or paired at line 19 by Lemma 6. If γ is added to PQzero, it is either paired at line 20 or declared critical at line 26. This also shows that, when $i = I(\sigma)$, every cell in $L_*(\sigma)$ enters at most once in PQzero and PQone.

It remains to show that (a) also holds for cells γ with

$$\texttt{num\_unclass\_facets}_\sigma(\gamma) \geq 2. \tag{3}$$

We prove (a) by induction on the dimension n of cells in $L_*(\sigma)$. The initial step is $\dim \gamma = \dim \sigma + 1$. But then γ is a cofacet of σ and, by Lemma 3, (2) holds. Assume by induction that all cells of $L_*(\sigma)$ with dimension smaller than n have entered either PQzero or PQone.

Let γ be a cell of dimension n in $L_*(\sigma)$. If (2) holds, we are done. Suppose that (3) holds. We show that γ eventually enters PQone. By induction, all faces of γ

eventually enter PQzero or PQone. We have earlier shown that those which enter PQone are classified or moved to PQzero. So all faces of γ which are not classified enter PQzero. All such faces which have a coface in PQone or get a coface in PQone at line 21 are classified at line 19-20. We remain with the faces of γ which are in PQzero but have no coface in PQone. Let r be the number of such faces. Necessarily $r > 1$, otherwise γ is in PQone. At lines 25-26 one of those faces is classified as critical, so we remain with $r - 1$ such faces. After passing other $r - 2$ times through lines 25-26, γ remains with only one unclassified face and it is added to PQone at line 27.

So we have proved that every cell in $L(\sigma)$ is processed exactly once by the algorithm while $i = I(\sigma)$ and it is paired with some other cell of $L(\sigma)$ or classified as critical.

Finally, since the number of cells in the complex $\mathcal{K}$ is finite and the union of $L(\sigma)$'s covers the complex, the proof is complete. □

The correctness proof is concluded by proving that the algorithm produces an acyclic partial matching of the complex compatible with the filtration of $\mathcal{K}$ induced by f.

Proposition 8 *$\mathsf{A}, \mathsf{B}, \mathsf{C}$ is a partition of the complex $\mathcal{K}$ and $\mathtt{m}$ is a bijective function from A to B. Moreover, if $\sigma \in \mathcal{K}^\alpha \cap \mathsf{A}$ then $\mathtt{m}(\sigma) \in \mathcal{K}^\alpha$.*

Proof By Lemma 7, $\mathsf{A} \cup \mathsf{B} \cup \mathsf{C} = \mathcal{K}$. We show that $\mathsf{A} \cap \mathsf{B} = \emptyset$. This statement is trivial for vertices since they cannot belong to B. Assume on the contrary that there exists a cell $\alpha^{(p)}$ with $p \geq 1$ such that $\alpha \in \mathsf{A} \cap \mathsf{B}$. Thus, there exist cells $\delta^{(p-1)} < \alpha < \gamma^{(p+1)}$ such that $\mathtt{m}(\alpha) = \gamma$ and $\mathtt{m}(\delta) = \alpha$. This means that α is paired twice by processing two different lower stars $L(\sigma_1)$ and $L(\sigma_2)$. By Lemma 2(4), there exists a cell β such that the cells δ, α and γ are all processed within $L(\beta)$. Thus α is processed twice within $L(\beta)$, which contradicts Lemma 7. If we assume that $\mathsf{A} \cap \mathsf{C} \neq \emptyset$ and contains a cell α, then α has been declared critical either at line 7 or at line 26. In the first case, α was not previously assigned to A because of line 5; on the other hand it cannot be assigned to A later because of Lemma 7. In the second case, when α is added to C, it comes from PQzero and PQone is empty. The only cells that may enter PQzero or PQone later are cofaces of α (see line 27). Therefore α cannot be added again to PQzero, and as a consequence it cannot be added to A. The proof that $\mathsf{B} \cap \mathsf{C} = \emptyset$ can be handled in much the same way. It follows that $\mathsf{A}, \mathsf{B}, \mathsf{C}$ is a partition of $\mathcal{K}$.

By construction, the map $\mathtt{m}$ is onto. We will show that $\mathtt{m}$ is injective. If two cells σ_1 and σ_2 are paired with the same cell α, it follows that α must belong to the intersection of two lower stars. Therefore, again by Lemma 2(4), there must exist a β such that α is processed twice by the algorithm within $L(\beta)$ which is again a contradiction to Lemma 7. Thus $\mathtt{m}$ is bijective.

By construction, σ is a face of $\mathtt{m}(\sigma)$ and they both belong to some $L(\beta)$. Thus, by Lemma 2(1), $f(\sigma) = f(\mathtt{m}(\sigma))$, and therefore if $\sigma \in \mathcal{K}^\alpha \cap \mathsf{A}$ then $\mathtt{m}(\sigma) \in \mathcal{K}^\alpha$. □

Theorem 9 *Algorithm 2 produces a partial matching $(\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathtt{m})$ that is acyclic.*

Proof A partial matching is acyclic if and only if there are no nontrivial closed $\mathtt{m}$–paths. We prove this by contradiction. Assume that

$$\sigma_0 \xrightarrow{\mathtt{m}} \tau_0 \xrightarrow{>} \sigma_1 \xrightarrow{\mathtt{m}} \tau_1 \xrightarrow{>} \ldots \xrightarrow{>} \sigma_n \xrightarrow{\mathtt{m}} \tau_n \xrightarrow{>} \sigma_0 \tag{4}$$

is a directed loop in the modified Hasse diagram. In particular, all σ_i in the loop have the same dimension, say, p and all τ_i have the same dimension $p+1$. The index i of σ_i is not the value of the indexing function I but it simply displays its position in the loop. From Lemma 2(1), it follows that

$$f(\sigma_0) = f(\tau_0) \succeq f(\sigma_1) = f(\tau_1) \succeq \ldots \succeq f(\sigma_n) = f(\tau_n) \succeq f(\sigma_0). \qquad (5)$$

If any of the inequalities $f(\tau_{i-1}) \succeq f(\sigma_i)$ is strict, then there exists a coordinate j such that $f_j(\tau_{i-1}) > f_j(\sigma_i)$ and so $f_j(\sigma_0) > f_j(\sigma_0)$, a contradiction. Hence f is constant on all the elements of the loop. Let us set $\bar{\sigma}$ equal to the cell such that

$$I(\bar{\sigma}) = \min \left\{ I(\alpha) \in \mathbb{N} : \alpha \subseteq \bigcap_{i=0}^{n} \sigma_i \cap \bigcap_{i=0}^{n} \tau_i \right\}.$$

The simplex $\bar{\sigma}$ exists by Lemma 2(3). This implies that σ_i and τ_i belong to $L(\bar{\sigma})$ for $i = 0, \ldots, n$. Now we have two cases: either $\bar{\sigma} = \sigma_j$ for some j, $0 \leq j \leq n$, or $\bar{\sigma}$ is a face of σ_j for every j. In the first case, without a loss of generality, we may assume that $\bar{\sigma} = \sigma_0$. Since σ_n has the same dimension as σ_0, it is in $L(\sigma_0)$ if and only if $\sigma_n = \sigma_0$, implying that the loop is trivial, a contradiction. In the second case, note that Algorithm 2 produces a pairing $\mathtt{m}(\sigma_i) = \tau_i$ only when $\mathtt{num\_unclass\_facets}_{\bar{\sigma}}(\tau_i)$ = 1, and in that case the unclassified face of τ_i is exactly σ_i. Therefore, we have that σ_0 is paired to τ_0 after that σ_1, also a face of τ_0, has been paired to τ_1.

