



Thesis for the Master degree in Mathematics.
Department of Mathematical Sciences,
University of Copenhagen

Computing Persistent Homology via Discrete Morse Theory

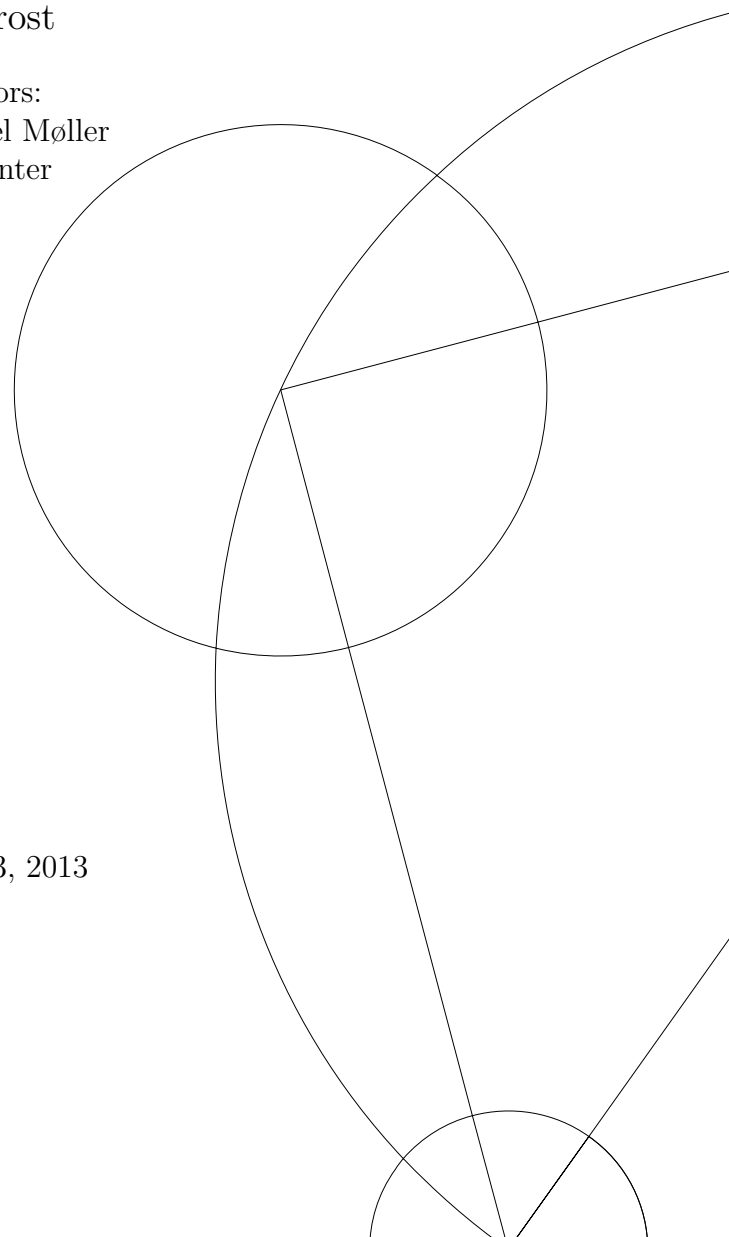
Brian Brost

Supervisors:

Jesper Michael Møller

Pawel Winter

December 13, 2013



Abstract

This report provides theoretical justification for the use of discrete Morse theory for the computation of homology and persistent homology, an overview of the state of the art for the computation of discrete Morse matchings and motivation for an interest in these computations, particularly from the point of view of topological data analysis. Additionally, a new simulated annealing based method for computing discrete Morse matchings is presented. For several problem instances this outperforms the best known heuristics for the task.

The computation of homology and persistent homology has become an important task in computational topology, with applications in fields such as topological data analysis, computer vision and materials science. Unfortunately computing homology is currently infeasible for large input complexes. Discrete Morse theory enables the preprocessing of homology computation by reducing the size of the input complexes. This is advantageous from a memory and performance point of view. The key to making efficient use of discrete Morse theory is the quick computation of optimal, or good, discrete Morse matchings.

Resume

Denne specialerapport giver et overblik over den aktuelle status for udregning af diskrete Morse matchings og motiverer interessen for disse udregninger, specielt ud fra synspunktet af topologisk dataanalyse. Yderligere præsenteres en ny metode for udregning af diskrete Morse matchings baseret på simulated annealing. I mange problemer er denne nye metode hurtigere end de bedst kendte heuristikker.

Udregning af homologi og persistent homologi har udviklet sig til et vigtigt område i algoritmisk topologi, med anvendelser indenfor områder som topologisk dataanalyse og computer vision. Desværre er udregning af homologi upraktisk for store inputkomplekser. Diskret Morse teori gør det muligt at forbehandle homologiudregninger ved at reducere størrelsen af inputkomplekset. Dette har fordele med hensyn til hukommelse og køretid. Det vigtigste element for at muliggøre en effektiv brug af diskret Morse teori er den hurtige udregning af optimale, eller i hvert fald gode, diskrete Morse matchings.

Acknowledgements

I would like to thank my mathematical supervisor Jesper Michael Møller for all his help and guidance, and in particular for helping me find such an interesting topic to investigate. I am also very grateful to Pawel Winter my computer science supervisor for all his help and guidance, and for the many very useful discussions. Frank Lutz was a tremendous help to me in understanding the state of the art with respect to computing discrete Morse matchings.

Finally, I would like to thank Jacob Andersen for helping with the translation of the abstract and proof reading.

Prerequisites

This report assumes a basic familiarity with algebra, corresponding to chapters 1-3 of [21], and with topology, corresponding to chapters 1-3 of [31]. Some familiarity with algebraic topology would also help, but is not strictly necessary. Furthermore knowledge of basic algorithmic concepts, particularly related to *NP*-completeness and integer programming is assumed.

Contents

1	Introduction	1
2	Introduction to Topology	5
2.1	Topological Spaces	5
2.2	Homeomorphisms	7
2.3	Invariants	7
2.3.1	Compactness	8
2.3.2	Homotopy Equivalence	9
2.3.3	Singular Homology	10
2.3.4	Euler Characteristic	11
2.4	Topological Data Analysis	11
3	Complexes	13
3.1	Simplicial Complexes	13
3.1.1	Geometric Simplicial Complexes	13
3.1.2	Abstract Simplicial Complexes	15
3.1.3	Geometric and Abstract Simplicial Complexes	16
3.2	Constructing Simplicial Complexes	19
3.2.1	Cech Complexes	20
3.2.2	Alpha Complexes	21
3.2.3	Vietoris-Rips Complexes	22
3.3	CW complexes	23
3.4	Topological Data Analysis	24
4	Surfaces	26
4.1	Manifolds and Surfaces	26
4.2	Orientability of Simplicial Complexes and Surfaces	27
4.3	Classification of Compact Surfaces	28
5	Homology	30
5.1	Simplicial Homology	30
5.2	Persistent Homology	35
5.3	Computing Homology and Persistent Homology	36
6	Discrete Morse Theory	38
6.1	Basic Concepts of Discrete Morse Theory	38
6.2	Simple Homotopy Equivalence	41

6.3	Proof of the Main Theorem of Discrete Morse Theory	42
6.4	Morse Inequalities	43
6.5	Gradient Vector Fields	44
6.6	Morse Matchings	46
6.7	Discrete Morse Complexes	48
	6.7.1 Example	50
6.8	Discrete Morse Theory and Persistence	51
7	Computing Morse Matchings	53
7.1	Optimal Morse Matchings	53
7.2	Hardness of Optimal Morse Matchings	54
7.3	Joswig and Pfetsch	58
	7.3.1 Integer Programming Formulation	58
	7.3.2 Branch and Cut	60
7.4	Naive heuristics	61
7.5	Lewiner et al	62
7.6	Benedetti and Lutz	63
7.7	Mrozek and Batko	63
7.8	Other approaches	64
	7.8.1 Iteration	65
	7.8.2 Cancelling Critical Simplices	65
	7.8.3 Lexicographic Ordering	65
	7.8.4 Engström	66
7.9	Simulated Annealing	66
7.10	Comparison of Methods	67
8	My Contributions	73
9	Future Work	74

1 Introduction

This report is intended to provide theoretical justification for the use of discrete Morse theory for the computation of homology and persistent homology and an overview of the state of the art for the computation of discrete Morse matchings. Motivation for an interest in these computations is provided in particular from the point of view of topological data analysis.

Since this report should be readable for computer scientists with only limited familiarity with topology, a brief introduction to the required topology is given in Section 2. One of the fundamental problems in topology is to decide if two spaces are homeomorphic or not. The obvious way of showing that two spaces are homeomorphic is to construct a homeomorphism between them, but if we wish to show that two spaces are not homeomorphic, we would have to show that there does not exist any homeomorphism between them.

It turns out the problem of homeomorphy is undecidable [26], meaning that no algorithm can always correctly determine if two topological spaces are homeomorphic. When we wish to investigate whether two spaces are homeomorphic, we often use topological invariants to show that they can not be homeomorphic. An example of a topological invariant is homotopy equivalence, but the problem of deciding whether two spaces are homotopy equivalent is also undecidable [26]. We therefore consider coarser invariants, and one of the main invariants used in computational topology is homology.

The primary motivation for this project is the field of topological data analysis, a more thorough introduction to topological data analysis will be given in Section 2.4. In topological data analysis we have two tasks, we first need to approximate the topological space from which our data originates, and secondly, we need to compute topological invariants to represent the topological properties of our approximated space. Both tasks are computationally difficult. Section 3 provides an overview of some of the most commonly used tools for solving the first task. For the second task, the most popular invariants to compute are homology and persistent homology. These will be described in detail in Section 5.

We will here give a brief example of topological data analysis, and in particular of the advantages of persistent homology. Figure 1 illustrates a possible problem in data analysis. We have a set of points such that the cluster on the left may have been sampled from a Gaussian, while the points on the right appear to have been sampled noisily from a circle, and we would like to understand where the points were sampled from. While this point set



Figure 1: Point set where the points on the left appear to have been sampled from a Gaussian and the points on right appear to have been sampled noisily from a circle.

was just generated as a simple illustration, an example of a practical problem that could be solved by the same methods is that of identifying the tunnels and voids in a protein.

If we attempted to perform cluster analysis, we could discover that there seem to be two clusters, but would learn nothing about the fact that the points on the right have been sampled from a circle. In Figure 2 we have illustrated so called alpha complexes for various values of the parameter α . These alpha complexes form our approximations for the space from which the points were sampled. As α grows, so does the alpha complex. If instead of considering the alpha complex, we just consider its shape, α can be considered as controlling the level of detail that is being examined.

The next task is to compute the homology of the alpha complexes in order to gain information about topological properties of their underlying spaces. In particular, the so called zeroth Betti number tells us that for $\alpha = 20$ and $\alpha = 30$, there are two connected components or clusters, and the first Betti number tells us that the connected component on the right has the homology of a circle, or more specifically, that it contains a hole. Unfortunately, without knowing more about our problem, there is no a priori way of identifying an appropriate α value, and for α values too small, we would identify too many connected components, whereas for α values too large, we would only detect one connected component. Even worse, for many applications there might not be a particular α value which is more appropriate than any other. The advantage of persistent homology over homology is that it enables us to automatically capture only those topological properties which persist as α grows. An underlying assumption of topological data analysis is that this

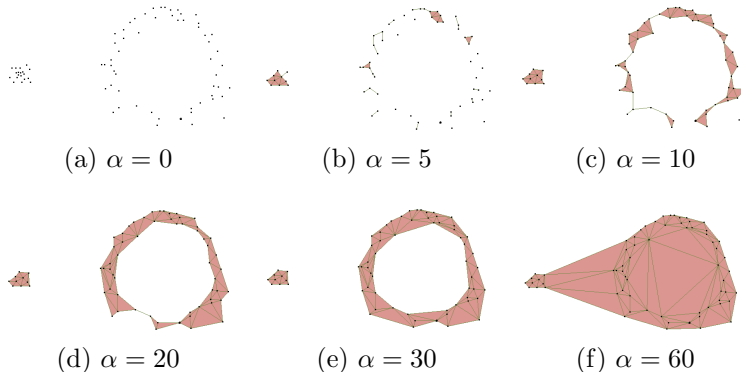


Figure 2: Alpha complexes for α between 0 and 60 for the point set from Figure 1.

will allow us to capture the important features of the data. Furthermore, the stability of persistent homology under perturbations makes it suitable for data analysis in the presence of noise [6].

An additional important advantage of topological data analysis is that the running times are generally dependent on the dimension of the complex representing the data, rather than the extrinsic dimension of the data.

Unfortunately, the best known running time for computing persistent homology is $O(n^\omega)$, where ω is the exponent for matrix multiplication [28]. The best current known estimate for ω is 2.3727 [37], and the best performing algorithm in practice is Strassen's algorithm, with a running time of $O(n^{2.807})$.

It might be possible to speed up the computation of homology by improving the running time of matrix multiplication, but improving this would solve a much broader and more difficult problem than what we are interested in, and improvements are therefore less likely to be easy to obtain. In particular, we can not use topological insights to improve the general running time of matrix multiplication.

Instead we focus on using discrete Morse theory as a preprocessing step to reduce the size of the input to our homology and persistent homology algorithms. Discrete Morse theory and its applications to homology computation will be introduced in Section 6. In order for this preprocessing via discrete Morse theory to be efficient, it turns out that we need to quickly compute discrete Morse matchings while minimizing the number of critical simplices.

Section 7 provides an overview of the state of the art for computing Morse matchings and provides a new approach based on simulated annealing which is faster than the best known heuristics for several problems. The optimal algorithm for computing Morse matchings for surfaces presented in Section 7.5 will also justify the inclusion of Section 4 introducing surfaces and presenting the theoretical background enabling Lewiner's optimal algorithm to work.

Section 7.10 provides experimental results showing the relative performances of the heuristics described in Section 7. We see that none of the tested heuristics perform better on all the simplicial complexes, but rather that the relative performances of the heuristics depend on the complex. We also see that for those complexes which are sufficiently difficult to find discrete Morse matchings for, such that the collapse and coreduction heuristics produce inconsistent results, simulated annealing performs better.

The report concludes with Section 9 describing some possible directions of future work related to this project and topological data analysis in general.

2 Introduction to Topology

This section provides a review of the most basic aspects of topology that will be required. If the reader is familiar with basic topology and algebraic topology, the only section that needs to be read is Section 2.4, the introduction to topological data analysis.

We will begin by introducing topological spaces, the building blocks of topology, along with some of the most important properties thereof. We will then proceed to give some topological invariants that are needed to understand discrete Morse theory and to classify the compact surfaces. The sections related to compact surfaces are introduced here because they form the theoretical basis for the optimal algorithm for computing discrete Morse matchings presented in Section 7.5. This section concludes with an introduction to topological data analysis. The material presented in this report has been significantly motivated by an interest in applications to topological data analysis.

2.1 Topological Spaces

Definition 2.1. A *topological space* is a set X together with a collection O of subsets called the *open sets*, such that

1. Any union of open sets is open
2. The intersection of any two open sets is open.
3. The empty set and X itself are open.

The collection O is called the *topology* on X .

A topological space is denoted by the pair (X, O) , although the O will usually be omitted, and X will be used to refer to the space and the set when there is no room for ambiguity. Note that while the definition of a topological space requires only that the intersection of any two open sets is open, it immediately follows by induction that this actually holds for any finite intersection of open sets. We call a set $A \subset X$ **closed** if $X \setminus A$ is open.

Definition 2.2. If X is a set, a *basis* for a topology is a collection \mathcal{B} of subsets of X such that for each $x \in X$ there is at least one $B_0 \in \mathcal{B}$ such that $x \in B_0$. Additionally, if $x \in B_0 \cap B_1$, then there exists a $B_2 \subset B_0 \cap B_1$ such that $x \in B_2$.

Given a basis $\mathcal{B} = \{B_i | i \in I\}$, we can then define the topology **generated** by \mathcal{B} to be the sets $U = \cup_{k \in K, K \subset I} B_k$. Bases are useful since topologies are generally too large to specify explicitly, however in this report we will only require bases so that we can later define manifolds.

Example 2.3. *The standard topology on \mathbb{R} is the topology generated by the **open intervals**, the intervals of the form (a, b) for $a, b \in \mathbb{R}$ with $b > a$.*

Definition 2.4. *If (X, O) is a topological space, with $A \subset X$, then A together with the collection of subsets $O|_A = \{U \cap A | U \in O\}$ is a topological space with the **induced** topology, and we call $(A, O|_A)$ a **subspace** of (X, O) .*

An open set containing a point $x \in X$ is called a **neighborhood** of x .

Definition 2.5. *A topological space X is called **Hausdorff** if for each pair of distinct points $x, y \in X$, there exist disjoint neighborhoods of x and y .*

An important class of topological spaces are the metric spaces. For these, the open sets are defined by a particular distance function satisfying the metric axioms.

Definition 2.6. *Let X be a set, and let $d : X \times X \rightarrow \mathbb{R}$ be a map, then d is a **metric** if the following three conditions are satisfied.*

1. $d(x, y) \geq 0$ and $d(x, y) = 0 \Leftrightarrow x = y$
2. $d(x, y) = d(y, x)$
3. $d(x, z) \leq d(x, y) + d(y, z)$

A set X together with a metric d is called a **metric space**.

We define an **open ball** around $x \in X$ for some $\epsilon > 0$ to be

$$B_\epsilon(x) = \{y \in X | d(x, y) < \epsilon\}.$$

It is simple to verify that the open balls are a basis, and the **metric topology** on X is defined to be the topology generated by the open balls. We can thus regard any metric space as a topological space, with topology generated by the open balls of the metric.

2.2 Homeomorphisms

Definition 2.7. Let X and Y be topological spaces, a map $f : X \rightarrow Y$ is called *continuous* if the preimage of any open set is open.

Definition 2.8. A map $f : X \rightarrow Y$ is called a **homeomorphism** if it is bijective, and f and f^{-1} are both continuous. In this case, we say that X and Y are homeomorphic and write $X \cong Y$.

Clearly homeomorphism of spaces is an equivalence relation. Homeomorphisms are the most important maps in topology and we consider two topological spaces which are homeomorphic to be topologically equivalent. This notion of equivalence can be understood by the fact that a topology is defined in terms of its open sets, and these are preserved by homeomorphisms.

Example 2.9. For $n \geq 1$, the punctured sphere $S^n - p$ is homeomorphic to \mathbb{R}^n . The homeomorphism $f : S^n - p \rightarrow \mathbb{R}^n$ given by

$$f(x_1, \dots, x_{n+1}) = \frac{1}{1 - x_{n+1}}(x_1, \dots, x_n)$$

is called a *stereographic projection*. This homeomorphism is visualized for $n = 1$ in Figure 3.

2.3 Invariants

We can show that two spaces X and Y are homeomorphic by constructing a homeomorphism between the two spaces. If, however, we want to show that two spaces are not homeomorphic, we use a **topological property** or **topological invariant**. This can be considered as a map f such that $X \cong Y$ implies $f(X)$ and $f(Y)$ are the same in some sense. Then we can find some invariant f_0 such that $f_0(X)$ and $f_0(Y)$ are not the same and conclude that X and Y are not homeomorphic. In other words, a topological invariant is any property which is invariant under homeomorphism.

