Computational Topology and Random Simplicial Complexes

Masterarbeit

vorgelegt von

Ivan Lothar Arbo Spirandelli

zur Erlangung des akademischen Grades

Master of Science
Mathematik

Betreuer: Priv.-Doz. Dr. Frank Lutz

Zweitgutachter: Prof. Dr. Michael Joswig

Technische Universität Berlin
Fakultät für Mathematik und Naturwissenschaften
Institut für Mathematik

Berlin, den 03.09.2020

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den 03.09.2020

Ivan Lothar Arbo Spirandelli

Abstract

In this thesis we will give a brief introduction on simplicial persistent homology and discrete Morse theory. We will discuss the mathematical background and the most popular algorithms for persistence computation in the Chapters 1–3. In Chapter 4 we will extend some recent discussions of canonically defined persistence pairs, which are closely related to discrete Morse functions. We will formulate and prove a theorem linking them to the number of column additions in the standard algorithm used to compute persistent homology. Furthermore we will introduce a new construction of examples that achieves the worst case bound for this algorithm and compare it to a previously known construction.

In the final Chapter 5 we will do computational experiments on random simplicial complexes with respect to the concepts established in Chapter 4. Here random simplicial complex is meant in the sense of analogues of the Erdös–Rényi random graph model and of alpha complexes constructed on random point clouds.

Zusammenfassung

In dieser Arbeit werden in den ersten Kapiteln die Grundlagen von simplizialer persistenter Homologie und diskreter Morse-Theorie eingeführt. Es werden sowohl die mathematischen Strukturen als auch die Standardalgorithmen der Themengebiete diskutiert. Im vierten Kapitel greifen wir jüngere Erkenntnisse zu kanonisch definierten Persistenzpaaren auf, die in engem Zusammenhang mit diskreter Morse-Theorie stehen, und erweitern diese. Wir formulieren und beweisen ein Theorem, dass sie mit der Laufzeit des Standardalgorithmus zur Berechnung von persistenter Homologie verknüpft. Des weiteren führen wir eine neue Konstruktion von Beispielen ein, die die schlechtest mögliche Laufzeit in Bezug auf den Standardalgorithmus haben. Wir vergleichen die Konstruktion mit einer bereits bekannten Konstruktion dieser Art.

Im letzten Kapitel untersuchen wir zufällige Simplizialkomplexe in Bezug auf die Erkenntnisse aus dem vierten Kapitel. Hierbei meint zufällig sowohl im Sinne von Analoga zu dem zufälligen Graphen Model von Erdös-Rényi, als auch im Sinne der Konstruktion von Alpha-Komplexen auf zufälligen Punktwolken.

Acknowledgements

I'd like to thank Frank Lutz and Paweł Dłotko for their support and guidance during the times of me working on this thesis.

Contents

Chapter 1

Simplicial Complexes

The fundamental concept, that all other concepts discussed in this work will rely upon, is that of simplicial complexes. Simplicial complexes are topological spaces we can describe in a discrete manner and they are of interest in a variety of different topics. In computer graphics, every triangular mesh is a simplicial complex. Meshes are also used to solve physical problems via finite element methods. Simplicial complexes play a role in network theory, in material sciences and in the topological analysis of point clouds. The latter will be one of the central topics in this work.

1.1 Basic Definitions

We assume that the reader is familiar with affine and convex combinations and hulls. For a thorough introduction in discrete geometry we refer you to [JT13] or [EH10], which also are the basis for this section. We begin by defining simplices.

Definition 1.1. A **geometric k-simplex** is the convex hull of $k+1$ affinely independent points in $\mathbb{R}^n$. We call k the **dimension** of the simplex. The empty set is a geometric simplex of dimension -1.

This means a geometric 0-simplex is a point, a geometric 1-simplex is a line, a geometric 2-simplex is a triangle and so on and so forth. We will often omit the number k and just talk about geometric simplices. See the following figure for an example of a geometric 2-simplex and a geometric 3-simplex.

Figure 1.1: A geometric 2-simplex and a geometric 3-simplex

Definition 1.2. [EH10, III.1] Consider some geometric simplex σ of dimension k, which is the convex hull of affinely independent points $S = \{s_0, \dots, s_k\}$. Let γ be the convex hull of a non-empty subset U of S. Then

γ is called a **face** of σ and σ is called **coface** of γ. A face is called **proper**, if $|U| \leq k$. Finally if $\dim(\gamma) = k - 1$, we say that γ is a **facet** of σ and σ is a **cofacet** of γ.

A popular notation, that we will also use, is to write $\tau \leq \sigma$ if τ is a face of σ and $\tau < \sigma$ if it is proper.

We will now introduce a particular kind of collection of simplices in which faces and intersections of simplices are also contained in the collection.

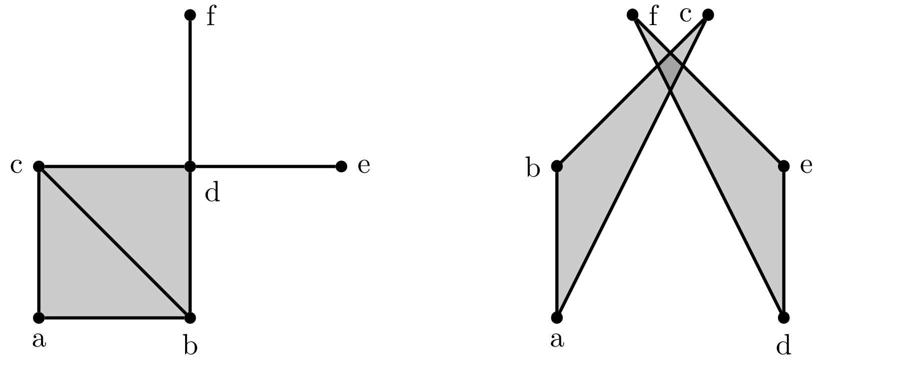
Definition 1.3. Let K be a finite collection of geometric simplices. K is a **geometric simplicial complex** if the following two conditions hold:

1. If $\sigma \in K$ and $\tau < \sigma$, then $\tau \in K$
2. If $\sigma_0, \sigma_1 \in K$ then $\sigma_0 \cap \sigma_1 \in K$

The **dimension** of K is the maximum dimension of any of its simplices. A subset $L \subset K$ is called a **geometric subcomplex** of K if it is a geometric simplicial complex itself. The ***k*-skeleton** of K is the set of all simplices of dimension k or less.

We will refer to the elements of a simplicial complex as its simplices or its **cells**.

The following Figure 1.2 gives (a) an example of a geometric simplicial complex in two dimensions and (b) of a set of geometric simplices, that is not a geometric simplicial complex.



(a) A geometric simplicial complex (b) Not a geometric simplicial complex

Figure 1.2: Two different sets of simplices, where the first collection fulfills the conditions of a geometric simplicial complex and the second does not.

Now when we look at the sets of vertices in each simplex of Figure 1.2 (a) and list them, we get:

Empty set:	$\emptyset,$
Vertices:	$\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\},$
Edges:	$\{a, b\}, \{a, c\}, \{b, c\}, \{b, d\}, \{c, d\}, \{d, e\}, \{d, f\},$
Triangles:	$\{a, b, c\}, \{b, c, d\}.$

Forgetting the explicit coordinates of each vertex and only considering the set of labels we can still see that there is some vertex $\{a\}$ that has two adjacent edges $\{a, b\}$ and $\{a, c\}$ while some other vertices $\{e\}$ and $\{f\}$ only have one adjacent edge. This type of combinatorial data is captured in what we call an abstract simplicial complex.

Definition 1.4. Consider a finite set $V = \{v_1, v_2, \ldots, v_n\}$, which we will call a **vertex set**. Let A be a set of subsets of V. We call A an **abstract simplicial complex** over V if the following condition holds:

- If $\alpha \in A$ and $\beta \subset \alpha$ then $\beta \in A$.

Note that $V = \bigcup_{\sigma \in A} \sigma$.

We call the sets in A **abstract simplices,** and the **dimension** of an abstract simplex σ is its cardinality minus one, i.e.,

$$\dim(\sigma) = |\sigma| - 1.$$

The notions of faces and cofaces are analogous to those of geometric simplicial complexes. The dimension of the abstract simplicial complex A is the maximal dimension of one of its simplices. A subset $B \subseteq A$ is called an (abstract) subcomplex of A if it is an abstract simplicial complex itself.

When comparing the definition of an abstract simplicial complex to the definition of a geometric simplicial complex, we can see that the second condition for geometric simplicial complexes is automatically fulfilled for abstract simplicial complexes.

Now consider a vertex set $V = \{a, b, c, d, e, f\}$. The following set

$$A = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{a, b\}, \{a, c\}, \{b, c\},$$
$$\{b, d\}, \{c, d\}, \{d, e\}, \{d, f\}, \{a, b, c\}, \{b, c, d\}\}$$

is an abstract simplicial complex over V.

If we interpret the sets with two elements in A as edges and those with three elements as triangles, this corresponds to the tabular above, in which we listed the simplices of the geometric simplicial complex in Figure 1.2.

The simplicial complex in Figure 1.2 is realizing the abstract simplicial complex A.

More formally, let

$$g : V \to \mathbb{R}^m$$

be a map that assigns linearly independent coordinates in $\mathbb{R}^m$ to the vertices of an abstract simplicial complex A. The map on the vertices induces a map on each individual abstract simplex $\sigma = \{v_1, \dots, v_i\}$ by sending σ to $\mathrm{conv}(\{g(v_0), \dots, g(v_i)\})$. We abuse notation slightly and also refer to this induced map as g.

In case g is injective and $g(\sigma) \cap g(\tau) = g(\sigma \cap \tau)$ for all pairs $\sigma, \tau \in A$, then g is called a **geometric realization** of A.

Given a geometric simplicial complex K, we obtain an abstract simplicial complex A, by taking the set of vertex labels for each geometric simplex in K. Then A is called a **vertex scheme** of K. Hence, every geometric simplicial complex yields an abstract simplicial complex by simply omitting the coordinate information for the vertices of K. It also holds that every abstract simplicial complex has a geometric realization.

One way to define a geometric realization of an abstract simplicial complex A over vertex set $V = \{v_1, \dots, v_m\}$ is to define $g : V \to \mathbb{R}^m$ as

- $g(v_i) = e_i$, for $i = 1, \dots, m$
- $g(\sigma) = \mathrm{conv}(\{g(v) \mid v \in \sigma\})$, for $\sigma \in A$.

Then $K \coloneqq g(A)$ is a subcomplex of an $(m-1)$-simplex embedded in $\mathbb{R}^m$. We will conclude our discussion of the relation between abstract and geometric simplicial complexes by stating the following theorem, which allows a lower dimensional realization. See [EH10, p.64].

Theorem 1.5 (Geometric Realization Theorem)**.** *Every d-dimensional abstract simplicial complex has a geometric realization in* $\mathbb{R}^{2d+1}$

We have seen how readily the concepts of abstract and geometric simplicial complex can be translated into one another. Therefore from now on, we omit the keywords abstract and geometric, and just talk about simplicial complexes, simplices, etc., unless we explicitly need to distinguish the two concepts. We are usually referring to abstract simplicial complexes. Often however we will see examples in which these abstract simplicial complexes are realized geometrically without discussing explicitly, that the realization is but one possible geometric realization of the abstract simplicial complex.

For brevity, we will introduce the following notation of writing abstract and geometric simplices in square brackets like so:

$$[a, b, c]$$

which stands for the convex hull $\mathrm{conv}(\{a, b, c\})$, i.e., a geometric simplex and/or the set $\{a, b, c\}$, i.e., an abstract simplex.

We have already talked about the notion of subcomplexes. Now we will consider sequences of subcomplexes, that are increasing with respect to inclusion.

Definition 1.6. Let K be a simplicial complex a **filtration** of K is a sequence of subcomplexes $F = \{K_0, \dots, K_n = K\}$ such that $K_i \subseteq K_{i+1}$ for $i = 0, \dots, n-1$.

Note that two consecutive elements of the filtration might be equal, or might differ substantially. An intuitively useful extension of the last definition is the following.

Definition 1.7. Let $F = \{K_0, \dots, K_n\}$ be a filtration. If $K_0 = \emptyset$ and $|K_{i+1} \setminus K_i| = 1$ for all $i \in 0, \dots, n-1$ we call the filtration **simplexwise** and denote it by F_*.

A notation we will use in this work is to write simplexwise filtrations as

$$F_* = (\sigma_1, \dots, \sigma_m),$$

where we mean that F_* is a filtration of $K = \{\sigma_1, \dots, \sigma_m\}$ and subcomplex K_i is the set of the first i simplices, i.e. $K_i = \{\sigma_1, \dots, \sigma_i\}$.

One way to define a filtration is by taking a function $f : K \to \mathbb{R}$ that is monotonously increasing with respect to inclusion, i.e, for $\sigma, \tau \in K$, $\sigma < \tau$ we have $f(\sigma) \leq f(\tau)$.

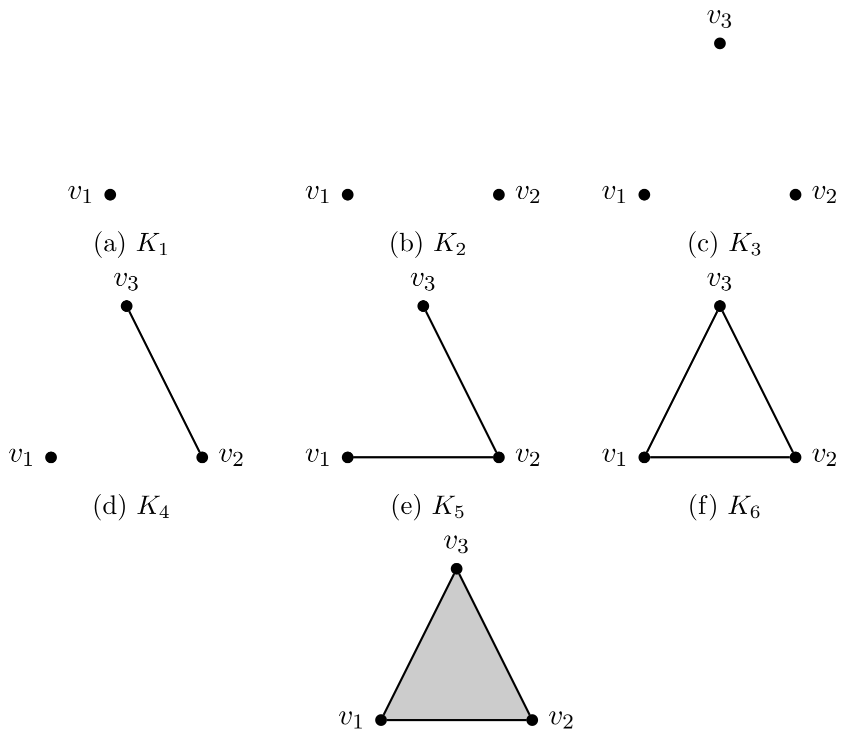
Given such a function we can define subcomplexes of K by looking at the sub level sets of f. Let $a \in \mathbb{R}$, then $f^{-1}((-\infty, a]) \subseteq K$ is a subcomplex of K. Now we can take a sequence of elements in $\mathbb{R}$, $a_0 < a_1 < \cdots < a_n$ and take $f^{-1}(a_0), \dots, f^{-1}(a_n)$ to define a filtration.

On the other hand, given a filtration $F = \{K_0, \dots, K_n\}$ we can define such a function by setting $f(\sigma) = 0$ for $\sigma \in K_0$, and $f(\sigma) = i$ for $\sigma \in K_i \setminus K_{i-1}$ for $i = 1, \dots, n$.

Furthermore for a filtration $F = \{K_0, \dots, K_n = K\}$ and a function f defined in that manner we call $F_* = (\sigma_1, \dots, \sigma_m)$ with $f(\sigma_i) \leq f(\sigma_{i+1})$ and $i < j$ if $\sigma_i < \sigma_j$ a **simplexwise refinement** of F.

Sometimes we want to consider an ordering of that type of the simplices of a filtration F. In this case we will just consider a simplexwise refinement F_* without further discussion.

The following figure gives a simple example of a simplexwise filtration.



(a) K_1 (b) K_2 (c) K_3

(d) K_4 (e) K_5 (f) K_6

(g) $K = K_7$

Figure 1.3: A simplexwise filtration of a triangle

1.2 Filtrations from Point Clouds

As we will see later on, we can use filtrations to analyze how the topology of a simplicial complex changes 'over time', i.e. with respect to the sequence of the filtration. A major practical use for this is to analyze the topology of point clouds. To this end, however, we first need to have a natural way to get from a point cloud to some simplicial complex and a filtration. More detailed information on this can be found in [EH10] and [R.G14].

The first construction we will discuss is the so called Vietoris-Rips complex.

Definition 1.8. Let $S \subset \mathbb{R}^n$ be a finite set of points and consider some $\epsilon > 0$. The set

$$\mathrm{VR}_\epsilon(S) := \{\sigma \subseteq S \mid \mathrm{dist}(s_i, s_j) \leq \epsilon,\, s_i, s_j \in \sigma\}$$

is called **Vietoris-Rips complex** of S.

In other words, $\mathrm{VR}_\epsilon(S)$ is the set of all simplices defined by all sets of finitely many points in S that have a pairwise distance smaller than ϵ.

It is easy to see, that this yields an abstract simplicial complex over the vertex set S. Since all subsets and intersections of sets in $\mathrm{VR}_\epsilon(S)$ are again in $\mathrm{VR}_\epsilon(S)$.

While the Vietoris-Rips complex is straightforward to compute, it will result in simplicial complexes, that obscure lower dimensional topological structure in our data set. This happens since, whenever there is a set of edges, that is the edge set of some high dimensional simplex, this simplex is also included.

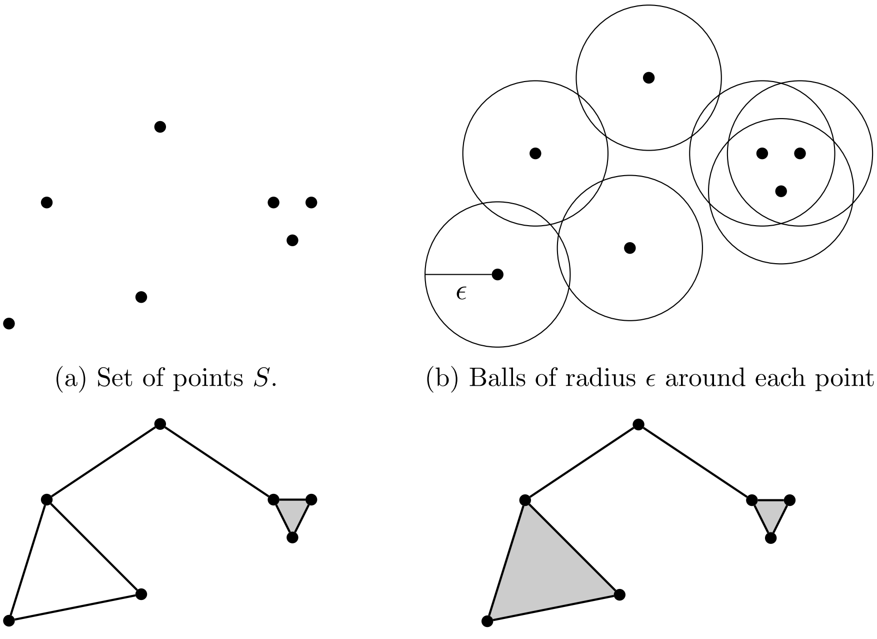
Another construction, that is similar to the Vietoris-Rips complex, but a little more aware of lower dimensional structure is the following.

Definition 1.9. Let $S \subset \mathbb{R}^n$ be finite and let $\epsilon > 0$. The set

$$\check{\mathrm{C}}_\epsilon(S) := \{\sigma \subseteq S \mid \bigcap_{s \in \sigma} B_\epsilon(s) \neq \emptyset\},$$

where $B_\epsilon(s)$ is the closed n-dimensional ball of radius ϵ with center s, is called **Čech complex** of S.

In other words, the Čech complex consists of the simplices, whose vertices have intersecting balls of radius epsilon centered on them. The following Figure 1.4 gives a small example of a Čech and a Vietoris-Rips complex.



(a) Set of points S.

(b) Balls of radius ϵ around each point.

(c) The resulting Čech complex $\check{\mathrm{C}}_\epsilon(S)$.

(d) The Vietoris-Rips complex $\mathrm{VR}_\epsilon(S)$.

Figure 1.4: Example of a Čech and a VR complex.

The above figure illustrates how a Čech and a Vietoris-Rips complex differ for the same value ϵ. It generally holds, that $\check{\mathrm{C}}_\epsilon(S) \subseteq \mathrm{VR}_\epsilon(S) \subseteq \check{\mathrm{C}}_{\sqrt{2}\epsilon}(S)$

which is proven in [EH10][Chapter III]. A notable property of both is that they not necessarily have a geometric realization in the dimension of the underlying point set. For example, let us consider the points in Figure 1.4(b). If we choose our radius ϵ large enough, the resulting simplicial complex will be a 6-simplex on seven vertices with all its faces, which has a geometric realization in spaces of dimension six or higher, while the underlying point set is in dimension two.

We will introduce a final construction, that has a geometric realization in the ambient space of the underlying point set. This is the so called Alpha Complex, which in a sense mixes the construction of Čech complexes with Voronoi diagrams.

Definition 1.10. Let $S \subset \mathbb{R}^n$ be finite set of points. The **Voronoi cell** of a point $q \in S$ is the set of points that is not closer to any other point in S than q, i.e.

$$V_q := \{x \in \mathbb{R}^n \mid \|x - q\| \leq \|x - s\|, s \in S \setminus \{q\}\}.$$

The **Voronoi diagram** of S is the set of all its Voronoi cells.

In other words, a Voronoi diagram is a partitioning of $\mathbb{R}^n$ into areas that are closest to a particular point in the set S. The intersections of the Voronoi cells, are areas, in which several points of S are equally close.

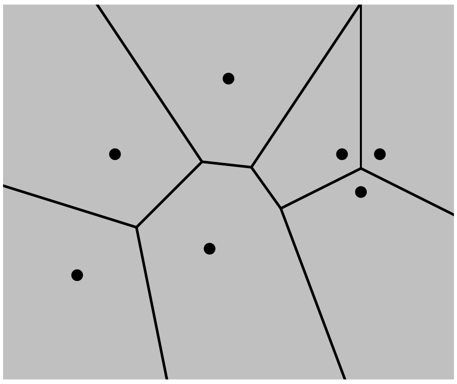


Figure 1.5: Voronoi diagram of the point set from Figure 1.4 (a).

