Turing Decidability and Computational Complexity of Morse Homology

Christopher Dare

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
in
Mathematics

William Floyd
Peter Haskell
Leslie Kay
Leonardo Mihalcea

July 20, 2019

Keywords: Morse Homology, Computational Complexity, Turing Machines

Turing Decidability and Computational Complexity of Morse Homology

Christopher Dare

ABSTRACT

This thesis presents a general background on discrete Morse theory, as developed by Robin Forman, as well as an introduction to computability and computational complexity. Since general point-set data equipped with a smooth structure can admit a triangulation [19], discrete Morse theory finds numerous applications in data analysis which can range from traffic control to geographical interpretation. Currently, there are various methods which convert point-set data to simplicial complexes or piecewise-smooth manifolds; however, this is not the focus of the thesis. Instead, this thesis will show that the Morse homology of such data is computable in the classical sense of Turing decidability, bound the complexity of finding the Morse homology of a given simplicial complex, and provide a measure for when this is more efficient than simplicial homology.

Turing Decidability and Computational Complexity of Morse Homology

Christopher Dare

GENERAL AUDIENCE ABSTRACT

With the growing prevalence of data in the technological world, there is an emerging need to identify geometric properties (such as holes and boundaries) to data sets. However, it is often fruitless to employ an algorithm if it is known to be too computationally expensive (or even worse, not computable in the traditional sense). However, discrete Morse theory was originally formulated to provide a simplified manner of calculating these geometric properties on discrete sets. Therefore, this thesis outlines the general background of Discrete Morse theory and formulates the computational cost of computing specific geometric algorithms from the Discrete Morse perspective.

Acknowledgements

I would first and foremost like to thank my advisor, William J. Floyd, for his constant support and mentorship throughout my graduate program. We both walked into a subject which neither of us had touched upon before; yet Professor Floyd took the time to keep up with my studies in addition to his own research and students. It is certainly no simple task to cover an entire textbook on a foreign topic within a semester, and thus I would like to express my sincere gratitude for his willingness to help.

I would also like to thank Peter Haskell, in addition to William J. Floyd, for helping me choose the topic of the thesis in the last year of his role as department chair. Despite his busy schedule, Professor Haskell was always eager to help me explore the numerous research opportunities related to differential geometry and algebra.

Contents

List of Figures

Chapter 1

Introduction

1.1 Preliminaries

Before we introduce both the intricacies of discrete Morse theory and the challenging development of Morse homology, we must first build up an understanding of simplicial complexes and how Morse functions fit into the subject. As a forward, the application of discrete Morse theory to simplicial complexes can be greatly simplified to piecewise-linear Morse theory, as developed by Thomas Banchoff [1] in the late 1960s and early 1970s. However, the computational analyses of piecewise-linear Morse theory and of discrete Morse theory are very similar (particularly on simplicial complexes), and thus we choose to formulate the computational efficiency in the more general case. This will allow us to analyze smooth manifolds (observed under a triangulation) as well as general simplicial complexes of point-set data.

We begin this chapter by introducing the basic concepts and definitions of discrete Morse theory and the relevant terms in algebraic combinatorics.

1.2 Simplicial Complexes and Discrete Morse Functions

At this point, little has been addressed regarding the significance of discrete Morse functions and why they are used in tandem with simplices. As noted by Robin Forman in *A User's Guide to Discrete Morse Theory*, "there is a close relationship between the topology of a smooth manifold M and the critical points of a smooth function f on M" [5]. The reader may consider, for example, how a rational map of degree 2 on S^2 (or equivalently the Riemannian sphere $\hat{\mathbb{C}}$) has only 2 critical points (as discussed by [12]). Forman himself considered Morse theory as a distant generalization of the extreme value theorem, since a compact manifold must always provide a maximum and minimum to a continuous function. In general, discrete Morse functions provide a powerful analogy to smooth Morse functions, in that they extract the same topological data without the need for stable and unstable manifolds or compactifications.

Throughout the remainder of this thesis, we will use $\mathcal{P}(A)$ to denote the power set of a set A. With that said, we may now formally define a (finite) simplicial complex, using the notation from [8]:

Definition 1.1 (Simplicial Complex)**.** *Given a finite set V (called the set of vertices) and $K \subseteq \mathcal{P}(V) \backslash \{\emptyset\}$, we say that K is a simplicial complex if, for all $\sigma \in K$, we have that $\emptyset \neq \tau \subseteq \sigma$ implies $\tau \in K$.*

We call the elements of K the faces of K, and define $\dim \sigma := \#\sigma - 1$ for all $\sigma \in K$. Furthermore, we define the dimension of K to be the maximum dimension of its faces; i.e.

$$\dim K := \max\{\dim \sigma \mid \sigma \in K\}$$

We write $\sigma \prec \tau$ if $\sigma \subset \tau$ and $\dim \sigma = \dim \tau - 1$.

From this definition, it should be clear that vertices are simply the 0-dimensional singleton elements of V while edges are the 1-dimensional sets connecting a pair of vertices. It is worth noting that an arbitrary simplicial complex need not be finite — however, the theorems and results of this thesis do not apply to such simplicial complexes and henceforth we assume all of our simplicial complexes are finite.

Since much of what we compute in this thesis will be dependent on some dimension $0 \leq p \leq \dim K$ (where K is our simplicial complex), we also wish to introduce the notion of face sets:

Definition 1.2. *If K is a simplicial complex and $0 \leq p \leq \dim K$, we refer to K_p as the set* *of all p-faces of K; that is:*

$$K_p = \{\sigma \in K \mid \dim \sigma = p\}$$

This will come into use later on, particularly when we want to find the most accurate bounds on computational complexity.

Before addressing discrete Morse functions, it is useful to first describe the role of Morse functions for smooth manifolds. In the smooth category, a Morse function is a smooth function whose values represent the "height" of such a manifold. Consider, for example, the function $h(x, y, z) = z$ on the sphere S^2. If we look at the sphere with the z-axis aligned vertically, then h gives us the height of any point on the sphere.

For an arbitrary smooth manifold, we anticipate that a height function should assign lower values to minima and higher values to maxima. The motivation for a discrete Morse function is to relay similar information on a simplicial or CW-complex where such concepts are less obvious. Specifically,

Definition 1.3 (Discrete Morse Function)**.** *Let K be a simplicial complex. A function $f : K \to \mathbb{R}$ is said to be a discrete Morse function if, for every $0 \le p \le \dim K$ and $\beta \in K_p$, f satisfies the following:*

(a) $\#\{\beta \prec \gamma \mid \gamma \in K_{p+1}, f(\beta) \ge f(\gamma)\} \le 1$

(b) $\#\{\alpha \prec \beta \mid \alpha \in K_{p-1}, f(\beta) \le f(\alpha)\} \le 1$

In simpler terms, a discrete Morse function is a function in which any given face cannot have more than one face with larger "height" value or more than one coface with smaller "height" value. We will next see that these two conditions cannot occur simultaneously for any given face $\alpha \in K$.

Lemma 1.4. *Given a simplicial complex K and discrete Morse function $f : K \to \mathbb{R}$, there cannot be both a coface of smaller value and a face of higher value for any given simplex $\beta \in K$.*

Proof. The proof is by contradiction. Let $\dim \beta = p$, and assume to the contrary that there exists a coface $\gamma \in K_{p+1}$ with $f(\gamma) \le f(\beta)$ as well as a face $\alpha \in K_{p-1}$ with $f(\alpha) \ge f(\beta)$.

By definition of a discrete Morse function, for any $\beta' \prec \gamma$ with $\beta' \neq \beta$, we cannot have both $f(\gamma) \leq f(\beta)$ and $f(\gamma) \leq f(\beta')$ so it must be the case that $f(\beta') < f(\gamma)$ and thus $f(\beta') < f(\beta)$.

Now clearly $K_{-1} = \emptyset$ since a simplex is defined on $\mathcal{P}(A)\backslash\{\emptyset\}$, so we must have that $p > 0$. In fact, we may assume without loss of generality that $p = 1$ since higher dimensional simplices are much finer in terms of edges and faces, and thus satisfy the following assumptions to a fuller extent. The only assertion left to show is that there exists some other $\beta' \prec \gamma$ with $\alpha \prec \beta'$. Let $\alpha = \{v_1\}$, $\beta = \{v_1, v_2\}$, and lastly $\gamma = \{v_1, v_2, v_3\}$. Then we may clearly choose $\beta' = \{v_1, v_3\}$ to satisfy the requirements above (a similar construction may be used in higher dimensions, as it simply builds off the definition of a simplicial complex).

By definition of a discrete Morse function, we cannot have both $f(\alpha) \geq f(\beta)$ and $f(\alpha) \geq f(\beta')$ so it must be the case that $f(\alpha) < f(\beta')$ and thus $f(\beta) < f(\beta')$. However, this is a contradiction since we proved $f(\beta') < f(\beta)$, so we must have that only one of the two conditions hold. $\square$

One thing to take away from this lemma is that there is a rigorous bound on how much a face may deviate from its neighboring faces or cofaces in terms of a discrete Morse function. However, that says nothing about what happens when both sets in conditions (a) and (b) in the definition above are empty. Intuitively this would mean that our function increases along higher-dimensional cofaces and decreases along its lower-dimensional faces. This gives us a natural analogy to gradient paths in the context of of smooth manifolds - for this reason, we give the following definition according to [4]:

Definition 1.5 (Critical Face)**.** *Let K be a simplicial complex, let $f : K \to \mathbb{R}$ be a discrete Morse function, and let $\beta \in K_p$. We say β is a critical face if the following two conditions hold:*

(a) $\#\{\gamma \succ \beta \mid \gamma \in K_{p+1}, f(\beta) \geq f(\gamma)\} = 0$

(b) $\#\{\alpha \prec \beta \mid \alpha \in K_{p-1}, f(\beta) \leq f(\alpha)\} = 0$

To give a physical interpretation, [8] notes that critical faces of dimension p correspond to critical points of rank p for smooth Morse functions. In terms of [11], these are exactly the p-handles of a smooth manifold.