A **complete invariant** is a map f such that $X \cong Y$ if and only if $f(X)$ and $f(Y)$ are the same. Homeomorphism can itself be regarded as a complete, but obviously trivial, invariant. Once we have introduced the Euler characteristic and orientability, we will be able to provide a complete

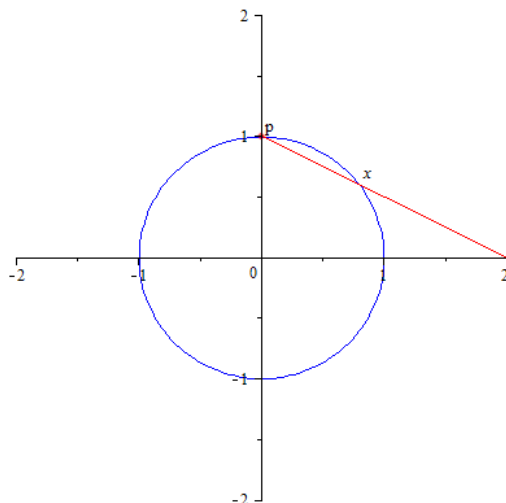


Figure 3: Stereographic projection of punctured circle onto real line. For a point $x \in S^1$, the straight line from the north pole p through x crosses the x -axis at $f(x)$.

invariant for the compact surfaces. This classification theorem of compact surfaces was one of the most important early results of algebraic topology and will be needed for the algorithm presented in Section 7.5.

Suppose we are given two invariants f and g , then we say that f is a **coarser** invariant than g if $g(X) = g(Y)$ implies $f(X) = f(Y)$. Thus, if f is coarser than g , then there are spaces that we can not tell apart using f , but that we might be able to tell apart using g .

2.3.1 Compactness

Compactness is one of the most important topological properties.

An **open cover** of a topological space X is a collection of open sets $C = \{U_\alpha | \alpha \in A\}$, such that $X \subset \cup_{\alpha \in A} U_\alpha$. If $B \subset A$ satisfies $X \subset \cup_{\alpha \in B} U_\alpha$, then we say that $\{U_\alpha | \alpha \in B\}$ is a **subcover** of C . Finally, we say that X is **compact** if any open cover of X has a finite subcover.

Example 2.10. *As a simple example of compactness, we note that the closed interval $[0, 1]$ is compact as a subspace of \mathbb{R} , whereas the open interval $(0, 1)$ is not.*

It is easy to see that compactness is a topological invariant.

Lemma 2.11. *Let $f : X \rightarrow Y$ be a homeomorphism, then X is compact if and only if Y is compact.*

Proof. Suppose, without loss of generality, that X is compact. Let $C = \{U_\alpha | \alpha \in A\}$ be an open cover of Y , then $D = \{f^{-1}(U_\alpha) | \alpha \in A\}$ is an open cover of X . Now since X is compact there exists a $B \subset A$ such that $D' = \{f^{-1}(U_\alpha) | \alpha \in B\}$ is a finite cover of X and it follows that $C' = \{U_\alpha | \alpha \in B\}$ is a finite subcover of C , and therefore that Y is compact. \square

2.3.2 Homotopy Equivalence

A central concept that will be needed when we introduce discrete Morse theory is that of homotopy equivalence.

Definition 2.12. *Given topological spaces X and Y , and continuous maps $f_0, f_1 : X \rightarrow Y$, we say that f_0 and f_1 are **homotopic**, and denote it by $f_0 \simeq f_1$, if there exists a continuous map $F : X \times I \rightarrow Y$, such that $f_0(x) = F(x, 0)$ and $f_1(x) = F(x, 1)$ for all $x \in X$. Here I denotes the unit interval $[0, 1] \subset \mathbb{R}$.*

Definition 2.13. *The spaces X and Y are **homotopy equivalent**, denoted $X \simeq Y$, if there are maps, called **homotopy equivalences**, $f : X \rightarrow Y$ and $g : Y \rightarrow X$, such that $g \circ f \simeq id_X$, and $f \circ g \simeq id_Y$, where id_X and id_Y are the identity maps on X and Y respectively.*

Homotopy equivalence is an equivalence relation of spaces and if two spaces are homotopy equivalent, we say that they have the same **homotopy type**. A space which has the homotopy type of a point is called **contractible**. It is immediate from the definition of homotopy equivalence that it is a topological invariant.

Observation 2.14. *If two spaces X and Y are homeomorphic, they are also homotopy equivalent.*

Proof. If $f : X \rightarrow Y$ is a homeomorphism with inverse $f^{-1} : Y \rightarrow X$, then $f^{-1} \circ f = id_X$, and $f \circ f^{-1} = id_Y$, so in particular, they are homotopy equivalences. \square

The converse is not generally true, as the below example illustrates

Example 2.15. *A d -ball is homotopy equivalent to a 1-point space, but clearly not homeomorphic to it, since they have different cardinalities.*

Recall that we said that homotopy equivalence is important to discrete Morse theory. This is because the central Theorem of discrete Morse theory, Theorem 6.5, tells us that given a simplicial complex X , there exists an often significantly smaller CW-complex which is homotopy equivalent to X . These complexes will be defined properly in Section 3.

Let X be a topological space and A a subspace of X , then a continuous map $r : X \rightarrow A$ is a **retraction** if the restriction of r to A is the identity map on A , that is, if $r \circ i = id_A$, where i is the inclusion map and id is the identity map. If additionally, we have that $i \circ r \simeq id_X$, r is a special type of homotopy equivalence called a **deformation retract**.

2.3.3 Singular Homology

We will not go into detail here, but will simply note that for every topological space X , we can define its **singular homology groups**. These are groups $H_n(X)$, defined for $n \in \mathbb{N}_0$. We will in Section 5 introduce simplicial homology in detail, along with the fact that when the simplicial homology groups are defined, they are isomorphic to the corresponding singular homology groups. For an introduction to singular homology, see chapter 2 of [19]. Homology is the most important topological invariant in computational topology, and the reason for this is that it provides a good balance between computational tractability and coarseness.

Having defined singular homology, we can state the following theorem, demonstrating the homotopy invariance of singular homology. This will be important later for applications of discrete Morse theory to computational homology, since discrete Morse theory allows us to simplify complexes while preserving homotopy type, and the below theorem ensures that preserving homotopy type also preserves homology.

Theorem 2.16. [19] *Let $f : X \rightarrow Y$ be a continuous map, then there exists an induced homomorphism $f_* : H_n(X) \rightarrow H_n(Y)$ for each n . Furthermore, if f is a homotopy equivalence, f_* is an isomorphism.*

An immediate consequence of this is the weaker statement that singular homology is a topological invariant.

Observation 2.17. *If X and Y are homeomorphic, $H_n(X)$ and $H_n(Y)$ are isomorphic for each n .*

We will use the following topological property frequently throughout this report.

Definition 2.18. *The n 'th **Betti number** is the rank of $H_n(X)$.*

2.3.4 Euler Characteristic

The Euler characteristic is an important topological invariant in its own right, but for this report we require the Euler characteristic because it will later allow us to prove the NP -hardness of finding optimal so called discrete Morse matchings. It will also allow us to classify the compact surfaces in Section 4.

Definition 2.19. *The **Euler characteristic** of X can be defined as*

$$\chi(X) = \sum_{i=0}^{\infty} (-1)^i b_i,$$

where b_i is the i 'th Betti number of X .

Since the Euler characteristic is here defined in terms of the singular homology groups, it follows from the definition that it is an invariant of homotopy equivalence, and therefore also a topological invariant. It is also clear that the Euler characteristic must be a coarser invariant than the homology groups.

Corollary 2.20. *If X and Y are homotopy equivalent, $\chi(X) = \chi(Y)$.*

2.4 Topological Data Analysis

We will now give a slightly more formal definition of what we mean by **topological data analysis**. In topological data analysis we assume that we have a finite set $S \subset Y$, where Y is a topological space, of points sampled with noise from an unknown topological space X . Our goal is to recover properties of the topological space X , given our dataset S .

We differentiate between **extrinsic** properties, which are properties of Y , and **intrinsic** properties, which are properties of X or our representation of X . This distinction is important, particularly in the analysis of high dimensional data, since often the intrinsic dimension of the representation of the data will be much lower than the extrinsic dimension in which the data is embedded.

In order to analyze data, we need to make additional assumptions beyond the data itself, since otherwise we have no basis to make inferences. Thus, for example in principal component analysis, we assume that the space X is a linear subspace of Y . In manifold learning we assume that X is a manifold. Since most real-world point sets are sampled from spaces that violate these assumptions, methods such as principal component analysis or manifold learning may not work as well as they do when their assumptions are met [41]. One of the core benefits of topological data analysis is that we make very weak assumptions about the intrinsic properties of the data. It is therefore potentially applicable to a much wider range of problems than the above mentioned methods, and can be an ideal first tool for data analysis, when we know very little about our data.

Given our finite set of points S , the general approach used in topological data analysis involves two steps. We first need to approximate the space X by a combinatorial structure K . This can for example be a simplicial or cubical complex. We will cover approaches to this step in Section 3.2. Secondly, we compute topological invariants of K , giving us approximate information about the topological properties of the space X . We have already provided some invariants, and will focus on the most important invariant in topological data analysis, persistent homology, in Section 5.

3 Complexes

We will begin this section by introducing geometric simplicial complexes, and then define abstract simplicial complexes, which are easier to work with. We will also show that given one type of simplicial complex, we can construct the other, and that these constructions are unique up to certain isomorphisms which will be defined below. Having introduced simplicial complexes, we will provide some of methods of constructing simplicial complexes which are particularly useful in topological data analysis. We conclude this section by introducing CW complexes which can be regarded as a generalization of simplicial complexes. The Morse complex introduced in Section 6.7 is an example of a CW complex.

3.1 Simplicial Complexes

3.1.1 Geometric Simplicial Complexes

A set $T = \{a_0, \dots, a_k\}$ of points in \mathbb{R}^d is called **affinely independent** if the only solution to the set of equations $\sum_{i=0}^k \alpha_i a_i = 0$ and $\sum_{i=0}^k \alpha_i = 0$ is the zero solution $\alpha_0 = \dots = \alpha_k = 0$. As a simple illustration of this definition, 3 points are affinely independent if they are not all on a line defined by any 2 of them. Similarly, 4 points are affinely independent, if they are not all on a plane defined by three of them.

Definition 3.1. *Given a set $T = \{a_0, \dots, a_k\}$ of $k + 1$ affinely independent points in \mathbb{R}^d for $k \leq d$, a k -**simplex**, denoted by σ_T , is the convex hull of T . The simplices σ_U , $U \subset T$ are the **faces** of σ_T .*

In the lower dimensions, the k -simplices are familiar shapes. A 0-simplex is simply a vertex, a 1-simplex is an edge, a 2-simplex is a triangle and a 3-simplex is a tetrahedron.

Definition 3.2. *A **geometric simplicial complex** in \mathbb{R}^d is a finite collection of simplices K in \mathbb{R}^d such that*

- (i) *Every face of a simplex of K is in K .*
- (ii) *The intersection of any two simplices of K is a face of each of them.*

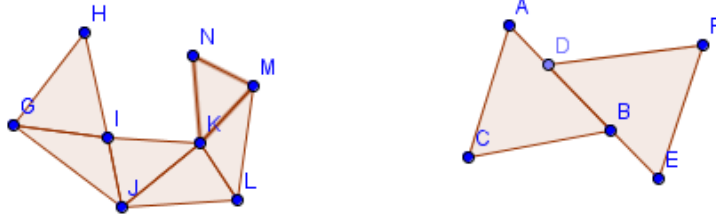


Figure 4: The first collection is a simplicial complex. The second collection is not a simplicial complex since the two triangles intersect only in part of an edge.

An important and immediate consequence of this definition is that the intersection of any two simplices is itself in K . See Figure 4 for some simple examples illustrating simplicial complexes.

A subcollection \mathcal{L} of K is called a **subcomplex** of K if it is itself a simplicial complex. The subcomplex consisting of all k -simplices for $k \leq p$ is called the p -**skeleton** of K , and is denoted by $K^{(p)}$. The **vertices** of K are then the simplices in $K^{(0)}$ and we call $K^{(1)}$ the **graph** of K . An important class of simplicial complexes in computational topology are the **clique complexes**, where the higher dimensional simplices are fully determined by the cliques in the 1-skeleton. In Section 3.2.3 we will encounter a type of clique complex called the Vietoris-Rips complex.

Definition 3.3. Let $|K|$ be the subspace of \mathbb{R}^d that is the union of all the simplices of K with the induced topology. We call the space $|K|$ the **underlying space** of K .

We say that a simplicial complex K is a **triangulation** of a topological space X if $|K|$ is homeomorphic to X .

We can now introduce simplicial maps.

Lemma 3.4. [32] Let K and L be complexes, and let $f : K^{(0)} \rightarrow L^{(0)}$ be a map. Suppose that whenever the vertices v_0, \dots, v_n of K span a simplex of K , the points $f(v_0), \dots, f(v_n)$ are vertices of a simplex of L , then f can be extended to a continuous map $g : |K| \rightarrow |L|$ such that

$$x = \sum_{i=0}^n t_i v_i \implies g(x) = \sum_{i=0}^n t_i f(v_i).$$

Definition 3.5. *In light of Lemma 3.4, we define g to be the **simplicial map** induced by the vertex map f .*

Lemma 3.6. [32] *Suppose $f : K^{(0)} \rightarrow L^{(0)}$ is a bijective correspondence such that the vertices v_0, \dots, v_n of K span a simplex of K , if and only if the points $f(v_0), \dots, f(v_n)$ span a simplex of L . Then the induced simplicial map $g : |K| \rightarrow |L|$ is a homeomorphism.*

Definition 3.7. *We call the map g a **simplicial isomorphism** of K with L .*

Geometric simplicial complex can be difficult to work with. The challenge presented is that for anything other than very small complexes, there are a huge number of simplices which have to intersect in very specific ways, as required by Definition 3.2(ii). Abstract simplicial complexes allow us to disregard this geometric complication.

3.1.2 Abstract Simplicial Complexes

Definition 3.8. *An **abstract simplicial complex** is a set of subsets S of a finite set of vertices V such that if $\alpha \in S$ and $\beta \subset \alpha$, then $\beta \in S$.*

Note that V need not be finite in general, but that we have used this definition since computational results are our main interest.

The subsets S are called **simplices**, and we call a simplex containing $p + 1$ elements a **p -dimensional** simplex and denote it by $\alpha^{(p)}$, or simply α when the dimension is not important. The **dimension** of the simplicial complex is the maximum dimension among its simplices. If $\alpha \subset \beta$, we say that α is a **face** of β , and write $\alpha < \beta$. Similarly, we say that β is a **coface** of α and write $\beta > \alpha$. If additionally $|\alpha| = |\beta| - 1$, we say that α is an **immediate face** of β and that β is an **immediate coface** of α . We call a face α of β a **free face** if α is not a face of any other simplex in S . We call the 0-dimensional simplices of the complex its **vertex set**. Finally, a simplex which is not a proper face of any simplex is called a **facet** and a simplicial complex is **pure** if all its facets have the same dimension. Note that our above definitions imply that the empty set is a (-1) -dimensional simplex, however we will generally disregard this simplex from consideration to simplify the presentation.

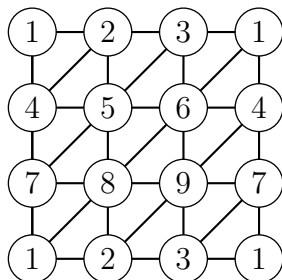


Figure 5: A triangulation of the torus. Here we identify the edges as indicated by the vertex numbers.

Example 3.9. *An example of a simplicial complex is*

$$\{\{1, 2, 3\}, \{1, 3\}, \{1, 2\}, \{2, 3\}, \{1\}, \{2\}, \{3\}\}$$

This is simply the abstract simplicial complex corresponding to a triangle. It is a triangulation of the 2-ball. A slightly more complicated example is given in Figure 5.

In practice we can provide a compact representation of a simplicial complex just by listing its facets, since they are the only simplices whose membership of the complex can not be inferred from the membership of another simplex. Thus we could represent the abstract simplicial complex from example 3.9 just by the set $\{\{1, 2, 3\}\}$.

An **isomorphism** between two abstract simplicial complexes S and S' is a bijective mapping f from the vertex set of S to the vertex set of S' such that $\{a_0, \dots, a_n\} \in S$ if and only if $\{f(a_0), \dots, f(a_n)\} \in S'$.

3.1.3 Geometric and Abstract Simplicial Complexes

Having introduced geometric and abstract simplicial complexes, we can explain the relationship between the two.

Definition 3.10. *Let K be a geometric simplicial complex with vertex set V , then the set of all subsets $\{a_0, \dots, a_n\}$ of V such that the vertices a_0, \dots, a_n span a simplex of K is called the **vertex scheme** of K .*

The vertex scheme of a geometric simplicial complex is an example of an abstract simplicial complex. It closely links geometric and abstract simplicial complexes, as Theorem 3.11 demonstrates.

Theorem 3.11. [32]

- Every abstract simplicial complex S is isomorphic to the vertex scheme of some geometric simplicial complex K .
- Two geometric simplicial complexes are simplicially isomorphic if and only if their vertex schemes are isomorphic as abstract simplicial complexes.

Definition 3.12. If the abstract simplicial complex S is isomorphic with the vertex scheme of the simplicial complex K , we call K a **geometric realization** of S . It is uniquely determined up to a simplicial isomorphism.

While we have seen above that any abstract simplicial complex has a geometric realization, Theorem 3.13 gives a guarantee on the dimension of the geometric realization.

Theorem 3.13. [12] Every abstract simplicial complex of dimension d has a geometric realization in \mathbb{R}^{2d+1} .

Note that a d -dimensional abstract simplicial complex can easily have a geometric realization in \mathbb{R}^d . Recall example 3.9, this has the obvious geometric realization in \mathbb{R}^2 , given in Figure 6.

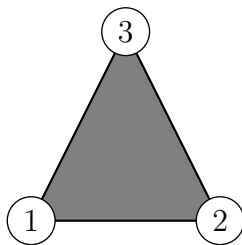


Figure 6: Geometric realization in \mathbb{R}^2 of $\{\{1, 2, 3\}\}$.

Example 3.14. The graph K_5 is an abstract simplicial complex of dimension 1 which does not have a geometric realization in \mathbb{R}^2 or lower. It is illustrated in Figure 7. More generally, any non-planar graph will not have a geometric realization in \mathbb{R}^2 or lower. Even more generally, the d -skeleton of the $(2d + 2)$ -simplex cannot be embedded in \mathbb{R}^{2d} according to the Van Kampen-Flores Theorem.

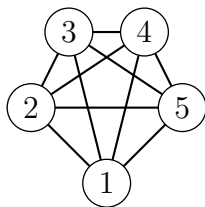


Figure 7: K_5

The proof of Theorem 3.13 is slightly complicated, and to get an upper bound on the embedding dimension we can also consider the following much more straightforward construction.