Iterating this argument for $i = 1, \ldots, n$, we deduce that σ_0 is paired to τ_0 after that σ_n has been paired to τ_n. But since σ_0 is also a face of τ_n, and σ_0 is still unclassified when σ_n is paired to τ_n, it follows that $\sigma_n = \sigma_0$, implying that the loop is trivial, again a contradiction. □

3.2 Filtration Preserving Reductions

Lefchetz complexes introduced by Lefschetz in [20] are developed further in [22] under the name S-complex. In our context, these complexes are produced by applying the reduction method [18,22,21] to an initial simplicial complex $\mathcal{K}$, with the use of the matchings produced by our main Algorithm 2. Both concepts of partial matchings and *sublevel set filtration* of $\mathcal{K}$ induced by $f : \mathcal{K} \to \mathbb{R}^k$ introduced in Section 2 naturally extend to Lefschetz complexes as proved in [2]. We denote by $H_*(\mathsf{S})$ the graded homology module of S with respect to a given principal ideal domain R. A choice of a ground ring R is made in applications. The dependence of persistent homology on that choice is discussed in [7]. As the computation of invariants based on ranks of homology modules is of concern, such as rank invariants in [8], it is sufficient to assume that R is a field, and most often $R = \mathbb{Z}_2$ is chosen. Persistence is based on analyzing the homological changes occurring along the filtration as the multiparameter $a \in \mathbb{R}^k$ varies. This analysis is carried out by considering, for $a \preceq b$, the homomorphism $H_*(j^{(a,b)}) : H_*(\mathsf{S}^a) \to H_*(\mathsf{S}^b)$ induced by the inclusion map $j^{(a,b)} : \mathsf{S}^a \hookrightarrow \mathsf{S}^b$. The image of the map $H_q(j^{(a,b)})$ is known as the *q-th multidimensional persistent homology group* of the filtration at (a, b) and we denote it by $H_q^{a,b}(\mathsf{S})$. It contains the homology classes of order q born not later than a and still alive at b.

If we assume that $(\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathtt{m})$ is an acyclic matching on a filtered Lefschetz complex S obtained from the original simplicial complex $\mathcal{K}$ by reduction, the following result holds which asserts that the multidimensional persistent homology of the reduced complex is the same as of the initial complex (see [2]).

Corollary 10 *For every* $a \preceq b \in \mathbb{R}^k$, $H_*^{a,b}(\mathsf{C}) \cong H_*^{a,b}(\mathcal{K})$.

The collection of homology groups $\{H_*^{a,b}(\mathcal{K}) \mid a \preceq b \in \mathbb{R}^k\}$ together with the maps $H_*(j^{(a,b)})$ induced by inclusions $j^{(a,b)}$ may be phrased in terms of multidimensional persistence modules introduced by [7]. Algorithms and programs computing such modules are constantly being improved and we refer for instance to [16] for recent contributions.

3.3 Complexity Analysis

We first describe the computational complexity of Algorithm 2 and then establish some comparisons with Algorithm 3.3 in [2]. We use the following definitions and parameters in estimating the computational cost of Algorithm 2.

1. Given a simplex $\sigma \in \mathcal{K}$, the coboundary cells of σ are given by
$$\mathbf{cb}\,(\sigma) := \{\tau \in \mathcal{K} \,|\, \sigma \text{ is a face of } \tau\}. \tag{6}$$
It is immediate from the definitions that $L_*(\sigma) \subset \mathbf{cb}\,(\sigma)$.
2. We define the coboundary mass γ of $\mathcal{K}$ as
$$\gamma = \max_{\sigma \in \mathcal{K}} \operatorname{card} \mathbf{cb}\,(\sigma), \tag{7}$$
where card denotes cardinality. While γ is trivially bounded by N, the number of cells in $\mathcal{K}$, this upper bound is a gross estimate of γ for many simplicial complexes such as simplicial manifolds or approximating surface boundaries of objects.
3. For the simplicial complex $\mathcal{K}$, we assume that the boundary and coboundary cells of each simplex are computed offline and stored in such a way that access to every cell is done in constant time.
4. Given an admissible function $f : \mathcal{K} \to \mathbb{R}^k$, the values by f of simplices $\sigma \in \mathcal{K}$ are stored in the structure that stores the complex $\mathcal{K}$ in such a away that they are accessed in constant time.
5. We assume that adding cells to the lists A, B, and C is done in constant time.

Algorithm 2 processes every cell σ of the simplicial complex $\mathcal{K}$ and checks whether it is classified or not. In the latter case, the algorithm requires a function that returns the cells in the strict lower star $L_*(\sigma)$ which is read directly from the structure storing the complex. In the best case, $L_*(\sigma)$ is empty and the cell is declared critical. Since $L_*(\sigma) \subset \mathbf{cb}\,(\sigma)$, it follows that $\operatorname{card} L_*(\sigma) \leq \gamma$. By Lemma 7, we can see that every cell in $L_*(\sigma)$ enters at most once in `PQzero` and `PQone`. It follows that the while loops in the algorithm are executed all together in at most 2γ steps. We may consider the operations such as finding the number of unclassified faces of a cell to have constant time except for the priority queue operations which are logarithmic

in the size of the priority queue when implemented using heaps. Since the sizes of `PQzero` and `PQone` are clearly bounded by γ, it follows that $L_*(\sigma)$ is processed in at most $O(\gamma \log \gamma)$ steps. Therefore processing the whole complex incurs a worst case cost of $O(N \cdot \gamma \log \gamma)$.

Algorithm 3.3 in [2] is based on processing recursively lower links of vertices to achieve a partial classification of the cells of the complex. The cells that are not classified at the end of the process are added to the list of critical cells. We have established in [2] that given a simplicial complex $\mathcal{K}$ with vertex set $\mathcal{K}_0$ of size N_0, the computational cost of Algorithm 3.3 for processing all the cells of the complex is bounded above by $2\gamma_0^d(d+1)!N_0$ where d is the dimension of the complex and $\gamma_0 = \max_{v \in \mathcal{K}_0} \text{card}\,\mathbf{cb}\,(v)$. It is easily seen that $\gamma = \gamma_0$ and $N \leq \gamma N_0$, where γ and N are as defined above. Unlike Algorithm 3.3, Algorithm 2 processes and classifies all the cells of the complex with a computational cost that cannot exceed $2N\gamma \log \gamma \leq 2\gamma^3 N_0$. It follows that when $d \geq 3$, we have $2N\gamma \log \gamma \leq 2\gamma^3 N_0 \leq 2\gamma^d(d+1)!N_0$. This suggests a possible improvement in the computational complexity of the second algorithm when the dimension of the complex is high. This comparison does not take into account the number of residual cells, which are the cells that are not classified by Algorithm 3.3 and added by default to the critical cells. This number is example dependent and can represent an important proportion when compared to the total number of cells of the complex as shown in column 4 of Table 1, where the number of residual critical cells for several complexes is displayed between parentheses. For example, the proportions of residual cells with respect to the original number of cells represent 42% for `sphere_1`, 61% for `sphere_2` and `torus_4608`, and 50% for `Klein_187`.