Example 1.6. *Consider a discrete Morse function $h : K \to \mathbb{R}$ on a simplicial complex K where K is the 1-skeleton of Δ_3. We represent the values of h in Figure 1.1, where bold-faced numbers correspond to edges. Note that all cofaces of $h^{-1}(0)$ have a greater value,*

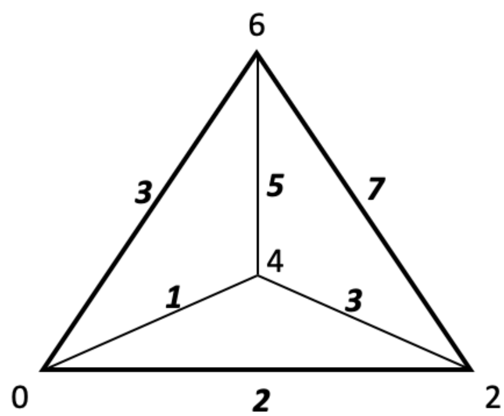


Figure 1.1: A discrete Morse function with a critical vertex $h^{-1}(0)$ and critical edge $h^{-1}(7)$

thus making $h^{-1}(0)$ a critical vertex. Similarly, all faces of $h^{-1}(7)$ have a lesser value, thus making $h^{-1}(7)$ a critical edge.

1.3 Hasse Diagrams and Discrete Vector Fields

At this point, there are numerous directions one could go in order to continue the basics of discrete Morse theory. Some authors such as [10] pursue simplicial collapses (which inevitably lead to homotopy equivalence), while others such as [5] pursue the discrete analogues of level sets and the Morse inequalities. However, we will pursue the order of [8] much more closely. At the beginning of this section, it may be unclear how some of the concepts are related to discrete Morse theory — nonetheless, we will quickly relate several of the concepts and show in the next section how they are applied to homology.

First and foremost, we will introduce the concept of a Hasse diagram in order to relay as much information about a simplicial complex as possible in terms of one dimension. Specifically:

Definition 1.7 (Hasse Diagram)**.** *Let K be a simplicial complex. The Hasse diagram $\mathcal{H}_K$ of K is the directed graph whose vertices represent the faces (of any dimension) of K. If $\alpha \prec \beta$ then there is a directed edge in $\mathcal{H}_K$ going from α to β.*

Though the Hasse diagram may seem like an unnecessary abstraction, it is actually essential when it comes to the theory of discrete vector fields. We give an example of a Hasse diagram below:

Example 1.8. *Consider the simplicial complex K given by the 2-skeleton of Δ_3. There are exactly four vertices, six edges, and four 2-faces (which we may simply refer to as faces). Since Δ_3 is a complete simplicial complex (i.e. no faces can be fit inside the existing*

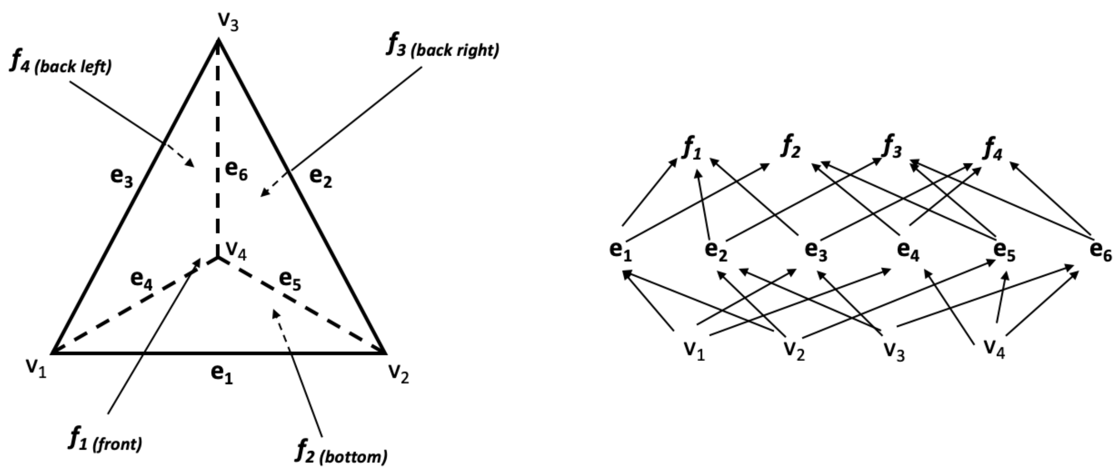


Figure 1.2: A visualization of Δ_2 along with $\mathcal{H}_{\Delta_2}$

vertices), the Hasse diagram $\mathcal{H}_K$ is clearly much finer than the Hasse diagram $\mathcal{H}_{K'}$ for any other simplicial complex K' with four vertices.

Now that we have established the definition of a Hasse diagram, we wish to give it some structure in order to better find applications on discrete vector fields.

Definition 1.9 (Matching)**.** *Let G be a graph. A set of edges $\mathcal{M} \subset G$ is called a matching if every pair of edges is non-adjacent (that is, they do not share a common vertex). A matching is said to be maximal if adding an edge not in $\mathcal{M}$ would make $\mathcal{M}$ no longer a matching.*

We now turn our focus to discrete vector fields. In the smooth case, a vector field indicates the direction that a point mass would tend towards when in a given position. Since we are no longer talking about "position" as a continuum, but instead as a set of states, we need to adjust our definitions.

Definition 1.10 (Discrete Vector Field)**.** *Let K be a simplicial complex. A discrete vector field V on K is a collection of disjoint pairs $\{\alpha \prec \beta\}$ (i.e. such that each $\sigma \in K$ is in at most one pair).*

It may not be immediately clear how a discrete vector field has anything to do with the previous two definitions. However, for each element in a discrete vector field V there must exist faces σ and τ with $\sigma \prec \tau$ and thus a directed edge in the Hasse diagram $\mathcal{H}_K$ going from the vertex representing σ to the vertex representing τ. In addition, since the pairs in V are disjoint, it must be the case that the edges in $\mathcal{H}_K$ corresponding to the pairs in V are non-adjacent (as otherwise adjacent pairs would imply two faces share the same coface or face). Therefore, we come upon the following lemma:

Lemma 1.11. *Let K be a simplicial complex. There is a one-to-one correspondence between discrete vector fields on K and matchings on $\mathcal{H}_K$;*

$$\{V \mid V \text{ is a discrete vector field}\} \longleftrightarrow \{\mathcal{M} \subset \mathcal{H}_K \mid \mathcal{M} \text{ is a matching}\}.$$

The proof will be omitted since the previous paragraph summarizes the logic without going into the full formality. From this point forward, we will let $\mathcal{M}_V$ denote the representation of a discrete vector field V as a matching in $\mathcal{H}_K$. When representing a discrete vector field on a Hasse diagram through some matching $\mathcal{M}_V$, we reverse the orientation of the edges in $\mathcal{M}_V$ (this will allow us to extend the notions of paths on a simplicial complex to paths on a Hasse diagram and vice versa).

In the case of vector fields on smooth manifolds, critical points represent positions where a point mass is in equilibrium. We wish to somehow extend this notion to discrete vector fields:

Definition 1.12. *Let K be a simplicial complex and V be a discrete vector field on K. A* *face $\beta \in K$ is said to be a critical face of V if there do not exist $\alpha \prec \beta$ or $\gamma \succ \beta$ such that $\{\alpha \prec \beta\} \in V$ or $\{\gamma \succ \beta\} \in V$.*

In terms of our Hasse diagrams, this extends to the following notation:

Corollary 1.13. *Let K be a simplicial complex and V a discrete vector field on K. A face β is a critical face of V if and only if the vertex corresponding to β in $\mathcal{H}_K$, call it v_β, has no adjacent edge in $\mathcal{M}_V$.*

At this point, we have not added any structure specific to Morse theory — Hasse diagrams, matchings, and discrete vector fields all exist in the realm of graph theory. However, it is worth noting that we have given competing definitions for what a critical face is — one in the context of discrete Morse functions and one in the context of discrete vector fields. Therefore, it is natural to introduce a bridge between the two subjects in order to reconcile these competing definitions:

Definition 1.14 (Discrete Gradient Vector Field)**.** *Let K be a simplicial complex and $f : K \to \mathbb{R}$ a discrete Morse function on K. We define the discrete gradient vector field $-\nabla f$ of f on K as follows: for every $\beta \in K$, if there exists $\alpha \prec \beta$ with $f(\alpha) \geq f(\beta)$ then $\{\alpha \prec \beta\} \in -\nabla f$.*

As the reader may observe, β is a critical face of f if and only if β is a critical face of $-\nabla f$. More importantly, the discrete gradient vector field allows either structure to be used in place of the other. Consequently, this provides a means of endowing a simplicial complex with a discrete dynamical system using only point-set data. Our next goal is to identify an analogy for discrete gradient vector fields in the Hasse diagram.

Now that we have our discrete gradient vector field to bridge discrete vector fields and discrete Morse functions, we are able to construct an analogue to cycles in the context of discrete Morse functions. The following definition is adapted from [10]:

Definition 1.15 (Discrete Path)**.** *Let K be a simplicial complex and V a discrete vector field on K. We say that a discrete path is a sequence*

$$\sigma_0 \prec \tau_0 \succ \sigma_1 \prec \cdots \succ \sigma_{n-1} \prec \tau_{n-1} \succ \sigma_n \prec \tau_n$$

such that:

(i) $\{\sigma_i \prec \tau_i\} \in V$ *for all* $0 \leq i \leq n$.

(ii) $\sigma_i \neq \sigma_{i+1}$ *for all* $0 \leq i < n$.

(iii) $\tau_i \neq \tau_{i+1}$ *for all* $i \leq i < n$.

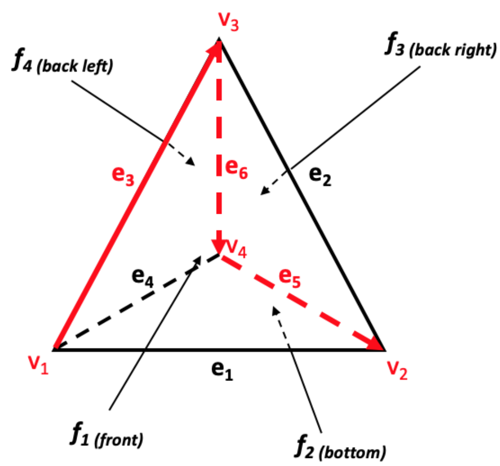
Moreover, σ_0 *or* τ_n *may be omitted.*

It is easy to see that a discrete path on a simplicial complex K induces a path on the Hasse diagram $\mathcal{H}_K$:

Example 1.16. *Consider the simplicial complex* K *given by the* 2*-skeleton of* Δ_3*. Furthermore, suppose we have the discrete path*

$$\gamma = v_1 \prec e_3 \succ v_3 \prec e_6 \succ v_4 \prec e_5 \succ v_2$$

on K*. The path* γ *can be expressed in Figure 1.3:*



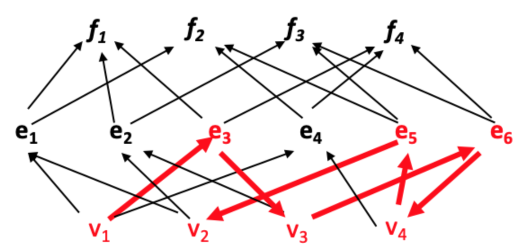


Figure 1.3: A visualization of Δ_3 along with $\mathcal{H}_{\Delta_3}$

Note that we reverse the orientation of some arrows in the Hasse diagram to make the orientation of the path more clear to the reader.