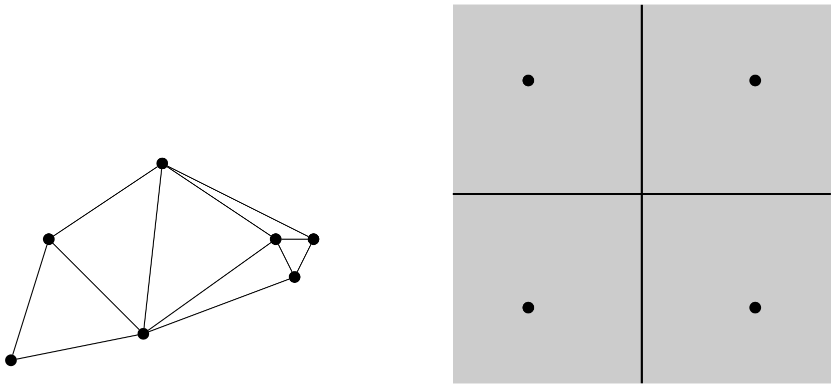
Definition 1.11. Let $S \in \mathbb{R}^n$ be a finite set of points. Let $V(S)$ be the Voronoi diagram of S. Then

$$D(S) := \{\sigma \subseteq S \mid \bigcap_{s \in \sigma} V_s \neq \emptyset\}$$

is the **Delaunay complex** of S.

If the points of a finite point set $S \subset \mathbb{R}^n$ are in general position, in the sense that no $n+2$ points lie on a common n-sphere, then the Delaunay complex is a triangulation of the convex hull of S and is called the **Delaunay triangulation**.

The following Figure 1.6 (a) shows the Delaunay triangulation corresponding to the Voronoi diagram from Figure 1.5. Figure 1.6 (b) shows a Voronoi diagram in two dimensions on a set of four points that lie on a circle. The corresponding Delaunay complex has a three dimensional cell, i.e., is not a triangulation of the convex hull of the point set.



(a) Delaunay triangulation corresponding to Figure 1.5.

(b) Voronoi diagram on points not in general position.

Figure 1.6: Illustrating Delaunay triangulations and complexes.

We will now combine Voronoi diagrams with the constructing idea of Čech complexes.

Definition 1.12. We define the **alpha complex** of a finite set of points $S \subset \mathbb{R}^n$ as

$$\text{Alpha}(\epsilon) := \{\sigma \subseteq S \mid \bigcap_{s \in \sigma} (B_\epsilon(s) \cap V_s) \neq \emptyset\},$$

with $B_\epsilon(s)$ being the ball of radius ϵ centered at s and V_s being the Voronoi cell of s.

It is apparent from the definition, that if we choose ϵ large enough, so that each ball $B_\epsilon(s)$ covers all points in S, all simplices will be defined by the incidences in the Voronoi diagram. Hence, for S in general position we can take the convex hulls of the abstract simplices defined by our Alpha complex and will get a geometric realization in the ambient space of the set of points. See the following example, where the resulting alpha complex is equal to the Čech complex from Figure 1.5.

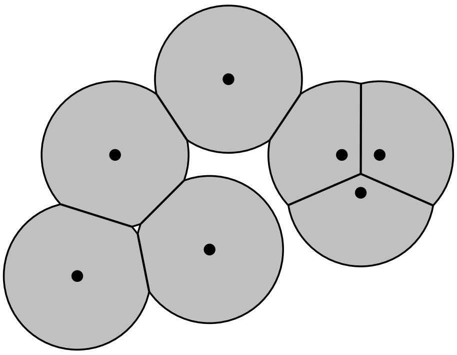


Figure 1.7: The intersection of balls depicted in figure 1.4 (b) and the Voronoi cells from figure 1.5.

Now for any of these constructions we can use different parameters $\epsilon_0, \dots, \epsilon_n$ to obtain a sequence of nested simplicial complexes, i.e., a filtration.

If the number of simplices at some ϵ_i actually increases, which only happens finitely many times, we can assign each new simplex the value ϵ_i to get a function $f : K \to \mathbb{R}$, which describes the given filtration, via the sub level sets as discussed before.

1.3 Simplicial Collapses

We conclude this chapter on simplicial complexes by introducing the notion of simplicial collapses. An early reference to these deformations can be found in [Whi39].

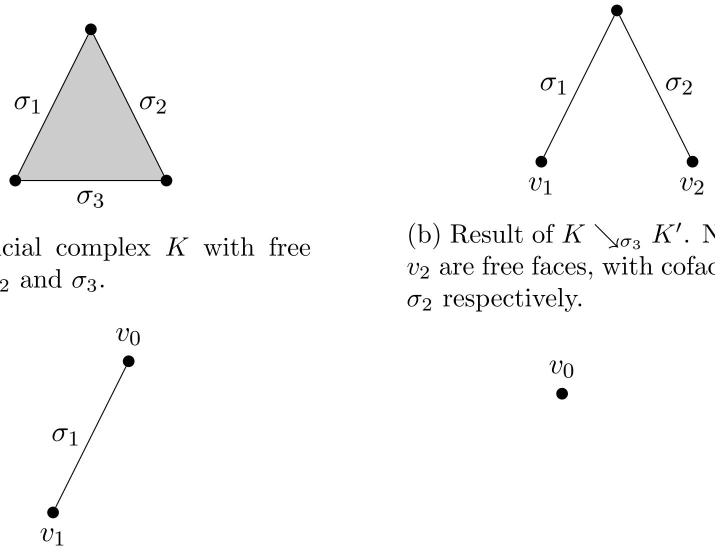
Definition 1.13. Let K be a simplicial complex and let $\sigma, \tau \in K$. Let σ be a facet of τ. We call σ a **free face** if the following conditions are fulfilled:

- σ has no cofacet except τ
- τ has no cofacet

Definition 1.14. Let K be a simplicial complex and let σ be a free face with coface τ.

Define $C_\sigma \coloneqq \{\gamma \in K \mid \sigma \leq \gamma \leq \tau\}$. Constructing $K' \coloneqq K \setminus C_\sigma$ by removing the simplices in C_σ from K is called a **simplicial collapse**. If $\dim(\sigma) = \dim(\tau) - 1$, we call the collapse **elementary**. We denote an (elementary) simplicial collapse by $K \searrow_\sigma K'$.

Note that K' is a subcomplex of K. In the following example, Figure 1.8, we start with a triangle as simplicial complex K and successively collapse free faces until we have reached a single point.



(a) Simplicial complex K with free faces σ_1, σ_2 and σ_3.

(b) Result of $K \searrow_{\sigma_3} K'$. Now v_1 and v_2 are free faces, with cofacets σ_1 and σ_2 respectively.

(c) Result of $K' \searrow_{v_2} K''$. Now v_0 and v_1 are free faces.

(d) Result of $K'' \searrow_{v_1} K'''$.

Figure 1.8: A sequence of elementary collapses.

If there is a sequence of collapses, such that some simplicial complex K collapses to a point by following that sequence, we say that K is a **collapsible**.

In topology, a space is **contractible**, if it is homotopy-equivalent to a point. Every collapsible complex is contractible. However contractible does not imply collapsible as the famous example of the „Dunce hat“, see Figure 1.9, illustrates. More on this topic can be found in [Coh73].

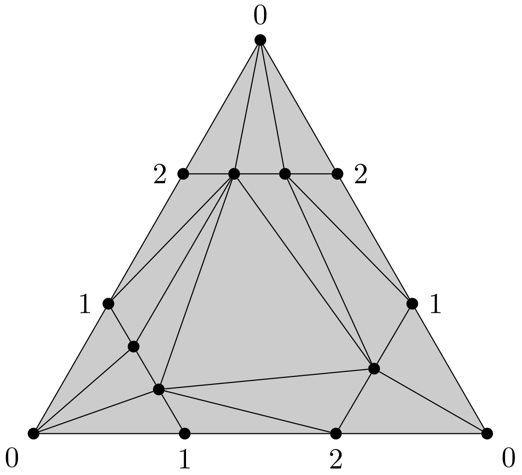


Figure 1.9: The dunce hat.

In Figure 1.9 vertices with the same label indicate that the respective vertices are identified. In this example the sides of the triangle we see here are therefore glued together.

It is easy to verify that the dunce hat has no free face and therefore is not collapsible. However it is contractible, c.f. [Zee63].

Chapter 2

Homology

2.1 Homology Groups

This section is mainly based on [EH10][Chapter IV]. Homology groups are topological invariants that describe holes in topological spaces. The topological spaces we will consider are simplicial complexes. Restricting the theory in this way is also referred to as simplicial homology. We will begin by introducing chains of simplices. A handy notation we will use in the following definition is to write $\sigma^{(k)} \in K$ for a k-dimensional simplex in some simplicial complex K. All concepts we will discuss work just as well when we include the empty face but since nothing that is of interest to us happens, when including it, we will omit it for brevity. Strictly speaking in all our examples and concepts we will discuss simplicial complexes without the empty face. Note that including the empty face yields reduced homology theory which is different on an algebraic level.

Definition 2.1. Let K be a simplicial complex. Let $K^{(k)} := \{\sigma^{(k)} \in K\}$ be the set of k-simplices in K. A formal sum

$$c = \sum_{i=1}^{|K^{(k)}|} a_i \sigma_i$$

with $a_i \in \mathcal{R}$, where $\mathcal{R}$ is a ring, and $\sigma_i \in K^{(k)}$, $i \in 1, \ldots, |K^{(k)}|$, is called a ***k*-chain**. We denote the set of all k-chains of K by $C_k(K)$.

In general, we will omit the bounds of the sum and just write $\sum a_i \sigma_i$. Furthermore we will omit the reference to the simplicial conmplex K and just write C_k.

Definition 2.2. We define an addition on the set of k-chains C_k:

$$+ : C_k \times C_k \to C_k,$$
$$\sum a_i \sigma_i + \sum b_i \sigma_i := \sum (a_i + b_i) \sigma_i.$$

Lemma 2.3. *The k-chains together with the above defined addition form an abelian group that we denote as* $(C_k, +)$

Proof. Associativity follows from the associativity of the coefficients from the ring $\mathcal{R}$. The neutral element is $0 := \sum 0\sigma_i^{(k)}$ and each chain $c = \sum a_i\sigma_i^{(k)}$ has an inverse $-c := \sum(-a_i)\sigma_i^{(k)}$. The group is abelian, since addition in $\mathcal{R}$ is abelian. □

As mentioned above, we want to describe holes in a simplicial complex. We do this by considering their boundary. The following definition introduces a boundary operator for simplices that naturally extends to k-chains.

Definition 2.4. Let $\sigma^{(k)} = [s_0, \dots, s_k]$ be some simplex. We define the **boundary operator**

$$\partial_k\sigma := \sum_{j=0}^{k}(-1)^j[s_0, \dots, \hat{s_j}, \dots, s_k]$$

where $[s_0, \dots, \hat{s_j}, \dots, s_k]$ is the facet of σ that does not contain s_j. For a k-chain $c = \sum a_i\sigma_i^{(k)}$ the boundary operator is defined as

$$\partial_k c = \sum a_i\partial_k\sigma_i.$$

The boundary operator maps k-chains to $(k-1)$-chains, and since

$$\partial_k(c_1 + c_2) = \partial_k c_1 + \partial_k c_2$$

the boundary operator

$$\partial_k : C_k \to C_{k-1}$$

is a group homomorphism between the groups of k-chains and $(k-1)$-chains. We can now introduce k-cycles and k-boundaries and their respective groups.

Definition 2.5. Let c be a $\boldsymbol{k}$**-chain**. If $\partial_k c = 0$, then c is called a $\boldsymbol{k}$**-cycle**. Consider a $(k+1)$-chain r. If $c = \partial_{k+1}r$, then c is called a $\boldsymbol{k}$**-boundary**.

By definition, ∂_k commutes with addition and hence we get the following groups. The group of $\boldsymbol{k}$**-cylces** denoted as $Z_k = Z_k(K)$ and the group of $\boldsymbol{k}$**-boundaries** $B_k = B_k(K)$. Both Z_k and B_k are subgroups of C_k. Another perspective on these groups is to interpret them as kernel and image of ∂, i.e. $Z_k = \ker(\partial_k)$ and $B_k = \operatorname{im}(\partial_{k+1})$.

Since C_k is abelian, so are Z_k and B_k. The following fundamental lemma further specifies the relation between C_k and B_k.

Lemma 2.6 (Fundamental Lemma of Homology)**.** $\partial_k\partial_{k+1}c = 0$ *for* $k \in \mathbb{N}_0$ *and* $(k+1)$*-chain* c*.*

Proof. It suffices to show that $\partial_k\partial_{k+1}\tau = 0$ for every $(k+1)$-simplex τ since a $(k+1)$-chain is just a sum of $(k+1)$-simplices.

Let $\tau = [s_0, \dots, s_{k+1}]$. Consider the boundary $\partial_{k+1}\tau$. It consists of all k-faces of τ.

Let $\gamma = [s_0, \dots, \hat{s_i}, \dots, \hat{s_j}, \dots, s_{k+1}]$, i.e., a $(k-1)$-face. It is adjacent to the two faces k-faces

$$\sigma_i = [s_0, \dots, \hat{s_i}, \dots, s_{k+1}] \text{ and } \sigma_j = [s_0, \dots, \hat{s_j}, \dots, s_{k+1}]$$

Considering the different possibilities for i and j to be even or odd, a straightforward calculation yields that γ cancels out in the sum corresponding to $\partial_k\partial_{k+1}\tau$. Hence $\partial_k\partial_{k+1}\tau = 0$. □

From the lemma we immediately get that B_k is a subgroup of Z_k. There is a popular way of depicting these group relations, found in most literature concerned with homology. The following Figure 2.1 is our version of it.

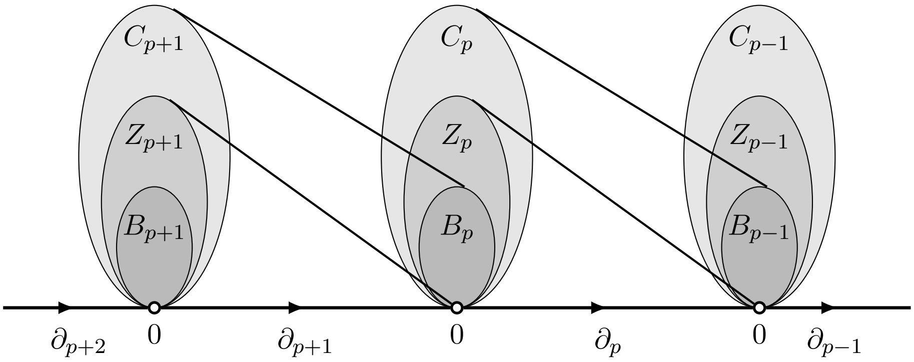


Figure 2.1: Connections between cycle and boundary groups, established by the group homomorphisms ∂.

Since B_k is a subgroup of Z_k, we are allowed to take the group quotient Z_k/B_k. This leads us to the definition of Homology groups and Betti numbers.

Definition 2.7. We define the ***k*-th Homology group** as

$$H_k := Z_k/B_k$$

and the ***k*-th Betti number** as

$$\beta_k := \text{rank}(H_k).$$

What we will see is that, informally speaking, β_k counts the number of k-dimensional holes in the underlying simplicial complex, except for β_0, which is counting the connected components.

The k-th homology group can be interpreted as the partitioning of the k-th cycle group Z_k into classes of cycles that differ by boundaries of the

k-th boundary group B_k. From algebra we know, that $H_k = \{zB_k \mid z \in Z_k\}$ forms a group with

$$+ : H_k \times H_k \to H_k, z_1 B_K + z_2 B_k = (z_1 + z_2)B_k$$

where the + on the right hand side is the addition of k-chains defined in 2.2.

From here on out we will restrict our coefficients to be from the field $\mathbb{Z}_2$, since this suffices for all further concepts discussed in this work and it simplifies some definitions and calculations. For example the coefficients of k-chains now just indicate if some simplex is contained in the chain or not. Furthermore we can drop the $(-1)^j$ in the definition of the boundary operator, since in $\mathbb{Z}_2$ we have $-1 = 1$. As we will discuss later on it also simplifies the standard algorithm to compute homology groups.

In the following example we will calculate the Betti number β_1 of the simplicial complex depicted in the following Figure 2.2 by computing the factorization of Z_1 by B_1. The classes resulting from this factorization are called **homology classes** and any two elements within one class are called homologous. Some of the simplifications arising from choosing coefficients from $\mathbb{Z}_2$ will become apparent.

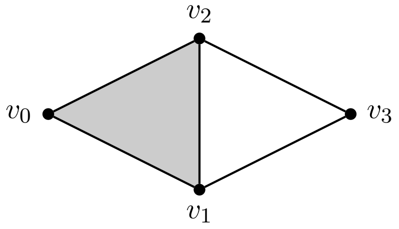


Figure 2.2: A simplicial complex.

To calculate H_1 we need to know what B_1 and Z_1 are. Let t_1 be the triangle with vertices v_0, v_1 and v_2. Then $\partial_2(t_1) = [v_0, v_1] + [v_0, v_2] + [v_1, v_2]$. Since t_1 is the only 2-simplex we get

$$B_1 = \{0, [v_0, v_1] + [v_0, v_2] + [v_1, v_2]\}.$$

To compute the cycle group, we need to find all 1-chains that are mapped to 0 by ∂_1, i.e. we have to compute the kernel. We can do this by solving:

$$\begin{aligned} 0 &= a_0\partial_1[v_0, v_1] + a_1\partial_1[v_0, v_2] + a_2\partial_1[v_1, v_2] + a_3\partial_1[v_1, v_3] + a_4\partial_1[v_2, v_3] \\ &= a_0(v_0 + v_1) + a_1(v_0 + v_2) + a_2(v_1 + v_2) + a_3(v_1 + v_3) + a_4(v_2 + v_3) \\ &= v_0(a_0 + a_1) + v_1(a_0 + a_2 + a_3) + v_2(a_1 + a_2 + a_4) + v_3(a_3 + a_4) \end{aligned}$$

for coefficients $a_i \in \mathbb{Z}_2$. So we get the set of equations:

$$\begin{array}{lll} (1) & 0 & = a_0 + a_1 \\ (2) & 0 & = a_0 + a_2 + a_3 \\ (3) & 0 & = a_1 + a_2 + a_4 \\ (4) & 0 & = a_3 + a_4 \end{array}$$

From equation (1) we get that $a_0 = a_1$ and from (4) we get that $a_3 = a_4$. Therefore equations (2) and (3) tell us if $a_2 = 0$ then $a_0 = a_3$ and $a_1 = a_4$. This implies two possible cycles. Namely the 0-cycle and

$$z_1 = [v_0, v_1] + [v_1, v_3] + [v_2, v_3] + [v_0, v_2].$$

On the other hand. If $a_2 = 1$, we get that either $a_0 = a_1 = 1$ or $a_3 = a_4 = 1$, which implies the cycles

$$z_2 = [v_0, v_1] + [v_0, v_2] + [v_1, v_2]$$

and

$$z_3 = [v_1, v_2] + [v_1, v_3] + [v_2, v_3].$$

These are all possible cycles, so overall we get

$$Z_1 = \{0, z_1, z_2, z_3\}.$$

By adding all elements of B_1 to each element of Z_1 we get

$$\begin{array}{rclcl} 0 + B_1 & = & \{0, z_2\} & & \\ z_1 + B_1 & = & \{z_1 + 0, z_1 + z_2\} & = & \{z_1, z_3\} \\ z_2 + B_1 & = & \{z_2 + 0, z_2 + z_2\} & = & \{z_2, 0\} \\ z_3 + B_1 & = & \{z_3 + 0, z_3 + z_2\} & = & \{z_2, z_3\} \end{array}$$

This means, that Z_1 is partitioned into two classes by factorisation with B_1. Namely $\{\{0, z_2\}, \{z_1, z_3\}\}$. Now we choose elements representing both classes in our example. We will pick $[0]$ and $[z_3]$ and hence

$$H_1 = \{[0], [z_3]\}.$$

Therefore $\beta_1 = \text{rank}(H_1) = 1$, which corresponds to the one-dimensional hole bounded by the edges between vertices v_1, v_2 and v_3.

2.2 Boundary Matrices and Smith Normal Form

At the end of the last section we did the calculations by hand to illustrate the discussed concepts. In practice, however, we are often only concerned with the Betti numbers and not with the different homology groups. Therefore we only need to compute the ranks of Z_k and B_k. As we will see, we can encode ∂ as a matrix over $\mathbb{Z}_2$. Then we can use standard linear algebra to extract the desired information. For more details we refer the reader to [EH10][Chapter IV.2], which this section is based on.

Definition 2.8. Let K be a simplicial complex and assume an arbitrary but fixed ordering of the simplices. Let n_{k-1} be the number of $(k-1)$-simplices and n_k be the number of k-simplices. Let $i \in \{1, \ldots, n_{k-1}\}$ and $j \in \{1, \ldots, n_k\}$, then we define a matrix $\partial_k = [a_{ij}]$ with $a_{ij} = 1$, if the i-th $(k-1)$-simplex is a face of the j-th k-simplex. Otherwise $a_{ij} = 0$. This matrix is called the ***k*-th boundary matrix**.

The 1-st boundary matrix of the example in Figure 2.2 is the following

$$\begin{array}{c} \\ [v_0] \\ [v_1] \\ [v_2] \\ [v_3] \end{array} \begin{array}{c} \begin{array}{ccccc} [v_0, v_1] & [v_0, v_2] & [v_1, v_2] & [v_1, v_3] & [v_2, v_3] \end{array} \\ \left[\begin{array}{ccccc} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{array} \right] \end{array}$$

The rows of the k-th boundary matrix, form a basis of C_{k-1} and the columns form a basis of C_k. From linear algebra we know that we can exchange and add rows and columns, without changing the rank of the matrix. Furthermore, as is illustrated in [EH10][Chapter IV] these alterations always yield rows and columns that represent a basis of the k and $k-1$-chains.

The standard algorithm to extract the ranks of the chain groups is to reduce the boundary matrix to the so called **Smith normal form** (SNF).

In the general case, the computation of the SNF allows to read off the Betti numbers. In case that the chosen ring $\mathcal{R}$ is a field, the groups $Z_k = \ker \partial_k$ and $B_k = \operatorname{im} \partial_{k+1}$ can be obtained by standard Gaussian elimination. In particular, this applies to the choice $\mathcal{R} = \mathbb{Z}_2$.

Now consider some k-th boundary matrix M_k. And let $\hat{M}_k$ be the matrix in Smith normal form, we get by reducing M_k via row and column operations. As stated, the number of columns n_k is the rank of C_k. Now let

$$n_k = b_{k-1} + z_k,$$

where b_{k-1} is equal to the number of columns containing a one in $\hat{M}_k$ and z_k the number of columns which have only zeroes for entries. The leftmost b_{k-1} columns of $\hat{M}_k$ represent k-chains that form $(k-1)$-boundaries and the rightmost z_k columns of $\hat{M}_k$ represent k-cycles that generate Z_k. This means the rank of Z_k is z_k and the rank of B_{k-1} is b_{k-1}. Therefore, if we have all the boundary matrices of some simplicial complex K and reduce them to Smith normal form, we get the Betti numbers $\beta_k = \text{rank}(Z_k) - \text{rank}(B_k)$.

2.3 Persistent Homology

Persistent homology is one of the central concepts in Topological Data Analysis. It enables us to extract and describe the topological structure of high-dimensional spaces.