4 Combinatorial Interpretation of the Matching Algorithm

4.1 Multidimensional Discrete Morse Function

As commented in Introduction, we provide here the first steps towards an extension of the Forman discrete Morse theory [14] to multidimensional functions. We relate the new definitions to the acyclic partial matching provided by Algorithm 2.

Given a function $g : \mathcal{K} \to \mathbb{R}^k$, for any $\alpha \in \mathcal{K}_p$, we introduce the notation

$$H_g(\alpha) = \{\beta \in K_{p+1} \mid \beta > \alpha \text{ and } g(\beta) \preceq g(\alpha)\};$$

$$T_g(\alpha) = \{\gamma \in K_{p-1} \mid \gamma < \alpha \text{ and } g(\alpha) \preceq g(\gamma)\}.$$

The letter H stands for *heads* and T for *tails*.

Definition 11 A function $g : \mathcal{K} \to \mathbb{R}^k$ is a *multidimensional discrete Morse function*, in short, an *mdm function*, if the following conditions hold for every $\alpha \in K_p$:

(1) $\text{card}\, H_g(\alpha) \leq 1$;
(2) $\text{card}\, T_g(\alpha) \leq 1$;
(3) If $\beta^{(p+1)} > \alpha$ is not in $H_g(\alpha)$, then $g(\alpha) \npreceq g(\beta)$;
(4) If $\gamma^{(p-1)} < \alpha$ is not in $T_g(\alpha)$, then $g(\gamma) \npreceq g(\alpha)$.

Proposition 12 *For any simplex $\alpha \in \mathcal{K}$, one of the sets $H_g(\alpha)$ or $T_g(\alpha)$ must have cardinality zero. That is, for any α,*

$$\operatorname{card} H_g(\alpha) \cdot \operatorname{card} T_g(\alpha) = 0.$$

Proof We recall that any simplicial complex has the property that whenever $\gamma^{(p-1)} < \alpha^{(p)} < \beta^{(p+1)}$, necessarily there exists $\alpha'^{(p)} \neq \alpha$ such that $\gamma < \alpha' < \beta$.

Let us assume that both $H_g(\alpha)$ and $T_g(\alpha)$ are nonempty for some $\alpha \in \mathcal{K}_p$. Then there exists a coface $\beta^{(p+1)}$ and a face $\gamma^{(p-1)}$ of the simplex α such that $g(\beta) \preceq g(\alpha) \preceq g(\gamma)$. Let $\alpha' \neq \alpha$ be a different face of β that contains γ. It follows from Definition 11 that $g(\gamma) \npreceq g(\alpha') \npreceq g(\beta)$, which is a contradiction. □

Definition 13 Let $g : \mathcal{K} \to \mathbb{R}^k$ be a multidimensional discrete Morse function. A simplex $\gamma \in \mathcal{K}$ is *critical* if both $H_g(\gamma)$ and $T_g(\gamma)$ are empty. A simplex that is not critical is called *regular*.

Essentially, discrete Morse functions are functions on $\mathcal{K}$ that increase with the dimension of the simplices, with at most one exception for each simplex α. The simplest example of a multidimensional discrete Morse function is the one given by the formula $g(\alpha) = (\dim(\alpha), \ldots, \dim(\alpha)) \in \mathbb{R}^k$. In this case, all simplices of $\mathcal{K}$ are critical. Any classical Forman's Morse function $f : \mathcal{K} \to \mathbb{R}$ gives rise to an `mdm` function $g : \mathcal{K} \to \mathbb{R}^k$ defined by $g(\sigma) = (f(\sigma), \ldots, f(\sigma)) \in \mathbb{R}^k$.

Recall that a *discrete vector field* V on $\mathcal{K}$ is a collection of pairs $\left(\alpha^{(p)}, \beta^{(p+1)}\right)$ of simplices of $\mathcal{K}$ with $\alpha^{(p)} < \beta^{(p+1)}$ such that each simplex of $\mathcal{K}$ is in at most one pair of V.

Let now $g : \mathcal{K} \to \mathbb{R}^k$ be an `mdm` function. It follows from Definition 11 and Proposition 12 that the sets

$$\mathsf{A} = \{\alpha \in \mathcal{K} \mid \operatorname{card} H_g(\alpha) = 1\}, \quad \mathsf{B} = \{\beta \in \mathcal{K} \mid \operatorname{card} T_g(\beta) = 1\},$$

and

$$\mathsf{C} = \{\gamma \in \mathcal{K} \mid \operatorname{card} H_g(\gamma) = 0 = \operatorname{card} T_g(\gamma)\}$$

form a partition of $\mathcal{K}$. Next, a map $\mathtt{m} : \mathsf{A} \to \mathsf{B}$ defined by

$$\mathtt{m}(\alpha) = \beta \in H_g(\alpha),$$

where β is the unique element of $H_g(\alpha)$, defines a discrete vector field $\{(\alpha, \mathtt{m}(\alpha))\}_{\alpha \in \mathsf{A}}$ which will be called the *gradient field* of g. It also follows that $(\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathtt{m})$ is a partial matching as defined in Section 2.

4.2 Linking MultiD Discrete Morse Functions to the Matching Algorithm

Let now $f : \mathcal{K} \to \mathbb{R}^k$ be the function used as input in Algorithm 2 and $(\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathtt{m})$ is the partial matching produced by that algorithm. It is instantly seen that, in general, f does not satisfy the assumptions of Definition 11. Indeed, the lower star $L(\alpha)$ of a simplex $\alpha^{(p)}$ may contain more than one simplex $\beta^{(p+1)}$ so the condition (1) of the definition fails. Condition (2) may fail as well. We do however have this:

Proposition 14 *Any function $f : \mathcal{K} \to \mathbb{R}^k$ used as input in Algorithm 2 satisfies conditions (3) and (4) of Definition 11.*

Proof Let $\beta = \beta^{(p+1)} > \alpha$ be any cofacet of a given $\alpha \in \mathcal{K}_p$. Since f is an extension of the data values on vertices:

$$f_i(\alpha) = \max_{v \in \mathcal{K}_0(\alpha)} f_i(v),$$

we have $f(\alpha) \preceq f(\beta)$. Hence such β is in $H_f(\alpha)$ if and only if $f(\alpha) = f(\beta)$. Thus a cofacet β of α is not in $H_f(\alpha)$ if and only if $f(\alpha) \npreceq f(\beta)$ and (3) follows. Condition (4) follows by the same argument. □

We now proceed toward the construction of an `mdm` function g from the function f obtained so that the partial matching produced by the algorithm for f coincides with the one for g.

Any simplex $\sigma \in \mathcal{K}$ is either classified by Algorithm 2 at the beginning of processing its own lower star (lines 7 or 10), that is when its index $I(\sigma)$ is considered in the algorithm, or as an element of a lower star of a distinct simplex. In the first case, we call σ a *primary simplex of lower star processing* or shortly *primary*. Otherwise, it is called *secondary*. In particular, every vertex is *primary*. It is easily seen that a simplex is *primary* if and only if it is not contained in a lower star of another simplex with smaller index. Any such simplex is always classified either at line 7 or at line 10 of Algorithm 2.

Let $P = \{\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_n}\}$ be the set of all *primary* simplices of $\mathcal{K}$ ordered by increasing values of their indices, i.e. $i_j = I(\sigma_{i_j})$ and $i_j < i_k$ if $j < k$. The following proposition proves that the lower stars of the primary simplices of $\mathcal{K}$ is a partition of the simplicial complex $\mathcal{K}$.