However, discrete paths do not yet fully incorporate the structure of discrete Morse functions — simply discrete vector fields. The following Lemma is adapted from [10]:

Lemma 1.17. *Let K be a simplicial complex and f be a discrete Morse function on K. A sequence*

$$\sigma_0 \prec \tau_0 \succ \sigma_1 \prec \cdots \succ \sigma_{n-1} \prec \tau_{n-1} \succ \sigma_n$$

is a discrete path for the gradient vector field $-\nabla f$ if and only if

$$f(\sigma_0) \geq f(\tau_0) > f(\sigma_1) \geq \cdots \geq f(\tau_{n-1}) > f(\sigma_n)$$

Proof. For the forward direction, we assume $\sigma_0 \prec \tau_0 \succ \sigma_1 \prec \cdots \succ \sigma_{n-1} \prec \tau_{n-1} \succ \sigma_n$ is a discrete path. For each $0 \leq i \leq n$ we have $\{\sigma_i \prec \tau_i\} \in -\nabla f$ so by definition $f(\sigma_i) \geq f(\tau_i)$. By definition of a discrete Morse function, τ_i cannot have more than one face with greater f-value than it. In other words for $0 \leq i < n$, $f(\sigma_{i+1}) \geq f(\tau_i)$ would be a contradiction since $\tau_i \succ \sigma_{i+1}$. Therefore, we must have $f(\tau_i) > f(\sigma_{i+1})$.

For the reverse direction, suppose $\sigma_0 \prec \tau_0 \succ \sigma_1 \prec \cdots \succ \sigma_{n-1} \prec \tau_{n-1} \succ \sigma_n$ satisfies

$$f(\sigma_0) \geq f(\tau_0) > f(\sigma_1) \geq \cdots \geq f(\tau_{n-1}) > f(\sigma_n).$$

For each $0 \leq i \leq n$, we have that $\sigma_i \prec \tau_i$ and $f(\sigma_i) \geq f(\tau_i)$ so by definition $\{\sigma_i \prec \tau_i\} \in -\nabla f$. In addition, we have that $f(\sigma_i) > f(\sigma_{i+1})$ for $0 \leq i < n$, so it must be the case that $\sigma_i \neq \sigma_{i+1}$ since f is well-defined. A similar argument shows that $\tau_i \neq \tau_{i+1}$ for $0 \leq i < n$. Therefore, the sequence is a discrete path. □

Definition 1.18. *Let K be a simplicial complex and V a discrete vector field on K. We call a discrete path*

$$\sigma_0 \prec \tau_0 \succ \sigma_1 \prec \cdots \succ \sigma_{n-1} \prec \tau_{n-1} \succ \sigma_n$$

a cycle if $\sigma_0 = \sigma_n$. Furthermore, we say that K is acyclic if there does not exist a cycle on K.

Note that this definition of an acyclic complex clearly extends to directed graphs since we need only consider the Hasse diagram of our complex. We now consider the following lemma, as posed by [10]:

Lemma 1.19. *A directed graph G is acyclic if and only if there exists a function $f : \mathrm{Vert}(G) \to \mathbb{R}$ that is strictly decreasing along each directed path.*

The proof for the above lemma will be omitted since it goes outside the domain of this thesis. One direction is fairly clear since a function that is strictly decreasing along each path would lead to an inequality $f(v) < f(v)$ if v is in some cycle. The interested reader may refer to [2] to find a proof for why the converse holds.

Recall that when representing a discrete vector field V on a Hasse diagram $\mathcal{H}_K$ through some matching $\mathcal{M}_V$, we reverse the orientation of the edges in $\mathcal{M}_V$. However, when used with Lemma 1.17, the Hasse diagram gives us a powerful redefinition of discrete gradient vector fields.

Corollary 1.20. *Let K be a simplicial complex and V be a discrete vector field on K. Then V is a discrete gradient vector field of a Morse function f if and only if the Hasse diagram $\mathcal{H}_K$ with orientation modified by $\mathcal{M}_V$ is a directed acyclic graph.*

Proof. For the forward direction, suppose that $V = -\nabla f$ for some discrete Morse function $f : K \to \mathbb{R}$. Moreover, suppose to the contrary that there exists some cycle

$$\sigma_0 \prec \tau_0 \succ \sigma_1 \prec \dots \tau_n \succ \sigma_0.$$

By Lemma 1.17, this implies that

$$f(\sigma_0) \geq f(\tau_0) > f(\sigma_1) \geq \cdots \geq f(\tau_n) > f(\sigma_0),$$

a clear contradiction. Conversely, suppose that $\mathcal{H}_K$ modified by $\mathcal{M}_V$ is a directed acyclic graph. By Lemma 1.19, we can find a function $f : \mathbf{Vert}(\mathcal{H}_K) \to \mathbb{R}$ that is strictly decreasing along each directed path. Note however, that if $\{\alpha \prec \beta\} \in V$, then the directed edge from v_β to v_α is flipped so that there is a path of length 1 from v_α to v_β. By definition, f must be decreasing along this path so $f(v_\alpha) \geq f(v_\beta)$. Suppose we define $f^* : K \to \mathbb{R}$ by $f^*(\sigma) := f(v_\sigma)$ (where v_σ represents the vertex corresponding to σ on $\mathcal{H}_K$). Then we have $\{\alpha \prec \beta\} \in V$ if and only if $f^*(\alpha) \geq f^*(\beta)$, which is precisely the definition of $-\nabla f^*$. $\square$

As we will see in the next chapter, a lack of cycles allows us to define an analogue of Morse homology on a Hasse diagram whenever a Morse function is present.

1.4 Morse Homology

By this point, the reader should understand that discrete Morse functions give meaningful insight to both the local structure and global structure of simplicial complexes. Since simplicial complexes (and CW complexes) are often used in conjunction with smooth manifolds, this allows us to observe many topological properties and dynamics of smooth manifolds. However, one crucial property which has yet to be covered is the homology of a topological space.

Since we are only discussing simplicial complexes in this paper, a trivial solution would be to present simplicial homology; however, this would give us little carryover between discrete Morse theory and homology. Furthermore, we will see that given the right discrete vector field, Morse homology is far more efficient to compute than simplicial homology. There is actually an enormous amount of theory building up the connection between discrete vector fields, discrete gradient flows, boundary maps, and homology — much of this theory is explained in [10, 4]. For the sake of brevity, this background will be skipped and we will take an approach much closer to [5] (it is worth noting, however, that [10, 4] show how the background theory implies the approach of [5]).

At this point, we will assume the reader has no familiarity with simplicial homology. Thus, before establishing any equivalence classes on chain complexes, it is natural to first define a chain on a simplicial complex:

Definition 1.21 (p-Chain)**.** *Let K be an oriented simplicial complex and $0 \leq p \leq \dim K$. Assuming $K_p = \{\sigma_1, \ldots, \sigma_n\}$, we define a p-chain to be a linear combination*

$$a_1\sigma_1 + \cdots + a_n\sigma_n$$

where each $a_i \in \mathbb{Z}$.

This allows us to define a group structure on the set of chains over some simplex K

through usual addition. In particular, we expect that the group of p-chains should have a basis K_p with induced orientation from K since each p-chain is a linear combination of elements over K_p.

Definition 1.22 (Group of p-Chains)**.** *Let K be a simplicial complex and $p \geq 0$ with $p \in \mathbb{Z}$. We define the group of p-chains, denoted $C_p(K;\mathbb{Z})$, to be the free abelian group with the binary operation of standard addition and basis K_p. Thus, elements of $C_p(K;\mathbb{Z})$ are linear combinations of elements in K_p with coefficients in $\mathbb{Z}$.*
We define $C_*(K;\mathbb{Z})$ to be the graded structure

$$C_*(K;\mathbb{Z}) = \bigoplus_{p=0}^{\dim K} C_p(K;\mathbb{Z})$$

This is the point at which simplicial homology and Morse homology begin to diverge. If we were to pursue simplicial homology, we would give a formal definition of the boundary of a simplex in order to establish our boundary map for the chain complex $C_*(K;\mathbb{Z})$. Our goal is to still establish some sort of boundary map for a chain complex; however, our boundary map will differ slightly from the boundary of a simplex and admit a much smaller group of p-chains.

Definition 1.23 (Morse Group)**.** *Let K be a simplicial complex, let V be a discrete gradient vector field on K, and let $p \geq 0$ with $p \in \mathbb{Z}$. We define the Morse group (of degree p), denoted $\mathcal{M}_p \subset C_p(K;\mathbb{Z})$, to be the free abelian group with the binary operation of standard addition and basis $\{\sigma \in K_p \mid \sigma \text{ is critical}\}$. Thus, elements of $\mathcal{M}_p$ are linear combinations of critical faces of degree p with coefficients in $\mathbb{Z}$.*
We define $\mathcal{M}_*$ to be the graded structure

$$\mathcal{M}_* = \bigoplus_{p=0}^{\dim K} \mathcal{M}_p$$

If we define $m_p = \#\{\sigma \in K_p \mid \sigma \text{ is critical}\}$ (which is known as the *Morse Number* is most literature) we have the following corollary:

Corollary 1.24. *Let K be a simplicial complex and $p \geq 0$ with $p \in \mathbb{Z}$. Then we have that*

$\mathcal{M}_p \simeq \bigoplus_{n=1}^{m_p} \mathbb{Z}$.