Example 3.15. *Let S be an abstract simplicial complex with vertex set V given by $\{v_1, \dots, v_{|V|}\}$. Then we can obtain a geometric realization of S in $\mathbb{R}^{|V|}$ by defining for an abstract simplex $\{v_1, \dots, v_t\}$ the corresponding geometric simplex to be the convex hull of the points $\{e_1, \dots, e_t\}$, where e_i is the i 'th standard basis vector. This is illustrated for $S = \{\{1, 2, 3\}\}$ in Figure 8.*

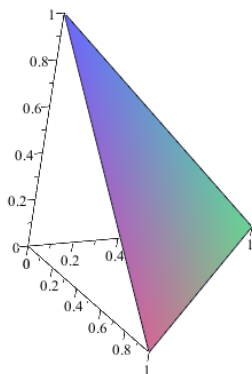


Figure 8: Geometric realization of $S = \{\{1, 2, 3\}\}$ in \mathbb{R}^3 .

In light of the close relationship between abstract and geometric simplicial complexes, we will refer simply to simplicial complexes, without needing to specify which type we mean.

3.2 Constructing Simplicial Complexes

Recall that in topological data analysis our input is not a simplicial complex, but rather a finite set $S \subset Y$ of points embedded in a topological space Y sampled noisily from some topological space X . Our task is therefore to construct an approximation of X and three possible approaches will be given in Sections 3.2.1, 3.2.2 and 3.2.3. These approaches will involve either constructing so called nerves of covers, or clique complexes.

Before moving on to a description of the approaches we use to approximate X , we need to introduce an important result in topological data analysis, Leray's Nerve Lemma.

Definition 3.16. *Given an open cover of S , $U = \{U_i\}_{i \in I}$, $U_i \subset Y$, where I is some indexing set, the **nerve** of U , denoted by N , is given by*

- $\emptyset \in N$ and
- If $\bigcap_{j \in J} U_j \neq \emptyset$ for $J \subset I$, then $J \in N$.

It follows immediately from the definition that the nerve of a cover is a simplicial complex.

In the following methods the union of the sets in an open cover of X forms our approximation of X and the nerve will function as our finite combinatorial representation of X . We define an open cover U to be **good** if all U_i are contractible, as are their nonempty finite intersections. We then have the following Nerve Lemma due to Leray.

Lemma 3.17. *[41] The underlying space of the nerve of a good cover is homotopy equivalent to the union of the sets in the cover.*

This motivates our definition of a good cover. It is one for which the nerve provides an accurate representation of the topology of the cover up to homotopy equivalence. Specific examples of covers and nerves will be given in Sections 3.2.1 and 3.2.2.

In order to deal with noise introduced by the sampling process, one possible solution is persistent homology which will be introduced in Section 5.2. For persistent homology, instead of trying to just construct a simplicial complex, we will construct what is known as a filtration of a simplicial complex, which will allow us to approximate the topology of X across a range of scales or resolutions.

Definition 3.18. A *filtration* on a simplicial complex K is a collection of subcomplexes

$$\emptyset \subset K^0 \subset K^1 \subset \dots \subset K^N = K$$

For the following techniques we will make the additional assumption that our embedding space Y is a metric space. Doktorova and Zomorodian provide an overview of various techniques where we do not even need to assume that Y is a metric space [10].

3.2.1 Cech Complexes

We denote by $B_\epsilon(x)$ the open ball of radius ϵ centered at x . Thus,

$$B_\epsilon(x) = \{y \in Y \mid d(x, y) < \epsilon\},$$

and we have the following open cover of S

$$U_\epsilon = \{B_\epsilon(x) \mid x \in S\}.$$

We define the **Cech complex** C_ϵ to be the nerve of U_ϵ .

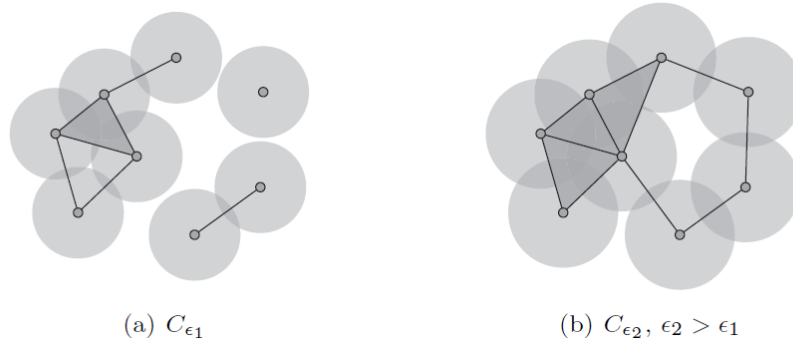


Figure 9: A set of 8 points in the plane with covers U_ϵ and Cech complexes C_ϵ for two difference values of ϵ . [41]

To see that U_ϵ is a good cover, we must show that any finite intersection of sets in U_ϵ is contractible. Since intersections of convex sets are convex and each ball in U_ϵ is convex, the fact that U_ϵ is a good cover follows from the fact that convex sets are contractible.

Since the cover is good, the Cech complex is homotopy equivalent to the union of the sets in U_ϵ according to the Nerve Lemma.

We have defined the Cech complex for a given value of ϵ , but since our points are sampled noisily, it is likely that some of the topological properties of C_ϵ will be caused by noise. We will therefore consider the simplicial complex C_∞ and its filtration. This is possible since for $\epsilon_1 \leq \epsilon_2$, C_{ϵ_1} is clearly a subcomplex of C_{ϵ_2} . As we increase our scale parameter ϵ , the Cech complex grows from $C_0 = \emptyset$ to eventually being just an $(|S| - 1)$ -simplex for C_∞ . Since an n -simplex has 2^{n+1} faces, the Cech complex has size exponential in the size of S and the dimension of the Cech complex can be significantly larger than the embedding dimension. The Cech complex is computationally infeasible to compute in practice, so we will now look at complexes which are more practical.

3.2.2 Alpha Complexes

Suppose now that for each of our points in $p \in S$ we assign a real valued weight w_p . We then define the **power region**, $R(p)$, of a point $p \in S$ to be the set of points $y \in Y$ closest to it in terms of the **power distance**, $\pi(p, y) = |py|^2 - w_p$. The **power diagram** is the subdivision of Y into power regions, and the dual of the power diagram is called the **regular triangulation** of S . Note that if the weights for all $p \in S$ are 0, or just equal to each other, the power diagram and regular triangulation are just the well known Voronoi diagram and Delaunay triangulation.

Recall that the Cech complex is the nerve of a cover where we took an open ball of radius ϵ centered at each point $x \in S$. We can now use the power regions to restrict the possible intersections between the elements of our cover. We define the following cover

$$U_\epsilon = \{B_\epsilon(x) \cap R(x) | x \in S\}$$

and let the **alpha complex**, A_ϵ , be the nerve of the cover. Clearly U_ϵ is a good cover since each power region is convex. As we increase our scale parameter ϵ , the alpha complex grows from $A_0 = \emptyset$ to eventually being the regular triangulation of S for A_∞ . Since the underlying spaces of the covers of the alpha and Cech complexes are the same, it follows immediately from the Nerve Lemma that the complexes are homotopy equivalent.

If the points in S are in what is known as **general position**, the maximum dimension of the alpha complex is no greater than the embedding

dimension of S . There are several assumptions required for this notion of general position to be fulfilled, the details of which can be found in [11]. For example, if S is embedded in \mathbb{R}^d , we require that any $d + 1$ or fewer points in S be affinely independent. The important thing to note is that even if the points in S are not in general position, we can simulate general position using **simulation of simplicity** [14], or we can simply perturb the points slightly to obtain a set of points which almost certainly is in general position.

The alpha complex of n points can be computed in $O(n^2)$ time in \mathbb{R}^3 , and the alpha complex is therefore a good choice for topological data analysis in dimension 3 or lower. Alpha complexes are particularly interesting from the point of view of low dimensional topological data analysis, since there exist fast algorithms for computing homology and persistent homology for them [8, 13].

An additional advantage of alpha complexes is that they allow us to weight points differently according to some notion of importance or size. For example, in protein structure prediction, our points may represent atoms, and we may want to give them different weights according to their atomic radii. Similarly, if we know that our points are not sampled uniformly, we can weight the points from sparsely sampled areas higher than those from densely sampled areas.

3.2.3 Vietoris-Rips Complexes

It is infeasible to compute Čech complexes in practice, while alpha complexes can currently only be efficiently computed in dimension 3 or lower. The Vietoris-Rips complex although not generally as fast to compute as the alpha complex in low dimensions, can be somewhat efficiently computed in higher dimensions.

The Vietoris-Rips complex will be defined slightly differently to the above two complexes, and we first need to define the ϵ -**neighborhood graph** on S , given by

$$E_\epsilon = \{\{u, v\} \mid d(u, v) \leq 2\epsilon, u \neq v \in S\}.$$

A **clique** in a graph is a subset of vertices such that their induced subgraph is complete and we say a clique is **maximal** if adding any vertex would make the induced subgraph incomplete. A **clique** or **flag** complex is a simplicial complex whose facets are the maximal cliques of a graph. The **Vietoris-Rips** complex V_ϵ is the clique complex of the ϵ -neighborhood graph. In particular, this means that it is fully defined by its graph, allowing a compact

representation of the simplicial complex. V_ϵ can, like C_ϵ , simply be an $(|S| - 1)$ -simplex, this occurs if the ϵ -neighborhood graph is complete. In practice, we restrict our computations to some maximum scale or dimension of the Vietoris-Rips complex to make computations feasible.

The Vietoris Rips complex is not necessarily homotopy equivalent to the Cech complex, and it may instead be regarded as an approximation of the Cech complex. In particular the filtrations of Cech and Vietoris-Rips complexes are closely related by the following result.

Theorem 3.19. [7] *For any $\epsilon > 0$, we have the following inclusions,*

$$V_\epsilon \subset C_{\epsilon\sqrt{2}} \subset V_{\epsilon\sqrt{2}}.$$

The persistent homologies, defined in Section 5.2, of filtrations of Cech and Vietoris-Rips complexes are therefore related.

As part of this project, the construction of Vietoris-Rips complexes was implemented in ProGAL, a computational geometry library in Java. This implementation was based on [39]. There are two steps to constructing Vietoris-Rips complexes. We must first construct the 1-skeleton of the complex, the goal is to establish for each pair of points if their distance is less than the parameter ϵ . This can be done with a simple all-pairs algorithm in time $O(n^2)$. Approximate algorithms can be faster, but at the price of only obtaining an approximately correct Vietoris-Rips complex. This may in some circumstances be acceptable, since the Vietoris-Rips complex is itself only an approximation of the underlying space of the point set. For the second step of constructing the Vietoris-Rips complex we can disregard the geometry of the point set and simply need to find the cliques of the graph constructed in the first step. Since the maximal clique problem is NP -hard, this step is hard and in practice we often restrict the number of dimensions for which we compute the Vietoris-Rips complex. The simplest algorithm, and the one implemented here, finds all the cliques of size k in order to determine the simplices of dimension $k - 1$. This takes $O(n^k k^2)$ time and is therefore exponential in the dimension, but polynomial if we fix the dimension parameter.

3.3 CW complexes

As for simplicial complexes, we will only consider finite CW complexes. We will not provide a detailed introduction to CW complexes, preferring to provide a brief sketch. We begin by defining a d -**cell** to be a space which is homeomorphic to B^d .

Definition 3.20. A **cell complex** or **CW complex** X is a topological space constructed inductively in the following manner.

- X^0 is a discrete, finite set of points which we regard as 0-cells
- We form the n -skeleton X^n from X^{n-1} by attaching n -cells via **attaching maps** from the boundary of the n -cell to the $n - 1$ skeleton, $S^{n-1} \rightarrow X^{n-1}$. For some $N \in \mathbb{N}$, this process ends and $X^N = X$.

For a formal introduction to CW complexes, see chapter 0 of [19]. For now, we simply note that CW complexes are a generalization of the simplicial complex already introduced, where we allow more general n -cells instead of n -simplices, and we allow more general attaching maps. We require CW complexes since the central result of discrete Morse theory will be a homotopy equivalence between a simplicial complex and a corresponding Morse complex which is a CW complex. The fact that this Morse complex can be significantly smaller than the simplicial complex, while still being homotopy equivalent to it, motivates our interest in discrete Morse theory from a computational point of view. This is because it is generally quicker to compute topological invariants for a smaller complex, but also because the memory requirements of the larger simplicial complexes can be a problem.

In Sections 5 and 6, we have focused on simplicial complexes only, in order to simplify the presentation since the complexes we are most interested in from the point of view of topological data analysis are simplicial complexes. We could have generalized those sections to CW complexes instead. A particularly interesting class of CW complexes for some aspects of topological data analysis such as computer vision, are the so called cubical complexes. See [23] for a detailed introduction to cubical complexes.

3.4 Topological Data Analysis

We have described constructing the various complexes as an attempt to approximately capture the topology of the space from which our point set was sampled. However we have not made any statements regarding the accuracy of these approximations. Progress has only recently begun to be made in the area of providing conditions that guarantee that our various constructions correctly reconstruct the topology of X . Some of the currently known conditions under which the Cech, alpha and Vietoris Rips complexes provide topologically correct reconstructions are given in [34, 2]. For example,

it has been proven that if the underlying shape is a smooth manifold and the sample is sufficiently dense, the Čech and alpha complexes will have the correct homotopy type [2]. In general, the Čech and alpha complexes better capture the topology of X , but this comes at the expense of their being computationally more expensive to construct. In low dimensions, since the alpha complex is actually fast to compute, it is the best choice to use.

4 Surfaces

We will now define the surfaces, an important class of topological spaces. The surfaces will be of particular interest when we introduce discrete Morse theory, since they are one of the few types of topological spaces for which we can optimally compute discrete Morse matchings efficiently. See Section 7.5 for further details.

We will begin by defining manifolds and surfaces and then define their orientability. We will then have introduced all the concepts needed to state the classification theorem for the compact surfaces.

4.1 Manifolds and Surfaces

Definition 4.1. An *m -manifold* is a locally euclidean Hausdorff space X with a countable basis, where *locally euclidean* means that for some $m \in \mathbb{N}$, each point in X has a neighborhood homeomorphic to \mathbb{R}^m . If instead each point in X has a neighborhood homeomorphic to \mathbb{R}^m or $\mathbb{R}^{m-1} \times \mathbb{R}_+$, X is an *m -manifold with boundary*. A 2-manifold is called a *surface* and a 2-manifold with boundary is called a *surface with boundary*.

Example 4.2. The 2-sphere, the torus are surfaces and the Möbius strip is a surface with boundary. They are illustrated in Figure 10.

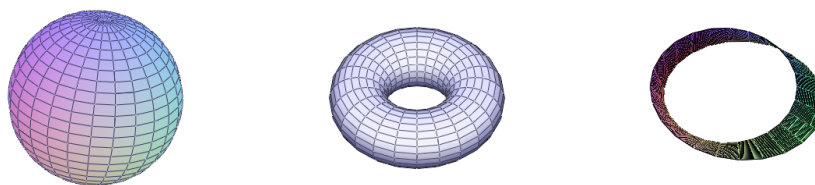


Figure 10: From left to right, the 2-sphere, the torus and the Möbius strip.

If we cut open disks out of any of the surfaces in Figure 10, they become surfaces with boundary.

4.2 Orientability of Simplicial Complexes and Surfaces

We delayed introducing surfaces until we had introduced simplicial complexes because rather than providing the standard definition of orientability of surfaces in terms of normal vectors, we will define orientability of their triangulations. We will then use the fact that every surface has a triangulation and that orientability is a topological invariant to see that our notion of orientability is well defined for surfaces in general. This method of defining orientable surfaces minimizes the amount of extraneous material that needs to be introduced.

Let σ be a simplex, then we say that two orderings of its vertices are equivalent if they differ by an even permutation. The equivalence classes of these orderings are called **orientations**. For simplices of dimension 0, there is only one orientation and for dimension greater than 0, there are 2 orientations. An **oriented simplex** is a simplex together with an orientation. For an oriented simplex σ , we denote by $-\sigma$ the oriented simplex that is the same simplex but with different orientation.

An oriented k -simplex $\sigma = [v_0, \dots, v_k]$ induces an orientation on its $(k - 1)$ -faces given by $(-1)^i[v_0, \dots, \hat{v}_i, \dots, v_k]$, where \hat{v}_i indicates that we are considering the face of σ obtained by excluding v_i . For example the oriented 2-simplex $\{1, 2, 3\}$ has the oriented faces $\{2, 3\}$, $\{3, 1\}$ and $\{1, 2\}$. We say that two k -simplices that share a $(k - 1)$ -face σ are **consistently oriented** if they induce different orientations on σ . We now note that each facet of a triangulation of a surface is of dimension 2 and we can therefore define the orientability of a triangulation in the below manner.

Definition 4.3. *A simplicial complex which is the triangulation of a surface is **orientable** if all its 2-simplices can be consistently oriented. Otherwise it is **nonorientable**.*

We wish to use the oriented simplices to define orientability of surfaces. We first require the following results.

Theorem 4.4. [36] *Every surface has a triangulation.*

Recall that we have assumed simplicial complexes to be finite, but for the above theorem we relax this restriction and allow a triangulation to be infinite. If we restrict our attention to the compact surfaces, the above theorem is valid for finite triangulations.

Theorem 4.5. [1, III.4.13] *If K_1 and K_2 are two triangulations of homeomorphic surfaces, then K_1 is orientable if and only if K_2 is orientable.*

We can therefore extend our definition of orientable triangulations of surfaces to one of orientable surfaces.

Definition 4.6. *A surface S is **orientable** if all the 2-simplices of a triangulation of S can be consistently oriented. Otherwise it is **nonorientable**.*

An additional benefit of the above definition is that it immediately gives us a simple algorithm for checking orientability. Given a triangulation, we just need to attempt to consistently orient the 2-simplices of the triangulation. If this is possible, the surface is orientable, otherwise it is nonorientable. Since for a given simplex we only have two choices of orientation, and this choice is fixed by an oriented neighbor, we can not fail to create a consistent orientation when this is possible.

4.3 Classification of Compact Surfaces

As mentioned earlier, we have the following classification theorem for the compact surfaces, where $\#$ indicates the **connected sum**, T^2 is the torus and $\mathbb{R}P^2$ is the real projective plane.

Theorem 4.7. [31] *Every compact surface is homeomorphic to precisely one of the following:*

- S^2
- $T^2 \# \dots \# T^2$
- $\mathbb{R}P^2 \# \dots \# \mathbb{R}P^2$

We have now developed the terminology to give a full characterization of the compact surfaces in terms of orientability and Euler characteristic.

Corollary 4.8. [31] *If a compact surface X is orientable it is homeomorphic to*

- S^2 if $\chi(X) = 2$
- $T^2 \# \dots \# T^2$, where there are g summands if $\chi(X) = 2 - 2g \neq 2$

If a compact surface X is not orientable, it is homeomorphic to

$$\mathbb{R}P^2 \# \dots \# \mathbb{R}P^2,$$

where there are g summands if $\chi(X) = 2 - g$.

This classification can be extended to a classification of surfaces with boundary [31] and we will in Section 7.5 see how this classification can be used to produce an optimal algorithm for the computation of discrete Morse matchings for compact surfaces.

5 Homology

Since the general homeomorphism and homotopy equivalence problems are undecidable, we will now look at homology, which is a coarser invariant, but readily computable. Recall that this coarseness means that while some spaces might have isomorphic homology groups but are not homotopy equivalent, homotopy equivalent spaces have isomorphic homology groups as we saw in Theorem 2.16. Note that simplicial homology is defined in terms of simplicial complexes, not topological spaces, but Theorem 5.5 states that when we can triangulate a topological space, the simplicial and singular homology groups are isomorphic.