Proposition 15 *The collection of subsets $\{L(\sigma_{i_j})\}_{1 \le j \le n}$ forms a partition of $\mathcal{K}$. That is*

(1) $L(\sigma_{i_j}) \cap L(\sigma_{i_k}) = \emptyset$ *if* $j \neq k$, *and*

(2) $\bigcup_{j=1}^{n} L(\sigma_{i_j}) = \mathcal{K}$

Proof (1) Assume that $L(\sigma_{i_j}) \cap L(\sigma_{i_k}) \neq \emptyset$. Then, by Lemma 2(4) there exists $\beta \in \mathcal{K}$ such that $L(\sigma_{i_j}) \cup L(\sigma_{i_k}) \subset L(\beta)$ and $I(\beta) < \min(i_j, i_k)$. Hence σ_{i_j} or σ_{i_k} are *secondary* simplices, which is a contradiction.

(2) Let $\sigma \in \mathcal{K}$. If σ is *primary*, then $\sigma = \sigma_{i_j}$ for some j and $\sigma \in L(\sigma_{i_j})$. Otherwise σ is *secondary* and therefore it is classified by Algorithm 2 as part of a lower star $L(\beta)$ of some simplex $\beta \in \mathcal{K}$. But, then β is *primary* and $\beta = \sigma_{i_j}$ for some j. Hence $\sigma \in \bigcup_{j=1}^{n} L(\sigma_{i_j})$. It follows that

$$\bigcup_{j=1}^{n} L(\sigma_{i_j}) = \mathcal{K}.$$

□

We now give sufficient conditions so that a given function g defined on $\mathcal{K}$ is an `mdm` function with a partial matching that coincides with the partial matching produced by Algorithm 2 for f. First, note that when σ_{i_j} is critical, its lower star is reduced to the singleton $L(\sigma_{i_j}) = \{\sigma_{i_j}\}$. When σ_{i_j} is not critical, then, σ_{i_j} is added to A at line 10 of Algorithm 2. Let us order the simplices α of $L(\sigma_{i_j})$ increasingly as they become classified by Algorithm 2, though placing $\mathtt{m}(\alpha)$ immediately before α, if α is added to A. Thus $L(\sigma_{i_j})$ can be presented as

$$L(\sigma_{i_j}) = \{\alpha_{1j}, \alpha_{2j}, \ldots \alpha_{l_j j}\}, \tag{8}$$

where the first two terms are $\alpha_{1j} = \delta = \mathtt{m}(\sigma_{i_j})$ and $\alpha_{2j} = \sigma_{i_j}$.

Definition 16 *A function $g : \mathcal{K} \to \mathbb{R}^k$ is called f-`compatible` if for every $\alpha, \beta \in \mathcal{K}$ it satisfies the conditions*

(1) *if $f(\alpha) \precneqq f(\beta)$ then $g(\alpha) \precneqq g(\beta)$, and*
(2) *if $\alpha, \beta \in L(\sigma_{i_j})$ for some primary simplex σ_{i_j} and α is classified earlier than β as defined by equation (8), then $g(\alpha) \precneqq g(\beta)$.*

We prove in the following that any f-`compatible` function is an `mdm` function whose partial matching on $\mathcal{K}$ coincides with the one produced by Algorithm 2 for f.

Theorem 17 *Let $g : \mathcal{K} \to \mathbb{R}^k$ be an f-`compatible` function. Then g is a multidimensional discrete Morse function whose partial matching on $\mathcal{K}$ coincides with that produced by Algorithm 2 when f is used as input.*

Proof We start with the case of critical simplices.

If α is *primary* and critical in the sense that Algorithm 2 assigns it to C, then $L_*(\alpha) = \emptyset$. If there exists $\beta \in H_g(\alpha)$, then $g(\beta) \preceq g(\alpha)$. The first condition in Definition 16 allows us to conclude that $f(\alpha) = f(\beta)$ and hence $\beta \in L_*(\alpha)$, which contradicts the fact that $L_*(\alpha) = \emptyset$. Hence, $H_g(\alpha) = \emptyset$. Assume there exists $\gamma \in T_g(\alpha)$. Then γ is a facet of α with $g(\alpha) \preceq g(\gamma)$. Again, we must have $f(\gamma) = f(\alpha)$ and $\alpha \in L_*(\gamma)$, which contradicts the fact that α is *primary*. Hence, $T_g(\alpha) = \emptyset$. Thus, α is a critical cell of g.

If α is classified as critical by Algorithm 2 but not *primary*, then there exists a *primary* simplex σ_{i_j} such that $\alpha \in L_*(\sigma_{i_j})$. This means α is classified as critical at step 26 of Algorithm 2 in which case all the facets of α in $L(\sigma_{i_j})$ are classified at an earlier time than α. Let γ be a facet of α. If $f(\gamma) = f(\alpha)$ then γ and α should belong to the same lower star of some *primary* simplex, i.e. $\gamma \in L(\sigma_{i_j})$ and classified at earlier time. Hence $g(\gamma) \precneqq g(\alpha)$ and $\gamma \notin T_g(\alpha)$. If $f(\gamma) \precneqq f(\alpha)$ then $g(\gamma) \precneqq g(\alpha)$ and $\gamma \notin T_g(\alpha)$. Hence $T_g(\alpha) = \emptyset$. Let β be a coface of α. If $f(\alpha) \precneqq f(\beta)$ then $g(\alpha) \precneqq g(\beta)$ and $\beta \notin H_g(\alpha)$. On the other hand, if $f(\alpha) = f(\beta)$ then $\alpha, \beta \in L(\sigma_{i_j})$. Then by Lemma 7, α must have been classified at earlier time than β (otherwise β would be paired with α when β is popped from`PQone`), and hence $g(\alpha) \precneqq g(\beta)$ and $\beta \notin H_g(\alpha)$. It follows that $H_g(\alpha) = \emptyset$. Thus, also in this case α is a critical cell of g.

Now we examine paired cells (α, β) where β is a coface of α.

If α is a *primary* cell, then α is paired with a coface β at step 10 of Algorithm 2. By definition of g, we have $g(\beta) \precneqq g(\alpha)$. It follows that $\beta \in H_g(\alpha)$ and $\alpha \in T_g(\beta)$.

Now, assume there exists $\gamma \in T_g(\alpha)$. Then since γ is a facet of α and $g(\gamma) \precneqq g(\alpha)$ it follows that $f(\gamma) = f(\alpha)$ which contradicts the fact α is *primary*. Hence $T_g(\alpha) = \emptyset$.

Let us prove now that $H_g(\alpha) = \{\beta\}$. We know that $\beta \in H_g(\alpha)$. Assume that some other coface τ of α is in $H_g(\alpha)$. This means we have $g(\tau) \preceq g(\alpha)$ and hence $f(\alpha) = f(\tau)$ and therefore $\tau \in L_*(\alpha)$. If $\tau \neq \beta$, then τ is classified (processed) later than β and α and therefore $g(\alpha) \precneqq g(\tau)$, which is a contradiction. Hence $H_g(\alpha) = \{\beta\}$.