Proof. Since $\mathcal{M}_p$ is defined to be the free abelian group with basis over $\{\sigma \in K_p \mid \sigma \text{ is critical}\}$, there are exactly $n = m_p$ degrees of freedom. Since the coefficients in the linear combination are integers, there is a clear canonical isomorphism given by

$$\sum_{i=1}^{n} a_i \sigma_i \mapsto (a_1, \ldots, a_n)$$

The direct sum follows from linearity. □

From now on, assume that K is an *oriented* simplicial complex. Let $\tilde{\partial}_p : \mathcal{M}_p \to \mathcal{M}_{p-1}$ denote the Morse-theoretical boundary map we wish to construct and $\partial_p : C_p(K; \mathbb{Z}) \to C_{p-1}(K; \mathbb{Z})$ denote the boundary map of simplicial homology. In particular, recall that we define a p-face σ over a set of vertices V to be a subset of V of cardinality $p + 1$. Thus, we define the boundary of σ, denoted $\partial_p(\sigma)$, to be a linear combination of all subsets $\tau \subset \sigma$ of cardinality p (i.e. $\tau \prec \sigma$). However, this linear combination must take the orientation of our simplex into account. Suppose σ is the ordered set $[v_1, \ldots, v_p]$ and let $\tau_i = \sigma \backslash \{v_i\}$. Then we may formally define

$$\partial_p(\sigma) = \sum_{i=1}^{p} (-1)^i \tau_i.$$

It will soon become clear why we need ∂_p in order to define $\tilde{\partial}_p$ - however, the fact that $\tilde{\partial}_p$ is only a map on $\mathcal{M}_p$ should be enough to convince the reader that the two are quite distinct.

A second distinction between simplicial homology and Morse homology is that the orientation of a simplicial complex K alone does not suffice for Morse homology. Given the additional structure of a Morse function — which, in turn, gives rise to our discrete gradient vector field — we want to determine whether a particle moving initially in the positive orientation will continue to move in the positive orientation as it travels along the gradient vector field (i.e. a path requiring minimal energy). It should be relatively intuitive how this allows us to distinguish the geometry of surfaces: given two isomorphic simplicial complexes with differing Morse functions, if a particle moving along a fixed path switches orientations on one complex and not the other, the induced geometries are clearly different.

We now attempt to formally define the concept in the above paragraph. Using the notation of [10, 4], we define an inner product $\langle\, , \rangle$ on $C_*(K;\mathbb{Z})$ by setting cells of the same dimension to be orthonormal. However, suppose that $\sigma \in K_p$ with $\sigma \prec \tau$ and that σ and τ have conflicting orientations. Then we set

$$\langle \partial_{p+1}\tau, \sigma \rangle = -1.$$

That is, we define our inner product to account for orientation.

Using our inner product as described above, we define the multiplicity of a path γ (denoted $m(\gamma)$) as the following:

$$m(\gamma) = \begin{cases} \langle \partial\tau, \sigma \rangle & \gamma = \sigma \prec \tau \text{ or } \gamma = \tau \succ \sigma \\ \langle \partial\tau, \sigma_n \rangle m(\gamma') & \gamma = \gamma' \cdot \tau,\ \sigma_n \in K_p \text{ last face of } \gamma' \\ \langle \partial\tau_n, \sigma \rangle m(\gamma') & \gamma = \gamma' \cdot \sigma,\ \tau_n \in K_{p+1} \text{ last face of } \gamma' \end{cases}$$

We use the symbol $\cdot$ above to denote concatenation of paths. For certain cases, we may simply express $m(\gamma)$ in an iterative expansion. For example, if we know the path γ begins with a p-face and ends with a p-face, e.g.

$$\gamma = \sigma_0 \prec \tau_0 \succ \sigma_1 \prec \cdots \succ \sigma_n$$

then we may collapse the recursive definition above to

$$m(\gamma) = \prod_{i=0}^{n-1} \langle \partial_{p+1}\tau_i, \sigma_i \rangle \langle \partial_{p+1}\tau_i, \sigma_{i+1} \rangle$$

However, this expansion is not well-defined unless the path γ is non-trivial, begins on a p-face, and ends on a p-face.

Example 1.25. *Consider the following simplex K and discrete vector field over K, as adapted from [5].*

Note that only relevant simplices are labeled and given an orientation. Consider the

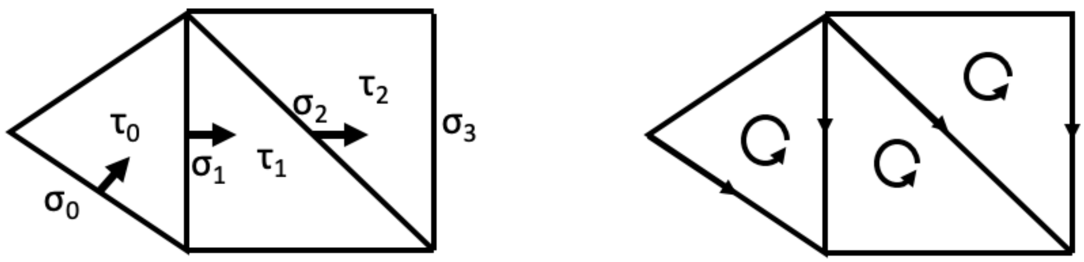


Figure 1.4: In the left diagram, the discrete vector field for K is given, whereas the right diagram shows the orientation for relevant simplices

discrete gradient path given by

$$\gamma = \sigma_0 \prec \tau_0 \succ \sigma_1 \prec \tau_1 \succ \sigma_2 \prec \tau_2 \succ \sigma_3$$

To simplify notation, let $\partial = \partial_2$. *From our definition above, we have that*

$$\begin{aligned}
m(\gamma) &= \prod_{i=0}^{n-1} \langle \partial_2 \tau_i, \sigma_i \rangle \langle \partial_2 \tau_i, \sigma_{i+1} \rangle \\
&= \langle \partial \tau_0, \sigma_0 \rangle \langle \partial \tau_0, \sigma_1 \rangle \langle \partial \tau_1, \sigma_1 \rangle \langle \partial \tau_1, \sigma_2 \rangle \langle \partial \tau_2, \sigma_2 \rangle \langle \partial \tau_2, \sigma_3 \rangle \\
&= (1)(-1)(1)(-1)(1)(-1) \\
&= -1
\end{aligned}$$

With the notion of accounting for orientation along a path via multiplicity, we are now able to formally define our boundary map $\tilde{\partial}_p : \mathcal{M}_p \to \mathcal{M}_{p-1}$. We follow the notation of [5] instead of [10] since the former provides a more intuitive explicit mapping.

Definition 1.26 (Morse Boundary Map)**.** *Let K be an oriented simplicial complex endowed with some discrete gradient vector field. Let $\{\tau_1, \ldots, \tau_n\}$ denote the basis for $\mathcal{M}_p$ (i.e. the critical faces of dimension p). We define the Morse boundary map $\tilde{\partial}_p : \mathcal{M}_p \to \mathcal{M}_{p-1}$ on the basis elements by*

$$\tilde{\partial}_p \tau_i = \sum_{\substack{\sigma \in K_{p-1} \\ \sigma \text{ critical}}} c_{\tau_i, \sigma} \sigma$$

where

$$c_{\tau_i,\sigma} = \sum_{\gamma \in \Gamma(\tau_i,\sigma)} m(\gamma).$$

Here we denote the set of all paths from τ_i to σ as $\Gamma(\tau_i, \sigma)$. We extend $\tilde{\partial}_p$ from $\{\tau_1, \ldots, \tau_n\}$ to $\mathcal{M}_p$ by defining it to be $\mathbb{Z}$-linear.

Note that $\tilde{\partial}_p(\tau)$ is defined to be a linear combination of *critical* cells — therefore, it is a well-defined map from $\mathcal{M}_p$ to $\mathcal{M}_{p-1}$.

We now begin our construction of the Morse homology groups. In general, we use the notation of [13]:

Definition 1.27 (Chain Complex). *A chain complex $\mathcal{C}$ is a sequence*

$$\ldots \to \mathcal{C}_{p+1} \xrightarrow{\partial_{p+1}} \mathcal{C}_p \xrightarrow{\partial_p} \mathcal{C}_{p-1} \to \ldots$$

of abelian groups $\mathcal{C}_i$ and homomorphishms ∂_i, indexed by the integers, such that $\partial_{i+1} \circ \partial_i = 0$ for all i.

We now present the following theorem, which will allow us to proceed in our construction of homology.

Theorem 1.28. *The sequence $\{(\mathcal{M}_i, \tilde{\partial}_i) : i \in \mathbb{N}\}$ given by*

$$... \xrightarrow{\tilde{\partial}_{i+1}} \mathcal{M}_i \xrightarrow{\tilde{\partial}_i} \mathcal{M}_{i-1} \xrightarrow{\tilde{\partial}_{i-1}} \ldots \xrightarrow{\tilde{\partial}_1} \mathcal{M}_0$$

is a chain complex.

A formal proof of the theorem above will be omitted since it requires several theorems proving the behavior of a helper function Φ. However, linearity in Definition 1.27 follows by definition of our Morse boundary map $\tilde{\partial}$ — thus, $\tilde{\partial}_i$ is a homomorphism. In particular, $\mathcal{M}_*$ is a $\mathbb{Z}$-graded algebra with components corresponding to dimensions of critical cells. Moreover, each $\mathcal{M}_p$ is a $\mathbb{Z}^{m_p}$ graded $\mathcal{M}_*$-module (by Corollary 1.24).