We will then present an extension of homology known as persistent homology. From the point of view of topological data analysis, this is useful since it provides a method of dealing systematically with noise. Finally, we will briefly describe how homology and persistent homology are computed.

With the algorithms presented for computing homology, we will see that we have a readily computable topological invariant for triangulated topological spaces. Recalling the methods from Section 3.2 for constructing simplicial complexes approximating the topology from which a set of points has been sampled, we will have covered all the steps needed for the use of persistent homology in topological data analysis. The poor running times and high memory requirements of our methods will help motivate our interest in discrete Morse theory.

Note that although we only consider simplicial homology here, the main results of this section apply also to homology of CW complexes.

5.1 Simplicial Homology

Recall that an oriented simplex is a simplex together with an orientation, as defined in Section 4.2.

Definition 5.1. *Let K be a simplicial complex. We define the p 'th chain group of K , denoted $C_p(K)$, to be the free abelian group with basis the oriented p -simplices, where $\sigma_1 = -\sigma_2$ if σ_1 and σ_2 are the same simplex with different orientations.*

Each chain $c_p \in C_p(K)$ can be written uniquely as a finite sum

$$c_p = \sum n_i \sigma_i,$$

where $n_i \in \mathbb{Z}$ and σ_i is an oriented simplex for each i . Here we have defined the chain groups with integer coefficients, and would sometimes denote this $C_p(K; \mathbb{Z})$. It is also common in computational topology to restrict the coefficients n_i to \mathbb{Z}_2 and to instead work with $C_p(K; \mathbb{Z}_2)$.

Let $\sigma = [v_0, \dots, v_p]$ be an oriented simplex in K . Then for $p > 0$ we define the **boundary operator** on σ by

$$\delta_p(\sigma) = \delta_p[v_0, \dots, v_p] = \sum_{i=0}^p (-1)^i [v_0, \dots, \hat{v}_i, \dots, v_p]$$

Here \hat{v}_i indicates that we are considering the face of σ obtained by excluding v_i . We can now extend this boundary operator linearly to a homomorphism between chain groups, $\delta_p : C_p(K) \rightarrow C_{p-1}(K)$.

It is simple to verify that if σ and σ' are the same simplex, but their orientations differ by a single transposition, then $\delta_p(\sigma) = -\delta_p(\sigma')$, and we can therefore conclude that for σ and σ'' differing by an even permutation, $\delta_p(\sigma) = \delta_p(\sigma'')$. It follows that δ_p is well-defined.

The kernel of δ_p is a subgroup of $C_p(K)$ called the group of p -**cycles**, and denoted by $Z_p(K)$. The image of δ_{p+1} is a subgroup of $C_p(K)$ called the group of p -**boundaries** and denoted by $B_p(K)$. Note that $Z_p(K)$ and $B_p(K)$ are both normal subgroups of $C_p(K)$ since it is abelian.

Example 5.2. Consider now the 2-simplex $[1, 2, 3]$, with the orientation indicated by the vertex numbers, then

$$\begin{aligned} \delta_2([1, 2, 3]) &= (-1)^0 [2, 3] + (-1)^1 [1, 3] + (-1)^2 [1, 2] \\ &= [2, 3] - [1, 3] + [1, 2]. \end{aligned}$$

Thus the 1-chain $[2, 3] - [1, 3] + [1, 2]$ is a 1-boundary. Similarly,

$$\delta_1([2, 3] - [1, 3] + [1, 2]) = [2] - [3] - [1] + [3] + [1] - [2] = 0.$$

Thus $[2, 3] - [1, 3] + [1, 2]$ is also a 1-cycle. This example can be seen in Figure 11, it illustrates a broader result which will be proven in Lemma 5.3.

As we see in Figure 11, and as suggested by the names, we can informally regard the p -cycles as closed p -chains, and the p -boundaries as the borders of $(p + 1)$ -chains. We wish to consider the quotient group $Z_p(K)/B_p(K)$, consisting of non-bounding cycles, but in order for this to be well-defined, we must first show the following lemma.

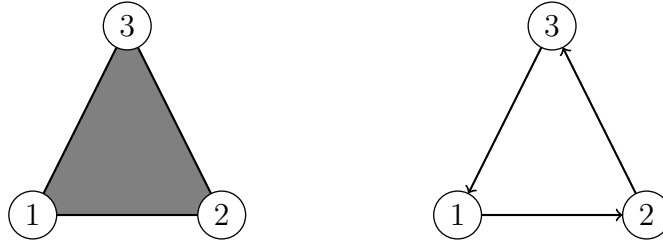


Figure 11: On the left, the 2-chain $[1, 2, 3]$, the orientation is counter clockwise in this example. On the right, the 1-chain $[2, 3] - [1, 3] + [1, 2]$, it is the image of $\delta_2([1, 2, 3])$. This is therefore a 1-boundary and 1-cycle.

Lemma 5.3. $B_p(K) \subset Z_p(K)$

Proof. It is sufficient to show that

$$\delta_p \circ \delta_{p+1}[v_0, \dots, v_{p+1}] = 0,$$

however this follows directly from the definition of δ_p as a homomorphism.

$$\begin{aligned} \delta_p \circ \delta_{p+1}[v_0, \dots, v_{p+1}] &= \sum_{i=0}^p (-1)^i \delta_p[v_0, \dots, \hat{v}_i, \dots, v_{p+1}] \\ &= \sum_{j < i} (-1)^i (-1)^j [\dots, \hat{v}_j, \dots, \hat{v}_i, \dots] \\ &\quad + \sum_{j > i} (-1)^i (-1)^{j-1} [\dots, \hat{v}_i, \dots, \hat{v}_j, \dots] \\ &= 0 \end{aligned}$$

The last equality follows from the fact that the sums consist of pairs with the same simplices, but different signs. \square

By Lemma 5.3, we can now define the p 'th **homology group** of K to be $H_p(K) = Z_p(K)/B_p(K)$. An immediate consequence of this definition is that homology groups are trivial for dimensions greater than the dimension of the simplicial complex.

We have thus defined the homology groups of a simplicial complex K . The p -chains together with their boundary operators form what is called a **chain**

complex, denoted by (C, δ) . This can be defined in any abelian category, and homology can be defined for any chain complex in the same way as the above. The definition used here is therefore a very specific example of the much broader concept of homology. In this report we will generally not concern ourselves with any other variants of homology, but simply note that homology with coefficients in \mathbb{Z}_2 is particularly important in computational topology since it can be faster to compute. The homology groups for coefficients in an abelian group such as \mathbb{Z}_2 are linked to integral homology by the Universal Coefficient Theorem. A useful property of homology is given in the following lemma.

Lemma 5.4. $H_p(K) = 0$ for p greater than the dimension of K .

This is in contrast to the so called **homotopy groups**, another important topological invariant, where the homotopy groups of even very simple spaces such as S^2 can be very complicated. For this reason, and the associated difficulty of computing the homotopy groups, they are not generally used in computational topology.

While simplicial homology can not be defined without reference to a simplicial complex, recall that singular homology is defined for any topological space X . It is a central theorem in algebraic topology that when simplicial homology is defined, it is equivalent to singular homology.

Theorem 5.5. [32] *Let K be a simplicial complex with underlying space $|K|$, then $H_n(|K|)$ and $H_n(K)$ are isomorphic, for $n \in \mathbb{N}$, where the former is the n 'th singular homology group of $|K|$ and the latter is the n 'th simplicial homology group of K .*

From now on we will therefore simply refer to homology, rather than singular or simplicial homology.

Recall that we have restricted our attention to finite simplicial complexes. The homology groups are therefore finitely generated in addition to being abelian. The fundamental theorem of finitely generated abelian groups now tells us how they can be written, and therefore provides an easy way of automatically checking if the homology groups of two spaces are isomorphic.

Theorem 5.6. [23] *Any finitely generated abelian group G is isomorphic to a group of the form:*

$$\mathbb{Z}^r \oplus \mathbb{Z}_{t_1} \oplus \dots \oplus \mathbb{Z}_{t_k}$$

where r is a nonnegative integer and t_i divides t_{i+1} for $i \in \{1, \dots, k-1\}$ for $k > 1$. The numbers r, t_1, \dots, t_k are uniquely determined by G .

The p 'th Betti number, b_p is the rank of the p 'th homology group, and is given by r in Theorem 5.6. The numbers t_i are called the **torsion coefficients**. As a result of the above characterization, we can represent a finitely generated abelian group just by a vector (b_p, t_1, \dots, t_k) , and two such groups are isomorphic if and only if the vectors representing them are the same.

In topological data analysis, there is a rough intuitive interpretation of the k 'th Betti number as representing the number of k -dimensional holes of a space. We can make this notion of a k -dimensional hole precise by defining it to be an equivalence class of non-bounding k -cycles. For an example illustrating this notion of a hole, the boundary of a 2-simplex has a 2-dimensional hole, whereas the 2-simplex itself does not, as the cycle is a boundary in this case. Since the 2-simplex is homeomorphic to a disk, and its boundary is homeomorphic to a circle, this reflects the hole contained in the circle but which is filled in for the disk.

Given that we have classified the compact surfaces in Theorem 4.7, and given that this classification can be extended to the surfaces with boundary, the following theorem can be obtained simply by computing the homology groups for the possible compact surfaces, with or without boundary, according to the classification. For our purposes, we require homology with coefficients in \mathbb{Z}_2 .

Theorem 5.7. [25] *For K a connected, compact surface with or without boundary,*

$$H_0(K; \mathbb{Z}_2) \cong \mathbb{Z}_2,$$

if K is with boundary

$$H_2(K; \mathbb{Z}_2) \cong 0,$$

otherwise,

$$H_2(K; \mathbb{Z}_2) \cong \mathbb{Z}_2.$$

The first homology groups aren't as simply characterized, but won't be needed in order to prove the optimality of the algorithm in Section 7.5.

5.2 Persistent Homology

Before introducing persistent homology, we will try to briefly motivate the concept. In Section 3.2 we introduced some methods for constructing simplicial complexes from point data. These complexes were all constructed with respect to some scale parameter, and we constructed a filtration to store the complexes across the entire range of possible values of the scale parameter. The problem with homology is that we compute it only for a single simplicial complex. The topological features of this simplicial complex may be due to noise or an inappropriately chosen scale parameter. As a simple example, if we let our scale parameter be close to 0, the Cech and Vietoris-Rips complexes will consist only of points, and b_0 will simply be the number of points. We will now see how persistent homology solves this problem. By allowing us to exclude short lived topological features, we can control how long a topological feature has to exist in the filtration before we consider it significant. Note that this implicitly assumes that this length of time, or persistence, of the feature reflects its importance.

Given a simplicial complex K , let $f : K \rightarrow \mathbb{R}$ be a non-decreasing function. Here non-decreasing means that if σ is a face of τ , $f(\sigma) \leq f(\tau)$. The **level subcomplexes** are then defined to be $K(a) = f^{-1}(-\infty, a]$. If we denote by a_i the values of f on the simplices of K in increasing order, the level subcomplexes define a filtration of K . Let $K^l = K(a_l)$, and L be the index of the largest value a_L , then we have a filtration

$$\emptyset \subset K^0 \subset K^1 \subset \dots \subset K^L = K,$$

This is another way of viewing our earlier construction of the Cech, alpha and Vietoris-Rips complexes. As the scale parameter increases, new simplices are added, until eventually we obtain the entire complex $K(\infty) = K$. For every $i \leq j$, we have an inclusion map from $|K^i|$ to $|K^j|$, and by Theorem 2.16 we therefore obtain an induced homomorphism $f_p^{i,j} : H_p(K^i) \rightarrow H_p(K^j)$ for every dimension p of K .

If we now consider the step from $H_p(K^i)$ to $H_p(K^{i+1})$ the following changes can occur: New homology classes can be created or already existing homology classes can merge or become trivial. The purpose of persistent homology is to keep track of those homology classes which are present throughout many steps in the filtration.

Definition 5.8. *The p -persistent k 'th homology group of K^l , denoted $H_k^{l,p}$ is*

$$H_k^{l,p} = Z_k^l / (B_k^{l+p} \cap Z_k^l).$$

Since B_k^{l+p} and Z_k^l are both subgroups of C_k^{l+p} , $B_k^{l+p} \cap Z_k^l$ is also a subgroup of Z_k^l and $H_k^{l,p}$ is therefore well-defined. Note that in the above the superscripts simply indicate the indices within the filtration, and do not denote cohomology. This definition reflects our desire to consider those non-bounding cycles which remain non-bounding for p steps in the filtration. An alternative definition of persistent homology is that $H_k^{l,p} = \text{im} f_k^{l,p}$. This definition is equivalent since the image of the induced homomorphism $f_k^{l,p}$ is isomorphic to $Z_k^l / (B_k^{l+p} \cap Z_k^l)$.

Let γ be a class in $H_p(K^i)$, we say that it is **born** at K^i if $\gamma \notin H_p^{i-1,i}$. If $f_p^{i,j-1}(\gamma) \notin H_p^{i-1,j-1}$ but $f_p^{i,j}(\gamma) \in H_p^{i-1,j}$ we say that γ **dies** entering K^j . This occurs when γ merges with an older class in the step from K^{j-1} to K^j . We define the **persistence** of γ to be $j - i - 1$.

Just like the Betti numbers and torsion coefficients fully characterized the homology of a simplicial complex, barcodes or persistence diagrams can be used to characterize the persistent homology of a filtration [5, 6]. A central result related to the applicability of persistent homology to data analysis is its stability with respect to perturbations, see [6] for details.

5.3 Computing Homology and Persistent Homology

We saw in Theorem 5.6 that we can represent the homology groups in terms of their Betti numbers and torsion coefficients. We will now see how we can determine the Betti numbers and torsion coefficients.

We first need to introduce matrix notation for homomorphisms of free abelian groups.

Definition 5.9. *Let G and G' be free abelian groups with bases a_1, \dots, a_n and a'_1, \dots, a'_m , respectively. If $f : G \rightarrow G'$ is a homomorphism, then*

$$f(a_j) = \sum_{i=1}^m \lambda_{ij} a'_i$$

for unique integers λ_{ij} . The matrix (λ_{ij}) is called the **matrix** of f relative to the given bases for G and G' .

Theorem 5.10. [32, Theorem 11.3] *Let G and G' be free abelian groups of ranks n and m , respectively and let $f : G \rightarrow G'$ be a homomorphism. Then*

there are bases for G and G' such that, relative to these bases, the matrix of f has the form

$$\left(\begin{array}{ccc|ccc} b_1 & & 0 & 0 & \cdots & 0 \\ & \ddots & & \ddots & \ddots & \vdots \\ 0 & & b_l & 0 & \cdots & 0 \\ \hline 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{array} \right)$$

where $b_i \geq 1$ and $b_1|b_2|\dots|b_l$ and all entries except the diagonal from b_1 to b_l are 0.

This is called the **Smith normal form** for the matrix of f and in order to compute homology, it suffices to compute the Smith normal form of matrices representing the boundary homomorphisms.

Theorem 5.11. *Given the boundary homomorphisms $\delta_p : C_p(K) \rightarrow C_{p-1}(K)$ and $\delta_{p+1} : C_{p+1} \rightarrow C_p$ then the non-zero entries of the Smith normal form of δ_{p+1} are the torsion coefficients of K in dimension p . Let z_p be the number of zero columns of the Smith normal form of δ_p and w_p the number of non-zero rows of the Smith normal form of δ_{p+1} . Then we have $b_p = z_p - w_p$, where b_p is the p 'th betti number of K .*

The computation of persistent homology can be reduced to the computation of homology and has the same worst case running time [42]. This is $O(n^\omega)$ for field coefficients, where ω is the exponent for matrix multiplication [28]. The best current estimate for ω is 2.3727 [37] and the best performing algorithm in practice is Strassen's algorithm with an exponent of 2.807. For coefficients in a principal ideal domain, such as the integers, the best currently known running time is $O(n^3)$ [28].

Thus, we see that using field coefficients such as \mathbb{Z}_2 gives a computational advantage, but even in this case the computation of homology and persistent homology is prohibitively expensive for large problem instances. There are special cases for which we can compute homology faster, for example for clique complexes [40] or for subcomplexes of triangulations of S^2 [8], but in general we have to accept the above slow running times.

The problem of slow algorithms, along with the memory requirements of storing simplicial complexes and their boundary matrices, will motivate our interest in discrete Morse theory.

6 Discrete Morse Theory

We will now give an introduction to the relevant aspects of discrete Morse theory. Discrete Morse theory was developed by Forman [17] as an adaptation of Morse theory to finite CW complexes. Morse theory will not be covered in this report, but the standard reference book is [27]. In addition, we will restrict our attention to finite simplicial complexes, rather than finite CW-complexes, to simplify the presentation.

Our first aim in this section is to state and prove Theorem 6.5, the main theorem of discrete Morse theory. We will then show that we can simplify the task of computing discrete Morse functions, by considering instead so called discrete vector fields and partial matchings in certain modified Hasse diagrams of simplicial complexes. Finally, we show how the homology of a simplicial complex and the persistent homology of a filtration can be computed via discrete Morse theory.

6.1 Basic Concepts of Discrete Morse Theory

The two building blocks of discrete Morse theory are the discrete Morse function and its critical points.

Definition 6.1. *Let K be a simplicial complex. A function $f : K \rightarrow \mathbb{R}$ is a **discrete Morse function** if for every $\alpha^{(p)} \in K$*

$$(i) |\{\beta^{(p+1)} > \alpha | f(\beta) \leq f(\alpha)\}| \leq 1$$

$$(ii) |\{\gamma^{(p-1)} < \alpha | f(\gamma) \geq f(\alpha)\}| \leq 1$$

A discrete Morse function therefore assigns higher numbers to higher dimensional simplices, with at most one immediate face and one immediate coface violating this condition for any one simplex. Lemma 6.4 will slightly restrict this definition by showing that both types of violations can not occur simultaneously for a given simplex.

We will use the terms discrete Morse and Morse interchangeably and when the Morse function being referred to is unambiguous, we will say that it assigns a value c to a given simplex, that simplex has **value** c .

For any simplicial complex, there is always the trivial Morse function, which assigns to each simplex its dimension. Some examples of Morse functions on a triangle are given in Figure 12.

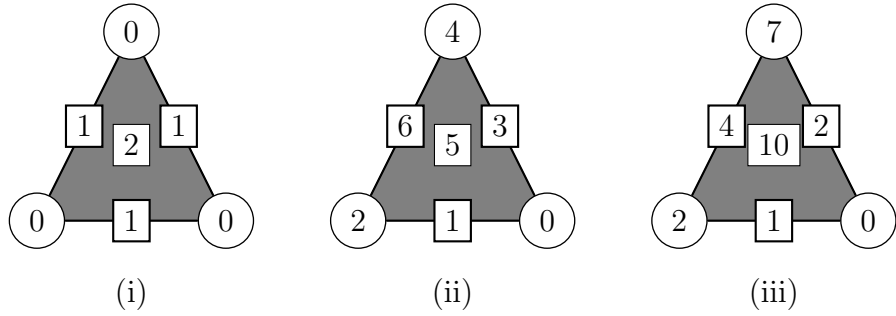


Figure 12: (i) is the trivial Morse function on the triangle, (ii) is a valid Morse function, and (iii) is not a Morse function, since the vertex with value 7 has two incident edges with smaller value.