Proving that $H_g(\beta) = \emptyset$ follows the same pattern. That is, if $\tau \in H_g(\beta)$, then $g(\tau) \preceq g(\beta)$ and therefore $f(\beta) = f(\tau)$ meaning that $\tau \in L_*(\alpha)$. But then τ is processed later than β and hence $g(\beta) \precneqq g(\tau)$ which is a contradiction. The fact that $T_g(\beta) = \{\alpha\}$ is a direct result of Lemma 3.
If α is not *primary*, then $\alpha, \beta \in L_*(\sigma)$ for some *primary* simplex σ. Assume that $\tau \in H_g(\beta)$, then $g(\tau) \preceq g(\beta)$ and therefore $f(\beta) = f(\tau)$ meaning that $\tau \in L_*(\sigma)$. By Definition 16 and Lemma 6, τ is processed earlier than β iff τ is paired with β, which is not the case. Hence τ is processed later than β and $g(\beta) \precneqq g(\tau)$, which shows that $H_g(\beta) = \emptyset$. Assume that $\gamma \in T_g(\alpha)$, then $g(\alpha) \preceq g(\gamma)$ and therefore $f(\alpha) = f(\gamma)$ meaning that $\gamma \in L_*(\sigma)$. By Definition 16 and Lemma 6, α is processed earlier than γ iff α is paired with γ, which is not the case. Hence γ is processed earlier than α and $g(\gamma) \precneqq g(\alpha)$, which shows that $T_g(\alpha) = \emptyset$.

It remains to show that $T_g(\beta) = \{\alpha\}$ and $H_g(\alpha) = \{\beta\}$. From Definition 16 (2), we can see immediately that $\beta \in H_g(\alpha)$ and $\alpha \in T_g(\beta)$. If we assume that $\tau \in H_g(\alpha)$ and $\tau \neq \beta$, then since $g(\tau) \preceq g(\alpha)$ we must have $f(\alpha) = f(\tau)$ and $\tau \in L_*(\sigma)$. Using again Lemma 6, we deduce that τ is processed later than α which contradicts $g(\tau) \preceq g(\alpha)$. Thus $H_g(\alpha) = \{\beta\}$.

Assume that $\gamma \in T_g(\beta)$ and $\gamma \neq \alpha$, then $g(\beta) \preceq g(\gamma)$ and therefore $f(\beta) = f(\gamma)$ meaning that $\gamma \in L_*(\sigma)$. By Definition 16 and Lemma 6, β is processed earlier than γ iff β is paired with γ, which is not the case. Hence γ is processed earlier than β and $g(\gamma) \precneqq g(\beta)$, which is a contradiction. Thus $T_g(\beta) = \{\alpha\}$. Hence g satisfies properties (1) and (2) of Definition 11.

Assume $\beta \notin H_g(\alpha)$. Then from what precedes in the proof, we can conclude that (α, β) are not paired cells. If $f(\alpha) \precneqq f(\beta)$ then by Definition 16 we have also $g(\alpha) \precneqq g(\beta)$. Otherwise $f(\alpha) = f(\beta)$ and $\alpha, \beta \in L(\sigma_{i_j})$ for some *primary* simplex σ_{i_j}. By Definition 16 the values of g are strictly increasing in $L(\sigma_{i_j})$. If $g(\beta) \precneqq g(\alpha)$ then by Lemma 6, α and β are paired which is a contradiction. Then, we must have $g(\alpha) \precneqq g(\beta)$, which proves property (3) of Definition 11.

Property (4) of Definition 11 is proved by a similar argument. □

4.3 Existence of f-`Compatible` Functions

In the following, we prove that there exists f-`compatible` functions by constructing one specific example. Let **cb** (σ) and γ be as defined in formulas (6) and (7). We define

$$\epsilon = 1/(\gamma + 1).$$

When σ_{i_j} is critical, its lower star is reduced to the singleton $L(\sigma_{i_j}) = \{\sigma_{i_j}\}$ and we define

$$\begin{aligned} g_1(\sigma_{i_j}) &= i_j, \\ g_i(\sigma_{i_j}) &= f_i(\sigma_{i_j}) \quad \text{if } i > 1. \end{aligned} \tag{9}$$

Consider the case where σ_{i_j} is not critical. Then, we use equation (8) to express the elements of $L(\sigma_{i_j})$ on which we define the function g as follows:

$$\begin{aligned} g_1(\alpha_{sj}) &= i_j + (s-2)\epsilon, \\ g_i(\alpha_{sj}) &= f_i(\sigma_{i_j}) \quad \text{if } i > 1. \end{aligned} \tag{10}$$

The same formula is used for all primary simplices to produce a function $g : \mathcal{K} \to \mathbb{R}^k$.

Theorem 18 *The function $g : \mathcal{K} \to \mathbb{R}^k$ defined by equation (9) and equation (10) is an f-`compatible` function.*

Proof It is clear that g is well defined on $\mathcal{K}$ since the lower stars of primary simplices form a partition of $\mathcal{K}$ by Proposition 15. We now prove that g satisfies the conditions of Definition 16. The second condition is trivially satisfied by construction since the values of g_1, the first component of g, are increasing on any given lower star of a non-critical *primary* simplex while the other components are maintained constant. To prove the first condition, let α and β be two simplices in $\mathcal{K}$ such that $f(\alpha) \precneqq f(\beta)$. It follows that α and β must belong to two different lower stars of *primary* simplices, that is there exist i_j and i_k such that $\alpha \in L(\sigma_{i_j})$ and $\beta \in L(\sigma_{i_k})$. Since $f(\sigma_{i_j}) = f(\alpha)$ and $f(\sigma_{i_k}) = f(\beta)$, we must have that $i_j < i_k$, that is $i_k - i_j > 1$. For $2 \leq i \leq k$, we have

$$g_i(\alpha) = f_i(\sigma_{i_j}) = f_i(\alpha) \leq f_i(\beta) = f_i(\sigma_{i_k}) = g_i(\beta).$$

On the other hand, there must exist $s_1 \in \{1, \ldots, l_j\}$ and $s_2 \in \{1, \ldots, l_k\}$ such that

$$g_1(\alpha) = i_j + (s_1 - 2)\epsilon \quad \text{and} \quad g_1(\beta) = i_k + (s_2 - 2)\epsilon.$$

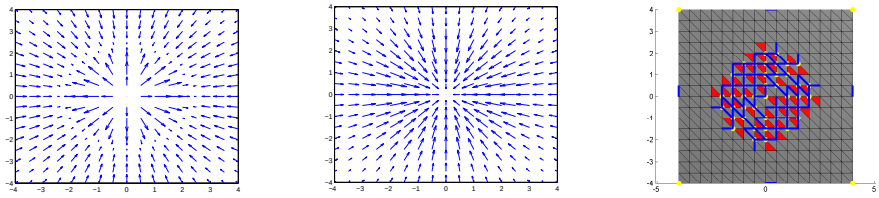
Since $s_1 \leq l_j \leq \gamma$ and $s_2 \geq 1$, we have

$$g_1(\beta) - g_1(\alpha) = i_k - i_j + (s_2 - s_1)\epsilon > 1 + (1-\gamma)\epsilon > 1 - \gamma\epsilon > 0,$$

since $\gamma\epsilon < 1$. It follows that $g(\alpha) \precneqq g(\beta)$. □

By combining Corollary 10, Theorem 17, and Theorem 18, we get the following result:

Corollary 19 *Let $g : \mathcal{K} \to \mathbb{R}^k$ be an f-`compatible` `mdm` function and `C` the set of its critical cells. Then, for every $a \preceq b \in \mathbb{R}^k$, $H_*^{a,b}(\mathtt{C}) \cong H_*^{a,b}(\mathcal{K})$.*



(a) (b) (c)

Fig. 2 (a): the gradient vector field of the scalar function $f_1(x,y) = e^{-(x^2+y^2)/8} - e^{-(x^2+y^2)/4}$. (b): the gradient vector field of the scalar function $f_2(x,y) = e^{-(x^2+y^2)/8}$. (c): the critical cells of dimension 0 in yellow, dimension 1 in blue, and dimension 2 in red, for the vector-valued function $f = (f_1, f_2)$, as retrieved by Algorithm 2 .