To show that $\tilde{\partial}_{p+1} \circ \tilde{\partial}_p = 0$, we follow the approach of [4]. First, we convert our discrete vector field V into a function $V : C_*(K; \mathbb{Z}) \to C_*(K; \mathbb{Z})$, defined by $V(\sigma) = \tau$ if $\{\sigma \prec \tau\} \in V$

and $V(\sigma) = 0$ otherwise. Next, we define a map $\Phi_p : C_p(K, \mathbb{Z}) \to C_p(K; \mathbb{Z})$ by:

$$\Phi_p(\sigma) = \sigma + \partial_p V(\sigma) + V(\partial_p \sigma)$$

for all $\sigma \in K_p$ (where ∂_p is the simplicial boundary map). This clearly extends to be linear by linearity of the simplicial boundary map ∂_p and V. If we let Φ^N denote the composition of Φ with itself N times, [4] proves the following two facts in Theorem 6.4(1) and Theorem 7.2, respectively:

Lemma 1.29. *If ∂_p is the simplicial boundary map, then $\Phi \circ \partial_p = \partial_p \circ \Phi$.*

Lemma 1.30. *For N large enough, $\Phi^N = \Phi^{N+1} = \cdots = \Phi^\infty$.*

Lemma 1.30 tells us that our map Φ^∞ is equal to the composition of Φ with itself a finite number of times. A direct corollary of Lemma 1.29 is that the finite composition of Φ with itself commutes with ∂_p — therefore, $\partial_p \circ \Phi^\infty = \Phi^\infty \circ \partial_p$. If we let $\tilde{\Phi}^\infty$ denote the restriction of Φ^∞ to the Morse group $\mathcal{M}_p$, then from Theorem 8.2 in [4] we have

$$\tilde{\partial}_p = \partial_p \circ \tilde{\Phi}^\infty = \tilde{\Phi}^\infty \circ \partial_p$$

Therefore, we may directly compute

$$\tilde{\partial}_p \circ \tilde{\partial}_{p+1} = \tilde{\Phi}^\infty \circ \partial_p \circ \partial_{p+1} \circ \tilde{\Phi}^\infty = \tilde{\Phi}^\infty \circ 0 \circ \tilde{\Phi}^\infty = 0$$

as desired.

The significance of the previous theorem is that it gives us the freedom to construct homology groups in the canonical manner, just as one would do for simplicial or persistent homology.

Definition 1.31 (Morse Homology)**.** *Let K be a simplicial complex endowed with a discrete gradient vector field, and let*

$$\ldots \xrightarrow{\tilde{\partial}_{i+1}} \mathcal{M}_i \xrightarrow{\tilde{\partial}_i} \mathcal{M}_{i-1} \xrightarrow{\tilde{\partial}_{i-1}} \ldots \xrightarrow{\tilde{\partial}_1} \mathcal{M}_0$$

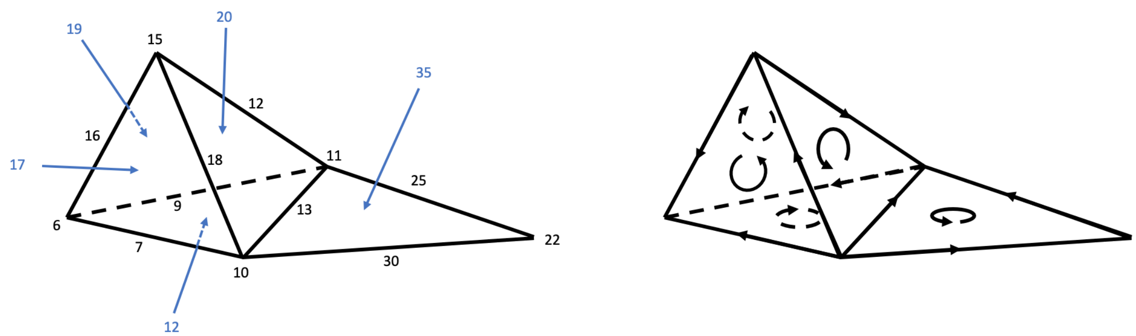
be the Morse chain complex associated to K. For each $i \in \mathbb{Z}$, we define the Morse homology group $\tilde{H}_i(K;\mathbb{Z})$ by

$$\tilde{H}_i(K;\mathbb{Z}) = \ker \tilde{\partial}_i / Im\, \tilde{\partial}_{i+1}$$

It is standard to let $\tilde{\partial}_0$ be the zero map so that $\tilde{H}_0(K;\mathbb{Z})$ is well-defined.

For the reader who wishes to make this notion concrete, we provide the following example:

Example 1.32. *Consider the 2-dimensional simplicial complex K and discrete Morse function $f : K \to \mathbb{R}$ given in the figures below. The second picture gives the orientation for the*



relevant simplices, while the first picture provides the values for f. To simplify notation, we let $\mathbf{u} = f^{-1}(6)$, $\mathbf{v} = f^{-1}(22)$, $\mathbf{a} = f^{-1}(16)$, $\mathbf{b} = f^{-1}(30)$, $\mathbf{c} = f^{-1}(25)$, $\mathbf{F_1} = f^{-1}(20)$, $\mathbf{F_2} = f^{-1}(19)$, and $\mathbf{F_3} = f^{-1}(35)$. The reader can check that the critical vertices are exactly those which are given some variable alias; that is,

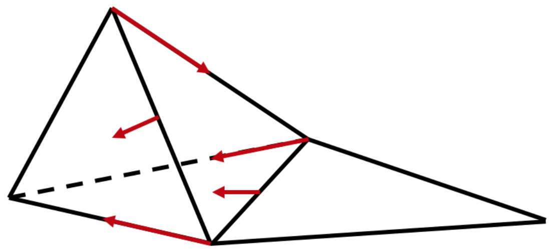
$$\mathcal{M}_0 = \{\mathbf{u}, \mathbf{v}\}$$

$$\mathcal{M}_1 = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$$

$$\mathcal{M}_2 = \{\mathbf{F_1}, \mathbf{F_2}, \mathbf{F_3}\}$$

We may represent the discrete gradient vector field of f as below:
where a red arrow emanating from some face α with arrow ending in a face β denotes $\{\alpha \prec \beta\} \in -\nabla f$.

In this example, we compute the second homology group of K denoted by $\tilde{H}_2(K;\mathbb{Z})$.



First, we wish to find the values of $\tilde{\partial}_2$. *Since there are only three critical faces, we calculate each explicitly. For* $\mathbf{F_1}$ *we have:*

$$\tilde{\partial}_2(\mathbf{F_1}) = c_{\mathbf{F_1},\mathbf{a}}\mathbf{a} + c_{\mathbf{F_1},\mathbf{b}}\mathbf{b} + c_{\mathbf{F_1},\mathbf{c}}\mathbf{c}$$

Looking at the discrete gradient vector field, there is exactly one discrete gradient path going from $\mathbf{F_1}$ *to* $\mathbf{a}$*, given by*

$$\gamma = \mathbf{F_1} \succ f^{-1}(18) \prec f^{-1}(17) \succ \mathbf{a}$$

Thus,

$$c_{\mathbf{F_1},\mathbf{a}} = m(\gamma) = \langle \partial\mathbf{F_1}, f^{-1}(18)\rangle\langle \partial f^{-1}(17), f^{-1}(18)\rangle\langle \partial f^{-1}(17), \mathbf{a}\rangle = -1$$

(note that the orientation on the edge $f^{-1}(18)$ *induced by* $\mathbf{F_1}$ *does not agree with the orientation on* $f^{-1}(18)$*). Finally, we note that there are no discrete paths going from* $\mathbf{F_1}$ *to* $\mathbf{b}$ *or* $\mathbf{c}$*, so* $c_{\mathbf{F_1},\mathbf{b}} = c_{\mathbf{F_1},\mathbf{c}} = 0$. *Thus,*

$$\tilde{\partial}_2(\mathbf{F_1}) = -\mathbf{a}$$

Next, we consider $\mathbf{F_2}$. *Since* $\mathbf{a}$ *is a maximal face of* $\mathbf{F_2}$*, we get that the only path from* $\mathbf{F_2}$ *to* $\mathbf{a}$ *is the trivial path* $\mathbf{F_2} \succ \mathbf{a}$ *and thus* $c_{\mathbf{F_2},\mathbf{a}} = \langle \partial\mathbf{F_2}, \mathbf{a}\rangle = -1$. *Again, we can see that there are no discrete paths from* $\mathbf{F_2}$ *to either* $\mathbf{b}$ *or* $\mathbf{c}$*, so* $c_{\mathbf{F_2},\mathbf{b}} = c_{\mathbf{F_2},\mathbf{c}} = 0$. *Thus,*

$$\tilde{\partial}_2(\mathbf{F_2}) = -\mathbf{a}$$

Lastly, we consider $\mathbf{F_3}$. *Both* $\mathbf{b}$ *and* $\mathbf{c}$ *are maximal faces of* $\mathbf{F_3}$*, so we must have*

$$c_{\mathbf{F_3},\mathbf{b}} = \langle \partial\mathbf{F_3}, \mathbf{b}\rangle = 1 = \langle \partial\mathbf{F_3}, \mathbf{c}\rangle = c_{\mathbf{F_3},\mathbf{c}}$$

However, there are no discrete paths from $\mathbf{F_3}$ *to* $\mathbf{a}$ *and thus we are left with*

$$\tilde{\partial}_2(\mathbf{F_3}) = \mathbf{b} + \mathbf{c}$$

In particular, since $\tilde{\partial}_2(\mathbf{F_1}) = \tilde{\partial}_2(\mathbf{F_2}) = -\mathbf{a}$, *we have that* $\mathbf{F_1} - \mathbf{F_2} \in \ker \tilde{\partial}_2$. *Since* $\tilde{\partial}_3 \equiv 0$, *we have that*

$$\tilde{H}_2(K;\mathbb{Z}) = \operatorname{span}\{\mathbf{F_1} - \mathbf{F}_2\} \cong \mathbb{Z}$$

In particular, the reader should note that the simplex K *is homotopy equivalent to the sphere with a disk attached. Since the singular homology of a sphere with a disk attached is simply the singular homology of a sphere by homotopy equivalence, the reader can check that*

$$\tilde{H}_1(K;\mathbb{Z}) \cong H_1^{sing}(S^2;\mathbb{Z}) \cong 0$$

and

$$\tilde{H}_0(K;\mathbb{Z}) \cong H_0^{sing}(S^2;\mathbb{Z}) \cong \mathbb{Z}$$

Though Morse homology is constructed in a different manner than simplicial homology, much of the homological algebra remains the same. In fact, we provide the following fact which will be referenced later in Chapter 3:

Proposition 1. *Let* K *be a simplicial complex. If* K *is equipped with the discrete gradient vector field* $V = \emptyset$ *(which can be given by* $f(\sigma) = \dim \sigma$*), the Morse boundary map is simply*

$$\tilde{\partial}_p = \partial_p$$

where ∂_p *is the standard boundary of a simplex. Consequently,*

$$\tilde{H}_p(K;\mathbb{Z}) = H_p(K;\mathbb{Z})$$

where $H_i(K;\mathbb{Z})$ *denotes the simplicial homology of* K.