In the last section, we established a way to talk about holes in simplicial complexes. We already established the notion of a filtration in Definition 1.6. What we will now discuss is a theory that allows us to talk about how holes behave with respect to the sequence of complexes we get from a filtration. Recall that the sequence encoded in a filtration is increasing with respect to inclusion. This section is based on [ELZ02][Chapter VII] and [ZC05].

Consider some filtration $F = \{K_0, \dots, K_n\}$, we denote the set of their chain groups by C_*^i, for $i \in 0, \dots, n$. Similarly we denote the corresponding cycle and boundary groups by Z_*^i and B_*^i respectively.

For $i, j \in 0, \dots, n$, $i \leq j$, we have an inclusion map from K_i to K_j, and therefore for every dimension k we have an induced group homomorphism

$$h_k^{i,j} : H_k(K_i) \to H_k(K_j).$$

This means that the sequence of simplicial complexes gives us a sequence of homology groups connected by group homomorphisms,

$$0 = H_k(K_0) \to \cdots \to H_k(K_n) = H_k(K).$$

Definition 2.9. For $0 \leq i \leq j \leq n$, we refer to the factor groups

$$H_k^{i,j} = \frac{Z_k^i}{Z_k^i \cap B_k^j}$$

as the ***k*-th persistent homology groups**. The ranks of these groups are the ***k*-th persistent Betti numbers**.

These groups are well-defined, since the denominator is an intersection of two subgroups of C_k^{i+p} and hence a subgroup of the numerator. We will refer to the set of chain groups C_*^i as the ***i*-th frame**.

Definition 2.10. Let $\gamma \in H_k(K_i)$. We say γ is **born** in frame i, if $\gamma \notin H_k^{i-1,i}$. We say γ **dies** in frame j, if

$$h_k^{i,j-1}(\gamma) \notin H_k^{i-1,j-1} \text{ but } h_k^{i,j}(\gamma) \in H_k^{i-1,j}.$$

We call the difference $j - i$ the **index persistence**. If some simplex never dies, its index persistence equals infinity.

When considering some function $f : K \to \mathbb{R}$, that defines a filtration via sublevel sets of $a_0, \dots, a_n$, we call the difference $a_j - a_i$ the **persistence** of the classes born at a_i and dying at a_j.

We now have a notion of how topological features of our space persist through different steps of a filtration of some simplicial complex K. In the following section we will introduce the standard algorithm with which persistent homology can be computed.

2.4 Matrix Reduction

The standard algorithm to compute persistent homology is a variant of the matrix reduction, used to compute homology via the Smith normal form.

Consider a simplexwise filtration F_* of some simplicial complex K. Let $\sigma_0, \ldots, \sigma_m$ be the ordering of all simplices in K, induced by F_*. Due to F_* being a filtration, we know that $\sigma_i < \sigma_j$ implies $i < j$. Note that every subsequence of simplices starting at σ_0 is a simplicial complex and an element of F_*. We define the boundary matrix

$$\partial[i,j] = \begin{cases} 1 & \text{if } \sigma_i \text{ is a facet of } \sigma_j, \\ 0 & \text{else.} \end{cases}$$

This matrix contains the same information as all of the k-th boundary matrices combined. Each column and row correspond to one simplex. Consider the column corresponding to σ_j, i.e., column j. Every row that has a one as entry in column j corresponds to a facet of σ_j. Conversely, the ones in some row i correspond to cofacets of σ_i.

Definition 2.11. Consider some matrix M with $m+1$ rows and elements in $\mathbb{Z}_2$. For each column j we define

$$\text{low}(j) = \begin{cases} \max\{i \in 0, \ldots, m \mid M[i,j] = 1\} & \text{if column } j \text{ is non-zero,} \\ -1 & \text{else.} \end{cases}$$

The operator low() gives us the index of the **lowest 1** of a column of a matrix, if it exists. What the following algorithm will do, is reduce the boundary matrix ∂ to a matrix, such that for any two columns i and j, it holds that $\text{low}(i) \neq \text{low}(j)$, if $i \neq j$ and if the respective lowest entries exist.

We will refer to this matrix as the reduced matrix and denote it as R. Simultaneously, the algorithm will keep track of the column additions we need to reduce the boundary matrix. We can store this information in a matrix which we will call A for addition matrix. We will write $R[*,j]$, when referring to column j of matrix R. Furthermore we write $\text{low}_R()$, to indicate which lowest elements we are interested in.

```
   Input  : boundary matrix $\partial$
   Output: reduced matrix $R$ and addition matrix $A$
1  $R = \partial$
2  $A = I_m$
3  for $j$ = 1,...,$m$ do
4      while $\mathrm{low}_R(i) \neq -1$ and
5      there exists $i < j$, with $\mathrm{low}_R(i) = \mathrm{low}_R(j)$ do
6          $R[*, j] = R[*, j] + R[*, i]$
7          $A[i, j] = 1$
8      end
9  end
10 return $R, A$
```

Algorithm 1: Column reduction algorithm

In lines 1 and 2 we just initialize our matrices. Line 3 starts a for-loop over all columns, which is in $\mathcal{O}(m)$. The while loop from line 4 to 8 is also in $\mathcal{O}(m)$. After the while loop, the column $R[*, j]$ is either all zero, or has a lowest entry differing from all lowest entries of preceding columns. To achieve this, we add columns $R[*, i]$ and $R[*, j]$, if they have the same lowest entry. This also runs in $\mathcal{O}(m)$ since each column has m entries. Overall we get a cubic running time in the worst case.

After adding two columns we are guaranteed to have differing lowest entries for them. Also the index of the lowest entry of a column decreases with each step or the column becomes all zero. Hence the while loop terminates. The matrices we get returned in line 10 fulfill:

$$R = \partial A.$$

Definition 2.12. For a boundary matrix ∂ and a column j representing simplex τ we denote the number of iterations of the while loop starting in line 4 of Algorithm 1 by $\mathrm{red}(\tau)$. We refer to this number as the number of **column reduction steps of $\boldsymbol{\tau}$**. We call the sum of all column reduction steps over all columns the **column reduction steps of $\boldsymbol{B}$** and denote it by $\mathrm{red}(B)$.

The column reduction steps do not count the total number of addition operations needed during the reduction of a column or boundary matrix. It only counts how often columns get added. In each individual column addition there can be $\mathcal{O}(n)$ addition operations needed to add the columns. We discuss an example later on in the section.

Similarly to what we discussed in Section 2.2, we can extract the Betti numbers of the simplicial complex K via the ranks of the cycle and boundary groups of K. The rank of Z_k is the number of zero columns corresponding to k-simplices in the reduced boundary matrix. We will denote these by $\#Zero_k$. The rank of B_k equals the number of rows corresponding to k-simplices that contain a lowest 1. We denote it by $\#Low_k$. Therefore for the kth Betti number we get

$$\beta_k = \#Zero_k - \#Low_k.$$

This however is not the primary goal of our algorithm. We are interested in the appearance and disappearance of cycles in our persistent homology groups. The lowest entries in our reduced matrix define a pairing between simplices. Namely, if $\mathrm{low}_R(j) = i$, the simplex corresponding to row i is paired with the simplex corresponding to column j. We will explain what these pairings mean, after formulating a lemma, which shows that this pairing depends on the boundary matrix ∂ and not on R, which is characterized by being reduced and calculated by left to right column operations. To this end we introduce the following notations.

The lower left sub matrix of some matrix M with top right corner element $M[i, j]$ is denoted as M_i^j, i.e., M_i^j consists of the first j columns of M and the last $m - i$ rows. Furthermore, we define

$$r_\partial(i, j) \coloneqq \mathrm{rank}(\partial_i^j) - \mathrm{rank}(\partial_{i+1}^j) + \mathrm{rank}(\partial_{i+1}^{j-1}) - \mathrm{rank}(\partial_i^{j-1}).$$

Note that the left to right column operations we do in our algorithm do not change the rank of a matrix, or a submatrix for that matter.

Therefore we get that $\mathrm{rank}(\partial_i^j) = \mathrm{rank}(R_i^j)$. Hence $r_\partial(i, j) = r_R(i, j)$ for all i and j. The following lemma and proof follow [EH10][VII.1] closely.

Lemma 2.13 (Pairing lemma). *It holds that $i = \mathrm{low}_R(j)$ if and only if $r_\partial(i, j) = 1$. This means the pairings defined by the lowest 1s of R do not depend on R.*

Proof. Any linear combination of non zero columns in R_i^j is again non zero, since the lowest 1s are unique. Hence we get that the rank of R_i^j is equal to its number of non zero columns. We will now consider several cases. In the first case $R[i, j]$, the top right corner element of R_i^j is the lowest 1 of $R[*, j]$. It follows that $\mathrm{rank}(R_{i+1}^j) = \mathrm{rank}(R_{i+1}^{j-1}) = \mathrm{rank}(R_i^{j-1}) = \mathrm{rank}(R_i^j) - 1$. This holds, since in all of these sub matrices, we either cut away row i or column j. So column j either becomes a zero column or is cut away. Let us set $x = \mathrm{rank}(R_{i+1}^j)$, then:

$$r_\partial(i, j) = r_R(i, j) = x + 1 - x + x - x = 1.$$

If $R[i, j]$ is not a lowest 1 there are two subcases to consider.

Firstly, let us assume, that none of the columns $1, \ldots, j - 1$ has a lowest 1 in row i. Then R_i^j and R_{i+1}^j have the same rank, because cutting away row i does not produce a new zero column. The same holds for R_i^{j-1} and R_{i+1}^{j-1}, and since in both cases column j is either a zero or a non zero column, their ranks are also the same. In this case we get $r_\partial(i,j) = r_R(i,j) = 0$.

Secondly, if some column $1, \dots, j-1$ has its lowest 1 in row i it follows that removing row i yields a zero column in any resulting sub matrix. Therefore R_{i+1}^{j-1} has one more zero column than R_i^{j-1} and R_{i+1}^{j} has one more zero column than R_i^j. This again results in $r_\partial(i,j) = r_R(i,j) = 0$. With this the proof is concluded. □

We will now discuss what these pairings mean. Either column j of R is zero. In this case the appearance of σ_j in our simplexwise filtration creates a new cycle and therefore gives birth to a new homology class. This is why we call these simplices positive.

Or column j of R has a lowest 1 in row i. Column j of R stores the boundary of the chain represented by column j of matrix A. This means, that the addition of the simplex corresponding to this column causes a homology class to die. Furthermore this class is born in frame i, since one of its cycles just died in column j and all other cycles that died with it, have lowest 1s below row i. If they had a lowest 1 in row i we could have reduced the matrix further and would have obtained $\text{low}(j) < i$, which is a contradiction to our algorithm.

This means, that the pair (i,j) gives us the index persistence $j-i$ of the respective homology class. Furthermore if there is some positive simplex σ_i that never gets paired with a negative one, the homology class appearing at i has an index persistence of infinity.

As an example consider the simplicial complex from Figure 2.2. And let

$$F_* = ([v_0],[v_1],[v_2],[v_3],[v_0,v_1],[v_0,v_2],[v_1,v_2],[v_1,v_3],[v_2,v_3],[v_0,v_1,v_2]),$$

be a simplexwise filtration. The following matrix B is the reduced boundary matrix of F_* in which we cut away all columns corresponding to vertices. See Matrix 2.2 for the relevant part of the initial boundary matrix.

$$\begin{array}{c|cccccc|}
 & [v_0,v_1] & [v_0,v_2] & [v_1,v_2] & [v_1,v_3] & [v_2,v_3] & [v_0,v_1,v_2] \\
[v_0] & 1 & 1 & 0 & 0 & 0 & 0 \\
[v_1] & 1 & 0 & 0 & 1 & 0 & 0 \\
[v_2] & 0 & 1 & 0 & 0 & 0 & 0 \\
[v_3] & 0 & 0 & 0 & 1 & 0 & 0 \\
[v_0,v_1] & 0 & 0 & 0 & 0 & 0 & 1 \\
[v_0,v_2] & 0 & 0 & 0 & 0 & 0 & 1 \\
[v_1,v_2] & 0 & 0 & 0 & 0 & 0 & 1 \\
[v_1,v_3] & 0 & 0 & 0 & 0 & 0 & 0 \\
[v_2,v_3] & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

Column $[v_1, v_2]$ is a zero column since when the simplex $[v_1, v_2]$ is added in the filtration a hole appears that has $[v_0, v_1]$, $[v_0, v_2]$ and $[v_1, v_2]$ as its boundary. The hole is closed when simplex $[v_0, v_1, v_2]$ is added. This means that $[v_1, v_2]$ and $[v_0, v_1, v_2]$ are a persistence pair, which is also indicated by the lowest 1 in column $[v_0, v_1, v_2]$. Another cycle appears when $[v_2, v_3]$ is added in the filtration. This hole never gets closed. This also implies that there is no column corresponding to a triangle which has a lowest 1 in row $[v_2, v_3]$. Note that the reduction of both columns requires additions with other columns. Column $[v_1, v_2]$ is reduced by adding columns $[v_0, v_2]$ and $[v_0, v_1]$. Column $[v_2, v_3]$ is reduced by adding columns $[v_1, v_3]$, $[v_0, v_2]$ and $[v_0, v_1]$. This means the number of column reduction steps $\text{red}(B)$ equals

$$\text{red}([v_1, v_2]) + \text{red}([v_2, v_3]) = 2 + 3 = 5.$$

2.5 Reduction with a Twist

As we have discussed in the last section, the lower ones are not dependent on our reduction strategy. A strategy developed in [CK11] is to do the reduction steps from higher to lower dimensions. Consider some simplicial complex K and filtration F. We first reduce the highest dimensional simplices of K in order of appearance in F and then work our way down to dimension zero.

We will refer to the algorithm using this strategy as the **twisted reduction algorithm**. The central observation that causes this strategy to yield an improvement in running time is the following.

When reducing a column j, such that $\text{low}(j) = i$ with $i < j$ we know that column j kills some class born at i, so column i has to be a zero column. This means we can just set column i to zero and have to do no further computational steps for the reduction of this particular column. This column operation is usually referred to as **clearing**.

Considering the example from the previous section we can see that if we start the reduction by processing the triangles, we consider column $[v_0, v_1, v_2]$ first. It is reduced from the beginning. Since it has a lowest one in row $[v_1, v_2]$ we now that column $[v_1, v_2]$ has to be a zero column and we can set it to zero without doing any further computations. Therefore we save some reduction steps compared to the standard reduction scheme.

The twisted algorithm yields the same asymptotic bound, as the standard algorithm, however it often performs much better. We will further discuss this in Chapter 5.

Chapter 3

Discrete Morse Theory

Discrete Morse theory is a toolkit that enables us to study and analyze simplicial complexes. It is a combinatorial adaptation of Morse Theory, which is a powerful tool to study manifolds. In general, discrete Morse theory can applied to finite regular CW complexes but we will restrict ourselves to simplicial complexes. The theory is used in a variety of applications like denoising, mesh compression, topological data analysis, and homology computations. Especially the latter will be of interest later on. This chapter is based on an overview by Robin Foreman who also developed the theory. See [For01].

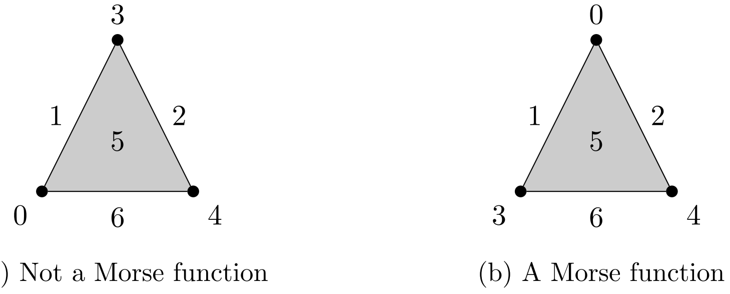
3.1 Discrete Morse Functions

Definition 3.1. Let K be a simplicial complex. A function $f : K \to \mathbb{R}$ is called a discrete Morse function, if for every $\sigma^{(k)} \in K$ the following two conditions are fulfilled:

1. $|\{\tau^{(k+1)} > \sigma^{(k)} \mid f(\tau) \leq f(\sigma)\}| \leq 1$,
2. $|\{\gamma^{(k-1)} < \sigma^{(k)} \mid f(\gamma) \geq f(\sigma)\}| \leq 1$.

This means that a Morse function assigns values to simplices in a simplicial complex, such that simplices of higher dimensions get larger values, with at most one exception for each simplex and its facets and cofacets.

The following Figure 3.1 illustrates two functions that assign values to the faces of a triangle. The numbers represent the assigned function values.



(a) Not a Morse function

(b) A Morse function

Figure 3.1: Two different functions that assign real values to the simplices of the 1-skeleton of a simplicial complex defined by a triangle.

In subfigure (a) the given function is not a Morse function. The vertex at the top gets assigned value 4, while its two adjacent edges get assigned smaller values. I.e., the first condition of Definition 3.1 is not fulfilled. Furthermore the edge on the right, which gets assigned a value of 2 has higher values assigned to its two vertices. This violates the second condition of the definition of a discrete Morse function.

The function depicted in Figure 3.1 (b) however is a Morse function. Note that the vertices with assigned values 3 and 4 have an adjacent edge with a smaller value, but in both cases it is just one. Also the triangle has only one edge with higher function value.

We now describe faces with exceptional values.

Definition 3.2. Let f be a discrete Morse function on a simplicial complex K. A simplex $\sigma^{(k)} \in K$ is called **critical** w.r.t f, if both of the following conditions hold:

1. $|\{\tau^{(k+1)} > \sigma \mid f(\tau) \leq f(\sigma)\}| = 0$,
2. $|\{\gamma^{(k-1)} < \sigma \mid f(\gamma) \geq f(\sigma)\}| = 0$.

In the example of Figure 3.1 (b) the vertex 0 is the only critical face.

One of the central theorems of discrete Morse theory are the so called Morse inequalities which link discrete Morse theory to homology. We will merely state and briefly discuss these results, since the proofs require further theory for which we refer the reader to the aforementioned „A Users Guide To Discrete Morse Theory“ by Robin Forman [For01].

Let K be a simplicial complex of dimension n with a discrete Morse function f. Let m_k denote the number of critical cells of dimension k with respect to f. Finally, recall the definition of the Betti numbers,

$$\beta_k = \operatorname{rank}(H_k(K)),$$

from the last chapter.

Theorem 3.3. *Let K, n, f, m_k and β_k be defined as above. Then the following hold.* ***Weak Morse Inequalities:***

1. *for each $k = 0, 1, \ldots, n$,*
$$m_k \geq \beta_k.$$
2. $m_0 - m_1 + m_2 - \cdots + (-1)^n m_n = \beta_0 - \beta_1 + \beta_2 - \cdots + (-1)^n \beta_n$.

Strong Morse Inequalities:
For each $k = 0, 1, 2, \ldots, n, n+1$,

$$m_k - m_{k-1} + \cdots + (-1)^k m_0 \geq \beta_k - \beta_{k-1} + \cdots + (-1)^k \beta_0.$$

The weak Morse inequalities tell us that the number of critical cells of some dimension with respect to a given discrete Morse function on K is bounded from below by the Betti number of the same dimension. Furthermore the alternating sums of the Betti numbers and the numbers of critical cells are equal.

The strong Morse inequalities even show, that the alternating sum of critical cells up to any dimension is an upper bound on the alternating sum of the Betti numbers up to the same dimension.

3.2 Gradient Vector Fields

In this section we discuss a concept that encodes the information of a discrete Morse function we are usually interested in, without the need to consider the actual function values.

We will begin by defining discrete vector fields on simplicial complexes.

Definition 3.4. Let K be a simplicial complex. Consider the set of pairs

$$V = \{(\sigma^{(k)}, \tau^{(k+1)}) \in K \times K \mid \sigma < \tau\}.$$

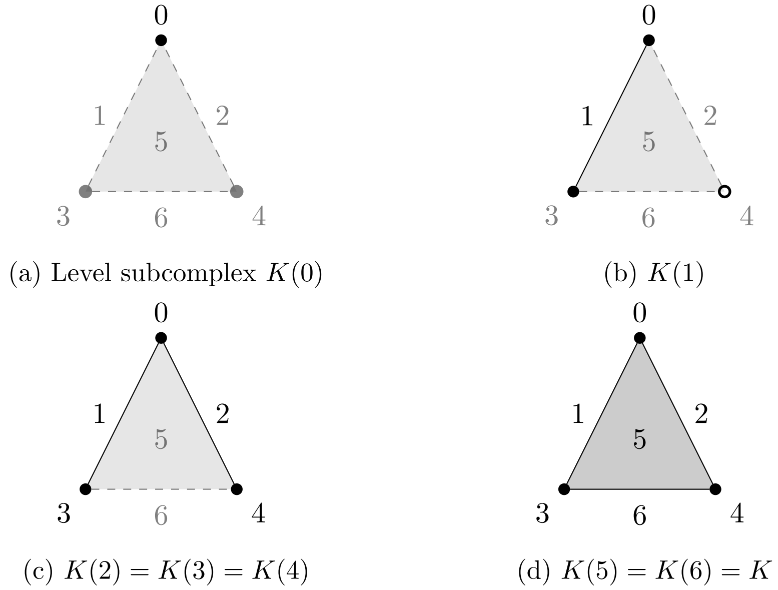
We call V a **discrete vector field** of K if each simplex in K is part of at most one pair in V.

As we will see, Morse functions can be encoded as a special type of discrete vector fields, which we will call **gradient vector fields** or **discrete gradients**. In the following we illustrate how the exceptions of a Morse function in the sense of Definition 3.1 form the pairs of a discrete vector field.

Let f be the Morse function in Figure 3.1 (b). Consider the preimage $f^{-1}((-\infty, 2])$. It is the set consisting of the top vertex and both its adjacent edges. This is not a simplicial complex. But it can be extended to one by including the missing vertices on both ends of the edges.

Definition 3.5. Let K be a simplicial complex and f a discrete Morse function on K. Let $c \in \mathbb{R}$. We define the **level subcomplex** as

$$K(c) = \bigcup_{f(\sigma) \leq c} \bigcup_{\gamma \leq \sigma} \gamma$$



(a) Level subcomplex $K(0)$

(b) $K(1)$

(c) $K(2) = K(3) = K(4)$

(d) $K(5) = K(6) = K$

Figure 3.2: Level subcomplexes of the previous example.

Note that in Figure 3.2 the level subcomplex $K(1)$ contains the vertex with assigned value 3, while being distinct form the level subcomplex $K(3)$.

Now look at simplicial complex K and discrete Morse function f as depicted in Figures 3.1(b) and 3.2(d). The only critical simplex is the top vertex with assigned value of 0. When considering the sequence of level subcomplexes, we realize that the non-critical simplices appear in pairs. Depending on the function values several pairs might appear in a single step or the subcomplexes of several levels might be equal. Nonetheless, if simplices appear, they appear in pairs.

In our example, the edge $f^{-1}(1)$ is not critical because it has vertex $f^{-1}(3)$ as a facet. At the same time vertex $f^{-1}(3)$ is not critical because it has the edge $f^{-1}(1)$ as a cofacet. This observation leads us to the following definition.