Definition 6.2. A simplex $\alpha^{(p)}$ is **critical** if

$$(i) |\{\beta^{(p+1)} > \alpha | f(\beta) \leq f(\alpha)\}| = 0$$

$$(ii) |\{\gamma^{(p-1)} < \alpha | f(\gamma) \geq f(\alpha)\}| = 0$$

Thus a simplex is critical if none of its immediate faces are assigned a greater or equal value, and none of its immediate cofaces are assigned a lesser or equal value. Simplices that are not critical are called **regular**. We consider a Morse function which minimizes the number of critical simplices **optimal**. The motivation for this definition of optimality will follow from Theorem 6.5. See Figure 13 for examples illustrating the critical simplices of a triangle given different Morse functions.

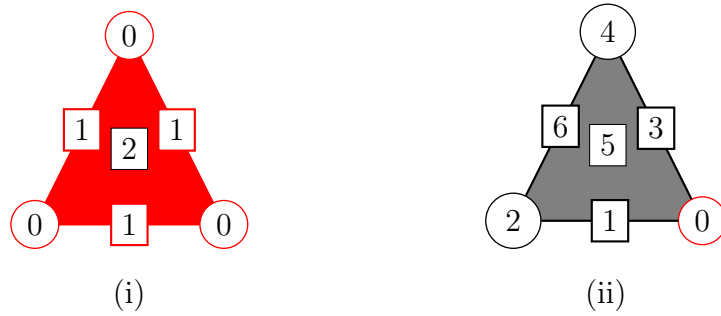


Figure 13: Critical simplices are red. The trivial Morse function, (i), has only critical simplices and (ii) is an optimal Morse function, with only one critical simplex.

Example 6.3. *Figure 13(ii) is an illustration of the more general fact that for an n -simplex, an optimal Morse function has exactly 1 critical simplex, a vertex, and that its boundary always has an optimal Morse function with exactly 2 critical simplices.*

Lemma 6.4. *[18] If K is a simplicial complex with a Morse function f , then for any simplex $\alpha^{(p)}$, at least one of the following will hold*

$$(i) |\{\beta^{(p+1)} > \alpha | f(\beta) \leq f(\alpha)\}| = 0$$

$$(ii) |\{\gamma^{(p-1)} < \alpha | f(\gamma) \geq f(\alpha)\}| = 0$$

Proof. Suppose that

$$|\{\beta^{(p+1)} > \alpha | f(\beta) \leq f(\alpha)\}| = 1$$

and

$$|\{\gamma^{(p-1)} < \alpha | f(\gamma) \geq f(\alpha)\}| = 1.$$

Then there exists a coface β of α such that $f(\beta) \leq f(\alpha)$ and a face γ of α such that $f(\gamma) \geq f(\alpha)$.

Consider now a different face α' of β , that has γ as a face. By the definition of a Morse function, $f(\beta) > f(\alpha') > f(\gamma)$.

Combining these inequalities, we obtain

$$f(\alpha) \leq f(\gamma) < f(\alpha') < f(\beta) \leq f(\alpha),$$

which is a contradiction. □

We now have the necessary definitions to state the main theorem of discrete Morse Theory.

Theorem 6.5. *[18] Suppose K is a simplicial complex with a Morse function f . Then K is homotopy equivalent to a CW complex with exactly one cell of dimension p for each critical simplex of dimension p .*

This result will later be strengthened from just a proof of existence when we introduce the **Morse complex** in Section 6.7. This again underlines the central importance of critical simplices in discrete Morse Theory, since fewer critical cells mean that we have proven homotopy equivalence to a smaller complex.

The proof of Theorem 6.5 requires some additional concepts which will be sketched out below.

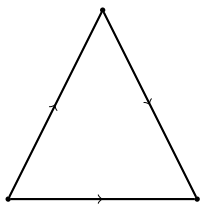


Figure 14: Dunce Hat

6.2 Simple Homotopy Equivalence

Suppose that $K_2 \subset K_1$ are simplicial complexes, and K_1 has exactly two simplices α and β that are not in K_2 , and where β is a free face of α . Then $|K_2|$ is a deformation retract of $|K_1|$ and hence $|K_1|$ and $|K_2|$ are homotopy equivalent [24]. This deformation retract is called a **simplicial collapse**, or just **collapse**. In this case we write $K_1 \searrow K_2$. Similarly we write $K_2 \nearrow K_1$ and say that K_1 is obtained from K_2 via a **simplicial expansion**, or just **expansion**.

Definition 6.6. *If a simplicial complex K_2 can be obtained from K_1 by a sequence of simplicial collapses or expansions, we say that they are **simple homotopy equivalent**.*

It is clear that simple homotopy equivalence is an equivalence relation and from the above paragraph that simple homotopy equivalence implies homotopy equivalence.

We say that a space which has a triangulation which is simple homotopy equivalent to a point is **collapsible**.

Example 6.7. *The Dunce hat is an example of a topological space which is contractible, but not collapsible. The Dunce hat is the space obtained by identifying all three sides of a triangle together as indicated in Figure 14, and it is not collapsible since no triangulation of the Dunce hat will have any free faces of a 2-simplex.*

We have introduced simple homotopy equivalence since it is needed for the proof of Theorem 6.5. This proof will show that simple homotopy equivalence is at the core of discrete Morse theory.

6.3 Proof of the Main Theorem of Discrete Morse Theory

Recall that in Section 5.2 we introduced level subcomplexes for non-decreasing functions. We now introduce a slightly modified definition for level subcomplexes, applicable to Morse functions.

Definition 6.8. *Let $c \in \mathbb{R}$, then the **level subcomplex**, $K(c)$ of K is the subcomplex consisting of all simplices α of K such that $f(\alpha) \leq c$, and their faces.*

This definition is illustrated in Figure 15.

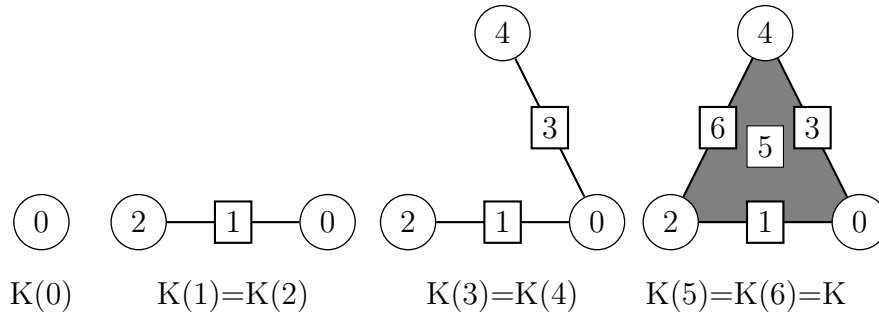


Figure 15: The level subcomplexes for a discrete Morse function

Theorem 6.5 will follow from the two below lemmas involving level subcomplexes. From now on to simplify the notation, we will not distinguish between a simplicial complex K and its underlying space $|K|$.

Lemma 6.9. [18] *If there are no critical simplices α with $f(\alpha) \in (a, b]$, then $K(b)$ is homotopy equivalent to $K(a)$.*

Lemma 6.10. [18] *If there is a single critical simplex α with $f(\alpha) \in (a, b]$ then there is a map $F : S^{(d-1)} \rightarrow K(a)$, where d is the dimension of α , such that $K(b)$ is homotopy equivalent to $K(a) \cup_F B^d$. In the above, $S^{(d-1)}$ is the $(d-1)$ -sphere and B^d is the d -ball.*

To understand why Lemma 6.9 holds, recall that a simplex has a free face if it has a face which is not the face of any other simplex.

When we move from $K(a)$ to $K(b)$, we add regular simplices in pairs consisting of a simplex and a free face thereof. Suppose now that $K_2 \subset K_1$

are simplicial complexes, and K_1 has exactly two simplices α and β that are not in K_2 , where β is a free face of α . This is precisely the definition of a simplicial expansion, so K_1 is obtained from K_2 by a simplicial expansion, and hence K_1 and K_2 are simple homotopy equivalent, and therefore homotopy equivalent.

To understand why Lemma 6.10 holds, we note that when we add a critical d -simplex α , it follows from our definition of a critical simplex that the faces of α have smaller discrete Morse values, and therefore that they all appear in an earlier level subcomplex. Therefore when we add α , we are attaching it along its entire boundary, corresponding to attaching the d -cell as in the statement of Lemma 6.10.

Finally, we can combine Lemma 6.9 and Lemma 6.10, to prove Theorem 6.5.

Proof of Theorem 6.5. For a simplicial complex K , and given a Morse function f with maximal value c and minimal value a , we have $K = K(c)$. To obtain $K(c)$, we can simply start from the minimal level subcomplex $K(a)$ and increase from a to c . As we pass regular simplices, Lemma 6.9 ensures that the homotopy type of the complex does not change, and when we pass a critical simplex of dimension p , Lemma 6.9 tells us that we need to attach a cell of dimension p .

Note that there can be a slight issue if f is not injective, but it is simple to perturb f , without changing which simplices are critical, to produce an injective Morse function. \square

6.4 Morse Inequalities

The Morse inequalities allow us to use discrete Morse theory to provide bounds on the Betti numbers of K . Recall firstly that the Euler characteristic was defined to be $\chi(K) = \sum_{i=0}^n (-1)^i b_i$ where b_p is the p 'th Betti number of K . Denote by m_p the number of critical simplices of dimension p for a simplicial complex K with Morse function f , then we have the following weak Morse inequalities.

Theorem 6.11. [18] For $p \geq 0$,

- $m_p \geq b_p$
- $\chi(K) = \sum_{i=0}^n (-1)^i m_i$.

We can strengthen the weak Morse inequalities to obtain the following strong Morse inequalities.

Theorem 6.12. [18] For $p \geq 0$,

- $m_p - m_{p-1} + \dots + (-1)^p m_0 \geq b_p - b_{p-1} + \dots + (-1)^p b_0$

If we minimize the number of critical simplices, we maximize the information provided by the Morse inequalities about the Betti numbers. This information is of general interest, but is particularly important in the context of topological data analysis, since the Betti numbers are a commonly used topological invariant. We can therefore obtain useful information about the topology of a simplicial complex, just by computing a Morse function on it. Note that the Morse inequalities for the trivial Morse function also imply that we can compute the Euler characteristic of a space easily given its triangulation.

An important notion in discrete Morse theory is that of a **perfect Morse function**, a Morse function such that the number of critical p -simplices is equal to the p 'th Betti number of the complex. Perfect Morse functions therefore give us the exact Betti numbers, but unfortunately Example 6.13 below illustrates that it is not always possible to obtain perfect Morse functions.

Example 6.13. *Recall from Example 6.7 that the Dunce hat is contractible, but not collapsible. Contractibility implies that a perfect Morse function would have 1 critical simplex of dimension 0, and no other critical simplices, however since it is not collapsible, no triangulation of the Dunce hat admits a Morse function with just 1 critical simplex of dimension 0, and we can therefore not obtain a perfect Morse function on a triangulation of the Dunce hat.*

For the same reason, we have the more general result that perfect Morse functions can not be obtained for any topological space which is contractible, but not collapsible.

6.5 Gradient Vector Fields

We see from the above Morse inequalities and the main theorem that the most important aspect of discrete Morse Theory is not really the Morse function, but rather the associated critical simplices. The Morse functions defined earlier can be difficult to compute, and in many cases, it is sufficient

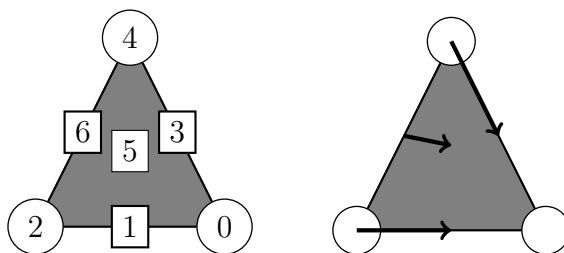


Figure 16: A discrete Morse function on the left with corresponding gradient vector field on the right.

to simply consider what is known as the gradient vector field of the Morse function.

We begin by noting that every regular simplex is paired with another regular simplex. For example, if $\alpha^{(p)}$ is a regular simplex with $\alpha^{(p)} < \beta^{(p+1)}$ satisfying $f(\beta) \leq f(\alpha)$, α and β are paired. We can represent these pairs with arrows. For a simplicial complex K with a discrete Morse function, Lemma 6.4 implies that every simplex is either the head of exactly one arrow, the tail of exactly one arrow, or neither the head or tail of an arrow. We call the diagram defined by the arrows the **gradient vector field** of the Morse function on the complex.

The arrows function like the level subcomplexes from earlier, indicating how to build up the entire simplicial complex from the empty complex. We either carry out a simplicial expansion where indicated by an arrow, or attach the critical simplices.

We wish to be able to find gradient vector fields without explicitly computing their corresponding Morse functions. We will therefore characterize the gradient vector fields as a particular type of discrete vector field, defined below.

Definition 6.14. A *discrete vector field* V on K is a collection of pairs $\{\alpha^{(p)} < \beta^{(p+1)}\}$ of simplices of K such that each simplex is in at most one pair of V .

Given a discrete vector field V on a simplicial complex K , a **V -path** is a sequence of simplices

$$\alpha_0^{(p)}, \beta_0^{(p+1)}, \alpha_1^{(p)}, \beta_1^{(p+1)}, \dots, \alpha_r^{(p)}, \beta_{r+1}^{(p+1)} \quad (1)$$

such that for each $i = 0, \dots, r$, $\{\alpha < \beta\} \in V$ and $\alpha_i \neq \alpha_{i+1} < \beta_i$. A V -path is **non-trivial closed** if $r \geq 0$ and $\alpha_0 = \alpha_{r+1}$. The V -paths of the gradient

vector field a Morse function f are also referred to as the **gradient paths** of f .

Theorem 6.15. [18] *Suppose V is the gradient vector field of a Morse function f . Then a sequence of simplices as in (1) is a gradient path if and only if $\alpha_i < \beta_i > \alpha_{i+1}$ for each $i = 0, \dots, r$ and*

$$f(\alpha_0) \geq f(\beta_0) > f(\alpha_1) \geq f(\beta_1) > \dots \geq f(\beta_r) > f(\alpha_{r+1}).$$

Intuitively, the gradient paths of f are precisely those connected sequences of simplices along which f is decreasing. We can now fully characterize the gradient vector fields of a Morse function by the following theorem.

Theorem 6.16. [18] *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.*

6.6 Morse Matchings

The following alternative characterization of gradient vector fields uses a modified version of the Hasse diagram of a simplicial complex K . We begin by defining a **matching** M of a graph to be a set of pairs of vertices sharing an edge, such that no vertex is contained in more than one pair. We define the **Hasse diagram** of K to be a directed graph, with vertices given by the simplices of K , and a directed edge from β to α if and only if α is an immediate face of β . This is also called the **face poset** of K . If we now let M be a matching of the Hasse diagram of K , and $\{\alpha < \beta\} \in M$, we modify the Hasse diagram by reversing the orientation of the edge between α and β so that it goes from α to β . The resulting graph is called the **modified Hasse diagram** of K with M and denoted $H(M)$.

We say that a matching M is **acyclic** if the corresponding modified Hasse diagram $H(M)$ does not contain any directed cycles.

With these definitions, we can therefore see that a discrete vector field V on K simply corresponds to a matching of the Hasse diagram. An example is illustrated in Figure 17.

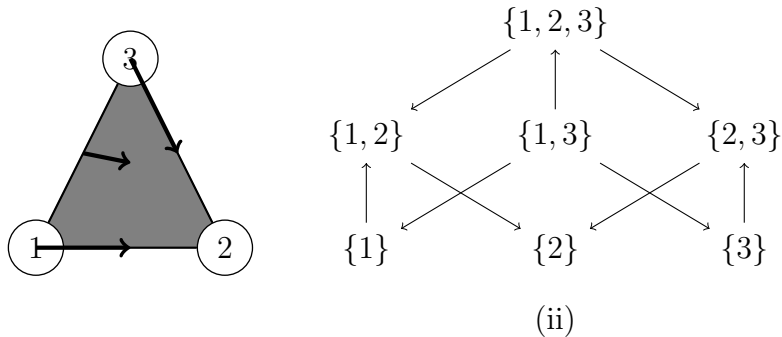


Figure 17: (i) The gradient vector field corresponding to the Morse function given in Figure 16 with vertices labelled and (ii) the corresponding modified Hasse diagram.

A V -path defines a particular directed path in this modified Hasse diagram. With this definition, we can characterize the gradient vector fields in terms of the modified Hasse diagram, since we have the following theorem.

Theorem 6.17. [18] *Let V be a discrete vector field. There are no nontrivial closed V -paths if and only if there are no nontrivial closed directed paths in the corresponding modified Hasse diagram.*

Thus as already noted, a discrete vector field corresponds to a matching of simplices in the Hasse diagram. This discrete vector field is a gradient vector field of a Morse function if and only if the matching is acyclic, that is, if the modified Hasse diagram contains no directed cycles.

The aim of finding an optimal Morse function for a simplicial complex K can be restated in this language as the attempt to find a maximal acyclic matching of the simplices of K . This formulation is most commonly used for attempting to design optimal Morse functions, and will also be the main focus in the section on computing discrete Morse matchings.

Definition 6.18. *A **Morse matching** M of a simplicial complex K is a matching of the simplices of K such that the corresponding modified Hasse diagram, $H(M)$, is acyclic. We will denote the matching as $M = (\mathcal{A}, w : \mathcal{Q} \rightarrow \mathcal{K})$, where \mathcal{A} , \mathcal{Q} and \mathcal{K} partition the simplices of K and $w : \mathcal{Q} \rightarrow \mathcal{K}$ is a bijection. \mathcal{A} denotes the critical simplices, \mathcal{Q} denotes simplices matched with a higher dimensional simplex and \mathcal{K} denotes simplices matched with a lower dimensional simplex. Finally, $w : \mathcal{Q} \rightarrow \mathcal{K}$ indicates how simplices from \mathcal{Q} are matched with simplices from \mathcal{K} .*

Before proceeding, we state the following lemma regarding the type of cycle that can prevent a matching from being a valid Morse matching. This will be useful when we prove that computing optimal Morse matchings is *NP*-hard.

Lemma 6.19. *Any cycle in the modified Hasse diagram corresponding to a matching will occur only between i and $i + 1$ dimensional simplices.*

Proof. This follows immediately from the fact that a given simplex can only be matched with at most one other simplex. There is therefore no way of moving up two dimensions in the modified Hasse diagram. \square

If we consider the empty simplex as an immediate face of each vertex, we have the following theorem. Note that allowing the empty simplex to be matched corresponds to considering so called reduced homology.

Theorem 6.20. [18] *Let M be a complete Morse matching of the Hasse diagram of K , then K is collapsible.*

6.7 Discrete Morse Complexes

We will now see how we can use discrete Morse Theory to calculate the homology of a simplicial complex.

Let K be a simplicial complex with Morse function f . Let $C_p(K)$ denote the p 'th chain group, and $\mathcal{M}_p \subset C_p(K)$ the span of the critical p -simplices. Then we can obtain the following theorem.