5 Experimental Results

We have successfully applied the algorithms from Section 3 to different sets of triangle meshes. In each case the input data is a 2-dimensional simplicial complex $\mathcal{K}$ and a function f defined on the vertices of $\mathcal{K}$ with values in $\mathbb{R}^2$. The first step is to slightly perturb f in order to ensure injectivity on each component as described in Section 2. The second step is to construct an index function defined on all the simplices of the complex and satisfying the properties of Lemma 1. Then we build the acyclic matching $\mathtt{m}$ and the partition $(\mathsf{A}, \mathsf{B}, \mathsf{C})$ on the simplices of the complex using Algorithm 2. In particular, the number of simplices in C out of the total number of simplices of $\mathcal{K}$ is relevant, because it determines the amount of reduction obtained by our algorithm to speed up the computation of multidimensional persistent homology.

Our experiments are aimed at understanding different phenomena:

- Geometric interpretation of retrieved critical cells;
- Invariance under subdivision of the mesh;
- Comparison with previous algorithm [2];

5.1 Geometric Interpretation of Retrieved Critical Cells

Our first dataset is a synthetic example. We triangulate the rectangle $[-4,4]^2$ by a triangle mesh with 81 vertices. We compute at each vertex the value of the scalar function $f_1(x,y) = e^{-(x^2+y^2)/8} - e^{-(x^2+y^2)/4}$ and that of the function $f_2(x,y) = e^{-(x^2+y^2)/8}$. These two functions have the property that their gradients agree outside the circle centered at $(0,0)$ with radius 2, while they disagree inside such circle. Next, we consider the vector-valued function $f = (f_1, f_2)$ and follow the above described procedure to apply Algorithm 2. The result of the computation is described in Figure 2. The critical cells retrieved by Algorithm 2 localize the region of the plane where the gradient vector fields of the scalar functions f_1 and f_2, that is the components of the vector-valued function f, disagree. The above experiments suggests a relationship between critical cells retrieved by our algorithm and Pareto critical points.

Pareto critical points for a vector-valued function are those points at which the convex hull of the gradients of the components of the function contains the null vector. They may be regarded as a generalization of critical points for a scalar function form the standpoint of optimization. It is important to notice that Pareto critical points on a surface generically form curves, and therefore they are not isolated. Thus, our next experiment concerns triangulations of well known surfaces (a sphere, a torus, a Klein bottle), and compares the location of the retrieved critical cells with the curves of Pareto critical points in the differentiable setting. The functions used for this experiment are obvious projections on a coordinate plane. The comparison is illustrated in Figure 3. The results are in accordance with the interpretation of critical cells as locations corresponding to Pareto critical points. a

5.2 Dependence on the Triangulation

This set of experiments aims at understanding the stability of our algorithm under changes of triangulations. More precisely, we aim at checking if the location of critical cells remains roughly the same even if we change the triangulation for example by subdivision. Moreover, we aim at checking if the ratio of number of cells found to be critical by the algorithm over the total number of cells remains roughly the same when we consider finer and finer meshes. It is important to notice that it is not to be expected that the number of critical cells remains the same because, as we have already mentioned, Pareto critical points for $\mathbb{R}^2$-valued functions on a surface generically form curves. Therefore, we expect that a refinement of the triangle mesh corresponds to a refinement of such curve of critical points. This is indeed what we find in Figure 4. However, our experiments also show that not all triangulations localize the Pareto critical region with the same precision. This phenomenon is shown in Figure 5 where a change in the spatial displacement of triangles of a sphere allows to localize much more precisely the Pareto critical curves of the function $f(x, y, z) = (x, z)$, and at the same time to reduce the percentage of critical cells from 10.2% to 2.5%. We have also repeated the experiment using triangulations of the same space that are not subdivisions of each other. For the sphere, we consider its triangulations of five different sizes and we take $f(x, y, z) = (x, y)$. The comparison with other triangulations of the sphere is shown in Table 1: the second column shows the number of simplices in each considered mesh $\mathcal{K}$; the third column shows the number of critical cells obtained by using our matching algorithm to reduce $\mathcal{K}$; the fourth column shows the ratio between the third and the second column in percentage points. In the cases of the torus and of the Klein bottle we again consider triangulations of different sizes and we take $f(x, y, z) = (x, y)$. The numerical results are shown in the same table. In conclusion, our experiments on synthetic data confirm that the current simplex-based matching algorithm scales well with the size of the complex whereas the precision in the localization of the curves of Pareto critical points depends on the triangulation.

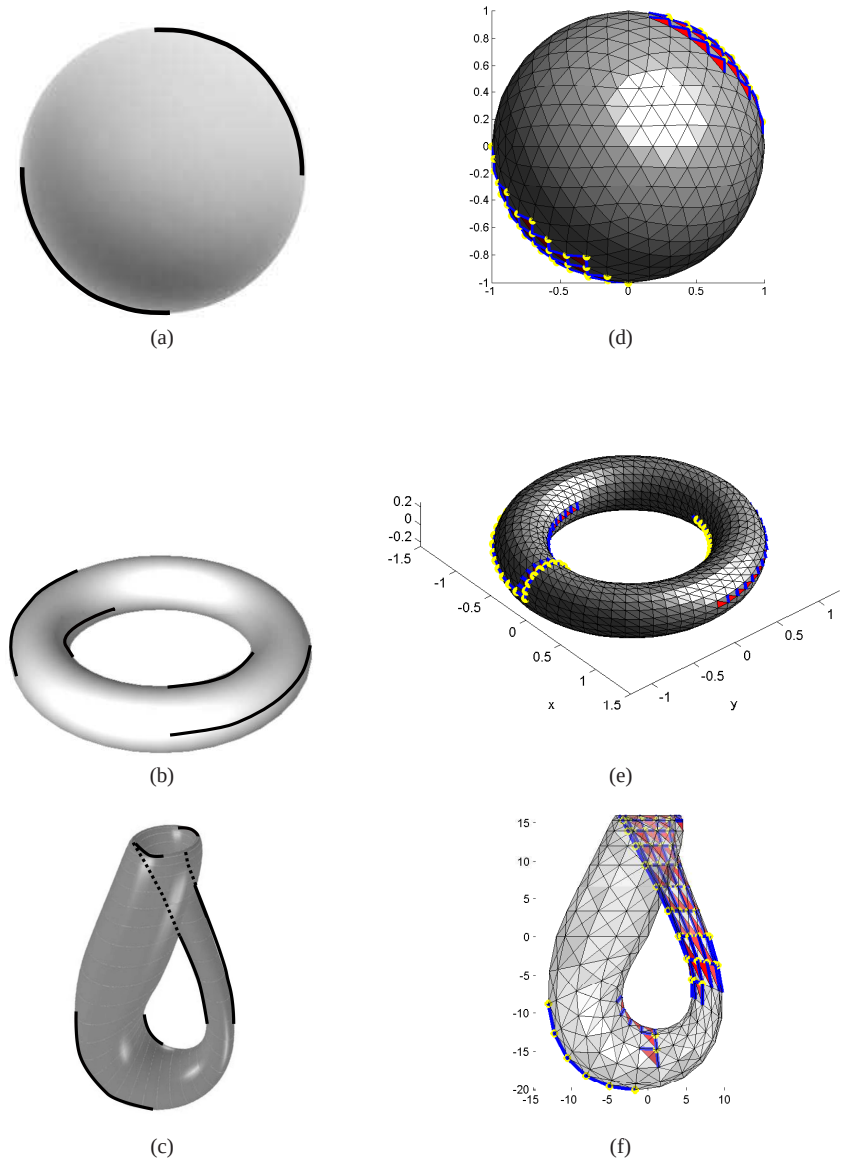
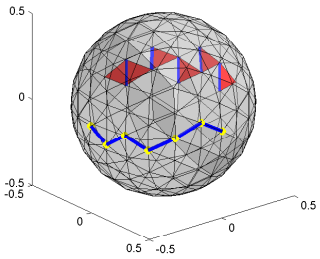
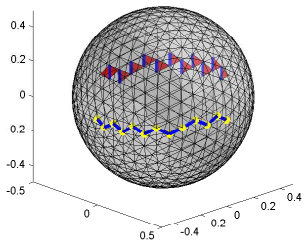