Proof. Since $V = \emptyset$, we must also have that our partial matching $\mathcal{M}_V$ on the Hasse diagram is also empty, which implies that every $\sigma \in K$ is critical. This implies that for every

$0 \leq p \leq \dim K$, $\mathcal{M}_p = K_p$. If we fix some $\tau \in K_p$, then the only paths emanating from τ are the paths of length 1 going to maximal faces $\sigma \prec \tau$. Each path clearly has multiplicity $m(\gamma) = \langle \partial_p \tau, \sigma \rangle$, which gives us

$$\begin{aligned}
\tilde{\partial}_p(\tau) &= \sum_{\sigma \in \mathcal{M}_{p-1}} c_{\tau,\sigma} \sigma \\
&= \sum_{\sigma \prec \tau} c_{\tau,\sigma} \sigma \\
&= \sum_{\sigma \prec \tau} \langle \partial_p \tau, \sigma \rangle \sigma \\
&= \partial_p(\tau)
\end{aligned}$$

By definition, this implies

$$\tilde{H}_p(K; \mathbb{Z}) = \ker \tilde{\partial}_p / \operatorname{Im} \tilde{\partial}_{p+1} = \ker \partial_p / \operatorname{Im} \partial_{p+1} = H_p(K; \mathbb{Z})$$

□

Our last goal for this chapter is to give some exposition on the common theorems used to compute homology by means of methods derived from the Fundamental Theorem of Finitely Generated Abelian Groups.

To begin, we cite the following theorem, adapted from [13]:

Theorem 1.33 (The Fundamental Theorem of Finitely Generated Abelian Groups)**.** *Let G be a finitely generated Abelian group. Then*

$$G \simeq \mathbb{Z}^{\beta_i} \oplus (\mathbb{Z}/t_1\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/t_m\mathbb{Z})$$

where $\beta_i > 0$ and the $t_i > 1$ satisfy $t_i | t_{i+1}$ for all $1 \leq i \leq m-1$. The cyclic groups $\mathbb{Z}/t_i\mathbb{Z}$ are known as the torsion subgroups and the t_i are known as the torsion coefficients.

From Definition 1.23 of the Morse group, we know that $\mathcal{M}_p$ is a free Abelian group for all $0 \leq p \leq \dim K$ (where K is our simplicial complex). We now consider a second theorem adapted from [13]:

Theorem 1.34. *Let G and G' be free Abelian groups with bases $a_1, \dots, a_n$ and $b_1, \dots, b_m$, respectively. Suppose $f : G \to G'$ is a homomorphism with*

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

Then there are bases for G and G' such that, relative to the bases, $M = \{\lambda_{ij}\}$ has the form

$$M = \begin{pmatrix} \mathbf{T} & 0 \\ 0 & 0 \end{pmatrix}, \qquad \mathbf{T} = \begin{pmatrix} t_1 & & 0 \\ & \ddots & \\ 0 & & t_r \end{pmatrix}$$

where $t_i \geq 1$ and $t_1 \mid t_2 \mid \cdots \mid t_r$.

In the theorem above, the matrix M is said to be in Smith normal form (SNF). Moreover, we know that the number r above is $r = \dim(f(G))$ (where $f(G)$ is the free Abelian group under the image of our homomorphism f) since the dimension of M is invariant under linear transformations. Lastly, by Theorem 11.5 in [13] all such $t_i > 1$ are precisely the torsion coefficients of the quotient group $G' \backslash f(G)$.

Since our Morse boundary map $\tilde{\partial}_p : \mathcal{M}_p \to \mathcal{M}_{p-1}$ is defined to be linear, the theorems above apply quite naturally. As before, we consider some oriented simplicial complex K and fixed dimension $0 \leq p \leq \dim K$. For simplicity, we assume $0 < p < \dim K$ and allow the reader to consider the edge cases of $p = 0$ and $p = \dim K$. Let $\{\mu_1, \dots, \mu_{m_p - 1}\}$ be the basis of the free group $\mathcal{M}_{p-1}$, $\{\sigma_1, \dots, \sigma_{m_p}\}$ be the basis of $\mathcal{M}_p$, and $\{\tau_1, \dots, \tau_{m_{p-1}}\}$ be the basis of $\mathcal{M}_{p-1}$. Then we have

$$\tilde{\partial}_{p+1}(\tau_j) = \sum_{i=1}^{m_p} a_{ij} \sigma_i$$
$$\tilde{\partial}_p(\sigma_j) = \sum_{i=1}^{m_{p-1}} b_{ij} \mu_i$$

so that we can find matrices D_p, D_{p+1} with $D_{p+1} = \{a_{ij}\}$ and $D_p = \{b_{ij}\}$. However, by Theorem 1.34 above, we need only consider the Smith normal forms of D_{p+1} and D_p — call them D'_{p+1} and D'_p.

Recall that we have

$$\tilde{H}_p(K;\mathbb{Z}) = \ker\tilde{\partial}_p/\operatorname{Im}\tilde{\partial}_{p+1}$$

Since $\tilde{\partial}_{p+1}$ is a homomorphism, Munkres' Theorem 11.5 tells us that the diagonal entries (greater than 1) of D'_{p+1} are precisely the torsion coefficients of $\mathcal{M}_p/\operatorname{Im}\tilde{\partial}_{p+1}$. Therefore, the torsion coefficients of $\tilde{H}_p(K;Z)$ are the exact same entries of D'_{p+1}.

To wrap up the chapter, we present a holistic way to represent our Morse homology groups:

Definition 1.35. *Let K be an oriented simplicial complex endowed with some discrete gradient vector field V, and let $p \geq 0$. We define the p^{th} Betti number of K to be*

$$\beta_p = \dim H_p(K;\mathbb{Z}) = \dim(\ker\tilde{\partial}_p) - \dim(Im\,\tilde{\partial}_{p+1})$$

By letting D_p and D_{p+1} be the matrix representations of $\tilde{\partial}_p$ and $\tilde{\partial}_{p+1}$ as above, it is easy to see that

$$\beta_p = \dim\ker(D_p) - \operatorname{rank}(D_{p+1}) = (m_p - \operatorname{rank}(D_p)) - \operatorname{rank}(D_{p+1})$$

Since the dimension and rank are invariant under linear transformations, we may simply consider the nullity and rank of the Smith normal forms of our two matrices D'_p and D'_{p+1}, respectively.

Corollary 1.36. *Let K be an oriented simplicial complex endowed with some discrete gradient vector field V, and let $p \geq 0$. If D'_p and D'_{p+1} are the Smith normal form representations of our boundary maps $\tilde{\partial}_p$ and $\tilde{\partial}_{p+1}$ then*

$$\tilde{H}_p(K;\mathbb{Z}) \simeq Z^{\beta_p} \oplus \bigoplus_i \mathbb{Z}/t_i\mathbb{Z}$$

where the t_i are the diagonal entries of D'_{p+1} with $t_i > 1$ and

$$\beta_p = m_p - \operatorname{rank}(D'_p) - \operatorname{rank}(D'_{p+1})$$

The reader should note that the formula above is not in any way specific to Morse homology; the Fundamental Theorem of Finitely Generated Abelian Groups always allows one to decompose a (non-trivial) homology group into its torsion elements along with some $\mathbb{Z}^k$ (assuming $\mathbb{Z}$ is the coefficient group). In fact, the last bit of information specific to Morse theory was given in the definition of the Morse boundary map $\tilde{\partial}_p$ from Definition 1.15. We will see in later chapters that the Morse boundary map provides a more arduous means of computing homology on a simplex, though also allows the potential for one to consider far fewer vertices (given the right discrete vector field).

Chapter 2

Turing Satisfiability

2.1 Preliminaries

With the basics of discrete Morse theory and homology covered, an important question to ask is whether it is always feasible to compute the homology of a given simplicial complex K endowed with a discrete Morse function $f : K \to \mathbb{R}$. If not, it would be meaningless to attempt finding the computational complexity of such an algorithm. In fact, the Church-Turing thesis [3] states that if there exists a simplicial complex with discrete Morse function such that the homology is incomputable, then no algorithm exists to compute the homology. Therefore, the focus of this chapter will be to first lay out the basic information of computability theory (from the perspective of Turing machines) and then proceed to apply concepts covered in sections 1.3 and 1.4. All notation and basic information are adapted from [15].

2.2 Turing Machines and Formal Languages

Though Turing completeness is often used as a characterization of whether a task is feasible by a computer or formal language, the underlying theory of Turing machines is actually a purely mathematical construct. In essence, a Turing machine is simply a finite state machine with memory. Moreover, a finite state machine is simply a directed graph (not necessarily acyclic) with inputs and outputs corresponding to the transition from one state to another along an edge, a distinguished starting vertex, and a subset of vertices for which the "machine" is allowed to halt.

However, before defining what a Turing machine truly is, one must make rigorous what is meant by "inputs" and "outputs" for a given machine.

Definition 2.1 (Language)**.** *Let A be an alphabet (a finite set with at least two elements). We define the graded structure A^* as*

$$A^* = \bigoplus_{k=0}^{\infty} A^k$$

where A^k denotes the Cartesian product of A exactly k times (and $A^0 = \{\epsilon\}$ where ϵ is the empty string). An element $s \in A^*$ is called a string over the alphabet A, and is said to have length p if $s \in A^p$.

We endow A^* with a monoid structure as follows: let $s = (s_1, \ldots, s_m) \in A^m$ and $t = (t_1, \ldots, t_n) \in A^n$. Then the binary operation

$$s \cdot t = (s_1, \ldots, s_m, t_1, \ldots, t_n)$$

known as concatenation makes A^* a monoid with identity ϵ. A language L is any subset of A^* (not necessarily a submonoid as $\epsilon \in L$ is not always true).

Thus, not much structure is required in order to define an arbitrary language. For that reason, most classes of languages are defined from a combinatorial perspective as opposed to an algebraic perspective. In particular, the class of languages that we care most about is decidable languages — that is, languages whose strings are accepted by a Turing machine.

Definition 2.2 (Turing Machine)**.** *A Turing machine M is a six-tuple $(\Gamma, \beta, Q, \delta, s, h)$ with*

Γ - *the tape alphabet*

β - *the blank symbol with $\beta \notin \Gamma$*

Q - *the set of states (or vertices)*

δ - the next-state function $\delta : Q \times (\Gamma \cup \{\beta\}) \to (Q \cup \{h\}) \times (\Gamma \cup \{\beta\} \cup \{\boldsymbol{L},\ \boldsymbol{R}\})$ such that if M is in state q with a under the tape-head and $\delta(q, a) = (q', \boldsymbol{C})$, then M enters state q' and either writes $\boldsymbol{C}$ if $\boldsymbol{C} \in \Gamma \cup \{\beta\}$, moves the tape-head left if $\boldsymbol{C} = \boldsymbol{L}$, or moves the tape-head right if $\boldsymbol{C} = \boldsymbol{R}$

s - the starting state

h - the distinguished halting state $h \notin Q$.