Theorem 6.21. [18] *There are boundary maps $\tilde{\delta}_d : \mathcal{M}_d \rightarrow \mathcal{M}_{d-1}$, for each d such that*

$$\tilde{\delta}_{d-1} \circ \tilde{\delta}_d = 0$$

and such that the resulting chain complex

$$0 \rightarrow \mathcal{M}_n \xrightarrow{\tilde{\delta}_n} \mathcal{M}_{n-1} \rightarrow \dots \xrightarrow{\tilde{\delta}_1} \mathcal{M}_0 \rightarrow 0 \quad (2)$$

calculates the homology of K . That is, if we define

$$H_d(\mathcal{M}, \tilde{\delta}) = \frac{\text{Ker}(\tilde{\delta}_d)}{\text{Im}(\tilde{\delta}_{d+1})}$$

then for each d

$$H_d(\mathcal{M}, \tilde{\delta}) \cong H_d(K).$$

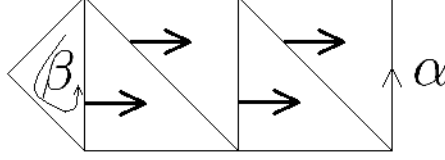


Figure 18: β and α are critical simplices. The gradient path from the edge of β to α induces the indicated orientation on α . The edge shared by β and the neighboring triangle inherits its orientation from β . This in turns induces the clockwise orientation on the neighboring triangle. This process continued along the gradient path gives α its orientation.

Once we have explicitly defined the boundary operators, this Morse complex provides an explicit complex, whereas the main Theorem of discrete Morse theory only ensured existence. The boundary operators of the Morse complex can be explicitly computed from the gradient paths of the corresponding Morse function.

We first note that given a gradient path from a face of a simplex β to a simplex α , the orientation of β induces an orientation on α by defining an orientation on each neighbor along the gradient path. See Figure 18 for an example of this induced orientation.

We can now state the formula for the boundary operators.

Theorem 6.22. [18] *Choose an orientation for each simplex of K . Then for any critical $(p + 1)$ -simplex β set*

$$\tilde{\delta}\beta = \sum_{\text{critical } \alpha^{(p)}} c_{\alpha,\beta} \alpha \quad (3)$$

where

$$c_{\alpha,\beta} = \sum_{\gamma \in \Gamma(\beta,\alpha)} m(\gamma)$$

where $\Gamma(\beta, \alpha)$ is the set of gradient paths which go from an immediate face of β to α . The multiplicity $m(\gamma)$ of any gradient path γ is equal to ± 1 , depending on whether given γ , the orientation on β induces the chosen orientation on

α , or the opposite orientation. With this differential, the complex in Theorem 6.21 computes the homology of K .

Example 6.23. To give a brief example of how to explicitly compute boundaries using Theorem 6.22, consider Figure 18. We first note that there is exactly one unique gradient path γ from an edge of β to α . Suppose that the orientation of the simplex α is upwards, then since the induced orientation is the same, $m(\gamma) = 1$, and $c_{\alpha,\beta} = 1$. Suppose that the orientation of α is downwards, then we have $m(\gamma) = -1$, and $c_{\alpha,\beta} = -1$.

Discrete Morse theory therefore provides us a tool to efficiently compute homology. If we are given a simplicial complex, we first try to find a good Morse function, and then calculate homology for the reduced Morse complex derived from this. Here, a good Morse function is as stated earlier one with as few critical simplices as possible, as these define the size of the reduced complex. Furthermore, since we only need the gradient paths, it is sufficient to compute a suitable Morse matching, rather than an explicit Morse function. For large simplicial complexes discrete Morse theory therefore has the potential to greatly speed up the computation of homology, provided good Morse functions or matchings can be found efficiently.

6.7.1 Example

We will here give an example to underline the importance of constructing good discrete Morse matchings and to give a simple example of a Morse complex. The triangulation **bing**, taken from the Lutz and Benedetti library of triangulations [4], is a triangulation of the 3-ball. The three heuristic strategies, along with experimental comparisons, will be explained in Section 7.

face vector	naive heuristic	collapse heuristic	Simulated annealing
(480, 2511, 3586, 1554)	605	3	1

From the above table we see that we started out with a triangulation with 8131 simplices. Even using naive heuristics we obtain only 605 critical simplices. For the collapse heuristic however, we obtain only 3 critical simplices, while we obtain only 1 critical simplex using simulated annealing. As we have seen above, the number of critical simplices determines the size of the

resulting Morse complex. Given that our homology computation algorithms have roughly cubic running times in the size of the input complex, the benefits of preprocessing using discrete Morse theory are therefore huge. We also see that the difference between the naive heuristics and the collapse heuristic are dramatic, and that the strategy used for computing Morse matchings is therefore of central importance in making discrete Morse theory useful for the computation of homology. Finally, we note that for this example, we were able to obtain a Morse matching with only one critical simplex. This corresponds to a Morse complex consisting of only one cell, and we therefore obtain the homology groups of the triangulation immediately. Denoting the simplicial complex *bing* by K , we obtain

$$H_0(K) = \mathbb{Z}$$

and

$$H_n(K) = 0$$

for $n \neq 0$.

6.8 Discrete Morse Theory and Persistence

With Theorem 6.21 we have established that discrete Morse theory can be used to preprocess homology computation by reducing the size of the complex. We wish to extend this theory so that we can preprocess the computation of persistent homology too. Since we have so far only developed discrete Morse theory for simplicial complexes, rather than filtrations of simplicial complexes, we might worry that in simplifying the simplicial complexes, even though we preserve the homology of the individual complexes, we may not preserve their persistent homology. Luckily the extension of a Morse matching of a simplicial complex to one on a filtration which preserves persistent homology is relatively simple.

Recall that a Morse matching on a simplicial complex K is an acyclic matching of the modified Hasse diagram of K .

Definition 6.24. *A **filtered Morse matching** on a filtration*

$$\emptyset \subset K^0 \subset \dots \subset K^N = K$$

consists of a Morse matching $M^n = (\mathcal{A}^n, w^n : \mathcal{Q}^n \rightarrow \mathcal{K}^n)$ on each K^n with the following additional requirements

- $\mathcal{A}^n \subset \mathcal{A}^{n+1}, \mathcal{Q}^n \subset \mathcal{Q}^{n+1}, \mathcal{K}^n \subset \mathcal{K}^{n+1}$
- $w^n = w|_{\mathcal{Q}^n}^{n+1}$

We denote the final Morse matching

$$(\mathcal{A}^N, w^N : \mathcal{Q}^N \rightarrow \mathcal{K}^N)$$

by

$$(\mathcal{A}, w : \mathcal{Q} \rightarrow \mathcal{K})$$

and let $\mathcal{M}, \tilde{\delta}$ be the corresponding Morse complex. Mischaikow and Nanda proved the two following results that ensure that we can use a filtered Morse matching to compute persistent homology [29].

Proposition 6.25. [29] *The critical simplices of a filtered Morse matching define a filtration of the Morse complex $\mathcal{M}, \tilde{\delta}$.*

$$\emptyset \subset \mathcal{A}^0 \subset \dots \subset \mathcal{A}^N = \mathcal{A} = \mathcal{M}$$

Theorem 6.26. [29] *Let*

$$\emptyset \subset K^0 \subset K^1 \subset \dots \subset K^N = K,$$

be a filtration of a simplicial complex K with filtered Morse matching

$$(\mathcal{A}^n, w^n : \mathcal{Q}^n \rightarrow \mathcal{K}^n),$$

and let $\mathcal{M}, \tilde{\delta}$ be the corresponding Morse complex with filtration

$$\emptyset \subset \mathcal{A}^0 \subset \dots \subset \mathcal{A}^N = \mathcal{A} = \mathcal{M}$$

then for all k, l and p we have

$$H_k^{l,p}(K^l) \cong H_k^{l,p}(A^l)$$

Our definition of a filtered Morse matching requires that if we match two simplices α and $w(\alpha)$, they must both appear in the filtration for the first time in the same complex K^n . In light of this, we can see that we can compute a filtered Morse matching by computing a Morse matching on K^N with the additional requirement that we don't match simplices whose birth times in the filtration differ, and then restrict the Morse matching to each complex in the filtration. This concludes our section on discrete Morse theory, we have seen that discrete Morse theory can be used to reduce the size of complexes while preserving homology and persistent homology. We will now look at how to actually compute the necessary Morse matchings.

7 Computing Morse Matchings

We have seen in the previous sections why computing homology is interesting, why we would like to be able to do it faster and the theory of how discrete Morse theory can be used to preprocess homology and persistent homology computation. With respect to persistent homology, recall that for a simplicial complex with a filtration, a Morse matching which respects the birth times of the simplices in the filtration, can be restricted on the subcomplexes of the filtration to obtain a filtered Morse matching. For both homology and persistent homology, we therefore regard our primary goal simply to compute good Morse matchings on simplicial complexes. This will then determine how effectively discrete Morse theory can be applied to preprocessing homology and persistent homology computations.

We start by showing that the problem of finding optimal Morse matchings is *NP*-hard in Section 7.2. We then provide an integer programming formulation for the problem, and describe a branch and cut algorithm for solving it in Section 7.3. Since this cannot solve large instances of the problem we turn to heuristic solutions and optimal algorithms for special cases. We begin by introducing some naive heuristic approaches in Section 7.4, before providing an optimal algorithm for the special case where the simplicial complex is the triangulation of a compact surface in Section 7.5. In Sections 7.6 and 7.7 we introduce two state of the art heuristics.

Section 7.9 provides a new simulated annealing based approach which outperforms the two state of the art heuristics mentioned earlier for many problem instances.

Finally, experimental results comparing the performances of some of the heuristics are provided in Section 7.10. Heuristics are usually compared experimentally in terms of their running times and the qualities of their solutions [35]. Since the homology computations with currently available hardware and algorithms are often intractable for sufficiently large complexes, our emphasis will be mainly on the quality of solution obtained.

7.1 Optimal Morse Matchings

Recall that a Morse matching of a simplicial complex K is a matching of the simplices of K such that the corresponding modified Hasse diagram is acyclic. An optimal Morse matching is then a Morse matching with the minimum number of critical simplices. As we have seen in the preceding

sections, this notion of optimality is justified by the fact that the number of critical simplices determines the size of the Morse complex and the tightness of the Morse inequalities.

Before proving the hardness of finding optimal Morse matchings, we will introduce an optimal algorithm for a special case that will be needed in the proof.

Let $c(M)$ be the number of critical simplices in a simplicial complex with Morse matching M , and $c_i(M)$ the number of critical simplices of dimension i .

Theorem 7.1. *We can compute an optimal Morse matching M for a connected simplicial complex S of dimension 1 in time linear in the size of the complex. For this matching, $c_0(M) = 1$.*

Proof. A simplicial complex of dimension 1 is simply a graph. We can therefore find a spanning tree of S in linear time using, for example, depth first search. Direct all the edges away from some vertex v of the spanning tree. The resulting directed subgraph on the spanning tree is clearly acyclic and the Morse matching obtained by matching each edge of the spanning tree with the vertex it points towards is optimal.

To see that it is optimal, we note that there is only one critical vertex, the vertex v , and since the weak Morse inequalities provide the bound $b_0 \leq c_0(M)$, the optimality follows from the fact that $b_0 = 1$.

Alternatively we can show that the matching is optimal by noting that if there are no critical 0-simplices the modified Hasse diagram must contain a cycle, since otherwise it would contain an infinite path. \square

Since we can process each connected component of S separately, we immediately obtain the following corollary.

Corollary 7.2. *We can compute an optimal Morse matching for a simplicial complex S of dimension 1 in time linear in the size of the complex.*

7.2 Hardness of Optimal Morse Matchings

We wish to show that for a simplicial complex S and a nonnegative integer k , it is NP -complete to decide whether there exists a Morse matching with at most k critical simplices. The standard way of showing that a problem is NP -complete is to show that it is in NP , and then to demonstrate a polynomial-time reduction from an NP -complete problem to it.

Egecioglu and Gonzalez proved that the following collapsibility problem is NP -complete [15] and this will be the problem we use for our reduction. This approach is due to Joswig and Pfetsch [22].

Theorem 7.3. [15] *Given a connected, pure 2-dimensional simplicial complex S which is embeddable in \mathbb{R}^3 , and a nonnegative integer k , it is NP -complete to decide whether there exists a subset K of the facets of S with $|K| \leq k$, such that there exists a sequence of simplicial collapses which transforms $S \setminus K$ to a 1-dimensional complex.*

Recalling Lemmas 6.4 and 6.4, the collapsibility problem can be restated in the language of discrete Morse theory.

Corollary 7.4. [15] *Given a connected pure 2-dimensional simplicial complex S which is embeddable in \mathbb{R}^3 , and a nonnegative integer k , it is NP -complete to decide whether there exists a Morse matching with at most k critical 2-faces.*

We wish to obtain a polynomial time computable transformation between the problem of deciding if there exists a Morse matching with at most k critical 2-simplices and the problem of deciding if there exist a Morse matching with at most c critical simplices in total. This transformation must satisfy that there exists a Morse matching with at most k critical 2-simplices if and only if there exists a Morse matching with at most c critical simplices in total.

Before proceeding, we need the following construction. Given a Morse matching M on a simplicial complex S , such that the dimension of S is greater than or equal to 1, let $\Gamma(M)$ be the graph obtained from the graph of S by removing all 1-simplices matched with 2-simplices.

To illustrate this definition, we return to our standard example. Consider the triangle with an optimal Morse matching, illustrated in Figure 16, $\Gamma(M)$ consists of two edges connecting the three vertices.

Lemma 7.5. [22] *Given a connected simplicial complex S and some Morse matching M , the graph $\Gamma(M)$ is connected and contains all the vertices of S .*

Proof. The fact that $\Gamma(M)$ contains all the vertices of S is clear since it was obtained by removing only edges from the graph of S .

In order to see that $\Gamma(M)$ is connected, we first note that if the dimension of S is smaller than 2, there are no 2-simplices. Hence $\Gamma(M)$ is just the graph of S , and is therefore connected by assumption.

Suppose now that the dimension of S is greater than or equal to 2 and that $\Gamma(M)$ is disconnected. Then we let N be the set of vertices in one of the connected components of $\Gamma(M)$ and let C be the set of 1-simplices with one incident vertex in N , and one outside N . These 1-simplices can be regarded as the cut edges of the partition $(S - N, N)$ of S . Since we assumed S to be connected, C must contain at least one 1-simplex. Furthermore, by the definition of $\Gamma(M)$ each 1-simplex of C is matched to a unique 2-simplex by the Morse matching M .

Let D be the subgraph of the modified Hasse diagram of S , which we will denote $H(M)$, consisting of the 1-simplices in C and their matched 2-simplices. We will construct a directed path in D starting at the node of D corresponding to some 1-simplex e_1 in C . Since $e_1 \in C$, it is matched to a 2-simplex f_1 , and we have an edge going from e_1 to f_1 in D . Since f_1 is a 2-simplex with at least one vertex in C and at least one vertex not in C , there must be a second 1-simplex, denoted e_2 , with one incident vertex in N , and one outside N . We can therefore go from f_1 to e_2 in D . Since e_2 is matched with a unique 2-simplex, this process can be continued and we obtain the directed path $e_1, f_1, e_2, f_2, \dots$ in D . Since we have defined a simplicial complex to be finite, we can't keep visiting new vertices of D , so we must visit a vertex of D that we have already visited at some point. Thus D contains a directed cycle, and we can therefore conclude that $H(M)$ contains a directed cycle. This is a contradiction since our assumption that M is a Morse matching implies that the modified Hasse diagram is acyclic. We can therefore conclude that $\Gamma(M)$ is connected. \square

We can use $\Gamma(M)$ to obtain the following corollary.

Corollary 7.6. [22] *Let S be a connected simplicial complex with Morse matching M . Then we can compute a Morse matching M' on S in polynomial time, such that $c(M') \leq c(M)$. Furthermore, M' is optimal on $\Gamma(M)$ with exactly one critical vertex, and the number of critical simplices of dimension 2 or higher is the same as for M .*

Proof. We begin by using Theorem 7.1 to compute an optimal Morse matching $M_{\Gamma(M)}$ on $\Gamma(M)$ in polynomial time. We now construct a new matching M' by setting it equal to $M_{\Gamma(M)}$ on $\Gamma(M)$ and equal to M on the rest of S . M and $M_{\Gamma(M)}$ are valid Morse matchings, and $M_{\Gamma(M)}$ only contributes matchings to M' between dimension 0 and dimension 1, whereas M doesn't contribute matchings between dimension 0 and dimension 1, the fact that M'

is acyclic, and therefore a Morse matching follows from Lemma 6.19 which states that cycles in the modified Hasse diagram only occur between two adjacent dimensions. \square

We are now in a position to prove the *NP*-completeness of the Morse matching decision problem.

Theorem 7.7. [22] *Given a simplicial complex S and a nonnegative integer c , it is *NP*-complete to decide whether there exists a Morse matching with at most c critical simplices. This holds even if S is connected, pure, 2-dimensional and embeddable in \mathbb{R}^3 .*

Proof. It's clear that the problem is in *NP*.

Let S be a connected, pure, 2-dimensional simplicial complex embeddable in \mathbb{R}^3 .

For k any non-negative integer, we wish to show that there exists a Morse matching with at most k critical 2-simplices if and only if there exist a Morse matching with at most $g(k) = 2(k+1) - \chi(S)$ critical simplices in total. Since the Euler characteristic, $\chi(S)$ can be computed in polynomial time, for example by using Theorem 6.11 on a trivial Morse matching, g is a polynomial-time computable function and the result will therefore follow from Corollary 7.4.

Suppose therefore that M is a Morse matching on S with at most k critical 2-simplices. Then by Corollary 7.6 we can compute a Morse matching M' in polynomial time such that

$$c_0(M') = 1, c_2(M') = c_2(M) \text{ and } c(M') \leq c(M).$$

Since S is 2-dimensional we note that

$$c(M') = c_0(M') + c_1(M') + c_2(M')$$

and by Theorem 6.11 we obtain

$$c_1(M') = c_0(M') + c_2(M') - \chi(S) = 1 + c_2(M') - \chi(S).$$

Combining our 3 above sets of equalities, we see that

$$k \geq c_2(M) = c_2(M') = \frac{1}{2}(c(M') + \chi(S)) - 1$$

and therefore that M' has at most $2(k+1) - \chi(S)$ critical simplices in total, which is what we wanted to show.

Conversely, suppose that there exists a Morse matching M with at most $g(k)$ critical simplices in total. We can again use Corollary 7.6 to obtain a Morse matching M' such that $c_0(M') = 1$, $c_2(M') = c_2(M)$ and $c(M') \leq c(M)$. We can then again obtain

$$c_2(M) = c_2(M') = \frac{1}{2}(c(M') + \chi(S)) - 1$$

Substituting in the fact that $c(M') \leq c(M) \leq g(k)$, we obtain

$$\begin{aligned} c_2(M) = c_2(M') &\leq \frac{1}{2}(g(k) + \chi(S)) - 1 \\ &\leq \frac{1}{2}(2(k+1) - \chi(S) + \chi(S)) - 1 \\ &\leq k \end{aligned}$$

□

Thus we have shown that the decision version of our problem is *NP*-complete, and this in turn implies that the optimization problem is *NP*-hard too.

In [15] it was proven that unless $P = NP$, there is no constant-ratio polynomial-time approximation algorithm for the optimization version of the collapsibility problem that we used for our reduction. Since the function g used in our proof is not approximation preserving, the question of whether there exists a constant-ratio polynomial-time approximation algorithm for computing Morse matchings is still open.

7.3 Joswig and Pfetsch

Joswig and Pfetsch provided the first non-trivial algorithm for solving the problem of finding optimal Morse matchings for general simplicial complexes. They formulated the problem as an integer linear program [22].