Definition 3.6. Let K be a simplicial complex and f a discrete Morse function on K. We call the set of pairs

$$V = \{(\sigma^{(k)}, \tau^{(k+1)}) \in K \times K \mid \sigma < \tau, f(\sigma) \geq f(\tau)\}$$

the **gradient vector field** of f on K.

Notice, that if there were two cofacets of some $\sigma^{(k)}$ with smaller function values, then f would not be a discrete Morse function. The same holds true if there were a face $\tau^{(k+1)}$ with two facets with higher function values.

We will visualize the pairs as arrows pointing from the lower dimensional simplex to the higher dimensional one. This visualization also motivates that we sometimes call the lower dimensional element of a pair in a discrete gradient its **tail** and the higher dimensional element its **head**. We will also refer to these pairs as the **gradient pairs** of f.

The following figure illustrates this in the case of our previous example.

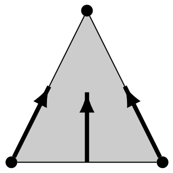


Figure 3.3: The discrete gradient of f and K.

This might remind us of the simplicial collapses discussed in Section 1.3 and indeed a sequence of collapses defines a discrete gradient.

A natural interest regarding vector fields is the analysis of the flows they induce. In our case this motivates the following definition.

Definition 3.7. Let V be a discrete vector field on some simplicial complex K. Then a sequence of simplices

$$P = \sigma_0^{(k)}, \tau_0^{(k+1)}, \sigma_1^{(k)}, \tau_1^{(k+1)}, \ldots, \sigma_m^{(k)}, \tau_m^{(k+1)}, \sigma_{m+1}^{(k)}$$

is called ***V*-path**, if $(\sigma_i, \tau_i) \in V$ and $\tau_i > \sigma_{i+1} \neq \sigma_i$ for $i = 0, \ldots, m$.

The **length** of the V-path is $m+1$ and is denoted as $\text{length}(P)$. We call the $(k+1)$-dimensional elements the **higher dimensional elements** of the path and the k-dimensional elements its **lower dimensional elements**. Furthermore, as the following theorem shows, a discrete gradient of a Morse function f is decreasing along its V-paths.

Theorem 3.8. *Let f be a discrete Morse function and V its gradient vector field. A sequence of simplices*

$$\sigma_0^{(k)}, \tau_0^{(k+1)}, \sigma_1^{(k)}, \tau_1^{(k+1)}, \ldots, \sigma_m^{(k)}, \tau_m^{(k+1)}, \sigma_{m+1}^{(k)}$$

is a V-path if and only if $\sigma_i < \tau_i > \sigma_{i+1}$ for each $i = 0, \ldots, m$ and

$$f(\sigma_0) \geq f(\tau_0) > f(\sigma_1) \geq \cdots \geq f(\tau_m) > f(\sigma_{m+1}).$$

Proof. The definitions of the discrete gradient and V-path imply that $f(\sigma_i) \geq f(\tau_i)$ for each $i = 0, \ldots, m$. Furthermore, the definition of V-paths also implies that $\tau_i > \sigma_{i+1}$. Now, for the sake of contradiction assume that $f(\tau_i) \leq f(\sigma_{i+1})$. This means τ_i has two facets with function values that are greater or equal to that of τ. This contradicts the assumption that f is a discrete Morse function. Hence $f(\tau_i) < f(\sigma_{i+1})$. □

This theorem implies that if f is a Morse function, then there are no nontrivial closed V-paths. The following theorem states that also the converse is true. This gives us a characterization of discrete vector fields, that are gradient vector fields of a Morse function.

Theorem 3.9. *A discrete vector field V is the gradient vector field of a discrete Morse function if and only if there are no non-trivial closed V-paths.*

We will get back to this theorem in the next section, since we will establish some further theory that simplifies the proof.

To conclude and summarize this section we state: Every gradient vector field is a discrete vector field and every discrete vector field without nontrivial closed V-paths is a gradient vector field. We will sometimes refer to V-paths of gradient vector fields as gradient paths. Note that a gradient vector field corresponds to a whole family of discrete Morse functions.

3.3 Morse Matchings

The combinatorial point of view, we will discuss now, was originally explored in [Cha00] and then also included and extended in [For01][Chapter 6].

Consider a simplicial complex K. The **Hasse diagram** $H_K = (W_K, E_K)$ of K is a directed graph with the elements of K being the vertices W_K and $E_K := \{(\tau^{(p+1)}, \sigma^{(p)}) \mid \sigma, \tau \in K, \sigma < \tau)\}$ the directed edges from a simplex to its facets.

We deviate from the popular notation of graphs, where vertex sets are denoted by V, since we stick to Forman's notation of gradient vector fields being denoted by V.

In the following Figure 3.4 we see the same simplicial complex as in Figure 2.2 and its corresponding Hasse diagram.

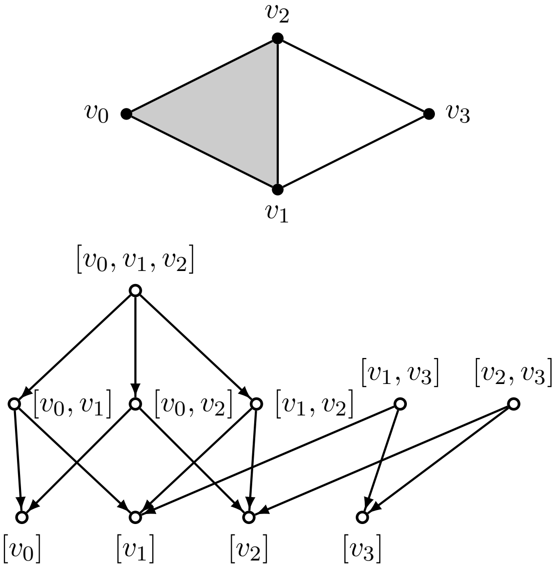


Figure 3.4: Simplical complex K and the corresponding Hasse diagram H_K.

Note that usually Hasse diagrams also include the empty set, which is excluded in our case. We will now modify the Hasse diagram with respect to a discrete vector field as specified in the following definition.

Definition 3.10. Let $H_K = (W_K, E_K)$ be the Hasse diagram of a simplicial complex K and let V be a discrete vector field on K. Define E_K^V as the set of edges where for each edge $e = (\tau^{(p+1)}, \sigma^{(p)}) \in E_K$ we include e in E_K^V if the pair $(\sigma^{(p)}, \tau^{(p+1)})$ is not in V and otherwise include e^{-1}. We call $H_K^V = (W_K, E_K^V)$ the **modified Hasse diagram** of K and V.

What this means is, that we just flip all edges in the Hasse diagram that correspond to pairs in the discrete vector field. Note, that a pair $(\sigma, \tau) \in V$ corresponds to an reversed edge from σ to τ in the modified Hasse diagram H_K^V. In the following figure we depict a gradient vector field on the simplicial complex from Figure 3.4 and its modified Hasse diagram. Looking at this example we can see that V-paths in our gradient vector field correspond to directed paths in the modified Hasse diagram.

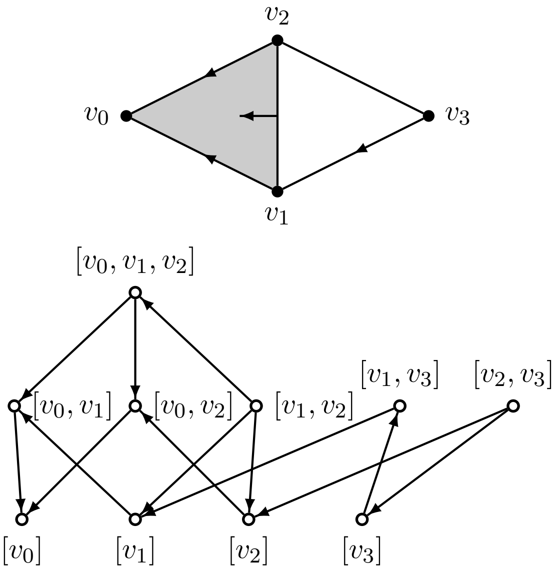


Figure 3.5: Simplical complex K and the corresponding Hasse diagram H_K.

Theorem 3.11. *For simplicial complex K, discrete vector field V and modified Hasse diagram H_K^V, it holds that there are no nontrivial closed V-paths if and only if there are no closed directed paths in H_K^V.*

Proof. We will prove the negated equivalence, i.e., there is a nontrivial closed V-path in V if and only if there is a closed directed path in H_K^V.

„$\Rightarrow$“: Assume we have a discrete vector field V on K and a nontrivial closed directed path $P = \sigma_0, \tau_0, \ldots, \tau_n, \sigma_0$. By definition of the V-path we know that $(\sigma_0, \tau_0) \in V$ hence there is a directed edge from σ_i to τ_i in H_K^V. Further we know by definition of the V-path, that $\tau_i > \sigma_{i+1}$, and since every simplex can only be part of a single pair in V we have that $(\sigma_{i+1}, \tau_i) \notin V$, hence we have a directed edge in H_K^V. In summary this means there is a directed edge in H_K^V between any two consecutive elements of P and therefore a closed directed path.

„$\Leftarrow$“ Whenever we have a directed edge $e = (\sigma^{(k)}, \tau^{(k+1)})$ pointing upwards with respect to dimension of the simplices corresponding to its vertices, then all outgoing edges of $\tau^{(k+1)}$ point to a vertex corresponding to a k-dimensional simplex. Otherwise, τ would need to be paired with a higher dimensional simplex in V, which would mean that τ is part of two pairs in V, which contradicts the definition of discrete vector fields.

Hence any closed directed path in H_K^V can only consist of vertices that correspond to simplices from two consecutive dimensions. We have already seen the one to one correspondence of these closed directed paths to nontrivial closed V-paths when proving the other direction. Therefore the existence of a closed directed path in H_K^V implies the existence of a nontrivial closed V-path. □

A standard result from graph theory states that for a directed graph G there exists a real valued function of the vertices that is strictly decreasing along each directed path if and only if there are no directed cycles in G. This together with Theorem 3.11 implies Theorem 3.9.

When only considering the edges in H_K^V that were flipped by some gradient vector field V it follows from the definition that each simplex is only part of one such edge. Furthermore, from Theorem 3.11 we get that there are no directed cycles. In a combinatorial sense, V defines a partial matching of H_K, which motivates the following definition.

Definition 3.12. For simplicial complex K and directed Hasse diagram $H_K = (W_K, E_K)$ let $E_* \subseteq E_K$ be a subset of edges and let $E_*^{-1} := \{e^{-1} \mid e \in E_*\}$. If every $w \in W_K$ is incident to at most one edge in E_*^{-1} and there are no closed directed paths in $H_K^V := (W_K, E_*^{-1} \cup (E_K \setminus E_*))$, we call E_* a **Morse matching** of K.

As we have already indicated in terms of notation, the edges of a Morse matching E_* correspond to a discrete vector field V and the resulting modified Hasse diagram has no closed directed paths. Therefore V is a gradient vector field. From now on we will not distinguish between a Morse matching and a gradient vector field, since they encapsulate the same information.

Imagine we have some simplicial complex K and a Morse matching on it. Let us denote the number of critical cells with respect to the Morse matching in dimension k by m_k. Then by the Morse inequalities, i.e., Theorem 3.3, we know that the Betti numbers β_k of this complex are bounded from above by m_k, i.e.,

$$\beta_k \leq m_k, \text{ for all } k \in \mathbb{N}_0.$$

Definition 3.13. Let K be a simplicial complex with Morse matching E_*. If $m_k = \beta_k$ for all k we say that E_* is a **perfect** Morse matching. If there is no other Morse matching with fewer critical cells, we say that the Morse matching is **optimal**.

Optimal Morse matchings need not be perfect. After stating a slight reformulation of Theorem 6.4 from [For01], we will discuss a particular example.

Theorem 3.14. *Let K be a simplicial complex and let E_* be a Morse matching of K, such that the single critical face of K with respect to E_* is a vertex. Then K is collapsible and hence contractible.*

Now recall the example of the Dunce hat from Figure 1.9. The Dunce hat is contractible, i.e., is homotopic to a point. Therefore all homology groups of dimension one or higher are trivial and the zeroth Betti number equals one [Mun84][Chapter 2], i.e., $\beta_0 = 1$ and $\beta_k = 0$, for all $k \in \mathbb{N}_{\geq 1}$.

This means a perfect Morse matching has a single critical vertex. For the sake of contradiction, assume that we have a perfect Morse matching for the Dunce hat. Theorem 3.14 implies that the Dunce hat is collapsible, which is a contradiction. Therefore there can be no perfect Morse matching for the Dunce hat or any other simplicial complex that is contractible but not collapsible. Figure 3.6 shows an optimal Morse matching for the Dunce hat.

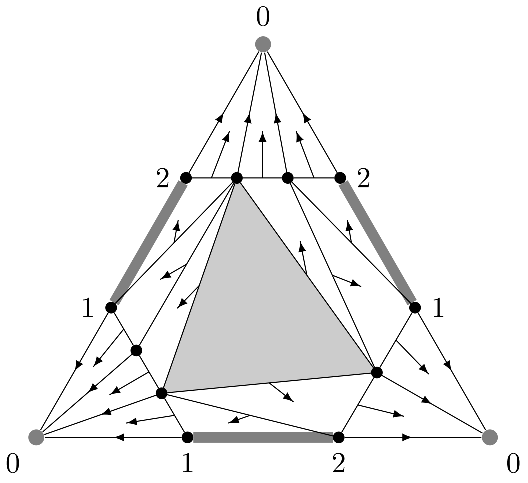


Figure 3.6: The dunce hat with an optimal Morse matching.

In Figure 3.6, the critical cells of the depicted discrete gradient are in gray, while all other cells are part of a pair indicated by an arrow. Since we know there can be no perfect Morse matching, the minimal number of critical cells has to be at least two. We also know that the alternating sums of Betti numbers and critical cells must be equal by the discrete Morse inequalities. Hence, if we have a critical edge, there also must be at least an additional critical vertex or a critical triangle. Yielding a total number of critical cells of at least three. This suffices to know that the matching in our example is indeed optimal.

As Joswig and Pfetsch have shown in [JP06], the computation of optimal Morse matchings is NP-hard. In the same article they introduce a linear program that computes optimal Morse matchings, which works well for small instances, but takes too long to solve for large instances, as would be expected. A random approach to compute Morse matchings was developed by Benedetti and Lutz [BL14], in which they try to match and collapse free faces until this is not possible, in which case a random maximum dimensional face is declared critical and deleted. This approach sometimes yields a perfect Morse matching, which is a certificate for the complex being collapsible. A Python version of the random discrete Morse algorithm, which was developed for this thesis, can be found on: [GitHub], also see [Spi20].

Chapter 4

Connecting Morse Theory and Persistence Computations

This chapter is dedicated to the connection between discrete Morse functions and persistent homology computations, which in general is a popular consideration. See [MN13] for an example of how Morse theory can be used to simplify persistent homology computations.

We will consider some direct translations between filtrations and discrete Morse functions and will try to answer the question if and how „complex“ Morse functions translate to filtrations with expensive persistent homology computations. Looking into this was motivated by talks with Frank Lutz and Paweł Dłotko, who suggested that this connection might be interesting to explore.

4.1 Discrete Morse Functions and Filtrations

We will begin by considering some insights Ulrich Bauer has recently discussed in [Bau19]. The definition of **apparent pairs**, the first two lemmas regarding this notion and their respective proofs follow [Bau19] closely. Afterwards we will build on and extend this theory. Algorithm 2 is a straight forward approach and Algorithm 3 was suggested by Paweł Dłotko. The lemmas concerned with the relation of these algorithms, namely Lemma 4.4 and Lemma 4.5, were developed for this thesis.

Definition 4.1. Let K be a simplicial complex and let F_* be a simplexwise filtration of K. We call a pair (σ, τ) of simplices in K an **apparent pair** if the following holds

- σ is the youngest facet of τ,
- τ is the oldest cofacet of σ.

Here, **young** means, coming late in the filtration and **old** means coming early in the filtration. Consider the simplexwise filtration of the simplicial complex of a triangle with all its faces. In this chapter, we will label vertices by 0 to number of vertices. With respect to a triangle this means we label the vertices 0, 1 and 2, and therefore get the following filtration for the simplicial complex of a 2-simplex:

$$F_* = ([0], [1], [2], [0, 1], [0, 2], [1, 2], [0, 1, 2]).$$

The youngest facet of the edge $[0,1]$ is the vertex $[1]$ and the oldest cofacet of vertex $[1]$ is the edge $[0,1]$. Hence $([1],[0,1])$ is an apparent pair. Consider the boundary matrix of this example, in which we have highlighted the lowest 1s in each column by a gray box:

$$\begin{array}{c|ccccccc|}
 & [0] & [1] & [2] & [0,1] & [0,2] & [1,2] & [0,1,2] \\
\hline
[0] & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
[1] & 0 & 0 & 0 & \colorbox{lightgray}{1} & 0 & 1 & 0 \\
[2] & 0 & 0 & 0 & 0 & \colorbox{lightgray}{1} & \colorbox{lightgray}{1} & 0 \\
[0,1] & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
[0,2] & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
[1,2] & 0 & 0 & 0 & 0 & 0 & 0 & \colorbox{lightgray}{1}
\end{array}$$

As we can see in this case, all three apparent pairs are also persistence pairs, since the respective columns are already reduced and have their lowest 1s in the rows corresponding to the other element of the apparent pair. This generally holds true.

Lemma 4.2. *An apparent pair of a simplexwise filtration is a persistence pair.*

Proof. Let K be a simplicial complex. Let F_* be a simplexwise filtration of K and let B be its boundary matrix. Consider an apparent pair (σ, τ) of F_*.

Assume $\beta \neq \sigma$ is the row with the lowest 1 in column τ. Then β is a facet of τ coming later in the filtration than σ. This contradicts (σ, τ) being an apparent pair. Similarly assume there is another column to the left of τ, which has a non-zero entry in row σ. This means, that τ is not the oldest cofacet of σ.

Again this contradicts (σ, τ) being an apparent pair.

Hence column τ is already reduced and by the pairing lemma it follows that (σ, τ) is a persistence pair. □

Furthermore the set of apparent pairs is a discrete gradient on K.

Lemma 4.3. *The apparent pairs of a simplexwise filtration $F_* = (\sigma_0, \ldots, \sigma_m)$ form a gradient vector field.*

Proof. The proof will be done in two steps. First we prove, that the apparent pairs form a discrete vector field. Then we define a Morse function that has the apparent pairs as gradient pairs.

Let $(\sigma^{(k)}, \tau^{(k+1)})$ be an apparent pair. Since τ is uniquely determined by σ, there can be no other apparent pair $(\sigma^{(k)}, \xi^{(k+1)})$ that contains σ. We will show that there can also be no apparent pair $(\phi^{(k-1)}, \sigma^{(k)})$ containing σ. Note that in this case $k \geq 1$. There is another k-simplex $\rho \neq \sigma$ that is a facet of τ and a cofacet of ϕ. By assumption σ is the youngest facet of τ, hence ρ is older than σ. In particular σ is not the oldest cofacet of ρ and hence (ρ, σ) is not an apparent pair. An analogous argument shows that τ also can not be in another apparent pair. Therefore the apparent pairs form a discrete vector field.

We will now show that the apparent pairs are the gradient pairs of a Morse function. Recall $F_* = (\sigma_0, \ldots, \sigma_m)$ and define

$$f(\sigma_j) = \begin{cases} i & \text{if there is an apparent pair}(\sigma_i, \sigma_j), \\ j & \text{else.} \end{cases}$$

We will verify that f is a Morse function. It holds that $f(\sigma_l) \leq l$. Furthermore let σ_i be a facet of σ_l, i.e., $i < j$. If (σ_i, σ_j) is not an apparent pair then $f(\sigma_i) \leq i < j = f(\sigma_j)$. In particular (σ_i, σ_j) is not a gradient pair of f.

Now assume (σ_i, σ_j) is an apparent pair, i.e., σ_i is the youngest facet of σ_j. Then we have $h \leq i$ for every σ_h that is a facet of σ_j. Thus $f(\sigma_h) \leq h \leq i = f(\sigma_j)$, where equality holds if and only if $h = i$. Meaning, there is exactly one facet of σ_j, namely σ_i, with function value not lower than $f(\sigma_j)$. Hence f is a Morse function and the gradient pairs of f are the apparent pairs of the filtration F_*.

□

We will also refer to this discrete gradient field as the **apparent gradient** of F_*. As with discrete vector fields we will call the lower dimensional element of an apparent pair its **tail** and the higher dimensional element its **head**. Note that from the proof we also get a definition for a Morse function, that has the pairs of the apparent gradient as gradient pairs.

The following example illustrates that the apparent gradient does not necessarily yield an optimal Morse matching. Consider the filtration

$$F_* = ([0], [1], [2], [3], [4], [5], [4, 5], [2, 3], [0, 1], [1, 2], [3, 5], [0, 4]).$$

Figure 4.1 shows a possible geometric realization and the apparent gradient of F_*. There are three critical cells and we know the induced Morse matching is not optimal since we can also match [2] and $[1, 2]$ as well as [4] and $[0, 4]$ without closing a V-path. Indeed adding these pairs yields a perfect Morse matching.

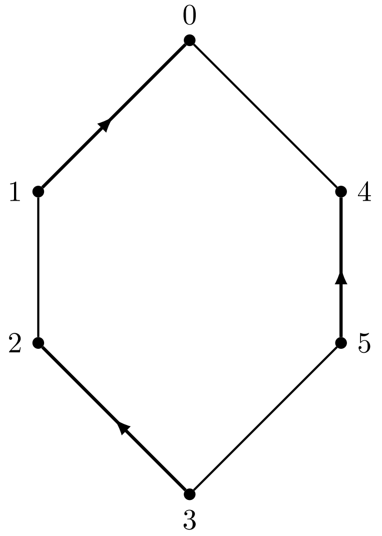


Figure 4.1: Geometric realization of the simplicial complex of F_* and its apparent gradient.

A simple algorithmic approach to construct this gradient vector field is to iterate over all simplices in a simplexwise filtration in order of appearance and for each element search for the youngest facet, then check if the current element is that facets oldest cofacet.
The following algorithm does exactly that while keeping track of all critical faces.

```
   Input  : Filtration $F_* = (\sigma_1, ..., \sigma_m)$, simplicial complex $K$
   Output: Discrete gradient $V$, Critical cells $C$
 1 Initialize $C$ as the set of vertices and $V$ as the empty set
 2 for $\sigma \in F$ in order of appearance do
 3     $\gamma$ = youngest facet of $\sigma$
 4     if $\sigma$ oldest cofacet of $\gamma$ then
 5         Add $(\sigma, \gamma)$ to $V$, remove $\gamma$ from $C$
 6     end
 7     else
 8         Add $\sigma$ to $C$
 9     end
10 end
11 return$(V, C)$
```

Algorithm 2: Filtration to discrete gradient field.