Given a string $\omega \in \Gamma^*$, we say that M accepts ω if when started in state s with some string ω' placed left-adjusted on its (otherwise blank) tape and the tape-head at the leftmost cell, the last state entered by M is h and the resulting string on the tape is ω. We define the language $L(M)$ over Γ as

$$L(M) = \{\omega \in \Gamma^* \mid M \text{ accepts } \omega\}$$

Moreover, we say that a language L is decidable if there exists some Turing machine M' such that $L = L(M')$.

This brings us to one of the first questions which this thesis seeks to answer: how does one produce a Turing machine that recognizes the homology groups $H_i(K; \mathbb{Z})$ of a given simplicial complex K?

Like the vast majority of mathematical theorems, one must build up the machinery to tackle such a problem. Indeed, constructing such a Turing machine explicitly would be onerous and its relation to homology would not be immediately obvious. Therefore, the next few lemmas and definitions will allow the use of more flexible approaches when proving the feasibility of an arbitrary task.

Lemma 2.3. *Let M_1 and M_2 be Turing machines over the the alphabets Γ_1 and Γ_2 respectively. Let $L_1 = L(M_1)$ and $L_2 = L(M_2)$. Then there exists a Turing machine M over the alphabet $\Gamma_1 \cup \Gamma_2$ that decides the language $L_1 \cdot L_2$.*

Proof. Let $M_1 = (\Gamma_1, \beta_1, Q_1, \delta_1, s_1, h_1)$ and $M_2 = (\Gamma_2, \beta_2, Q_2, \delta_2, s_2, h_2)$. Without loss of generality we may assume $\beta_1 = \beta_2$ and take $\beta = \beta_1 = \beta_2$. We construct our new Turing machine M with states $Q = Q_1 \cup Q_2$ such that Q_1 and Q_2 are glued together at $h_1 = s_2$ (let this state be called q so that $\pi_1(q) = h_1$ and $\pi_2(q) = s_2$ where $\pi_1 : Q \to Q_1$ and $\pi_2 : Q \to Q_2$ are the projection maps). Furthermore, construct a transition function δ such that $\delta|_{Q_1 \setminus \{q\}} \equiv \delta_1$ and $\delta|_{Q_2} \equiv \delta_2$ — in particular, if there exists some input $a \in \Gamma_2$ such that $\delta_2(s_2, a) = (q', \mathbf{C})$ with $q' \in Q_2$, we set $\delta(q, a) = (q', \mathbf{C})$. Lastly, we add an additional state f which we will call the "fail" state. If we are in a state $q' \neq q$ which corresponds to a non-halting state in M_1 and we encounter a character from our second alphabet Γ_2, we enter state f. Similarly, if we are in a state $q'' \neq q$ which corresponds to a non-starting state of M_2 and encounter a character from the first alphabet Γ_1, we wish to enter state f. If we enter the fail state, we wish to remain in the fail state — thus we set $\delta(f, a) = (f, \mathbf{L})$. This ensures that δ is well-defined on $\Gamma_1 \cup \Gamma_2$. We set our starting state as $s = s_1$ and halting state as $h = h_2$. Let $M = (\Gamma_1 \cup \Gamma_2, \beta, Q, s, h)$.

Fix some $\omega_1 \in L_1$ and $\omega_2 \in L_2$. For the forward direction, we wish to show that $\omega_1 \cdot \omega_2 \in L(M)$. Since $\omega_1 \in L_1$, M_1 halts at the end of the string ω_1. Thus, ω_2 starts at state $q = \pi_1^{-1}(h_1) = \pi_2^{-1}(s_2)$. Since $\delta|_{Q_2} \equiv \delta_2$ and M_2 halts on input ω_2, we have that M halts on input $\omega_1 \cdot \omega_2$.

Conversely, suppose that $\omega \in L(M)$. By definition, M halts on input ω and thus ultimately winds up in state $h = h_2$. By construction, our Turing machine must pass through q to transition from Q_1 to Q_2. Thus, there exists some input $a \in \Gamma_2$ such that $\delta(q, a) = (q', \mathbf{C})$ with $q' \in Q_2$ — without loss of generality we may assume a is the first input which takes M into Q_2. Then we can partition our input ω as

$$\begin{aligned}
\omega &= (\omega^1, \ldots, \omega^n, a, \ldots, \omega^m) \\
&= (\omega^1, \ldots, \omega^n) \cdot (a, \ldots, \omega^m) \\
&= \omega_1 \cdot \omega_2
\end{aligned}$$

for some $m > n$. Thus, M enters state $q = \pi_1(h_1)$ (from some other non-halting state) on input ω^n. By construction of our halting state f, we have that M would enter f if any character of ω_1 were in Γ_2 — therefore, $\omega_1 \in \Gamma_1^*$. Therefore, we must have that M_1 accepts ω_1 so $\omega_1 \in L_1$. Similarly, by selection of a, ω_2 takes M from $q = \pi_2^{-1}(s_2)$ to $h = h_2$. By the same logic, ω_2 must only contain characters of Γ_2 so M_2 halts on input ω_2 and thus $\omega_2 \in L_2$. Therefore, M decides $L_1 \cdot L_2$. $\square$

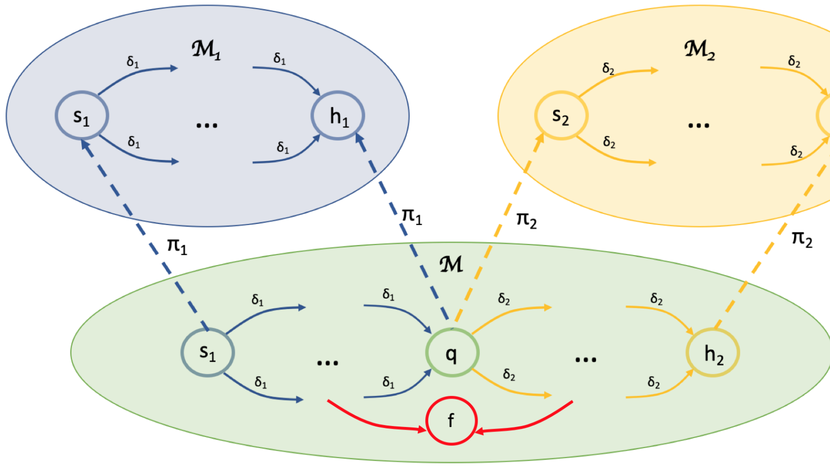


Figure 2.1: Representation of Turing machine concatenation. Dotted arrows represent projection while solid arrows represent next-state function

Corollary 2.4. *The finite concatenation of decidable languages is decidable.*

Though the above corollary may seem trivial, it is a key component in proving that certain iterative algorithms are indeed computable by a Turing machine. For example, if a certain algorithm T_1 is computable via a Turing machine, then any finite loop that executes T_1 in its body is also computable by a (likely different) Turing machine. Based on the previous chapter, the reader should note that the computation of the Morse boundary map is heavily dependent on iteration-based computations (primarily through finite summation). Therefore, if we are able to prove that each component in a finite summation is computable via a Turing machine, then the summation is computable as well.

We next turn our attention to a construct known as the multi-tape Turing machine. Since Turing machines are (in the simplest of terms) finite state machines with memory, we wish to simplify how to keep track of memory. Indeed, the restrictions on how a Turing machine iterates over its tape make it difficult to separate uncorrelated information. For example, if we wanted to examine all strings over the alphabet $\mathcal{A} = \{a, b\}$ and accept those strings with exactly one more a than b, it would be much more intuitive to simply separate the a's and b's onto two separate tracks.

Definition 2.5 (Multi-Tape Turing Machine)**.** *A k-tape Turing machine M is a $(k+5)$-tuple $(\Gamma_1, \dots, \Gamma_k, \beta, Q, \delta, s, h)$ with Γ_i being the tape alphabet for the i^{th} tape, and next-state function δ of the form*

$$\delta(q, a_1, \dots, a_k) = (q', a'_1, \dots, a'_k)$$

with $q, q' \in Q$, and each $a_i, a'_i \in \Gamma_i$. Our β, Q, s, and h are as before.

As we will see in the next theorem (as adapted from [15]), every computation feasible with a multi-tape Turing machine is also feasible with the standard Turing machine. This result will allow us to assume henceforth that the Turing machines we use can separate their data into finitely many tracks.

Theorem 2.6. *Let M_k be a multi-tape Turing machine with exactly $k \in \mathbb{N}$ tracks. Then there exists a single-tape Turing machine M such that $L(M_k) \cong L(M)$ (that is, each string in $L(M_k)$ has a corresponding string ω' in $L(M)$ such that M_k halts on ω if and only if M halts on ω).*

Proof. Suppose there is a distinct alphabet for each track on our k-tape turing machine, Γ_i for $1 \leq i \leq k$. For each i, we construct a second alphabet $\tilde{\Gamma_i}$ such that each $a \in \Gamma_i$ is in one-to-one correspondence with some $\tilde{a} \in \tilde{\Gamma_i}$ (this second alphabet will allow us to collapse k tape-heads into one tape head). Furthermore, since each alphabet is finite we may find some symbol, call it #, that is not in Γ_i for any i (this will serve to distinguish the k tapes when positioned adjacently). Take $\Gamma = \{\#\} \cup \bigcup_{i=1}^{k} (\Gamma_i \cup \tilde{\Gamma_i})$. Suppose M_k recognizes some $\omega_1, \dots, \omega_k$ such that each ω_i is placed on the i^{th} tape. Then we can form the string $\omega \in \Gamma^*$ as follows:

$$\omega = \# \cdot \omega_1 \cdot \# \cdot \omega_2 \cdot \dots \cdot \# \cdot \omega_k \cdot \#$$

Thus, each string on each track is padded by our unique symbol #. Now suppose our multi-tape Turing machine M_k performs the following state-transition when in state q:

$$\delta_k(q, a_1, \dots, a_k) = (p, \mathbf{C}_1, \dots, \mathbf{C}_k)$$

That is, our machine M_K transitions to state p and, for each $1 \leq i \leq k$, the i^{th} tape-head is over the letter $a_i \in \omega_i$ and performs $\mathbf{C}_i$ (that is if $\mathbf{C}_i = b_i \in \Gamma_i \cup \{\beta\}$ then b_i is written onto the i^{th} tape — otherwise the i^{th} tape-head moves either left or right). We modify ω in the following way: if the i^{th} tape head is over $a_i \in \omega_i \subset \omega$, we replace that letter in ω with $\tilde{a}_i$. We construct our next-state function on M by considering the following two cases for each $1 \leq i \leq k$:

Case 1 : ($\mathbf{C}_i = b_i \in \Gamma_i \cup \{\beta\}$). Then M_k writes b_i in place of a_i on the i^{th} track. Thus, we set $\delta(q, \tilde{a}_i) = (q', \tilde{b}_i)$. Note that the alternative alphabet for Γ_i is preserved since the tape-head does not move. If $i = k$ then we set $q' = p$ — otherwise we keep $q' = q$. Thus, we do not actually transition to the next state until all k stages are complete.