7.3.1 Integer Programming Formulation

We begin by noting that the problem of finding an optimal Morse matching for a simplicial complex K is the same as the problem of finding a Morse matching of maximal cardinality. Joswig and Pfetsch formulated this problem as selecting a maximal cardinality subset of the edges in the Hasse diagram of K , such that the edges correspond to a matching of the simplices

of K , and that this matching does not result in any directed cycles in the modified Hasse diagram of K . Together these 2 sets of constraints ensure that we find a maximum cardinality valid Morse matching.

Before giving the integer programming formulation, we need to introduce some notation. Until now we have only referred to the Hasse diagram of a simplicial complex as being directed, it will now be beneficial to also consider the undirected Hasse diagram. We denote by A the arc set of the undirected Hasse diagram H of the simplicial complex K and let $x = (x_0, \dots, x_{|A|}) \in \{0, 1\}^{|A|}$ be a vector such that each x_i represents an arc in the Hasse diagram. If $x_i = 0$, this corresponds to not matching the endpoints of x_i and if $x_i = 1$, the endpoints are matched. Let I be an index set, then we define $x(I) = \sum_{i \in I} x_i$. For a vertex v in the Hasse diagram, we denote by $\delta(v)$ the arcs incident to v .

Then x is a matching of the simplices of K if and only if $x(\delta(v)) \leq 1$ for all $v \in H$, since this constraint ensures precisely that we don't attempt to match any one simplex with more than one other simplex.

We say that all the vertices of the Hasse diagram which correspond to simplices of dimension i are on the i 'th level of the Hasse diagram. Denote by C_i the set of cycles between levels i and $i + 1$ in the undirected Hasse diagram. According to Lemma 6.19 these are the only cycles that can exist in the modified directed Hasse diagram. Furthermore, given that a directed cycle in the modified Hasse diagram has to alternate between simplices of dimension i and $i + 1$, half the edges must go up, and we therefore see that if for every cycle $c \in C_i$, we choose fewer than half the edges, the modified Hasse diagram will be acyclic. Conversely, if the modified Hasse diagram is acyclic, there would exist some undirected cycle for which at least half the edges had been chosen. We therefore obtain the constraints $x(c) \leq \frac{1}{2}|c| - 1$ for all $c \in C_i$ and $i = 0, \dots, d - 1$, where d is the dimension of K .

Combining the constraints, we see that $x \in \{0, 1\}^{|A|}$ satisfies the following integer linear program if and only if it corresponds to a valid Morse matching.

$$\begin{aligned}
& \max. \quad 1^T x \\
& \text{s.t.} \quad x(\delta(v)) \leq 1 \text{ for all } v \in H \\
& \quad \quad x(c) \leq \frac{1}{2}|c| - 1 \text{ for all } c \in C_i, i = 0, \dots, d - 1 \\
& \quad \quad x \in \{0, 1\}^{|A|}
\end{aligned}$$

It is important to recall that this optimization problem is *NP*-hard. Furthermore the number of constraints required to ensure acyclicity is exponential in the size of the complex.

7.3.2 Branch and Cut

Joswig and Pfetsch used a branch and cut algorithm to solve this problem. We will here give a brief introduction to branch and bound, and branch and cut methods. For an in depth description, see [38].

Suppose that we wish to maximize an objective function $f(x)$ for $x \in S$, where S is the set of feasible solutions. In the above integer programming formulation, the objective function is to maximize the cardinality of a subset of the edges in the Hasse diagram of K , and the feasible set S is defined by the constraints ensuring that the subset of edges is a valid Morse matching. In branch and bound we decompose S into subsets S_i , such that

$$S = S_1 \cup \dots \cup S_N$$

We can then attempt to compute upper and lower bounds for the objective function value on S and the sets S_i . A lower bound is usually found by finding a feasible solution, while upper bounds can be found by solving a relaxation of the original problem. We can then process a subset S_i by either solving it to optimality, or by obtaining an upper bound on S_i which is lower than our lower bound on S . This use of bounds to process a subset S_i can allow us to eliminate subsets of the feasible set without having to fully solve our original problem to optimality on them.

In the case of our problem, we can use branch and bound using the bounds obtained from the linear programming relaxation of the problem, but we still have the problem that there are an exponential number of acyclicity constraints.

Branch and cut is an extension of branch and bound where we combine a cutting plane algorithm with the branch and bound approach. In this approach, instead of just solving the linear programming relaxation, we also relax the acyclicity constraints. We therefore obtain a much smaller linear program which can be solved more quickly. The disadvantage is that the bounds provided by the relaxations are likely to be very poor. Assuming the solution to the relaxed problem is infeasible, we therefore add so called cuts each time we solve our relaxed problem. These cuts are simply constraints violated by the obtained solution, but not by any feasible solution.

The advantage of branch and cut over branch and bound is therefore that we don't have to solve our LP relaxation for all the acyclicity constraints, but can instead add only those constraints that are needed at a given iteration. This comes at the cost of solving a so called separation problem for generating the additional constraints. This separation problem is described in detail in [22], but the basic idea is to transform the modified Hasse diagram to a weighted graph and try to find a shortest cycle in this weighted graph. To solve the separation problem we then have to establish if the shortest cycle has weight greater than or equal to 1 in order to establish if a given solution is feasible or if we must add a cut to eliminate the found cycle.

The branch and cut algorithm developed by Joswig and Pfetsch is faster than a simple exhaustive enumeration of all the possible Morse matchings, but it is still far too slow to be used as an effective preprocessing step for homology computation. For example, even for a triangulation with only 392 simplices an optimal solution could not be found in a week in [22], whereas implementations of the collapse and coreduction heuristics that will be introduced in Sections 7.6 and 7.7 usually solve the problem to optimality in less than 30 milliseconds.

It is not surprising that the Joswig and Pfetsch algorithm is inappropriate for preprocessing simplicial complexes for the computation of homology. This is because the computation of homology takes roughly cubic time in the size of the complex in the worst case, whereas we have already established that the problem of finding optimal Morse matchings is *NP*-hard. If we want to be able to use discrete Morse theory effectively as a preprocessing step, we are therefore forced to either obtain faster optimal algorithms for special cases, fast approximation algorithms, or fast heuristic algorithms.

7.4 Naive heuristics

Potentially the easiest way of finding a Morse matching is simply to randomly match simplices in a manner that doesn't cause a cycle until we can't match any simplices without causing a cycle. One simple implementation of this idea would be to randomly make one pass through the simplices matching each with one of its cofaces whenever doing so doesn't cause a cycle. Checking for acyclicity using depth first search can be done in linear time, but if we do this after each iteration, we obtain a quadratic running time. Given that the early matchings are unlikely to cause cycles we can speed up the heuristic by only checking for acyclicity infrequently initially, and then undoing the matchings

since the last check if a cycle is found. Implemented in this manner, the heuristic is relatively fast. The Lewiner, collapse and coreduction heuristics that will be introduced in subsequent sections all achieve linear running times in part by choosing the matchings in a manner that makes it unnecessary to check for cycles.

A few variants of this approach were implemented in Java, but unfortunately they performed poorly. As well as performing poorly in practice, it turns out that the poor performance of these heuristics based on completely random choices is to be expected from a theoretical point of view [33].

Some brief experimental comparisons in Section 7.10 will demonstrate that for any but the simplest of complexes, this sort of naive approach is uncompetitive with the heuristics that will be described later in this section.

7.5 Lewiner et al

Lewiner et al developed a linear time algorithm for computing optimal Morse matchings on triangulations of compact surfaces [25].

Specifically, they showed that for a triangulation K of a compact surface, their linear time algorithm produces a Morse matching M with $c_0(M) = 1$ and such that if K has no boundary, $c_2(M) = 1$ and if K has a boundary, $c_2(M) = 0$.

Combining Theorems 5.7 and the weak Morse inequalities then immediately proves optimality, since the the number of critical p -simplices is bounded from below by the p 'th Betti numbers.

The Lewiner algorithm relied on constructing a spanning tree on a certain graph structure created based on the input complex. Their algorithm was later extended to a heuristic for general CW -complexes, rather than just triangulations of compact surfaces. In [4] it was remarked that this heuristic produces somewhat inconsistent results for problem instances that were complicated. This suggests that this heuristic is perhaps not as good as the collapse and coreduction heuristics, but an interesting future project might be to implement this heuristic and establish when the Lewiner heuristic works well, and under what circumstances it doesn't perform so well.

7.6 Benedetti and Lutz

Benedetti and Lutz designed a heuristic for testing if a simplicial complex is collapsible, and if so, how easy it is to collapse. This heuristic turned out to be very good at computing Morse matchings with very few critical cells [4]. We will refer to it as the **collapse heuristic** in this report.

The basic idea can be understood in terms of Lemmas 6.9 and 6.10. We try to find free faces as in the proof of Lemma 6.9. If a free face is found, delete the free face and its coface. In terms of our Morse matching, this implicitly corresponds to matching the free face with its coface. If at any point we can't find a free face, we delete one of the remaining simplices of maximum dimension and delete it. The simplices deleted in this manner correspond to the critical simplices of our Morse matching and this step can be understood as representing the attachment given in Lemma 6.10. We repeat this process until every simplex has been deleted.

It is simple to verify that the resulting modified Hasse diagram is acyclic, and therefore that the matching produced by this heuristic is a valid Morse matching. This follows from the fact that we have only matched faces when they were free in the remaining complex.

If we store the unprocessed simplices, the free faces and the pairs of matched simplices we can implement this heuristic in linear time in the number of simplices for fixed dimension. Each iteration either removes one or two simplices, and we only need to update a constant number of faces of the simplices removed. This heuristic was implemented in Java and along with the heuristic due to Mrozek and Batko, it will be the main heuristic with which simulated annealing approach will be compared in Section 7.10.

While this heuristic performs very well in practice, it is simple to construct simplicial complexes where the heuristic can be made arbitrarily likely to fail to produce an optimal solution, see Figure 19 [4].

7.7 Mrozek and Batko

Mrozek and Batko developed a so called coreduction based method for computing Morse matchings in [30]. We will in this report use a slight modification of this method, effectively making it a reversed version of the collapse heuristic. Thus, instead of finding free faces, we attempt to find **free cofaces**, simplices which only have one face. Additionally, we delete simplices from the bottom up, instead of from the top down as in the collapse heuristic.

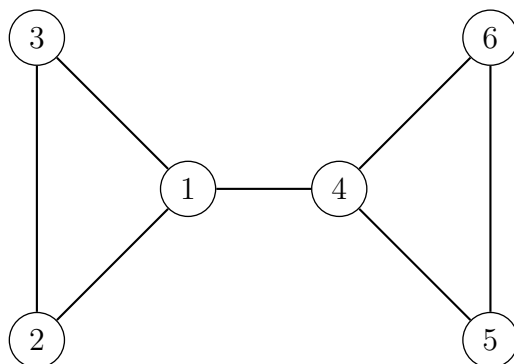


Figure 19: Since there are no free faces in the beginning, we start by deleting an edge. There is a $1/7$ chance of choosing the middle edge and obtaining a suboptimal Morse matching. By creating a chain of vertices and edges between vertices 1 and 4, we can make the chance of obtaining an optimal Morse matching arbitrarily small.

We will refer to this heuristic as the coreduction heuristic.

Note that initially there can be no free cofaces in a simplicial complex, so we always initialize this heuristic by deleting a vertex. This doesn't cause any problems, since there is always at least one critical vertex for any simplicial complex. Similar arguments hold for the correctness and running time of the collapse and coreduction heuristics. Like the collapse heuristic, the coreduction heuristic can be made arbitrarily unlikely to produce an optimal solution on simplicial complexes as in Figure 19.

The coreduction heuristic was implemented in Java and it will be one of the main heuristics investigated in Section 7.10. One of the main things we can conclude from the comparisons is that the methods complement each other, with some instances being more amenable to the collapse heuristic, and others to the coreduction heuristic.

7.8 Other approaches

We will briefly mention some other approaches, one of which can be used to improve other heuristics for computing Morse matchings, and one of which produces Morse matchings, even though the theoretical basis is slightly different.

7.8.1 Iteration

We noted earlier that the theory we have developed could have been extended to finite CW-complexes rather than just simplicial complexes. Since the Morse complex that we obtain is itself a finite CW-complex, this means that we can iterate our methods. Instead of stopping after we have obtained a Morse complex from our Morse matching, we can compute a new Morse matching on the Morse complex.

This process can then be repeated until improvements stop or become too small. Slight modifications are required to the heuristics, since for arbitrary finite CW-complexes, we have to restrict the permitted Morse matchings, but this only means that we have to ignore some matchings that we would otherwise have chosen. Assuming we start with a simplicial complex, these modifications are not needed for the first iteration, and there is therefore no disadvantage to iterating our procedure. An example of this approach can be found in [9]. Since it is possible, with some minor modifications, to carry out this iterated method of computing Morse complexes for all the above mentioned heuristics, we have not carried out this extension or made experimental comparisons using it. It would however be an interesting future extension of this project to see if some of the heuristics benefit more than others from iteration.

7.8.2 Cancelling Critical Simplices

Another approach to improving Morse matchings can be obtained from the following theorem.

Theorem 7.8. *[18] Suppose M is a Morse matching on a simplicial complex K such that $\beta^{(p+1)}$ and $\alpha^{(p)}$ are critical, and there is exactly only gradient path from the boundary of β to α . Then there exists another Morse matching M' with the same critical simplices, except that β and α are no longer critical.*

The way we obtain M' from M is to reverse the unique gradient path from β to α in the theorem. This could for example be applied to remove the critical simplices in Figure 18.

7.8.3 Lexicographic Ordering

In many of the heuristics we have described, random choices are made when choosing which simplices to match or to mark as critical. An alternative

approach is to make choices based on a lexicographic ordering. Lexicographic and reverse lexicographic versions of the collapse heuristic performed very well for some problem instances, although the completely randomized version was also superior for other instances [4]. Since implementing lexicographic versions of the collapse and coreductions heuristics is only a small task, it would be interesting in the future to try to investigate for what types of complexes each method is superior.

Note that this approach can still be randomized by randomizing the vertex numbers from which the lexicographic ordering is obtained. The theoretical justification for this approach is given in [3, 20].

7.8.4 Engström

Finally, in [16], Engström used what he called **Fourier-Morse theory** to compute a special type of Morse matching. There are complexes which are collapsible, for which this method can not produce an optimal Morse matching [4]. Some basic experimental comparisons involving this method will be made in Section 7.10, however since Engström's computations are not randomized, these comparisons will only be superficial. Generally the collapse and coreduction heuristics seem to produce Morse matchings with fewer critical simplices than Engström's method.

7.9 Simulated Annealing

Simulated Annealing has until now not been applied to the problem of computing Morse matchings. For this project simulated annealing based on the collapse and coreduction heuristics was implemented, and experimental results suggest that this approach has potential. For several instances the simulated annealing approach works better than the aforementioned heuristics, which are two of the best currently known heuristics for the problem. Unfortunately a bug in the the implementation of simulated annealing based on the coreduction heuristic prevented this version from being used in the experimental comparisons. However preliminary results suggest that it performs better than the coreduction heuristic, in the same way that it will be shown in Section 7.10 that simulated annealing based on the collapse heuristic performed better than the collapse heuristic.

Simulated annealing is inspired by the process of annealing in metallurgy. Suppose we have an optimization problem to minimize an objective function

$f(s)$, for $s \in S$, where S is the feasible set of solutions. The basic idea of simulated annealing is to initialize a temperature parameter t and generate an initial solution s using some heuristic, and then to iteratively replace s with another solution s' in the neighborhood of s . Here the concept of a neighborhood of a solution needs to be defined for a given implementation of simulated annealing, but the principle is to find a solution that is similar to the original solution. If $f(s') \leq f(s)$, we accept s' and continue the process using s' . If however $f(s') > f(s)$, we accept s' with a probability $p(t, f(s), f(s'))$, and reject it with probability $1 - p(t, f(s), f(s'))$. This temperature parameter is reduced for each iteration, and $p(t, f(s), f(s'))$ declines as t is reduced.

The idea therefore is that in the beginning, when t is still large, we are able to explore a large part of the feasible set, and as t declines, the simulated annealing process begins to resemble a simple hill climbing method. When some criteria for stopping the procedure are fulfilled, we terminate the procedure and output the best solution encountered.

Simulated annealing is a widely used technique in combinatorial optimization problems, and is particularly important when there are no known efficient exact or approximation algorithms. Since this is the case for the problem of computing optimal Morse matchings, simulated annealing was implemented for the problem.

For this implementation, the definition of neighboring solution is that given a matching M , generated by either the collapse or coreduction heuristic, we undo all the matchings made either above or below a randomly chosen dimension. We then compute a new Morse matching M' such that for some of the dimensions M and M' consists of the same matchings, and for the other dimensions M and M' are different. This can be regarded as a very broad type of neighboring solution, and the success of the implementation might be improved by improving the definition of the neighborhood.

7.10 Comparison of Methods

As noted earlier, heuristics are usually compared experimentally in terms of either their running times or the qualities of their solutions, and usually some combination of the two factors [35]. For our purposes, the most important factor is the quality of solution obtained, since the homology computations with currently available hardware and algorithms are often intractable for complexes that are too large.

The heuristics will be tested on triangulations taken from the Benedetti and Lutz library of triangulations, [4] and a selection of so called (m, n) -chessboard complexes for different values of m and n . The Benedetti and Lutz library was chosen because it contains some triangulations for which it is relatively difficult to obtain good Morse matchings, even though the complexes are small, and because the Engström heuristic has also been tested on some of these triangulations. This will allow a limited comparison even though the Engström heuristic wasn't implemented for this project. The chessboard complexes have been chosen because they were also tested for the Engström heuristic, but more importantly because for larger values of m and n their homology is unknown and an improved method of finding discrete Morse matchings might therefore help in future attempts to compute their homology.

We begin by comparing the performance of the best performing of the three naive heuristic implementations to that of the collapse heuristic on a selection of complexes. For each complex, we give its name, the number of simplices and a lower bound on the number of critical simplices that can be obtained. An asterisk is used to indicate that the lower bound is known to be feasible, and therefore optimal. The below table shows for each complex and heuristic a triple of numbers, the mean number of critical simplices obtained over 30 iterations, the minimum number of critical simplices obtained and the standard deviation over the 30 iterations.

complex	Naive	Collapse
bing (8131, 1*)	(651.4, 585, 42.11)	(3, 3, 0)
rudin (215, 1*)	(16.2, 9, 3.29)	(1, 1, 0)
S_3 -50-1033 (4232, 2*)	(386.8, 358, 15.10)	(3.53, 2, 2.46)
chessboard55 (1545, 57)	(173.6, 157, 8.02)	(67.33, 61, 4.04)

It's clear that the naive heuristics perform as poorly as expected and we therefore disregard them for the rest of this section.