Note that simplices are added to the set of critical cells and removed later on, if they are paired with a cofacet.

The outer loop starting in Line 2 runs in $\mathcal{O}(m)$, where m is the number of simplices in the filtration. At the time of considering σ there are two possibilities.

Either its youngest facet γ is not already paired and σ is the oldest cofacet of γ. See Line 4. In this case we pair σ and γ and remove γ from the set of critical simplices.

Or the condition is not fulfilled. This causes σ to be declared critical for now.

Checking the conditions is in $\mathcal{O}(m)$ each, and assuming that adding and removing elements from V and C takes constant time we get overall quadratic running time in the number of simplices of the filtration.

At the end of the algorithm all simplices remaining in the set C are critical and all others are part of an apparent pair.

Applying this algorithm to the filtration

$$F_* = ([0], [1], [2], [0,1], [0,2], [1,2], [0,1,2]).$$

yields the discrete vector field depicted in Figure 4.2.

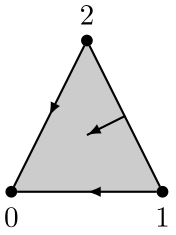


Figure 4.2: A simplicial complex with discrete gradient

A natural question is, how to invert this process, i.e., how to get from a discrete gradient to a filtration in which the pairs of the gradient correspond to apparent pairs.

Recall the modified Hasse diagram from the last chapter. The modified Hasse diagram of the simplicial complex and discrete Morse function from Figure 4.2 is depicted in the following Figure 4.3.

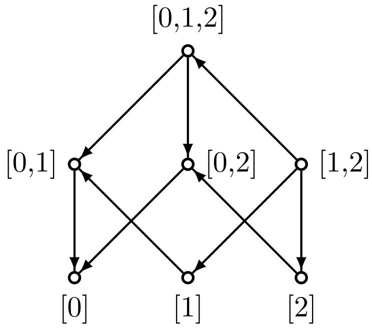


Figure 4.3: The modified Hasse diagram of the previously seen complex and discrete vector field.

The following algorithm to obtain a simplexwise filtration from a discrete vector field V and a simplicial complex K was suggested by Paweł Dłotko. We assume the reader is familiar with topological sorting.

In the version of it we are using, the set of all elements is partitioned such that the first set has no dependencies, the elements of the second set depend on the first etc. Within these sets we will order the elements lexicographically.

Note that the usual convention regarding topological sorting is, that a directed edge from some node a to some node b implies $a \prec b$. In our case it is the other way around.

```
  Input  : Simplicial complex $K$, Discrete gradient field $V$
  Output: Simplexwise filtration $F_*$
1 $H_K^V$ = modified Hasse diagram of $K$ and $V$
2 $Filtration$ = topological sorting of $(H_K^V)$
3 Sort $Filtration$ by dimension in a stable manner.
4 return($Filtration$)
```

Algorithm 3: Discrete gradient field to filtration

By sorting the simplices by dimension in Line 3 we are guaranteed to get a valid simplexwise filtration. Sorting in a stable manner means, that two elements of the same size, i.e. dimension, remain in the same internal order as before.

This algorithm only generates filtrations that are ordered by dimension which is quite restrictive in a sense. If we are mainly interested in the number of reduction steps needed to reduce the corresponding boundary matrix, however, this is no restriction at all. Any filtration F and a filtration F' which is equal to F sorted by dimension in a stable manner, yield the same persistence pairs and need their boundary matrices to be reduced by the same steps.

For an example of the algorithm consider the modified Hasse diagram of

Figure 4.3. The topological ordering we get in line 2 is

$$([0],[0,1],[0,2],[0,1,2],[1],[2],[1,2]).$$

At this point we do not have a valid filtration. But after the algorithm executes Line 3, in which we sort by dimension, while keeping the order within the dimensions intact, we get

$$F_* = ([0],[1],[2],[0,1],[0,2],[1,2],[0,1,2]).$$

We will now formally describe in which way the two Algorithms 2 and 3 are inverse to one another. We will write $F_* = \mathcal{F}(V)$ for the filtration that is generated by the discrete vector field V in Algorithm 3. For brevity, we will omit a reference to the underlying simplicial complex K. Conversly we write $V = \mathcal{V}(F)$ for the gradient vector field generated by Algorithm 2. The following Lemma proves that Algorithm 3 generates a filtration with the desired properties.

Lemma 4.4. *Let V be a gradient vector field on simplicial complex K and let $F_* = \mathcal{F}(V)$ be the generated filtration. Every pair in V is an apparent pair in F_*.*

Proof. Let $(\sigma^{(p-1)}, \tau^{(p)})$ be a pair in V. Recall that, by definition, τ and σ can not be part of any other pair. Now consider H_K^V, the modified Hasse diagram of K and V. Let γ be a facet of τ, with $\gamma \neq \sigma$.

In H_K^V, τ has a directed edge to γ, since (γ, τ) is not a pair in V. But σ has a directed edge to its cofacet τ, since (σ, τ) is a pair in V, so in H_K^V we get a directed edge from σ to τ and therefore a directed path from σ to γ. Hence, in the topological order $\gamma \prec \sigma$ and therefore in the filtration every such γ comes before σ. It follows that σ is the youngest facet of τ.

Analogously, let θ be a cofacet of σ, $\theta \neq \tau$. Now in H_K^V we have a directed edge from σ to τ and another directed edge from θ to σ. Again it follows, that $\tau \prec \theta$, meaning that τ is the oldest cofacet of σ.

The pair $(\sigma, \tau) \in V$ was chosen arbitrarily, hence the claim follows. □

Lemma 4.5. *Let V_0 be a discrete vector field and let C_0 be the set of critical cells of V_0. Now let $V_1 = \mathcal{V}(\mathcal{F}(V_0))$ be the discrete vector field generated by concatenating the respective algorithms. Let C_1 be the set of critical cells of V_1. Then $C_1 \subseteq C_0$.*

Proof. By Lemma 4.4, every pair in V_1 is an apparent pair in $\mathcal{F}(V_1)$, which again, is a pair in $V_2 = \mathcal{V}(\mathcal{F}(V_1))$ by Lemma 4.3. Since every simplex that is not part of a pair, is critical, the claim follows. □

Corollary 4.6. *Let V_0 be a discrete vector field and define $V_i = \mathcal{V}\mathcal{F}((V_{i-1}))$ for $i \in \mathbb{N}$. Then after finitely many steps k, we get that $V_k = V_{k-1}$.*

This means that after repeatedly applying the algorithms all apparent pairs of the filtration are pairs in the apparent gradient and vice versa. For an example of an iteration that actually decreases the number of critical cells, consider the simplicial complex from Figure 4.1 again. The corresponding filtration is

$$F^1_* = ([0], [1], [2], [3], [4], [5], [4, 5], [2, 3], [0, 1], [1, 2], [3, 5], [0, 4]).$$

Using F^1_* as input for Algorithm 2 yields the apparent gradient

$$V_1 = \{([1], [0, 1]), ([5], [4, 5]), ([2], [2, 3])\}.$$

Using V_1 and the underlying simplicial complex of F^1_* as input for Algorithm 3 yields the filtration

$$F^2_* = ([0], [2], [4], [1], [3], [5], [0, 1], [0, 4], [2, 3], [4, 5], [1, 2], [3, 5]),$$

which has apparent gradient

$$V_2 = \{([1], [0, 1]), ([4], [0, 4]), ([5], [4, 5]), ([2], [2, 3])\}.$$

This means the number of critical cells actually decreased and two critical cells remain.

This stabilization process yields altering filtrations and discrete gradients, but we never loose an apparent pair. This means that the filtrations remain close in the sense, that every pair that is an apparent pair once remains a persistence pair in every iteration of the process. Analogously no apparent gradient ever looses a pairing.

It remains to be uncovered if this relates to other notions of closeness of persistence computations we do not discuss in this work or if there are bounds on how many iterations we have to do until the process stabilizes.

4.2 V-Paths and Standard Reduction

In this section we will explore relations between V-paths of the apparent gradient of a filtration and the number of conflicts in the standard reduction scheme. All lemmas and the theorem discussed in this section are not based on previously known works but were developed for this thesis with guidance by Paweł Dłotko.

Recall Definition 2.12, i.e., the definition of the number of column reduction steps $\text{red}(B)$ during the reduction of some boundary matrix B corresponding to filtration F_*. We will now link this to the length of V-paths of the apparent gradient of F_*.

The following example will provide an intuition for the connection. Consider the following filtration:

$$F_* = ([0], [1], [2], [3], [4], [0, 1], [0, 2], [1, 3], [2, 4], [1, 2], [3, 4]).$$

A geometric realization of the underlying abstract simplicial complex is depicted in the following figure. It also shows the apparent gradient of F_*.

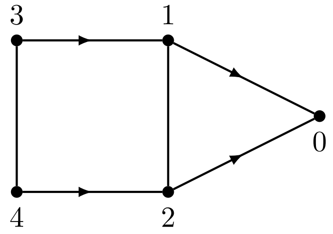


Figure 4.4: An example we will use to illustrate the connection between V-Paths and conflicts in the standard reduction scheme

The cropped boundary matrix of F_* looks as follows:

$$\begin{array}{c|cccccc|}
 & [0,1] & [0,2] & [1,3] & [2,4] & [1,2] & [3,4] \\
[0] & 1 & 1 & 0 & 0 & 0 & 0 \\
[1] & \colorbox{lightgray}{1} & 0 & 1 & 0 & 1 & 0 \\
[2] & 0 & \colorbox{lightgray}{1} & 0 & 1 & \colorbox{lightgray}{1} & 0 \\
[3] & 0 & 0 & \colorbox{lightgray}{1} & 0 & 0 & 1 \\
[4] & 0 & 0 & 0 & \colorbox{lightgray}{1} & 0 & \colorbox{lightgray}{1}
\end{array}$$

We have highlighted the lowest 1s in each column with a small gray box. As we can see every column corresponding to the head of an apparent pair is already reduced, i.e., the lowest 1 is in a row such that no column to the left of it has its lowest 1 in the same row.

Now lets try to resolve the conflicts. Column $[1,2]$ is reduced by first adding column $[0,2]$ and then column $[0,1]$. This corresponds to following the V-paths originating in the facets of edge $[1,2]$ to the critical vertex $[0]$.

Similarly, reducing column $[3,4]$ requires us to add colum $[2,4]$, $[1,3]$, $[0,2]$ and finally $[0,1]$ which again corresponds to following the V-paths originating in the facets of edge the $[3,4]$ and ending in the critical vertex $[0]$.

We will now generalize this example. For some simplexwise filtration $F_* = (\mu_1, \ldots, \mu_m)$ on a simplicial complex K, we define $f : K \to \mathbb{R}$ as $f(\mu_i) = i$ and call f the **filtration values** of F_*.

Let V be the apparent gradient of F_*. We will refer to simplices that are part of a pair in V as **paired** and to simplices that are not part of a pair in V as **unpaired**. Consider some $\tau^{(k+1)} \in K$ and let us denote by $\mathcal{P}_\tau$ the set of all V-paths originating in a facet of τ and ending in an unpaired simplex or a simplex that is the head of an apparent pair. We will call them the **complete facet paths** of τ.

In the above example the complete facet paths of edge $[1,2]$ are the paths $P_1 = [1],[0,1],[0]$ and $P_2 = [2],[0,2],[0]$, hence we write $\mathcal{P}_{[1,2]} = \{P_1, P_2\}$.

Lemma 4.7. *With K, F_*, V and f as above. Let τ be an unpaired simplex, and if there are unpaired elements that are the last elements of a path in $\mathcal{P}_\tau$, let $\sigma^{(k)}$ be the one with highest filtration value. If such a σ, exists it holds that:*

(1) During the reduction of column τ, every lowest 1 with higher filtration value than σ is the tail of an apparent pair and an element of a path in $\mathcal{P}_\tau$.

(2) The reduced column τ is zero or has a lowest 1 in row σ or a row with lower filtration value.

Proof. Initally all ones in column τ correspond to facets of τ. Let α be the row corresponding to the lowest 1. If $\alpha = \sigma$, we are done, so let $\alpha \neq \sigma$. We will show that α has to be the tail of an apparent pair.

Assume for the sake of contradiction that α is unpaired. Then the trivial path $P = \alpha$ is in $\mathcal{P}_\tau$. This contradicts the assumption that σ is the unpaired simplex with highest filtration value that is the last element of a path in $\mathcal{P}_\tau$.

Therefore α has to be part of an apparent pair. Assume α is the head of an apparent pair. Then α is part of a persistence pair with some lower dimensional simplex and can not be part of a persistence pair with some higher dimensional simplex. This means, no column can have its lowest 1 in row α. In particular, it can never be the lowest 1 of column τ during its reduction. This is a contradiction to the assumption that α corresponds to the lowest 1 in column τ. Hence, α is the tail of an apparent pair (α, β). Furthermore, α is an element of all paths in $\mathcal{P}_\tau$ that begin with the sequence α, β. In particular α is an element of a path in $\mathcal{P}_\tau$.

To reduce the lowest 1 in row α we have to add column β. Since β is the head of an apparent pair its column is reduced from the beginning. Therefore the new ones appearing in column τ correspond to facets of β and are themselves lower dimensional elements of a path in $\mathcal{P}_\tau$.

By the same reasoning as before, every lowest 1 with higher filtration value than row σ, corresponds to some tail α_* of an apparent pair and all appearing ones correspond to elements in a path of $\mathcal{P}_\tau$. This proves (1). Furthermore, (2) is a direct consequence of (1), since every lowest 1 that is the tail of an apparent pair is reduced by adding the column of its head. This continues until the column is zero or the lowest 1 appears in row σ. Either τ and σ get paired persistence-wise and the lowest 1 remains in row σ, or if σ is already paired with another column the reduction continues and after the next reduction step the column is zero or has its lowest 1 in a row with lower filtration value than σ. □

Consider the example from Figure 4.4. Reducing column $[1,2]$ follows along the previously stated paths in $\mathcal{P}_{[1,2]}$ and reducing column $[3,4]$ follows along the paths $P_1 = [3],[1,3],[1],[0,1],[0]$ and $P_2 = [4],[2,4],[2],[0,2],[0]$. In the next section we will see an example in which we have a lowest 1 in a row with lower filtration value than the σ from the lemma.

In the following we will prove one-to-one correspondence between lowest 1s during the reduction of column τ and a subset of $\mathcal{P}_\tau$. For some simplex $\mu \in K$, we define $\mathcal{P}_\tau^\mu$, as the set of V-paths originating in a facet of τ and ending in μ, and call them the **facet paths from $\boldsymbol{\tau}$ to $\boldsymbol{\mu}$**. We define the number of these paths

$$x_\tau^\mu := |\mathcal{P}_\tau^\mu|,$$

to be the **$\boldsymbol{\tau}$-multiplicity of $\boldsymbol{\mu}$**. Note that the elements of $\mathcal{P}_\tau^\mu$ are subsequences of the elements of $\mathcal{P}_\tau$.

When considering the above example and looking at the multiplicities of the vertices with respect to the different edges, we see that even multiplicity implies that the vertex is never a lowest 1 in the reduction of a column while odd multiplicity implies the opposite. For example consider the reduction of column $[1,2]$. The vertices $[1]$ and $[2]$ both have odd $[1,2]$-multiplicity and appear as lowest 1s while vertex $[0]$ has even $[1,2]$-multiplicity and does not appear as a lowest 1.

Lemma 4.8. *With the same notations and assumptions as in Lemma 4.7, i.e., $\tau^{(k+1)}$ an unpaired simplex and if there are unpaired elements that are the last elements of a path in $\mathcal{P}_\tau$, let $\sigma^{(k)}$ be the one with highest filtration value. Assuming such a σ exists, let $\mu^{(k)} \in K$ with $f(\mu) > f(\sigma)$. Then the lowest 1 of column τ is in row μ at some point during the reduction if and only if x_τ^μ is odd.*

Proof. We split the proof into the two directions of the equivalence. Both directions are proven by inductive arguments.

„$\Rightarrow$“: We will prove this direction by an induction over the number of appearing lowest 1s.

Let the initial lowest 1 be in row μ. Then $\mathcal{P}_\tau^\mu$ contains the single path $P = \mu$, i.e., x_τ^μ is odd. Assume this is true for the first n appearing lowest 1s. We will now show that it is then also true for the next lowest 1.

Consider μ with $f(\mu) > f(\sigma)$ to be the row that contains the lowest 1 after n reduction steps. The entry in row μ changes its value whenever some column corresponding to a cofacet of μ gets added to column τ. Let $\{\beta_1, \dots, \beta_l\}$ be all l columns that are added to column τ during its reduction that change the value in row μ. Since $f(\mu) > f(\sigma)$, we know by Lemma 4.7 that each of the β_i is part of an apparent pair (α_i, β_i) and α_i was a lowest 1 at some previous point in the reduction. Therefore, by assumption, $x_\tau^{\alpha_i}$ is odd. Furthermore $x_\tau^\mu = \sum_{i=1}^{l} x_\tau^{\alpha_i}$.

We have to consider two cases.

Case one: the entry in row μ of column τ initally equals one. Then l has to be even for the entry to be one after the first n reduction steps. If $l > 0$, x^{μ}_{τ} is the sum of an even number of odd values. Hence it is odd. If $l = 0$, i.e., μ is a facet of τ that is reached only by the trivial path $P = \mu$, then $x^{\mu}_{\tau} = 1$.

Case two: the entry in row μ of column τ initally equals zero. Then l has to be odd for the entry to be one after the first n reduction steps. Therefore x^{μ}_{τ} is the sum of an odd number of odd values. Hence it is odd.

„$\Leftarrow$“: We will prove the claim by induction over x^{μ}_{τ} and begin with $x^{\mu}_{\tau} = 1$. This means we have some path $P = \alpha_0, \beta_0, \dots, \alpha_r, \beta_r, \mu$ with α_0 being a facet of τ. This means the entry in row α_0 of column τ initially equals 1, and by Lemma 4.7 (2) we know it has to be set to zero at some point. Assume that the entry in row α_0 gets set to zero when some column ξ gets added to column τ. By Lemma 4.7 (1) we know that ξ is part of an apparent pair and a path in $\mathcal{P}_{\tau}$. This implies there is another path from a facet of τ to μ that contains P as a subsequence. This is a contradiction to the assumption $x^{\mu}_{\tau} = 1$. Therefore α_0 is the lowest 1 at some point and reducing it creates a 1 in row α_1. By the same reasoning as with α_0 we get that α_1 has to be a lowest 1, and following the V-path along to μ yields that it also has to be a lowest 1 at some point.

Assume this holds true for $x^{\mu}_{\tau} = n$ with n being odd. We will show it is also true for $x^{\mu}_{\tau} = n + 2$.

Let $x^{\mu}_{\tau} = n + 2$ and let γ be the simplex with highest filtration value in which paths in $\mathcal{P}^{\mu}_{\tau}$ intersect that are not equal on at least one apparent pair of which the tail has higher filtration value than γ. This means we are considering paths that might or might not originate in the same facet of τ, are not equal at some point and then intersect (again) in simplex γ. Note that it is possible that $\gamma = \mu$.

Let P and Q be two of these paths. Assume Q and P are initially the same sequence $\alpha_0, \beta_0, \dots, \alpha_l, \beta_l$. By the assumptions on γ, there can be no other path in $\mathcal{P}^{\mu}_{\tau}$ intersecting this sequence. Therefore by the same reasoning as for the case $x^{\mu}_{\tau} = 1$ at some point column β_l gets added to column τ and the rows corresponding to the first distinct elements of Q and P get set to 1. Let us denote these sequences by $Q' = \omega_1, \xi_1, \dots, \omega_y, \xi_y, \gamma$ and $P' = \theta_1, \psi_1, \dots, \theta_z, \psi_z, \gamma$. As neither Q' nor P' get intersected due to the assumptions on γ, we know that at some point of the reduction columns ξ_y and ψ_z get added to column τ. The first addition causes the entry in row γ of column τ to be set to 1. The second one causes it to be set to 0. This means that Q and P have no influence on whether there is a lowest 1 in row μ at some point. Hence μ being a lowest 1 depends on the odd n other paths in $\mathcal{P}^{\mu}_{\tau}$, which, by the induction assumption, tells us that μ indeed is a lowest 1 at some point. □

With this we also know how lowest 1s appear with respect to V-paths originating in a facet of τ. Again assuming the existence of unpaired last elements of a path in $\mathcal{P}_\tau$ we choose σ to be the one with highest filtration value and define:

$$\mathcal{O}_\tau^\sigma := \{(\alpha, \beta) \in V \mid f(\alpha) > f(\sigma),\, x_\tau^\alpha \bmod 2 = 1\},$$

the **odd pairs** between σ and τ.

Theorem 4.9. *Let F_*, K, V, and f be defined as above. Let $\tau^{(k+1)} \in K$ be a simplex that is either not part of an apparent pair or the tail of an apparent pair. If there is an unpaired cell that is the last element of a path in $\mathcal{P}_\tau$, let σ be the one with highest filtration value. If σ exists, then:*

$$\operatorname{red}(\tau) \geq |\mathcal{O}_\tau^\sigma|,$$

with equality holding if τ and σ are a persistence pair or if column τ gets reduced to zero.

Proof. By Lemma 4.7 (2) we know that every lowest 1 in column τ must be reduced until column τ is zero or has a lowest 1 in row σ. By Lemma 4.8 we know that each lowest 1 corresponds to an element in $\mathcal{O}_\tau^\sigma$. If σ is not yet part of a persistence pair or column τ is reduced to zero, we get that $\operatorname{red}(\tau) = |\mathcal{O}_\tau^\sigma|$. If σ is part of a persistence pair, the reduction continues and $\operatorname{red}(\tau) > |\mathcal{O}_\tau^\sigma|$. □

This theorem naturally extends onto the whole boundary matrix B of the filtration F_*. Consider the set $\{\tau_0^{(k+1)}, \dots, \tau_r^{(k+1)}\}$ of all simplices such that each τ_i is either unpaired or the tail of an apparent pair and for which an unpaired simplex that is the last element of a path in $\mathcal{P}_{\tau_i}$ exists. Let $\sigma_i^{(k)}$ be the ones with highest filtration value for the respective τ_i. Then

$$\operatorname{red}(B) \geq \sum_{i=0}^{r} |\mathcal{O}_{\tau_i}^{\sigma_i}|,$$

where equality holds if all columns τ_i get reduced to zero or τ_i gets paired with σ_i.

Note that in a practical application one might just set the columns of lower dimensional elements of apparent pairs to zero before doing any reduction. In this case we would still get a lower bound on the needed reduction steps for the cleared matrix by only considering the critical simplices with respect to the apparent gradient.

The special case we have seen at the beginning of this section follows from Theorem 4.9 and some further insight. It holds that for a connected simplicial complex K and any Morse matching M on K we can construct a Morse matching M' with exactly one critical vertex and at most the number of higher dimensional critical cells. See [JP06][Lemma 2.2].

This implies, that an optimal Morse matching of a connected simplicial complex always has a single critical vertex.