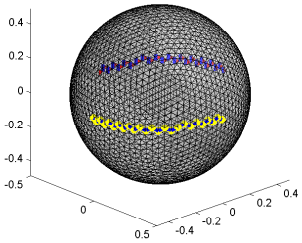
Fig. 3 (a), (b), (c): Curves of Pareto critical points for projection maps onto $\mathbb{R}^2$ for a sphere, a torus, and a Klein bottle are depicted by thick black lines. (d), (e), (f): Critical cells retrieved by Algorithm 2 on triangle meshes. Critical vertices are in yellow, critical edges in blue, and critical triangles in red.



(a)

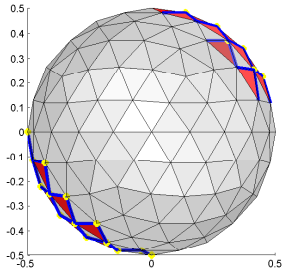


(b)

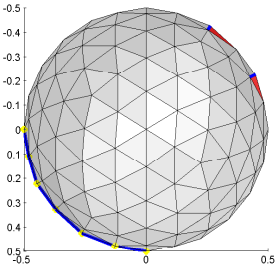


(c)

Fig. 4 (a): Critical cells of a triangle mesh. (b), (c) Two Loop subdivisions of it.



(a)



(b)

Fig. 5 Orientation of triangles impacts the precision of localization of Pareto critical curves.

5.3 Comparison with Previous Algorithms

Our experiments confirm that the current simplex-based matching algorithm produces a fair rate of reduction for simplices of any dimension on both synthetic and real data.

As for synthetic data, the comparison between the third and fifth columns of Table 1 shows that our new algorithm achieves an improvement in the rate of reduction

Table 1 Percentage of critical cells over the total number of cells achieved by Algorithm 2 on some triangulations of a sphere, a torus, and a Klein bottle, not obtained by refinement, and comparison with the performance of the algorithm in [2]. For the latter, the value between parenthesis shows the number of residual critical cells.

	$\lvert\mathcal{K}\rvert$	$\lvert\mathtt{C}\rvert$	%	$\lvert\mathtt{C}\rvert$ [2]	% [2]
sphere_1	38	4	10.53	22 (16)	57.89
sphere_2	242	20	8.26	162 (148)	66.94
sphere_3	962	98	10.19	650 (588)	67.57
sphere_4	1538	178	11.57	986 (904)	64.11
sphere_5	2882	278	9.65	1950 (1794)	67.66
torus_96	96	8	8.33	72 (64)	75.00
torus_4608	4608	128	2.78	3030 (2822)	65.75
torus_7200	7200	156	2.17	4592 (4308)	63.77
klein_89	89	19	21.3483	39 (27)	43.82
klein_187	187	35	18.7166	113 (93)	60.43
klein_491	491	59	12.0163	273 (232)	55.60
klein_1881	1881	257	13.6629	861 (771)	45.77

between 50% and 96% with respect to the vertex-based and recursive matching algorithm presented in [2]. Moreover, the fourth column of the same table shows the total number of cells classified as critical by Algorithm 3.3 of [2] followed, between parentheses, by the number of those cells residually classified as critical by the same algorithm without actually processing them. We see that the number of such residual critical cells found by Algorithm 3.3 of [2] is always between 30% and 95%. We underline that, on the contrary, our current algorithm processes all the cells making its results much more reliable in terms of significance.

As for real data, we now consider four triangle meshes (available at [1]). For each mesh the input 2-dimensional measuring function f takes each vertex v of coordinates (x, y, z) to the pair $f(v) = (|x|, |y|)$. In Table 2, the first row shows on the top line the number of vertices in each considered mesh, and in the middle line the percentage of critical vertices achieved by our algorithm. Finally, it also displays in the bottom line the analogous ratio achieved by our previous algorithm [2]. The second and the third rows show similar information for the edges and the faces. Finally, the fourth row shows the same information for the total number of cells of each considered mesh $\mathcal{K}$.

Again the current simplex-based matching algorithm produces a fair rate of reduction for simplices of any dimension also on real data, especially in comparison to the algorithm of [2].

The comparison with the analogous results obtained in [9] with reference to the reduction of vertices and edges in multi-valued graphs, shows a similar performance for vertices and edge reduction. However, it has to be noticed that the algorithm of [9] is designed to reduce graphs, and not generally simplicial complexes, in such a way that homology degree 0 multidimensional persistence is preserved. Hence, it can discard also all those edges that are important for the computation of homology in degree 1, whereas we aim at preserving multidimensional persistent homology in any degree.

Table 2 Percentage of critical cells over the total number of cells achieved by Algorithm 2 on some natural triangle meshes compared to that of [2], and that of [9] whenever applicable.

Dataset	tie	space_shuttle	x_wing	space_station
$\lvert\mathcal{K}_0\rvert$	2014	2376	3099	5749
%	27.5	9.5	19.8	30.8
%[2]	11.3	5.1	5.6	32.7
%[9]	29.2	11.0	18.4	33.7
$\lvert\mathcal{K}_1\rvert$	5944	6330	9190	15949
%	20.1	3.8	13.4	16.0
%[2]	56.2	58.4	39.2	70.0
%[9]	13.9	5.2	9.2	17.4
$\lvert\mathcal{K}_2\rvert$	3827	3952	6076	10237
%	14.1	0.4	9.9	8.0
%[2]	78.7	90.5	56.2	56.2
$\lvert\mathcal{K}\rvert$	11785	12658	18365	31935
%	19.4	3.8	13.3	16.1
%[2]	55.9	58.4	39.2	70.0

5.4 Discussion

The experiments on synthetic data confirmed two aspects: (1) The discrete case seems to behave much as the differentiable case for two functions [26] because critical cells are still localized along curves; (2) The number of critical cells scales well with the total number of cells, indicating that we are not detecting too many spurious critical cells.