We will instead compare the collapse heuristic, coreduction heuristic and Engström heuristic on those triangulations tested by Engström which are either chessboard complexes or in the Benedetti and Lutz library. Additionally we will carry out experiments for some other complexes to compare the collapse and coreduction heuristics. For the Engström heuristic, the results are obtained from [16], and are for a single iteration. The heuristic was later randomized, but no results have been published with experimental results

for the randomized version. The below table is formatted as before, but for the Engström heuristic, we just present the value listed in [16]. In the below table, for each row, we list an identifier of the simplicial complex, and then its number of simplices and a lower bound on the number of critical simplices. We then list for the collapse and coreduction heuristics a triple representing the mean and minimum numbers of critical simplices obtained in 30 trials, and the standard deviation. For the Engström heuristic we simply present the value given in [16].

complex	Collapse	Coreduction	Engström
dunce hat (49, 3*)	(3, 3, 0)	(3, 3, 0)	3
poincare (392, 6*)	(6.08, 6, 0.39)	(6.04, 6, 0.28)	12
rudin (215, 1*)	(1, 1, 0)	(1.04, 1, 0.28)	5
chessboard33 (33, 5*)	(5, 5, 0)	(5, 5, 0)	5
chessboard44 (208, 16*)	(16.08, 16, 0.39)	(16, 16, 0)	18
chessboard55 (1545, 57)	(67.92, 59, 4.49)	(60.4, 59, 2.43)	83
chessboard66 (13326, 236)	(521.44, 460, 33.21)	(389.72, 348, 20.98)	470
non42colorable (5982, 2*)	(29.52, 14, 6.97)	(37.8, 26, 7.49)	N/A
HP2 (16383, 3*)	(5.36, 3, 3.95)	(15.92, 3, 14.67)	N/A

The Engström heuristic was better than the collapse heuristic for the largest of the tested chessboard complexes, but worse for all other non-trivial complexes, and also worse than the coreduction heuristic for all non-trivial complexes.

We can use Welch’s t test to see that the difference in means between the collapse and coreduction heuristics is statistically significant for the last 4 complexes in the above table. Note that Welch’s t test assumes that the results are normally distributed for both heuristics, this assumption is not met and a more appropriate statistical test if the differences weren’t so clear would be a non parametric test where we compare the ranks of the results of the heuristics, rather than the results themselves.

With respect to the collapse and coreduction heuristics, from our experiments we can conclude that neither of the 2 heuristics dominates the other. Instead each outperforms the other on some input complexes. Additionally, in [4] it was found that for some problem instances randomized lexicographic versions of the collapse heuristic outperformed the standard randomized version, while again there were other complexes for which the standard randomized version performed best. Similarly, while the Engström heuristic is

dominated by the collapse heuristic, given the limited number of complexes tested, it seems very possible that there are complexes for which the Engström heuristic performs better than either of the other 2 tested heuristics. In the absence of a dominant heuristic it therefore seems potentially very important to have a variety of the above mentioned heuristics available.

It is very likely that modifying the above heuristics with the lexicographic and iterated modifications described in Section 7.8 would significantly improve their performances for some complexes. Since these improvements have not been implemented for this project, they couldn't be tested, but an obvious future extension would be to implement and test lexicographic and iterated versions of the heuristics.

Before investigating the benefits of using simulated annealing over simply using the collapse heuristic, we note that there are only very few of the complexes in the Benedetti and Lutz library that are candidates for improvements through simulated annealing. This is because the vast majority of the triangulations are easily solved by the collapse heuristic or too large for the experiments to be run. We therefore need to look at those relatively small complexes for which the heuristics produce inconsistent results. In addition to the Benedetti and Lutz library, the chessboard complexes are also suitable, since the collapse heuristic produces inconsistent results for these complexes too. Unfortunately, even though the collapse and coreduction heuristics could have been implemented in linear time in terms of the number of simplices, the actual implementations exhibit superlinear growth, in addition some of the larger complexes are very slow to construct. Due to time constraints, the number of suitable complexes which could be tested for a reasonable number of iterations is therefore very limited. The candidate complexes for this comparison are therefore only the following two complexes: **non42colorable** and **chessboard66**.

In Figure 20 we compare the rate at which the solution qualities converge as we increase the computation time for the collapse heuristic and simulated annealing for the complex **non42colorable**. Simulated annealing is consistently better, but this difference is quite small.

Finally, we compare the performance of the collapse heuristic to that of simulated annealing based on the collapse heuristic for the suitable complexes as in the previous tables. This comparison could only be done properly on two complexes, while we have preliminary results on two larger complexes that produce similar results. Preliminary results also suggest that simulated annealing based on the coreduction heuristic for the same complexes outper-

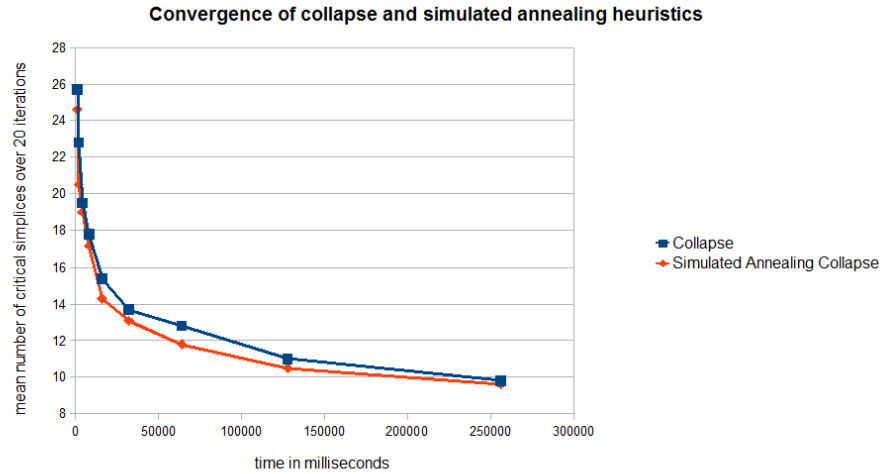


Figure 20

forms the coreduction heuristic. In the below table, for each row, we list an identifier of the simplicial complex, its number of simplices, a lower bound on the number of critical simplices, and the number of trials. We then list for the collapse heuristic and simulated annealing a triple representing the mean and minimum numbers of critical simplices obtained in the trials, and the standard deviation. Each trial was allowed to run for 20 seconds for non42colorable and 120 seconds for chessboard66.

complex	Collapse	Simulated Annealing
non42colorable (5982, 2*, 30)	(15.27, 10, 2.94)	(13.2, 8, 2.66)
chessboard66 (13326, 236, 60)	(461.27, 430, 15.81)	(455.83, 416, 14.21)

We can again apply Welch's t -test to see that the differences in means are statistically significant for both complexes. Thus simulated annealing outperforms the collapse heuristic for the tested complexes, but the benefits are not large. Even though the differences are relatively small, they seem to grow as the size of the input complexes grows. In particular limited test runs on the (6, 7)- and (7, 7)-chessboard complexes gave larger benefits to using simulated annealing. Unfortunately due to the above mentioned time constraints, it was not possible to test the performance on these complexes

repeatedly. We can therefore conclude that there is a small but clear benefit to using simulated annealing on the 2 tested complexes. Furthermore, based on the tests on the $(6, 7)$ - and $(7, 7)$ -chessboard complexes, it is very possible that there is a larger practical benefit to using simulated annealing for larger complexes, although more testing would be needed to verify this. As noted earlier, the definition of a neighboring solution used in the simulated annealing implementation is quite primitive, and it seems likely that improvements to the performance of simulated annealing could be made in this respect.

8 My Contributions

This report has provided an overview of the theoretical justification for the use of discrete Morse theory for the computation of homology and persistent homology, and a motivation for our interest in these topics. The most important contributions beyond this are the new simulated annealing based methods for computing Morse matchings, and an experimental comparison of some of the most important heuristics currently available for the computation of Morse matchings. In particular this comparison showed that simulated annealing outperforms the collapse heuristic for several problem instances. Comparing just the collapse and coreduction heuristics, it was found that for some problems the collapse heuristic is preferable, while for others the coreduction heuristic performs better.

Additionally this project has resulted in the implementation in Java of an algorithm for constructing Vietoris-Rips complexes and of algorithms for constructing explicit simplicial complexes allowing for experimental comparison of computational topology algorithms. Finally several heuristics for computing, and improving, Morse matchings were implemented, including the collapse and coreduction heuristics, simulated annealing based methods using these two heuristics and the cancellation method covered in Section 7.8.2.

9 Future Work

If we consider topological data analysis our motivation, there are several obvious areas where improvements are needed to enable the robust application of topological data analysis to large, high dimensional data sets. Recall that our general approach to topological data analysis has two main steps, that of approximating the topological space from which the data has been sampled, and that of computing topological invariants for our approximation. We will here briefly mention the main issues related to these two problems, before restricting our attention to the main focus of this report, the attempt to apply discrete Morse theory to the computation of homology and persistent homology.

The success of our attempts to approximate the topological space from which our data has been sampled can be measured in terms of three factors. The computation speed and memory requirements of our approximations, and how similar these approximations are to the true underlying spaces of the data topologically. In order to improve the speed of construction of the complexes, it seems necessary to either approximate the Vietoris-Rips complex, or use a different type of complex. From the point of view of memory, the commonly used tool for high dimensional problems, the Vietoris-Rips complex is too large to compute for huge data sets. Witness complexes have been proposed as a possible solution. It might also be possible to apply discrete Morse theory during the construction of the complex, thus never allowing the complex to grow dramatically in the first place. A similar approach proposed by Zomorodian is to compute what he called the tidy set of a simplicial complex, which can be used to efficiently compute the homology of a Vietoris-Rips complex [40]. The problem of establishing the accuracy of our approximations has only recently begun to be investigated and some results are given in [34, 2].

In topological data analysis and image analysis, we are not just interested in constructing simplicial complexes, but also other types of CW-complexes, particularly so called cubical complexes. It would be interesting to investigate what benefits and disadvantages cubical complexes have compared to simplicial complexes for different types of applications.

Persistent homology is the most important topological invariant in topological data analysis. An extension of persistent homology known as zigzag persistent homology has the potential to provide more topological information, and to allow less restrictive sequences of complexes than filtrations,

without significantly increasing the computational requirements.

We now return to the main focus of this report. The computation of Morse matchings, with a view to their use in preprocessing homology computations. An interesting extension of this project from an implementation point of view would be the implementation of the standard algorithms for the computation of homology and persistent homology. Even though the running time of these algorithms is significantly determined by the size of the input complex, this is not the only factor, and it would be interesting to see to what extent Morse matchings with similar numbers of critical simplices result in similar computation times for homology. Furthermore, implementing these algorithms would mean that all the steps necessary for the application of topological data analysis have been implemented. The computation of Vietoris-Rips complexes was implemented for this project along with the various heuristics for computing Morse matchings. The only missing step from this point of view are algorithms for computing homology and persistent homology.

Another interesting extension of this project would be to modify the various heuristics implemented to enable the iterated computation of Morse matchings. This would again be interesting from the experimental side because we could see if some of the heuristics are better suited to iteration, while it would be useful from the application point of view since it would likely improve the number of critical simplices obtained for large complexes significantly.

The collapse and coreduction heuristics investigated in this report could for some problem instances be improved by replacing random choices with choices determined by lexicographic order. It therefore seems possible that there could also be other methods of guiding the choices that would improve the quality of the solutions provided by these heuristics. It would also be advantageous to reimplement them to run in linear time. In particular, this would allow the experiments to be run for larger complexes.

The problem of computing Morse matchings is quite similar to the feedback arc set problem. There exist interesting approximation algorithms for the feedback arc set problem, whereas none are yet known for computing Morse matchings. An obvious future research area would be to try to obtain some results related to approximating Morse matchings, either in the form of algorithms, or proofs of hardness.

Returning to the broader field of topological data analysis, the greatest potential bottleneck for the more wide spread application of topological

methods in data analysis seems likely to be the difficulty of computing complexes that approximate the topology of the data. It could therefore be very interesting to investigate methods for approximating Vietoris-Rips complexes in the future.

References

- [1] P. S. Alexandrov, *Combinatorial topology. Vol. 1, 2 and 3*, Dover Publications Inc., Mineola, NY, 1998, Translated from the Russian, Reprint of the 1956, 1957 and 1960 translations. MR 1643155 (99g:55001)
- [2] Dominique Attali, André Lieutier, and David Salinas, *Vietoris-Rips complexes also provide topologically correct reconstructions of sampled shapes*, *Comput. Geom.* **46** (2013), no. 4, 448–465. MR 3003901
- [3] Eric Babson and Patricia Hersh, *Discrete morse functions from lexicographic orders*, *Transactions of the American Mathematical Society* **357** (2005), no. 2, 509–534.
- [4] Bruno Benedetti and Frank H Lutz, *Random discrete morse theory and a new library of triangulations*, arXiv preprint arXiv:1303.6422 (2013).
- [5] Gunnar Carlsson, Afra Zomorodian, Anne Collins, and Leonidas J Guibas, *Persistence barcodes for shapes*, *International Journal of Shape Modeling* **11** (2005), no. 02, 149–187.
- [6] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer, *Stability of persistence diagrams*, *Discrete Comput. Geom.* **37** (2007), no. 1, 103–120. MR 2279866 (2008i:68130)
- [7] Vin de Silva and Robert Ghrist, *Coverage in sensor networks via persistent homology*, *Algebraic & Geometric Topology* **7** (2007), no. 339-358, 24.
- [8] Cecil Jose A. Delfinado and Herbert Edelsbrunner, *An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere*, *Comput. Aided Geom. Design* **12** (1995), no. 7, 771–784, Grid generation, finite elements, and geometric design. MR 1365107 (96h:55004)
- [9] Paweł Dłotko and Hubert Wagner, *Computing homology and persistent homology using iterated morse decomposition*, arXiv preprint arXiv:1210.1429 (2012).
- [10] Milka Doktorova and Afra Zomorodian, *Constructing simplicial complexes over topological spaces*, Preliminary Draft (2012).

- [11] Herbert Edelsbrunner, *Weighted alpha shapes*, (1992).
- [12] Herbert Edelsbrunner and John L. Harer, *Computational topology*, American Mathematical Society, Providence, RI, 2010, An introduction. MR 2572029 (2011e:00001)
- [13] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian, *Topological persistence and simplification*, Discrete Comput. Geom. **28** (2002), no. 4, 511–533, Discrete and computational geometry and graph drawing (Columbia, SC, 2001). MR 1949898 (2003m:52019)
- [14] Herbert Edelsbrunner and Ernst Peter Mücke, *Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms*, Proceedings of the Fourth Annual Symposium on Computational Geometry (Urbana, IL, 1988) (New York), ACM, 1988, pp. 118–133. MR 1213465
- [15] Ömer Egecioglu and Teofilo F. Gonzalez, *A computationally intractable problem on simplicial complexes*, Comput. Geom. **6** (1996), no. 2, 85–98. MR 1390392 (97a:68159)
- [16] Alexander Engström, *Discrete morse functions from fourier transforms*, Experimental Mathematics **18** (2009), no. 1, 45–53.
- [17] Robin Forman, *Morse theory for cell complexes*, Adv. Math. **134** (1998), no. 1, 90–145. MR 1612391 (99b:57050)
- [18] ———, *A user's guide to discrete Morse theory*, Sémin. Lothar. Combin. **48** (2002), Art. B48c, 35. MR 1939695 (2003j:57040)
- [19] Allen Hatcher, *Algebraic topology*, Cambridge University Press, Cambridge, 2002. MR 1867354 (2002k:55001)
- [20] Patricia Hersh, *On optimizing discrete morse functions*, Advances in Applied Mathematics **35** (2005), no. 3, 294–322.
- [21] Thomas W. Hungerford, *Algebra*, Graduate Texts in Mathematics, vol. 73, Springer-Verlag, New York, 1980, Reprint of the 1974 original. MR 600654 (82a:00006)
- [22] Michael Joswig and Marc E. Pfetsch, *Computing optimal Morse matchings*, SIAM J. Discrete Math. **20** (2006), no. 1, 11–25 (electronic). MR 2257241 (2007h:90067)

- [23] Tomasz Kaczynski, Konstantin Mischaikow, and Marian Mrozek, *Computational homology*, Applied Mathematical Sciences, vol. 157, Springer-Verlag, New York, 2004. MR 2028588 (2005g:55001)
- [24] Dimitry Kozlov, *Combinatorial algebraic topology*, vol. 21, Springer, 2008.
- [25] Thomas Lewiner, Hélio Lopes, and Geovan Tavares, *Optimal discrete Morse functions for 2-manifolds*, *Comput. Geom.* **26** (2003), no. 3, 221–233. MR 2005300 (2004h:57041)
- [26] A. Markov, *The insolubility of the problem of homeomorphy*, *Dokl. Akad. Nauk SSSR* **121** (1958), 218–220. MR 0097793 (20 #4260)
- [27] J. Milnor, *Morse theory*, Based on lecture notes by M. Spivak and R. Wells. *Annals of Mathematics Studies*, No. 51, Princeton University Press, Princeton, N.J., 1963. MR 0163331 (29 #634)
- [28] Nikola Milosavljević, Dmitriy Morozov, and Primož Škraba, *Zigzag persistent homology in matrix multiplication time*, *Computational geometry (SCG'11)*, ACM, New York, 2011, pp. 216–225. MR 2919613
- [29] Konstantin Mischaikow and Vidit Nanda, *Morse Theory for Filtrations and Efficient Computation of Persistent Homology*, *Discrete Comput. Geom.* **50** (2013), no. 2, 330–353. MR 3090522
- [30] Marian Mrozek and Bogdan Batko, *Coreduction homology algorithm*, *Discrete & Computational Geometry* **41** (2009), no. 1, 96–118.
- [31] James R. Munkres, *Topology: a first course*, Prentice-Hall Inc., Englewood Cliffs, N.J., 1975. MR 0464128 (57 #4063)
- [32] ———, *Elements of algebraic topology*, Addison-Wesley Publishing Company, Menlo Park, CA, 1984. MR 755006 (85m:55001)
- [33] Liviu I Nicolaescu, *Combinatorial morse flows are hard to find*, arXiv preprint arXiv:1202.2724 (2012).
- [34] Partha Niyogi, Stephen Smale, and Shmuel Weinberger, *Finding the homology of submanifolds with high confidence from random samples*, *Discrete & Computational Geometry* **39** (2008), no. 1-3, 419–441.

- [35] Ronald L Rardin and Reha Uzsoy, *Experimental evaluation of heuristic optimization algorithms: A tutorial*, Journal of Heuristics **7** (2001), no. 3, 261–304.
- [36] Carsten Thomassen, *The Jordan-Schönflies theorem and the classification of surfaces*, Amer. Math. Monthly **99** (1992), no. 2, 116–130. MR 1144352 (92k:57026)
- [37] Virginia Vassilevska Williams, *Multiplying matrices faster than Coppersmith-Winograd [extended abstract]*, STOC’12—Proceedings of the 2012 ACM Symposium on Theory of Computing, ACM, New York, 2012, pp. 887–898. MR 2961552
- [38] Laurence A Wolsey, *Integer programming*, IIE Transactions **32** (2000), no. 273-285, 2–58.
- [39] Afra Zomorodian, *Fast construction of the Vietoris-rips complex*, Computers & Graphics **34** (2010), no. 3, 263–271.
- [40] ———, *The tidy set: a minimal simplicial set for computing homology of clique complexes [extended abstract]*, Computational geometry (SCG’10), ACM, New York, 2010, pp. 257–266. MR 2742959
- [41] ———, *Topological data analysis*, Advances in applied and computational topology, Proc. Sympos. Appl. Math., vol. 70, Amer. Math. Soc., Providence, RI, 2012, pp. 1–39. MR 2963600
- [42] Afra Zomorodian and Gunnar Carlsson, *Computing persistent homology*, Discrete Comput. Geom. **33** (2005), no. 2, 249–274. MR 2121296 (2005j:55004)