Corollary 4.10. *Let K be a finite one-dimensional connected simplicial complex and F_* a simplexwise filtration, such that the apparent gradient is an optimal Morse matching on K. Let C be the set of critical edges with respect to V and let μ_c be the single critical vertex. Then*

$$\mathrm{red}(B) = \sum_{\tau \in C} |\mathcal{O}_\tau^{\mu_c}|.$$

Proof. Every critical edge in K has to be reduced to zero. By Theorem 4.9 the claim follows. □

This corollary extends to higher dimensional simplicial complexes in the sense that any optimal Morse matching V on a simplicial complex K induces an optimal Morse matching V^1 on the one-skeleton K^1 of K. To see this, consider that V matches all except one vertex, which is the best we can achieve.

Hence, if we have some filtration F for which the apparent gradient V is an optimal Morse matching, we know that the number of column additions in the boundary matrix of F that correspond to edges is equal to $\sum_{\tau \in C_e} |\mathcal{O}_\tau^{\mu_c}|$, where C_e is the set of critical edges and edges that are the tail of an apparent pair.

An interesting example for a similar behaviour in dimension two is the Dunce hat from Figure 3.6. Let us denote the discrete gradient depicted there by V and let $F_* = \mathcal{F}(V)$. Reducing the boundary matrix B of F_* takes a total of 42 reduction steps. Of those, 34 happen in columns corresponding to edges. To be more precise, consider one of the 17 edges that is not the head of an apparent pair or is critical. The corresponding column is reduced to zero by exactly two column additions which correspond to the V-paths of length one, originating in the facets of the edge and ending in the critical vertex $[0]$.

Furthermore there are eight reduction steps needed to reduce the column corresponding to the critical triangle, which correspond to the V-paths from the triangles boundary to the critical edge $[1, 2]$.

4.3 Worst Case Example for the running time of the standard reduction algorithm

As we have discussed in Chapter 2, the worst case running time for the standard reduction scheme lies in $\mathcal{O}(N^3)$, where N is the number of faces of the simplicial complex.

A popular construction of filtrations that achieve a bound of $\Theta(N^3)$ was introduced by Dimitriy Morozov in [Mor05]. We will introduce a new series of examples that is inspired by the series of Dimitriy Morozov but achieves the bound for slightly different reasons. Two advantages of the new series of examples are that it is easy to construct algorithmically and that it grows faster than the previously known examples.

Consider the vertices of an n-gon $\{[1], \ldots, [n]\}$, which we will call the **outer vertices**. Furthermore we have a **center vertex** $[0]$ inside the n-gon. We will call the edges of the n-gon $[i, i+1]$ for $i = 1, \ldots, n-1$ and $[1, n]$ the **outer edges** and we will include edges $[0, i]$ between each bounding vertex and the center which we will call the **base edges**.

We add the triangles $[0, i, i+1]$ and call them **base triangles**. Furthermore we add the triangle $[0, 1, n]$ which we call the **closing triangle**. The following figure illustrates our construction so far for $n = 3$, i.e., three bounding vertices.

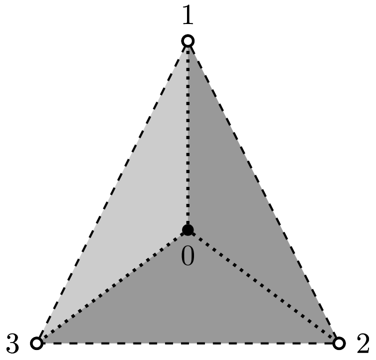


Figure 4.5: First part of the construction of a series of worst case examples.

In Figure 4.5 the outer vertices are white, the center vertex is black, the outer edges are dashed lines, the base edges are dotted lines. The base triangles are dark gray, while the closing triangle is depicted in a light gray.

We will now extend the construction. We place (in 3-space) a vertex above each of the base edges. This means, we get another n vertices $\{[n+1], \ldots, [n+n]\}$ which we will call the **fin vertices**. Then we will connect these vertices with the central vertex and the other end of the base edge they are placed above. This means, we include the edges $\{[0, n+1], \ldots, [0, n+n]\}$ and $\{[1, n+1], \ldots, [n, n+n]\}$. We will refer to them as **fin edges**. Finally, we add the triangles $\{[0, 1, n+1], \ldots, [0, n, n+n]\}$, which we call **fin triangles**. We call this construction an ***n*-gon with center and fins**.

Those familiar with the example by Dimitriy Morozov might recognize the similarities with respect to the construction and naming. With this the construction of our space is complete. See the following figure for an illustration of the complete construction for $n = 3$.

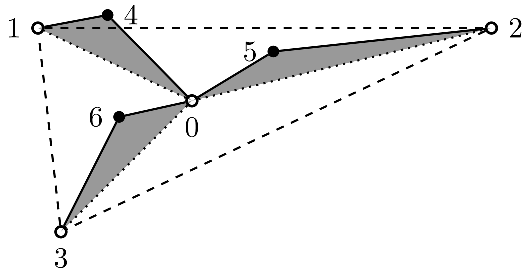


Figure 4.6: Adding the fins.

In Figure 4.6 we can see the fin vertices in black, the fin edges as solid lines, and the fin triangles in dark gray.

For any n, this simplicial complex has $1 + 2n$ vertices, $4n$ edges and $2n$ triangles. In particular, the number of simplices lies in $\Theta(n)$. The following table lists all simplices for some n in the categories we created:

Central vertex:	$[0]$,
Outer vertices:	$[1], \ldots, [n]$,
Fin vertices:	$[n+1], \ldots, [n+n]$,
Bounding edges:	$[1,2], [2,3], \ldots, [n-1,n], [1,n]$,
Fin edges:	$[0,1+n], \ldots, [0,n+n], [1,1+n], \ldots, [n,n+n]$,
Base edges:	$[0,1], [0,2], \ldots, [0,n]$,
Base triangles:	$[0,1,2], \ldots, [0,n-1,n]$,
Closing triangle:	$[0,1,n]$,
Fin triangles:	$[0,1,n+1], \ldots, [0,n,n+n]$.

Consider the filtration F_* defined by concatenating all rows of the above table in order of appearance. Consider the relations between the base edges and the base triangles. Edge $[0,1]$ is not part of an apparent pair, since its oldest cofacet $[0,1,2]$ has edge $[0,2]$ as a youngest facet. But this means $([0,2],[0,1,2])$ is an apparent pair. Similarly $([0,i],[0,i-1,i])$ is an apparent pair for $i = 3, \ldots, n$. So all base edges, except $[0,1]$ are paired with a base triangle.

We will now prove that the number of additions with respect to the standard reduction scheme for filtrations specified by this construction lie in $\Theta(n^3)$. Note that we consider $n \geq 3$.

To illustrate our reasoning we will follow along an example for $n = 5$. In the following figure we see boundary matrices corresponding to this example. We have cropped them such that only the relations between triangles and edges are visible. The light squares indicate 0s while the dark squares indicate 1s.

	[0, 1, 2]	[0, 2, 3]	[0, 3, 4]	[0, 4, 5]	[0, 1, 5]	[0, 1, 6]	[0, 2, 7]	[0, 3, 8]	[0, 4, 9]	[0, 5, 10]
[1, 2]	■									
[2, 3]		■								
[3, 4]			■							
[4, 5]				■						
[1, 5]					■					
[0, 6]						■				
[0, 7]							■			
[0, 8]								■		
[0, 9]									■	
[0, 10]										■
[1, 6]						■				
[2, 7]							■			
[3, 8]								■		
[4, 9]									■	
[5, 10]										■
[0, 1]	■				■	■				
[0, 2]	■	■					■			
[0, 3]		■	■					■		
[0, 4]			■	■					■	
[0, 5]				■	■					■

(a) Submatrix of edges to triangles before any reduction steps are done.

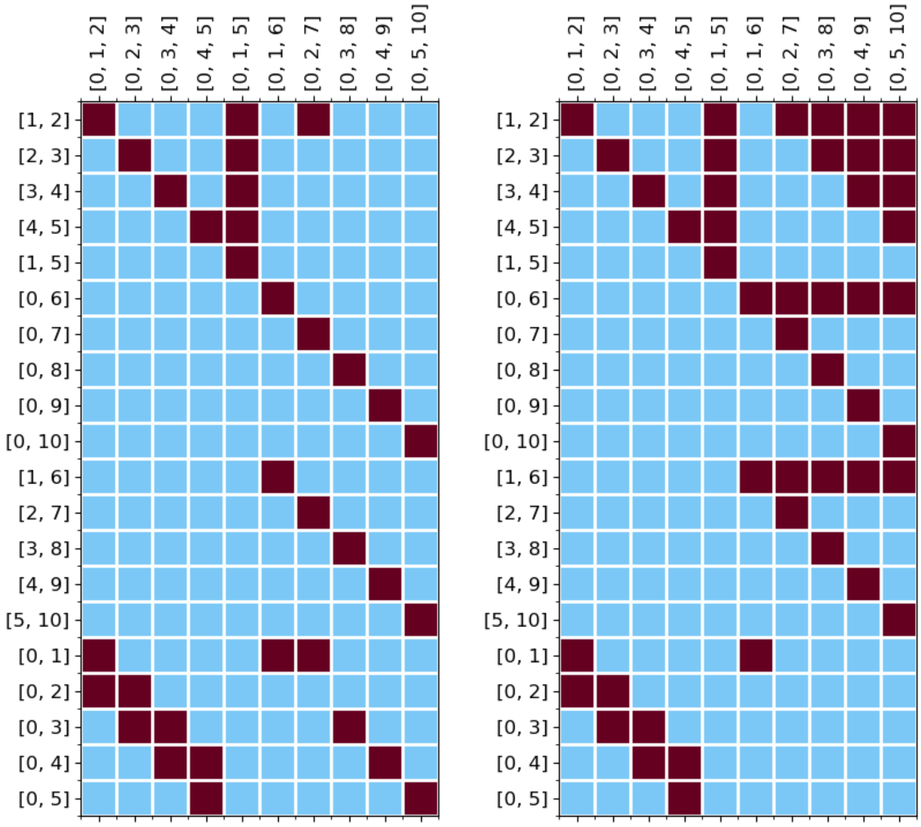
	[0, 1, 2]	[0, 2, 3]	[0, 3, 4]	[0, 4, 5]	[0, 1, 5]	[0, 1, 6]	[0, 2, 7]	[0, 3, 8]	[0, 4, 9]	[0, 5, 10]
[1, 2]	■				■					
[2, 3]		■			■					
[3, 4]			■		■					
[4, 5]				■	■					
[1, 5]					■					
[0, 6]						■				
[0, 7]							■			
[0, 8]								■		
[0, 9]									■	
[0, 10]										■
[1, 6]						■				
[2, 7]							■			
[3, 8]								■		
[4, 9]									■	
[5, 10]										■
[0, 1]	■					■				
[0, 2]	■	■					■			
[0, 3]		■	■					■		
[0, 4]			■	■					■	
[0, 5]				■						■

(b) Submatrix of edges to triangles after the closing triangle has been reduced.

Figure 4.7: Cropped boundary matrices of the n-gon with center and fins for $n = 5$.

Consider Figure 4.7 (a). The base triangle columns have lowest 1s in the rows of the respective other part of the apparent pair they are an element of. The first conflict appears in row $[0, 5]$ of column $[0, 1, 5]$. Removing it by adding column $[0, 4, 5]$ creates a new conflict in row $[0, 4]$ which is again resolved by adding the column of the corrseponding apparent pair. After four steps, or in the general case after $n - 1$ steps, we arrive at the situation depicted in Figure 4.7 (b). The closing triangle is paired with the last bounding edge $[1, 5]$, or $[1, n]$ in the general case. We now have to reduce the fin triangles. In the example, these are the triangles $[0, 1, 6], \dots, [0, 5, 10]$.

The first fin triangle has no conflicts. It is indeed already paired with the base edge $[0, 1]$. The second fin triangle has a conflict with the first base triangle and resolving it creates a conflict in row $[0, 1]$, which is resolved by adding the column of the first fin triangle. Since it is sufficient for our proof, we will only count the 1s in the columns we are reducing. Consider the number of additions in these first two reduction steps. When resolving the first conflict there are three 1s in the column of the second fin triangle. This means we get three additions. See Figure 4.8 (a) for the boundary matrix of our example after this addition.



(a) Submatrix of edges to triangles before any reduction steps are done.

(b) Submatrix of edges to triangles after the final triangle has been reduced.

Figure 4.8: Reducing the fin triangles.

Whenever we add a base triangle $[0, i, i+1]$ to some fin triangle to resolve the conflict in row $[0, i+1]$ there is a new lowest 1 appearing in row $[0, i]$. This procedure continues until we reach a lowest 1 in row $[0, 1]$ which is resolved by adding the column of the first fin triangle. After this step each fin triangle is reduced. Furthermore, whenever we add a base triangle there is a 1 appearing in the bounding edge that is a facet of the base triangle. See the entry in row $[1, 2]$ of column $[0, 2, 7]$ in Figure 4.8 (a).

This means there are 1s amassing during the reduction of a fin triangle. For $k = 2, \ldots, n$, the kth fin triangle is reduced in k steps. The first $k - 1$ correspond to base triangles and the last one to the first fin triangle. Since each addition with a base triangle increases the number of 1s in the column we are reducing and since we start at three ones for each fin triangle for the reduction of the kth fin triangle, we get

$$3 + 4 + \cdots + (3 + (k-1)) > 1 + 2 + \cdots + (1 + k - 1) = \sum_{j=1}^{k} j$$

additions. Considering our example we can see these amassed 1s in the top right corner of the matrix in 4.8 (b).

Summing up the additions to reduce all the fin triangles starting at the second one, we get

$$\sum_{k=2}^{n}\sum_{j=1}^{k} j$$

additions. Recall that we also had to reduce the closing triangle in more than one step. Hence for the total number of additions a it holds that:

$$\begin{aligned}
a &\geq 1 + \sum_{k=2}^{n}\sum_{j=1}^{k} j \\
&= \sum_{k=1}^{n}\sum_{j=1}^{k} j \\
&= n1 + (n-1)2 + ... + 2(n-1) + 1n \\
&= \sum_{l=0}^{n}(n-l)(l+1) \\
&\geq \sum_{l=0}^{n}(n-l)l \\
&= \sum_{l=0}^{n} nl - \sum_{l=0}^{n} l^2.
\end{aligned} \tag{4.1}$$

With $\sum_{l=0}^{n} l = \frac{n(n+1)}{2}$ and $\sum_{l=0}^{n} l^2 = \frac{n(n+1)(2n+1)}{6}$ we get

$$\begin{aligned}
\sum_{l=0}^{n} nl - \sum_{l=0}^{n} l^2 &= n\frac{n(n+1)}{2} - \frac{n(n+1)(2n+1)}{6} \\
&= n(\frac{n(n+1)}{2} - \frac{(n+1)(2n+1)}{6}) \\
&= n\frac{3n(n+1) - (n+1)(2n+1)}{6} \\
&= \frac{n(n+1)(3n-2n-1)}{6} \\
&= \frac{(n-1)n(n+1)}{6} \in \Theta(n^3).
\end{aligned} \tag{4.2}$$

This means we get cubic running time for the construction of the n-gon with center and fins for any $n \geq 3$.

The following figure compares this construction to the one of Dimitriy Morozov. On the x-axis we have the number of faces of the underlying simplicial complex and on the y-axis the number of additions in the standard reduction scheme.

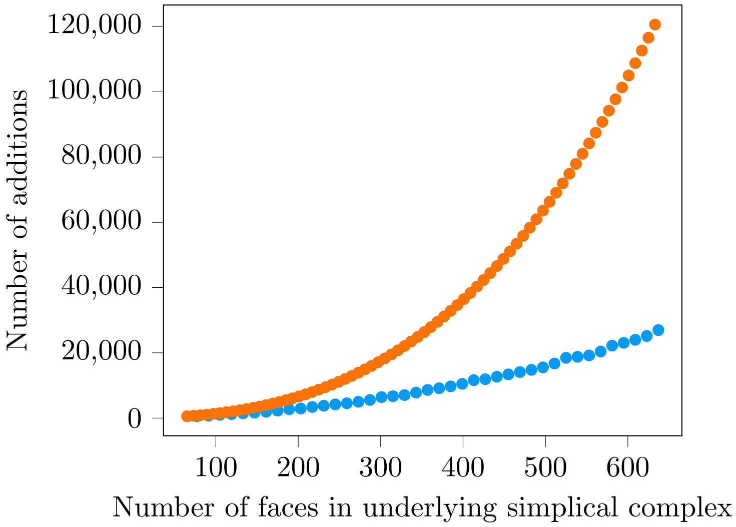


Figure 4.9: Comparision of the two constructions that yield cubical running time. Construction by Dimitriy Morozov in blue and the one propsed in this section in orange.

As we can see the construction propsed here grows much faster. As we have previously discussed the number of vertices, edges and triangles of the n-gon with fins depends on n. To be precise the f-vector of the construction is $(1 + 2n, 4n, 2n)$, i.e. all entries depend linearly on n, in particular all entries depend linearly on the number of vertices. For the construction of Dimitriy Morozov it also holds that the number of edges and triangles depends linearly on the number of vertices.

Python code, to generate the filtrations of both constructions can be found on [GitHub], see also [Spi20].

When considering the n-gon construction in terms of Theorem 4.9 we see that for the reduction of fin triangles we follow V-paths defined by the apparent pairs of base triangles and base edges. Indeed, we do not get equality since the final reduction step always corresponds to an addition of the first fin triangle, which is unpaired with respect to the apparent gradient.

The lower bound we get by Theorem 4.9 equals $\sum_{r=1}^{n-1} r = \frac{(n-1)n}{2}$ which is not the exact number of column additions but at least asymptotically correct.

However we can generate an almost identical filtration by reversing the order of the base edges which by the same reasoning results in the same asymptotic bound of additions but has a single apparent pair. In this case the bound we get by Theorem 4.9 is as bad as asymptotically possible.

4.4 Conclusions

The initial question that motivated this chapter was how complicated discrete Morse functions correspond to difficult persistent homology computations. While we can not give a complete answer to this we have shed some light on the relation between filtrations and gradient vector fields. We have proven a new theorem (Theorem 4.9), which states how the apparent gradients give a lower bound on the complexity of persistent homology computations. On the other hand we have also seen that two different filtrations of the same simplicial complex can yield an asymptotically perfect bound or a trivial constant bound that has an arbitrarily large difference to the actual number of column additions. Nevertheless, we hope that the results we discussed here prove fruitful for future research on the topic.

Chapter 5

Random Simplicial Complexes

In this chapter we carry out some experiments regarding apparent pairs on random simplicial complexes. We will consider two different notions of random simplicial complexes. Namely, alpha complexes constructed on randomly generated point clouds and two-dimensional analogues of the Erdös–Rényi random graph model as introduced by Meshulam and Linial in [LM03] and extended by Meshulam and Wallach in [MW09]. An extensive discussion of results regarding these was done by Kahle in [Kah16]. For some insights and experiments regarding apparent pairs and Vietoris–Rips complexes see [Bau19].

5.1 Erdös–Rényi Analogues

Consider the set $G(n)$ of all graphs on vertex set $\{1, \dots, n\}$. The Erdös–Rényi random graph model $G(n, p)$ is the probability distribution on $G(n)$, „where every edge is included with probability p jointly independently.“ [Kah16][page 1].

A general analogue in terms of simplicial complexes is the random k-complex $Y_k(n, p)$ introduced in [MW09]. $Y_k(n, p)$ „contains the complete $(k-1)$-skeleton of a simplex on n vertices, and every k-dimensional face appears independently with probability p.“ [Kah16][page 4].

In our case $Y_k(n, p)$ will have $n + 1$ vertices since we want to keep it consistent with our previous notations for filtrations.

We will construct filtrations by ordering simplices by dimension first and lexicographically within the respective dimensions. Then we take the full $(k-1)$-skeleton in that order. Finally we include each k-face with probability p. The following Figure 5.1 gives an example of a random 2-complex on four points, i.e., for $n = 3$.

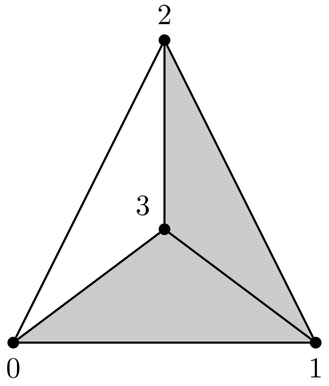


Figure 5.1: Random 2-complex on four points.

We have not explicitly stated a value for p with respect to Figure 5.1, since the complex might result from any $p > 0$ with varying probability. The corresponding filtration is

$$F_* = ([0],[1],[2],[3],[0,1],[0,2],[0,3],[1,2],[1,3],[2,3],[0,1,3],[1,2,3]).$$

5.2 Apparent Pairs on Random 2-Complexes

In this section we will analyze apparent pairs in the context of random 2-complexes $Y_2(n,p)$ for fixed n and varying p. We consider filtrations as specified in the previous section.

An interesting observation is that for $n \in \mathbb{N}$, $p \in \{0,1\}$, and simplices of $Y_2(n,p)$ in lexicographical order, the apparent gradient always yields a perfect Morse matching.

For $p = 0$ and n arbitrary but fixed we get the following filtration:

$$F_* = ([0],[1],\ldots,[n],[0,1],\ldots,[0,n],[1,2],\ldots,[n-1,n]).$$

Looking at the apparent pairs we see that $[0]$ is not paired, i.e., is a critical cell. Vertex $[1]$ gets paired with $[0,1]$, $[2]$ gets paired with $[0,2]$, and so on until $[n]$ gets paired with $[0,n]$. All other edges remain unpaired, and there are no triangles or higher-dimensional faces. Furthermore, each edge that does not get paired, corresponds to a one-dimensional hole in the simplicial complex. Therefore, we get that the number of critical cells is equal to the Betti number in the respective dimensions.

If $p = 1$, every edge that is not paired with a vertex gets paired with a triangle. Edge $[1,2]$ is the youngest facet of triangle $[0,1,2]$ which is the oldest cofacet of $[1,2]$. The same holds for $[1,3]$ and $[0,1,3]$ and so on until we reach $[1,n]$ and $[0,1,n]$. Then $[2,3]$ to $[2,n]$ get paired with $[0,2,3]$ to $[0,2,n]$. This pattern continues until $[n-1,n]$ gets paired with $[0,n-1,n]$. This means all vertices get paired with the edge between this vertex and $[0]$, while all edges that do not contain vertex [0] get paired with the triangle containing this edge and vertex [0].

For the following figure we sample $Y_2(10, p)$ for $p = 0, 0.01, \ldots, 0.99, 1$, calculate the Betti numbers for the resulting complex and subtract the number of critical cells in the apparent gradients. Then we count how often this difference equals zero and divide it by the total number of generated complexes, i.e., we calculated the ratio of the apparent gradient being a perfect Morse matching for different values of p. For each p we constructed 1000 simplicial complexes to average over.