We should point here a fundamental difference between Morse theory for one function whose critical points are isolated and extensions of Morse theory to vector-valued functions where, even in the generic case, critical points form stratified submanifolds. For example, for two functions on a surface, they form curves. Hence the topological complexity depends not on the number of critical points but on the number of such curves. As a consequence, the finer the triangulation, the finer the discretization of such curves and the larger number of critical cells we get.

On the other hand, experiments on real data show the improvement with respect to our previous matching algorithm [2] already observed in the toy example of Figure 1. We think that the new algorithm performs better because it is simplex-based rather than vertex-based. So, the presence of many non-comparable vertices has a limited impact on it.

6 Conclusion

The point of this paper is the presentation of Algorithm 2 to construct an acyclic partial matching from which a gradient compatible with multiple functions can be obtained. As such, it can be useful for specific purposes such as multidimensional persistence computation, whereas it is not meant to be a competitive algorithm to construct an acyclic partial matching for general purposes.

Some questions remain open. First, since indexing map is not unique, one may ask what is the effect of its choice on the output. We believe that the size of the resulting complex should be independent. This is a subject for future work.

A deeper open problem arises from the fact that the optimality of reductions is not yet well defined in the multidimensional setting, although the improvement is observed in practice. As commented earlier, even in the classical smooth case, the singularities of vector-valued functions on manifolds are not isolated. An appropriate application-driven extension of the Morse theory to multidimensional functions is not much investigated yet. The definitions proposed in Section 4.1 are the first step towards that extension. Some related work is that of [10] on Jacobi sets and of [23] on preimages of maps between manifolds. However there are essential differences between those concepts and our sublevel sets with respect to the partial order relation.

The experiments presented in this paper show an improvement with respect to the algorithm in [2] that, to the best of our knowledge, is still the only other algorithm available for this task. Such experiments were obtained with a non-optimized implementation. For an optimized implementation of it, we refer the reader to [17], where experiments on larger data sets can be found.

References

1. The GTS Library. Http://gts.sourceforge.net/samples.html
2. Allili, M., Kaczynski, T., Landi, C.: Reducing complexes in multidimensional persistent homology theory. Journal of Symbolic Computation **78**, 61–75 (2017)
3. Allili, M., Kaczynski, T., Landi, C., Masoni, F.: Algorithmic construction of acyclic partial matchings for multidimensional persistence. In: Discrete Geometry for Computer Imagery, eds. W.G. Kropatsch, N.M. Artner, I. Janusch, Proc. 20th IAPR Intl. Conf., DGCI2017, pp. 375–387. Springer (2017)
4. Biasotti, S., Cerri, A., Frosini, P., Giorgi, D., Landi, C.: Multidimensional size functions for shape comparison. J. Math. Imaging Vision **32**(2), 161–179 (2008)
5. Biasotti, S., Cerri, A., Giorgi, D., Spagnuolo, M.: Phog: Photometric and geometric functions for textured shape retrieval. Computer Graphics Forum **32**(5), 13–22 (2013)
6. Carlsson, G.: Topology and data. Bull. Amer. Math. Soc. **46**(2), 255–308 (2009)
7. Carlsson, G., Zomorodian, A.: The theory of multidimensional persistence. In: SCG '07: Proceedings of the 23rd annual Symposium on Computational Geometry, pp. 184–193. ACM, New York, NY, USA (2007). DOI http://doi.acm.org/10.1145/1247069.1247105
8. Cavazza, N., Ethier, M., Frosini, P., Kaczynski, T., Landi, C.: Comparison of persistent homologies for vector functions: from continuous to discrete and back. Computers and Mathematics with Applications **66**, 560–573 (2013). DOI doi:10.1016/j.camwa.2013.06.004
9. Cerri, A., Frosini, P., Krospatch, W.G., Landi, C.: A global method for reducing multidimensional size graphs - graph-based representations in pattern recognition. Lecture Notes in Computer Science **6658**, 1–11 (2011)
10. Edelsbrunner, H., Harer, J.: Jacobi sets of multiple Morse functions. In: Foundations of Computational Mathematics : FoCM'02 Minneapolis 2002, pp. 37–57. Cambridge Univ. Press, Cambridge (2004)
11. Edelsbrunner, H., Harer, J.: Persistent homology—a survey. In: Surveys on discrete and computational geometry, *Contemp. Math.*, vol. 453, pp. 257–282. Amer. Math. Soc., Providence, RI (2008)
12. Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological persistence and simplification. Discrete & Computational Geometry **28**(4), 511–533 (2002)
13. Escolar, E.G., Hiraoka, Y.: Persistence modules on commutative ladders of finite type. In: Mathematical Software ICMS 2014, pp. 144–151. Lecture Notes in Computer Science 8592 (2014)
14. Forman, R.: Morse theory for cell complexes. Advances in Mathematics **134**, 90–145 (1998)
15. Frosini, P.: Measuring shapes by size functions. In: Proc. of SPIE, Intelligent Robots and Computer Vision X: Algorithms and Techniques, vol. 1607, pp. 122–133. SPIE, Boston, MA, USA (1991)

16. Gäfvert, O.: Algorithms for multidimensional persistence. Master's thesis, KTH, Stockholm, Sweden (2016)
17. Iuricich, F., Scaramuccia, S., Landi, C., De Floriani, L.: A discrete morse-based approach to multivariate data analysis. In: SIGGRAPH ASIA 2016 Symposium on Visualization, SA '16, pp. 5:1–5:8. ACM, New York, NY, USA (2016). DOI 10.1145/3002151.3002166. URL http://doi.acm.org/10.1145/3002151.3002166
18. Kaczynski, T., Mrozek, M., Slusarek, M.: Homology computation by reduction of chain complexes. Computers and Mathematics with Applications **35**(4), 59–70 (1998)
19. King, H., Knudson, K., Mramor, N.: Generating discrete Morse functions from point data. Experimental Mathematics **14**(4), 435–444 (2005)
20. Lefschetz, S.: Algebraic Topology. Colloquium Publications, Vol. 27. Amer. Math. Soc., Providence (1942)
21. Mischaikow, K., Nanda, V.: Morse theory for filtrations and efficient computation of persistent homology. Discrete and Computational Geometry **50(2)**, 330–353 (2013)
22. Mrozek, M., Batko, B.: Coreduction homology algorithm. Discrete and Computational Geometry **41**, 96–118 (2009)
23. Patel, A.: Reeb spaces and the robustness of preimages. Ph.D. thesis, Duke Univ. (2010)
24. Robins, V., Wood, P.J., Sheppard, A.P.: Theory and algorithms for constructing discrete Morse complexes from grayscale digital images. IEEE Transactions On Pattern Analysis And Machine Intelligence **33**(8), 1646–1658 (2011)
25. Sedgewick, R., Wayne, K.: Algorithms. Addison-Wesley (2011)
26. Smale, S.: Global analysis and economics. I. Pareto optimum and a generalization of Morse theory. In: Dynamical systems (Proc. Sympos., Univ. Bahia, Salvador, 1971), pp. 531–544. Academic Press, New York (1973)
27. Turner, K., Murkherjee, A., Boyer, D.M.: Persistent homology transform for modeling shapes ans surfaces. Information and Inference **3**(4), 310–344 (2014)
28. Xia, K., Wei, G.W.: Multidimensional persistence in biomolecular data. Journal of Computational Chemistry **36**(20) (2015)