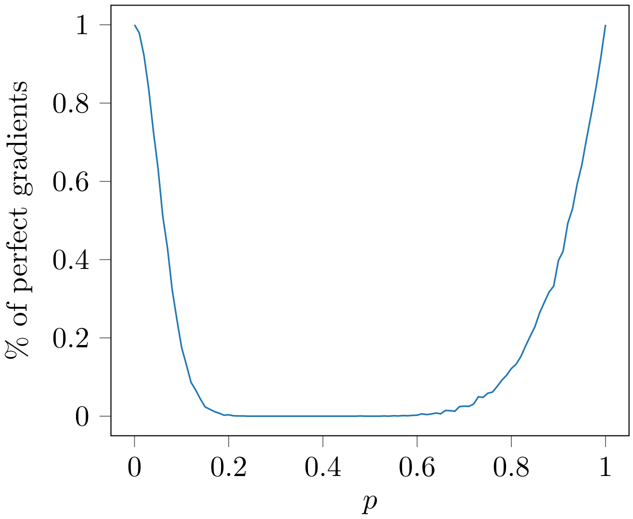


Figure 5.2: Probability of the apparent gradient being a perfect Morse matching for $Y_2(10, p)$ with different values of p.

As we have previously argued for $p = 0$ the apparent gradient is perfect. In order for an apparent gradient to be not perfect, the number of critical cells in some dimension l has to be larger than the Betti number β_l.

This can happen for example, if two triangles $[a, b, c]$ and $[b, c, d]$ have edge $[b, c]$ as a youngest facet but $[b, c]$ can only have one of them as the oldest cofacet. Without loss of generality it is $[a, b, c]$. This means that triangle $[b, c, d]$ is critical, i.e., the number of critical cells in dimension two is higher than β_2. Furthermore there also has to be some edge which can not be paired, hence the number of critical edges equals $\beta_1 + 1$. Up to a certain point it holds that the higher number of simplices in a complex drawn from $Y_2(10, p)$ causes situations to occur in which triangles can not be paired. This could explain the inital decline of the curve.

If more and more simplices appear, we get closer to the structure of the full simplex in which the apparent gradient is a perfect Morse matching again, hence the increase after $p = 0.6$.

For the next figure we compute

$$d = \sum_{i=0}^{2} c_i - \sum_{i=0}^{2} \beta_i,$$

for each sampled complex. Here c_i is the number of critical cells in dimension i. We average these values over 1000 sampled complexes for $Y_2(10,p)$ and 200 sampled complexes for $Y_2(15,p)$ and $Y_2(20,p)$.

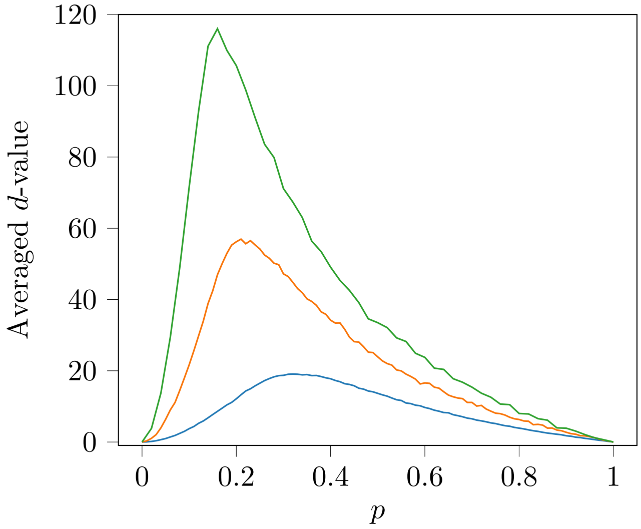


Figure 5.3: Average difference of summed Betti numbers and summed critical cells with resepct to the apparent gradient. $Y_2(10,p)$ in blue, $Y_2(15,p)$ in orange and $Y_2(20,p)$ in green.

It is interesting to see, that the peak in averaged d-values is achieved earlier for larger n and at low p-values in general. This implies that after a certain value of p is reached there are more triangles that break up the previously discussed situations in which triangles can not get paired.

5.3 Perfect Random Discrete Morse

Consider the previously discussed random discrete Morse algorithm as specified by Lutz and Benedetti in [BL14]. Their construction of random discrete Morse functions is very different, in particular random, while the apparent gradient construction is deterministic. Yet both approaches are interesting to analyze in a similar manner. The following figure shows plots similar to the one from Figure 5.2. This time for each $p = 0, 0.01, \ldots, 0.99, 1$, we draw a simplicial complex from $Y_2(15,p)$, 200 times and for each one we calculated a random discrete Morse function 300 times. Then we compute the percentage of how often the random discrete Morse function was perfect and again averaged this over the 200 simplicial complexes drawn for some fixed p. Figure 5.4 then shows this curve plotted against the percentage of perfect apparent gradients.

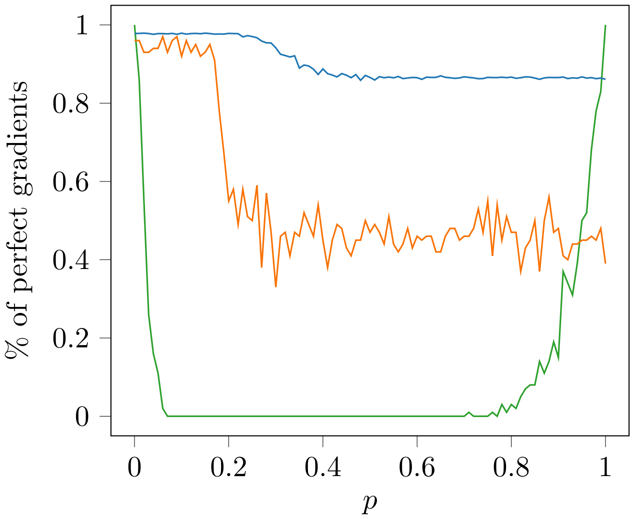


Figure 5.4: How often do apparent gradients of $Y_2(15, p)$ yield perfect Morse matchings (green) compared to how often the random discrete Morse function yields perfect Morse matchings for $Y_2(10, p)$ (blue) and $Y_2(15, p)$ (orange).

Figure 5.4 allows two interesting observations. Firstly in terms of the apparent gradients, we see the same behaviour for $Y_2(15, p)$ as for $Y_2(10, p)$, although for $Y_2(15, p)$, the initial decline is steeper and the percentage of perfect apparent gradients also starts to rise again later. Indeed for any $p \notin \{0, 1\}$ and n large enough we expect there to be at least one configuration in the filtration where some simplex can not get paired with respect to apparent pairs. Hence the apparent gradient is not perfect. Therefore the larger the value for n the lower the value of p for which we might get a perfect apparent gradient. A similar argument explains the behavior for values of p close to 1. This means that for $n \to \infty$ we get perfect gradients if and only if $p \in \{0, 1\}$.

Secondly we see that initially the random discrete Morse algorithm has a very high probability of finding a perfect Morse matching for $Y_2(10, p)$ and $Y_2(15, p)$. Then we get a decline at around $p = 0.25$ and $p = 0.19$ respectively. Afterwards fewer of the found Morse matchings are perfect. It is surprising to see these different levels for different n which seem to be quite stable over large ranges of values for p.

In some future work we would like to find some explanation for this sudden jump in the percentage of perfect Morse matchings found by the random discrete Morse algorithm. The following figure could give a potential clue or starting point for further exploration. It shows a plot of the sum of Betti numbers of random 2-complexes averaged over 50 runs.

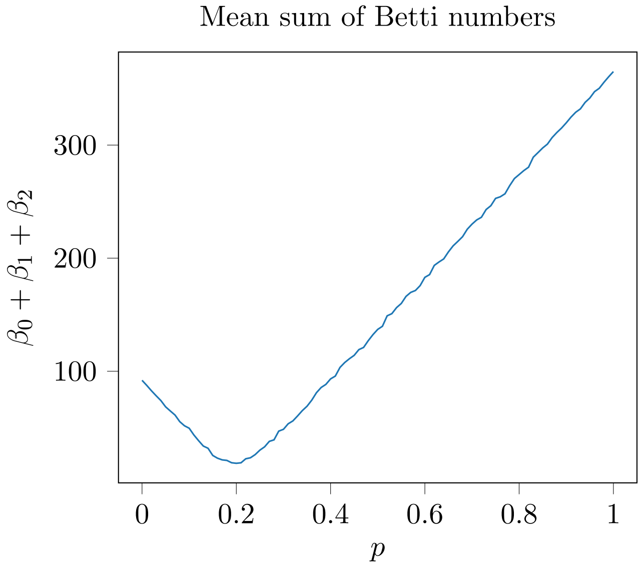


Figure 5.5: Sum of Betti numbers for $Y_2(15, p)$.

As we can see the sum of Betti numbers declines until p reaches a value of about 0.2 which is close to the value at which we see a jump in the percentage of perfect Morse matchings found by the random discrete Morse function.

If in some practical situation one is interested in a perfect Morse matching on a random k-complex, in most cases searching for one via the random discrete Morse algorithm is the better way to go. For very small and very large values of p however it might be a good idea to check the apparent gradient.

5.4 Alpha Complexes on Random Point Clouds

Recall the definition of alpha complexes from Section 1.2. The set of points we construct the complex on can have a variety of origins. Maybe it is real world data of some kind or a handpicked set of points or, as in our case, some randomly generated set of points. By randomly generated we mean that each point is sampled via some random distribution.

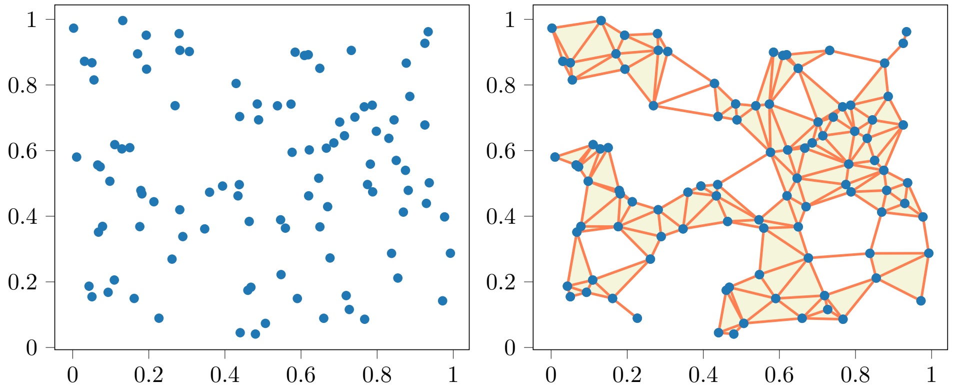
We generate point clouds using the **numpy.random** library for **Python3**, then we calculate the (inclusion-wise) largest possible Alpha complex on the point cloud and a simplexwise filtration of it using the **GUDHI** library. The exact construction is specified in [GUD20].

We chose the following distributions, since they also appear in real world situations.

5.4.1 Uniform Distribution

In this case, we will sample from the k-dimensional unit cube. For a point $p = (p_1, \dots, p_k)$ this means that each p_i is drawn from the interval $[0, 1]$ and that each value in the interval is equally likely. In other words the probability density we draw from equals one on the k-dimensional unit cube and zero elsewhere. Figure 5.6 gives an example of a two dimensional point cloud consisting of 100 points drawn like this. Subfigure (b) shows the simplicial complex $K_{\lfloor \frac{n}{2} \rfloor}$ of the filtration $F_* = \{K_1, \dots, K_n\}$, where K_n is the Alpha complex, where $\epsilon > 0$ is so large, that further increasing its value does not change the resulting simplicial complex, which we will call the **maximal alpha complex**.

Due to the discrete nature of floating point precision it is possible, although very unlikely, that several simplices appear at the same distance ϵ. In this case we sort the simplices by dimension first and lexicographically second, hence we always get a simplexwise filtration.



(a) Point cloud S

(b) Intermediate step of the filtration of the largest possible Alpha complex on S.

Figure 5.6: Sampled uniform distribution and simplicial complex.

The code that generated the point clouds and the visualizations in the previous and following figures can be found on [GitHub], see [Spi20].

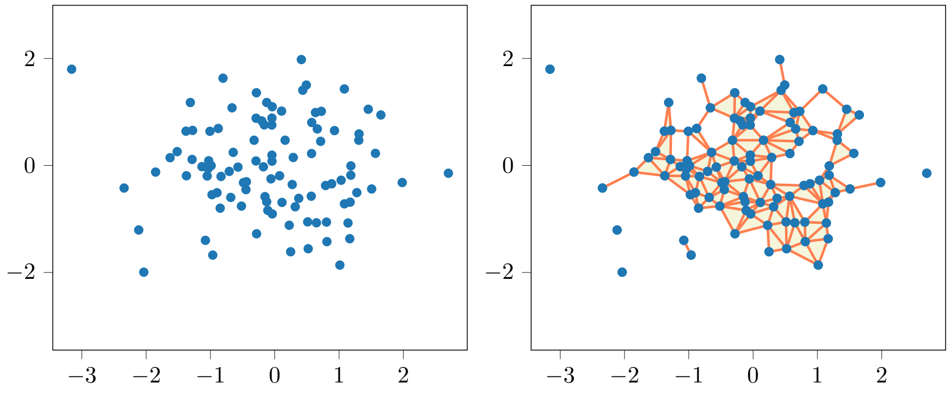
5.4.2 Multivariate Gaussian Distribution

In this case we will sample points from a multivariate Gaussian distribution. A k-dimensional random variable X is multivariate normal distributed, if its density function is of the following form:

$$f_X(x) = \frac{\exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))}{\sqrt{(2\Pi)^k|\Sigma|}},$$

where $x \in \mathbb{R}^k$, Σ is the covariance matrix, which we require to be positive definite, and μ is the mean vector. We write $X \sim \mathcal{N}(\mu, \Sigma)$.

Figure 5.7 gives an example on 100 points and the median simplicial complex generated in the same manner as before.



(a) Point cloud S.

(b) Intermediate step of the filtration of the largest possible Alpha complex on S.

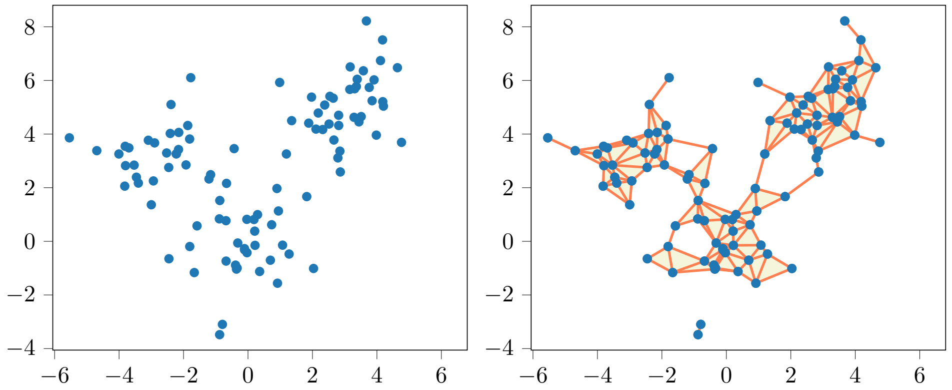
Figure 5.7: Sampled multivariate distribution.

5.4.3 Gaussian Mixture

Finally we will consider point clouds which are drawn from a mixture of m Gaussian distributions. We do this by repeating the following steps:

- Choose a value $i = 1, \ldots, m$ uniformly.
- Sample a point from $\mathcal{N}_i(\mu_i, \Sigma_i)$.

Figure 5.8 gives an example on 100 points and the median simplicial complex generated in the same manner as before.



(a) Point cloud S.

(b) Intermediate step of the filtration of the largest possible Alpha complex on S.

Figure 5.8: Sampled Gaussian mixture.

There are a lot of variables here which can be tweaked. We could choose i in a weighted manner or draw means and covariances from some distribution. In the following we will however restrict ourselves to handpicked means, identity matrices for covariance and uniform distribution when choosing the Gaussian to be drawn from.

5.5 Apparent Pairs on Random Alpha Complexes

The goal of this section is to explore the relation between expensive persistent homology computations and apparent pairs from an experimental perspective.

The first question we will discuss is: how many pairs of our filtrations are part of an apparent pair? Or alternatively how many simplices of some filtration F of an alpha complex are critical with respect to the apparent gradient V?

Each of our constructions has many different possible simplicial complexes and filtrations they can result in, given the same initial parameters. For example, two instances, of drawing 50 points uniformly and then constructing an alpha complex on them, will most likely not yield the same simplicial complex and filtration.

That is why we will look at the percentage of elements in our filtrations that are in apparent pairs. This means, given some filtration F_* with m elements and r apparent pairs we calculate the percentage q of elements in F_* that are part of an apparent pair by:

$$q := \frac{200r}{m}.$$

Since every maximal alpha complex is contractible, the q-value also gives a good indication of how close the apparent gradient is to the theoretically possible perfect Morse matching with a single critical vertex.

In our first experiment we draw 300 two dimensional points uniformly 2000 times. Then we construct the largest possible alpha complexes and the corresponding filtration on each of the point sets. Afterwards we calculate q for each complex. Since we do this 2000 times we are able to analyze how different q-values are distributed. To this end we fit a Gaussian curve to the results using **scipy.stats.norm.fit** from the Python library **scipy** which calculates the mean μ and the standard deviation σ^2 of the passed data. The Gaussian fits the data quite nicely and furthermore we have observed similar fits in all dimensions and for different sizes of point clouds, which indicates that the q-values are normally distributed.

The following plot shows the fitted Gaussian for the aforementioned setup and the calculated q values as a histogram with 25 bins. This means that several results are combined in the same bin, i.e., we have some information loss here, but it is a good way of visualizing the results. On the x-axis we have the values of q and on the y-axis we have the values for the density function of the fitted Gaussian. The histogram is scaled such that the integral over it equals 1. Just like that over the fitted Gaussian.

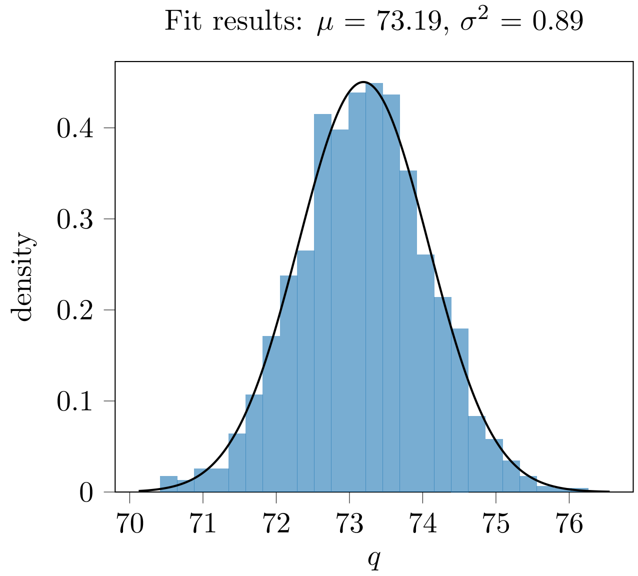


Figure 5.9: 2000 alpha complexes on 300 points drawn from a uniform distribution.

What we can take from the plot and the generated data is that we have an average q-value of 73.19% and values ranging from 70% to 76%. Meaning that more than two thirds of simplices are part of an apparent pair in all filtrations we constructed, while on average almost three quarters are part of an apparent pair.

The following figure shows a plot where we map the size of the two dimensional point clouds to the average mean q-values. We observe an almost monotone downtrend, which indicates that the percentage of simplices in apparent pairs decreases with increasing length of the filtrations. In the following two figures the scattered points are the actually calculated values and the line segments between them are linearly interpolated.

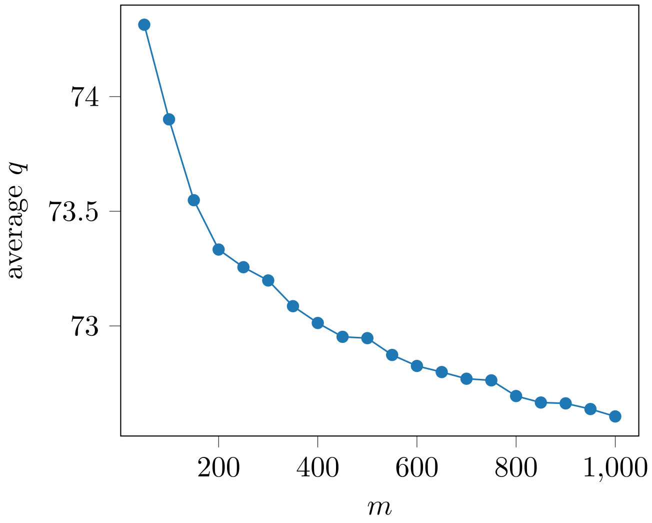


Figure 5.10: Means of 2000 averaged alpha complexes on differently sized points clouds drawn from a uniform distribution.

Since the q-values are well fitted by normal distributions, we generated a similar plot for the standard deviations.

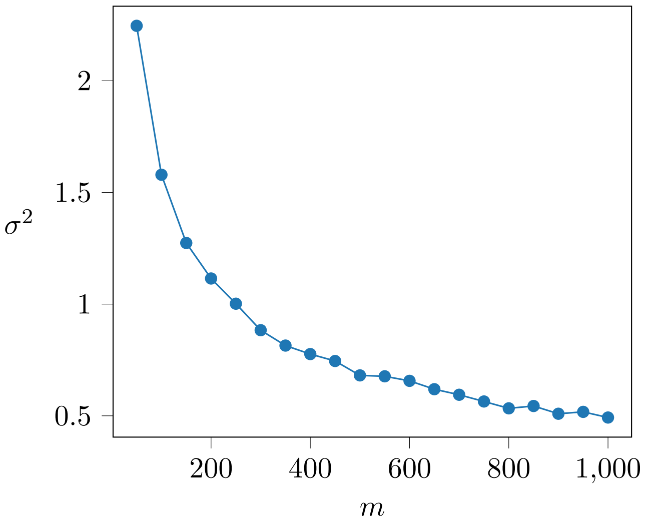


Figure 5.11: Standard deviations of 2000 averaged alpha complexes on differently sized points clouds drawn from a uniform distribution.

As we can see, the larger the complex, the less variance we have with respect to the q-values. This could be explained by some local configurations containing lots of apparent pairs while others are containing only a few. The larger the complex, the more likely that we have these evenly distributed, while in a smaller complex the probability for extreme cases is higher.

Overall this means the larger the complex the fewer of its elements are part of an apparent pair percentage wise, however larger complexes are more similar to one another in that regard than smaller complexes.

We have also calculated an average q-value for two-dimensional alpha complexes on 5000 points and got a mean of 72.20 and a standard deviation of 0.20. It would be interesting to explore if and to what value these values converge. Or if they are bounded somehow in a random setting.

As we have previously seen, it is theoretically possible to have a filtration of any length with a single apparent pair between edges and triangles.

One might expect a similar looking curve when increasing the dimension of the points instead of their number, since an alpha complex constructed on 100 points in dimension two is expected to be much smaller than one in dimension three. The mean value of the q-values however increases substantially.

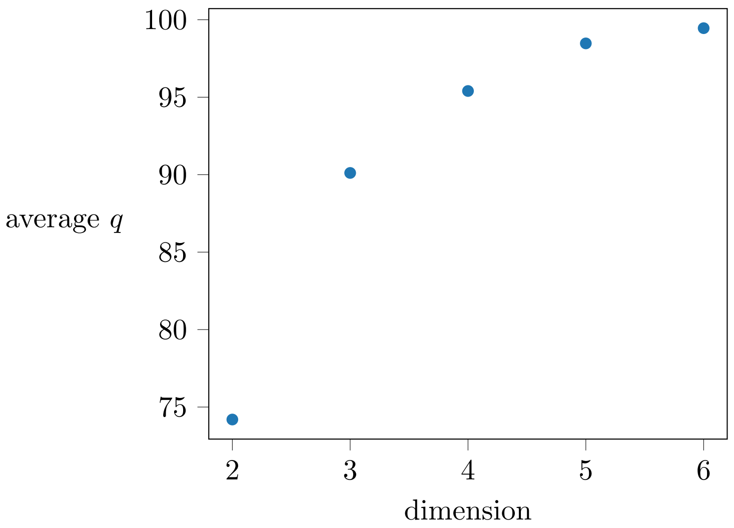


Figure 5.12: Means of alpha complexes on 30 points in varying dimensions, averaged over 50 runs.

While in dimension two, on average, about three quarters of all simplices are part of an apparent pair on an alpha complex on thirty points, in dimension six we have an average q-value of 99.46%. A partial explanation is that with increasing dimension more simplices can be paired with lower or higher dimensional simplices and the number of maximal faces of the complex decreases.

Note that in the last experiment we averaged the values over 50 runs instead of 2000. This is partially caused by the fact that simplicial complexes grow quite fast in higher dimensions and that running time was a limiting factor when performing the experiments. In the remainder of this chapter we will see values averaged over different amounts of tries due to this.

We carried out the same experiments for point clouds sampled from a multivariate Gaussian and a Gaussian mixture model. For the multivariate Gaussian we have taken the point $(0, 0)$ as mean and the identity matrix as covariance matrix. For the mixture model we chose $(0, 0)$, $(1, 1)$ and $(0, 3)$ as mean values and each covariance matrix was again the identity matrix. The values for the Gaussian mixture were picked to get some overlap between the areas of high density but not too much, since if they are too close we expect a similar set of points to that of a single multivariate distribution and if they are too far away they locally would also have the same structure as a point cloud drawn from a single multivariate distribution.

We observe the same general behavior for all three distributions. However there are noteable differences in the values of the means. Figure 5.13 shows the development of the means for all three generated data sets in one diagram. The continuous line represents the uniform distribution, the dotted line corresponds to the multivariate Gaussian and the dashed line corresponds to the Gaussian mixture model.

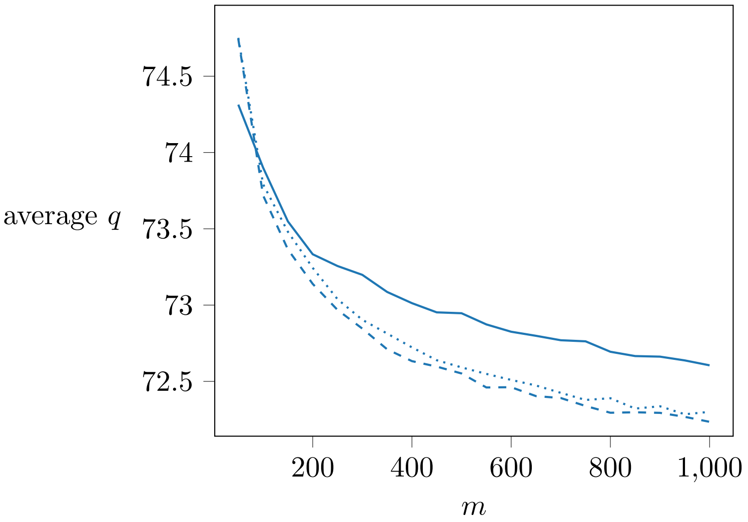


Figure 5.13: Mean q-values compared between differently drawn point clouds.

Let us consider the difference we see between the means of the two samples from Gaussian distributions and the uniform distribution. Note that the difference in number of simplices is quite small for all cases we considered. For example the average number of elements in filtrations on 1000 points is 5957.4 in case of the uniform distribution and 5971.16 in case of the Gaussian mixture. Therefore the gap between the curves is not explicable by a difference in number of elements of the filtrations.

A possible explanation for the differing developments of means is that point sets with one or several clusters and some outliers yield bad local structures with respect to simplices being a part of an apparent pair. Although the difference between the single multivariate Gaussian and the Gaussian mixture model is quite small it is nevertheless consistent for different sizes of point clouds. What exactly this potentially „bad“ structure is remains to be uncovered.

The similar curves in standard deviation strengthen the intuition, that these mainly depend on the size of the complex and that „good“ and „bad“ local configurations appear more evenly distributed.

5.6 Apparent Pairs in Standard Reduction

A natural question is if there is some kind of connection between the value q for some filtration F of simplicial complex K and the number of additions add(B) needed to reduce the boundary matrix B of F. We can guess from what we discussed in Chapter 4 that only considering the number of apparent pairs does not suffice to explain the number of additions. Nevertheless we checked if there is some linear correlation between the value of q and $\frac{\text{add}(B)}{m}$, where m is the number of elements in K. We calculated the correlation coefficient via **numpy.corrcoef** of the Python library **numpy**. The following figure shows a scatter plot of 100 alpha complexes on 100 points. On the x-axis we have the values of q and on the y-axis we have $\frac{\text{add}(B)}{m}$. Above the plot we display the calculated linear correlation coefficient.

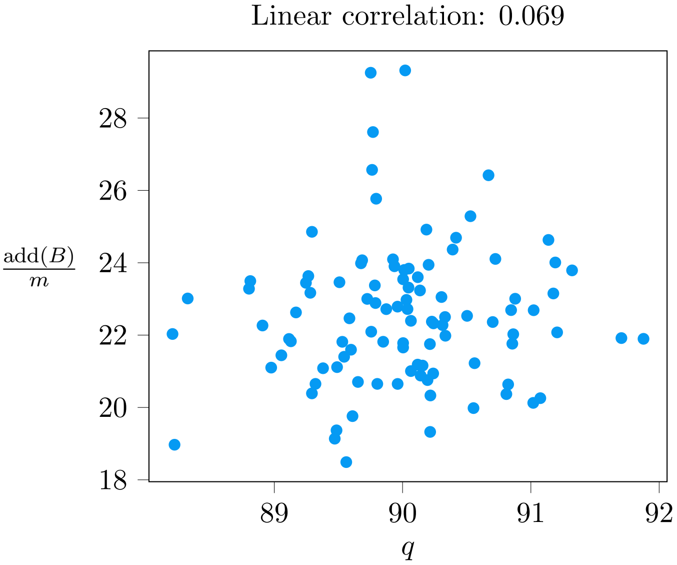


Figure 5.14: Value of q related to the number of additions per column in the standard reduction scheme for some filtration F with m elements.

Unfortunately but not unsurprisingly there is no or only a very small linear correlation between the number of additions divided by the number of elements in the complex and q. This means that the complicatedness of persistent homology computations is not explicable by the number of critical cells of the apparent gradient.

In the following we will try to illustrate how much of the work during reduction is done on apparent pairs. To be more precise, this means how much work is done on the tails of apparent pairs since the columns corresponding to the heads of the apparent pairs are reduced from the beginning. See the proof of Lemma 4.2. Note that we are not counting the number of column additions as in Definition 2.12 but the total number of addition operations, i.e. the number of additions of non-zero elements during the addition of columns.

As a preprocessing step for reduction we can set all tails of apparent pairs to zero. This will save us all additions needed to reduce the respective columns. It comes at the cost of finding the apparent pairs first however. Furthermore there are other more potent preprocessing steps that can be done.

For example for each simplex σ of a filtration one can just set the column of the youngest facet to zero, since it creates a cycle with the other facets of the simplex that has to be closed by simplex σ at the latest. This procedure also implicitly sets the columns corresponding to tails of apparent pairs to zero.

In Section 2.5 we already discussed the so called twisted reduction scheme which starts in the highest dimension and works its way down. Preprocessing with apparent pairs yields no improvement for this procedure, as every column corresponding to the tail of an apparent pair will just be set to zero without doing any calculations when its head is processed. Nevertheless the analysis we do in this section still tells us how much of the improvements of the twisted reduction scheme over the standard reduction scheme stem from columns corresponding to tails of apparent pairs. Recall that we refer to the process of setting columns to zero when we know that they correspond to a simplex creating a homology class as **clearing**. See Section 2.5.

We have already seen that in the setting of alpha complexes on random point clouds of a fixed size that the percentage of simplices that are part of an apparent pair increases with dimension. This means that we would expect higher speedups for the standard reduction scheme, see Algorithm 1, in higher dimensions. We will try to measure the speedups in the following way.

For given filtration F, let SRT (**standard reduction time**) denote the time it takes to construct and reduce some boundary matrix via the standard reduction scheme. Let ART (**apparent reduction time**) denote the time it takes to find all apparent pairs, clear columns in the boundary matrix corresponding to tails of apparent pairs, and do the standard reduction. We will consider

$$s := \frac{SRT}{ART},$$

i.e., the factor by which the overall reduction time decreases when we do clearing in the preprocessing via apparent pairs. If $s > 1$, we save time, if $s < 1$ we loose time, i.e., the cost for finding the apparent pairs and clearing the columns is higher than the gain compared to the standard reduction.

The implementations of the algorithms can be found on [GitHub], see [Spi20].

We will average each setup of parameters over hundreds of runs to make sure the values we are seeing are no anomaly caused by some external factor.

All experiments were done on an „Intel Core i7-3770 CPU 3.40GHz x 8“ processor.

Note that the implemented algorithms are not optimized for speed since the primary interest was not to develop a fast persistent homology computation but to analyse and compare examples with respect to additions in the boundary matrices.

Hence, the consideration of how many additions we save by clearing the apparent columns is more meaningful. For a given filtration F, let SRA (**standard reduction additions**), denote the number of additions in the standard reduction process and let ARA (**apparent reduction additions**) denote the number of additions on the boundary matrix which was cleared by setting tails of apparent pairs to zero. Then we define

$$a \coloneqq 100(1 - \frac{ARA}{SRA}),$$

i.e., the percentage of additions that are not necessary when reducing the cleared boundary matrix. Or in other words: A high a-value implies that the clearing yields a big improvement while a low value implies that even after clearing we still have lots of additions to do. The following table shows the average s and a values for alpha complexes on 30 points in dimensions two and three drawn from the previously mentioned distributions. We averaged the values over 200 runs.

	Execution Time Speedup		
Sample distribution	Uniform	Multivariate Normal	Gaussian Mixture
dimension two	1.96	2.09	2.13
dimension three	3.02	3.37	3.67

Table 5.1: s-value for alpha complexes on randomly generated point clouds.

	Percentage of Saved Additions		
Sample distribution	Uniform	Multivariate Normal	Gaussian Mixture
dimension two	66.64	69.90	70.98
dimension three	68.98	73.64	76.75

Table 5.2: a-value for alpha complexes on randomly generated point clouds.

Looking at the tables we see that there is substantial time and additions saved in all cases, while the speedup is generally higher on point clouds sampled from Gaussian distribution or a Gaussian mixture.

When reevaluating Figure 5.13 we might come to the conclusion that this stems from the fact that for alpha complexes on few points the q-value is higher for the Gaussian distribution and mixture than for the uniform distribution. However, as we will see later on, this is not the case.

Note that for the Gaussian mixture model in dimension three, we chose the points $(0, 0, 0)$, $(1, 1, 1)$, and $(0, 3, 0.5)$ as means and the identity matrix as covariance matrix in all cases.

When looking at the percentages of saved additions and speedups in running time in dimension three in more detail, an interesting pattern emerges. Consider the following plot in which we count how often a percentage of saved additions occurs for the multivariate Gaussian distribution in three dimensions.

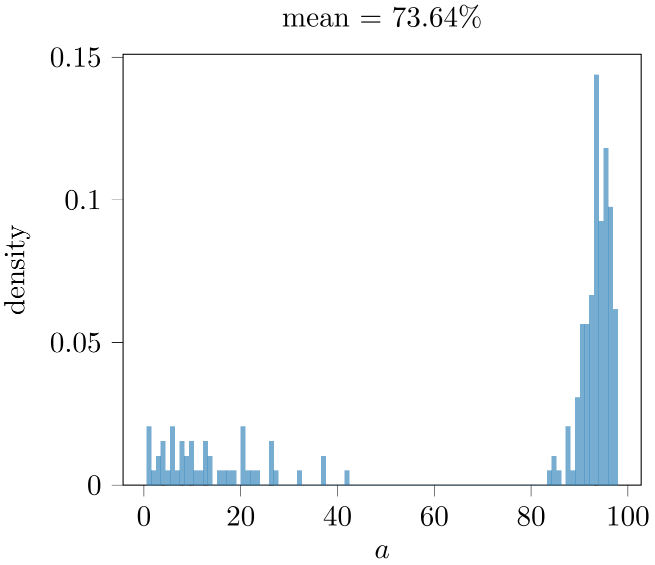


Figure 5.15: Occurences of percentages of saved additions in three dimensions.

As we can see in Figure 5.15, while we have a mean percentage of saved additions of about 73% there is actually a majority of cases with a-values of more than 90%. On the other hand there are a lot of cases with a-values below 20%. This is remarkable or even surprising, since it looks like our examples more or less fall into two categories in three dimensions. Namely one, where almost all the work is done on the tails of apparent pairs, and one where almost none of the work is done on the tails of apparent pairs. However this behaviour does not occur in two dimensions as Figure 5.16 illustrates. Note in particular that the values on the x-axis only go from 55 to 90 and not from 0 to 100 as in 5.15.

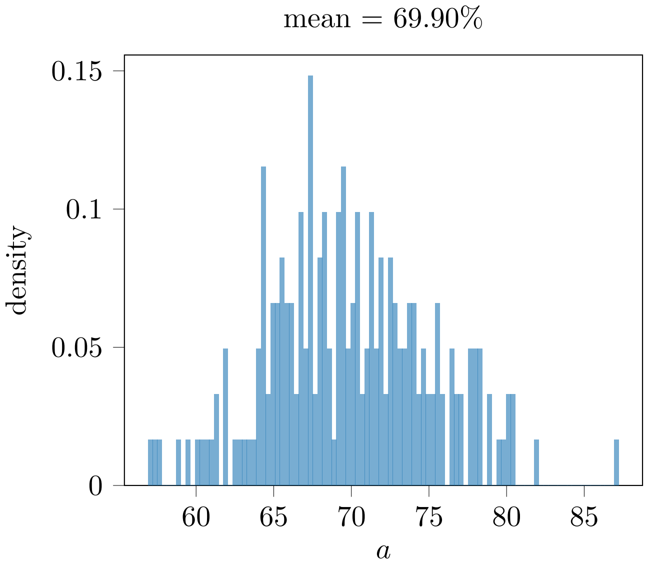


Figure 5.16: Occurences of percentages of saved additions in two dimensions.

Similar behavior showed for point clouds drawn from the uniform distribution and the Gaussian mixture. However, when increasing the number of points in three dimensions the distribution of a-values changes. Consider the following figure.

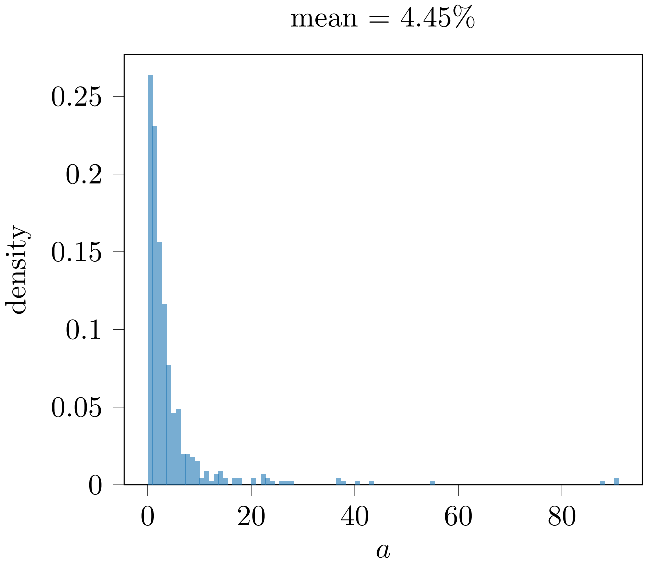


Figure 5.17: Occurences of percentages of saved additions in three dimensions on point clouds of 200 points.

Figure 5.17 in the vast majority of cases, there are very few additions saved. We still see some rare cases around the 90% mark and some values in between 10% and 60%, but we end up with a mean for saved additions of 4.45% which is substantially lower than the 73.64% we saw in Figure 5.15. Note that all constructed complexes had elements that were part of apparent pairs between 89.5% and 91%. Indeed some more detailed analysis in which we split the filtrations into two classes, namely those with a-values above and below 50% revealed no other readily accessible differences. The values of the means of the size of the filtrations, the percentage of apparent pairs, and the number of additions needed in the standard reduction without clearing were within 1% of each other for all examples generated on a fixed number of points.

The following table has the same columns and rows as Table 5.2 but this time we are looking at the averaged values of alpha complexes on 300 points.

	Percentage of Saved Additions		
Sample distribution	Uniform	Multivariate Normal	Gaussian Mixture
dimension two	57.53	60.63	60.91
dimension three	1.58	2.21	3.73

Table 5.3: a-value for alpha complexes on 300 points.

As we can see, the percentage of saved additions in dimension two is still significant, while in dimension three it is very small. Assuming that this trend continues as complexes get larger, implies that preprocessing or clearing schemes that implicitly contain the apparent pairs of some filtration generate their speedup outside of them, at least in three dimensions on alpha complexes. This might be of particular interest with respect to the widely used twisted variant of the standard reduction algorithm.

Another interesting thing to note is that again we see bigger improvements for the Gaussian distribution and mixture. But we also know that the average q value in filtrations on 300 points is lower than for the uniform case. This implies that the percentage of additions saved does not depend on the percentage of elements in apparent pairs but on some structural differences between the filtrations generated on differently sampled point clouds. Furthermore, there is some difference between points drawn from a single or multiple multivariate distributions. This might be within the margin of error for our experiments but might also be caused by structural differences between the two. Further experiments are required to shed further light on this.

5.7 Conclusions

The experiments we considered in this section are initial observations on the relation between apparent gradients, Betti numbers and complicated persistent homology computations that could give an indication in which direction a closer look might yield results.

We have seen that the majority of elements in the filtrations we generated by constructing an alpha complex on a random point cloud is part of an apparent pair. Previously, we showed in Chapter 4 how the reduction of critical cells with respect to an apparent gradient has a lower bound dependent on V-paths defined by the apparent pairs. Perhaps it is possible to find descriptions of filtrations based on their apparent gradients that already give us a good indication on whether the filtration will result in expensive or cheap persistent homology computations. A desired result would be a better understanding of why persistent homology computations in practice show linear or slightly super linear growth although the worst case bound is in $\mathcal{O}(n^3)$ [ZC05].

The behavior of the random discrete Morse function as seen in Figure 5.4 is intriguing.

Finally, the partitioning into two classes for three dimensional alpha complexes on few points, as seen in Figure 5.15, might be an interesting starting point for some further analysis.

Bibliography

[Bau19] Ulrich Bauer. Ripser: efficient computation of Vietoris-Rips persistence barcodes. 2019, arXiv 1908.02518.

[BL14] Bruno Benedetti and Frank H. Lutz. Random discrete Morse theory and a new library of triangulations. *Experimental Mathematics*, 23(1):66–94, Jan 2014. doi:10.1080/10586458.2013.865281.

[Cha00] Manoj K. Chari. On discrete Morse functions and combinatorial decompositions. *Discrete Mathematics*, 217(1):101 – 113, 2000. doi:https://doi.org/10.1016/S0012-365X(99)00258-7.

[CK11] Chao Chen and Michael Kerber. Persistent homology computation with a twist. In *Proceedings 27th European Workshop on Computational Geometry*, 2011.

[Coh73] Marshall M. Cohen. *A Course in Simple-Homotopy Theory.* Springer-Verlag, New York, 1973. doi:10.1007/978-1-4684-9372-6.

[EH10] Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction.* Jan 2010. doi:10.1007/978-3-540-33259-6_7.

[ELZ02] Herbert Edelsbrunner, David Letscher, and Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 28:511–533, Jan 2002. doi:10.1007/s00454-002-2885-2.

[For01] Robin Forman. A user's guide to discrete Morse theory. *Sém. Lothar. Combin.*, 48, Dec 2001.

[GUD20] GUDHI. Alpha complex user manual, 2020. URL `https://gudhi.inria.fr/python/latest/alpha_complex_user.html`.

[JP06] Michael Joswig and Marc E. Pfetsch. Computing optimal Morse matchings. *SIAM Journal on Discrete Mathematics*, 20(1):11–25, Jan 2006. doi:10.1137/s0895480104445885.

[JT13] Michael Joswig and Thorsten Theobald. *Polyhedral and algebraic methods in computational geometry.* Universitext. Springer, London, 2013. doi:10.1007/978-1-4471-4817-3. Revised and updated translation of the 2008 German original.

[Kah16] Matthew Kahle. Random simplicial complexes, 2016, arXiv 1607.07069.

[LM03] Nathan Linial and Roy Meshulam. Homological connectivity of random 2-complexes. *Combinatorica*, 26, 06 2003. doi:10.1007/s00493-006-0027-9.

[MN13] Konstantin Mischaikow and Vidit Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete and Computational Geometry*, 50, 09 2013. doi:10.1007/s00454-013-9529-6.

[Mor05] Dmitriy Morozov. Persistence algorithm takes cubic time in worst case. *BioGeometry News, Dept. Comput. Sci., Duke Univ*, 2005.

[Mun84] James R. Munkres. *Elements Of Algebraic Topology*. Boca Raton, 1984. doi:10.1201/9780429493911.

[MW09] R. Meshulam and N. Wallach. Homological connectivity of random k-dimensional complexes. *Random Structures and Algorithms*, 34(3):408–417, May 2009. doi:10.1002/rsa.20238.

[R.G14] R.Ghrist. *Elementary Applied Topology*. Createspace, 1.0 edition, 2014.

[Spi20] Ivan Spirandelli. Github for master thesis: Computational Topology and Random Simplicial Complexes, 2020. URL `https://github.com/IvanSpirandelli/Masterarbeit`.

[Whi39] J. H. C. Whitehead. Simplicial spaces, nuclei and m-groups. *Proceedings of the London Mathematical Society*, s2-45(1):243–327, 1939. doi:10.1112/plms/s2-45.1.243.

[ZC05] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005. doi:10.1007/s00454-004-1146-y.

[Zee63] E.C. Zeeman. On the dunce hat. *Topology*, 2(4):341 – 358, 1963. doi:https://doi.org/10.1016/0040-9383(63)90014-4.