Case 2 : ($\mathbf{C}_i \in \{\mathbf{L}, \mathbf{R}\}$). Then M_k moves the i^{th} tape-head left or right. Suppose b_i is the letter to the direct left or right of a_i. Then we perform 3 transitions for M:

$$\delta(q, \tilde{a}_i) = (q, a_i)$$
$$\delta(q, a_i) = (q, \mathbf{C}_i)$$
$$\delta(q, b_i) = (q', \tilde{b}_i)$$

Thus, we unmark a_i as being under the tape head, move over to b_i and mark b_i as being under the tape-head. Again, if $i = k$ then we set $q' = p$ — otherwise we keep $q' = q$.

We can assume without loss of generality that $\beta_i = \beta$ for all i — there need not be any distinction on how a blank character is interpreted. Furthermore, the set of states is not altered by our above algorithm. Therefore, if M_k recognizes $\{\omega_1, \ldots, \omega_k\}$ we have that M recognizes ω. Without loss of generality, we can assume $\{\omega_1, \ldots, \omega_k\}$ is already in the form of ω so it follows that $L(M_k) \subset L(M)$.

To see that $L(M) \subset L(M_k)$, we may simply partition ω at the # markers. Since M and M_k have the same states, it follows by construction that M_k halts on ω if and only if M halts on ω. □

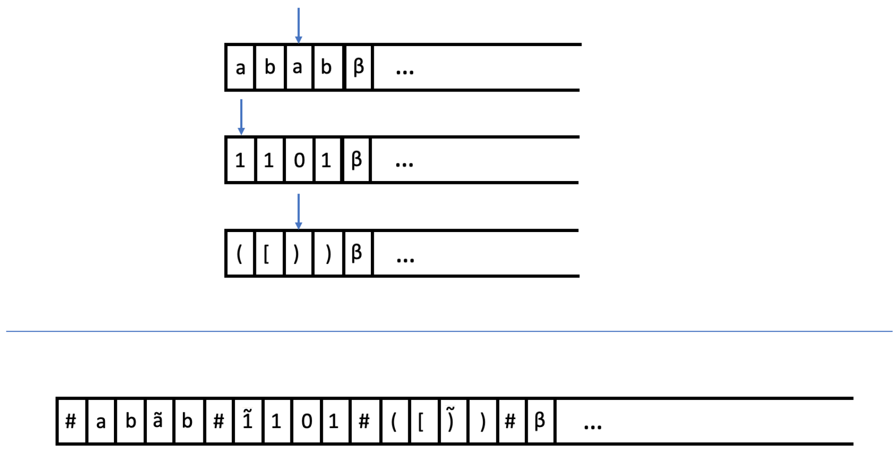


Figure 2.2: Representation of how a word recognized by a 3-tape Turing machine is translated to a word recognized by the corresponding single-tape Turing machine.

2.3 Decidability of Morse Homology

We now have the machinery to compute the main results of this thesis. The decidability of Morse homology alone is a straightforward algorithm which follows from its definition — the difficulty is the extensive prerequisite material required. In particular, we seek to fully utilize the correlation between Hasse diagrams and discrete Morse theory in order to fit graph-theoretical approaches into our algorithm.

Recall from Chapter 1 that we extended the notion of a discrete vector field V on a simplex to the Hasse diagram by associating a matching $\mathcal{M}_V$ with the directed edges reversed. Moreover, our critical points are precisely the vertices of our Hasse diagram $\mathcal{H}_K$ such that no adjacent edge is an element of $\mathcal{M}_V$. If we visualize our Hasse diagram with the vertices corresponding to lowest dimension simplices at the bottom and vertices corresponding to top dimensional simplices at the top, we are able to redefine our sets K_p to be the "level sets" of our Hasse diagram $\mathcal{H}_K$. We seek to make this idea rigorous:

Definition 2.7 (Level sets of $\mathcal{H}_K$)**.** *Let K be a simplicial complex and $\mathcal{H}_K$ be its corresponding Hasse diagram. Define $V \subset \mathcal{H}_K$ to be the set of vertices in $\mathcal{H}_K$ which correspond to vertices in K. We refer to V as the set of root elements or leaves of $\mathcal{H}_K$. For any vertex*

$v \in \mathcal{H}_K$, we define the depth of v to be

$$dep(v) := \min_{\substack{v' \in V \\ \gamma \in \Gamma(v,v')}} len(\gamma)$$

That is, the depth of a vertex is the length of the shortest path to a root element. For $0 \le p \le \dim K$, we redefine the level set K_p to be a level set on $\mathcal{H}_K$.

Clearly, we have that $K_0 = V$ since the singleton path is the shortest path from any root element to itself. It is not hard to see that the two representations of K_p correspond to the same elements:

Lemma 2.8. *Let K be a simplicial complex and $\mathcal{H}_K$ be its corresponding Hasse diagram. Let $K_p \subset K$ denote the set of p-faces and K'_p denote the level set of $\mathcal{H}_K$ as defined above. Then K'_p is the image of K_p in the Hasse diagram.*

Proof. Fix some $0 \le p \le \dim K$ and $\sigma_p \in K_p$. By definition of a simplex, σ_p is a set of $p+1$ vertices so

$$\sigma_p = \{v_1, \dots, v_{p+1}\}.$$

For any v_i we can find a decreasing chain of sets by removing v_j for $j \neq i$ at each step. Without loss of generality, we may assume this sequence is

$$\sigma_p = \{v_1, \dots, v_{p+1}\} \subset \cdots \subset \{v_1, v_2, v_3\} \subset \{v_1, v_2\} \subset \{v_1\}$$

since our enumeration of the vertices does not affect the length of our chain. Let each set of $i+1$ elements be denoted σ_i. Then we have the chain

$$\sigma_0 \prec \sigma_1 \prec \cdots \prec \sigma_p.$$

If we let V denote the set of root elements of $\mathcal{H}_K$, then $\sigma_0 = \{v_1\}$ corresponds to a root element of V. Moreover, if we let v_{σ_i} denote the vertex in $\mathcal{H}_K$ corresponding to σ_i, then $v_{\sigma_p} \dots v_{\sigma_0}$ is a shortest path to a root element in $\mathcal{H}_K$. Thus, $v_{\sigma_p} \in K'_p$.

The converse argument is identical and therefore omitted. □

Though the level sets are likely not going to be used, Corollary 1.13 gives us a simple way to define the Morse groups over the Hasse diagram.

Definition 2.9 (Morse Basis over $\mathcal{H}_K$). *Let K be a simplicial complex and V a discrete vector field over K. Then for $0 \leq p \leq \dim K$, we define the Morse basis of degree p to be*

$$B_{\mathcal{M}_p} = \{v_\sigma \mid \text{every adjacent edge of } v_\sigma \text{ is in } \mathcal{H}_K \backslash \mathcal{M}_V\}$$

We now turn our attention to proving whether certain constructs given so far are computable with a Turing machine.

Lemma 2.10. *The construction of the modified Hasse diagram is decidable for any simplicial complex K and discrete vector field V on K. That is, there is a Turing machine which recognizes an encoding of $\mathcal{H}_K$ with edges flipped along $\mathcal{M}_V$ for any simplicial complex K and discrete vector field V.*

Proof. We may construct an enumeration for any simplicial complex K such that if $\alpha \prec \beta$ then the index of α is strictly less than the index of β. Thus, we consider the alphabet $K^* \cup \{\prec, ;\}$ where $K^* = \{\sigma_1, \sigma_2, \dots\}$. We may thus represent any simplicial complex K as a string over the alphabet $K^* \cup \{\prec, ;\}$. Let $k = \epsilon$ initially denote the empty string. Starting with the vertices (lowest-dimensional faces), if σ_i is a vertex and $\sigma_i \prec \sigma_j$, then we concatenate $\sigma_i \prec \sigma_j$; to k. We apply the same rule to edges once all vertices have been exhausted. Following this pattern, we get a string k with all relations $\sigma_i \prec \sigma_j$ separated by ; and increasing along the length of k in terms of dimension.

For any K, we consider the alphabet $V_{K^*} \cup \widetilde{V_{K^*}} \cup \{\leftarrow, \rightarrow, ;\}$ where $V_{K^*} = \{v_{\sigma_i} \mid \sigma_i \in K^*\}$ and

$$\widetilde{V_K} = \{\widetilde{v_{\sigma_i}} \mid v_{\sigma_i} \in V_K\}$$

The latter set will allow us to distinguish critical faces. Recall that for any V, the discrete vector field is already of the form $\{\{\alpha_i \prec \beta_i\} \mid 1 \leq i \leq m\}$ for some $m \in \mathbb{N}$ — thus, we can always find a subset v of k which gives an encoding of V. We construct a 3-tape Turing machine, call it $M_{\mathcal{H}_K}$, to recognize the Hasse diagram $\mathcal{H}_K$ with edges flipped along $\mathcal{M}_V$ for arbitrary K and V.

On the first tape, we put our string representation of K which we denote as k. On the next tape, we put the string $h_k = \epsilon$ (which is initially empty), and on the third tape we put the substring v of k corresponding to V.

We start in state s with the tape-head positioned along the leftmost cell in each tape. Iterating over the third tape, if $\sigma_1 = \alpha_i$ for some $1 \leq i \leq m$ we enter some state q_1. In q_1 we concatenate $v_{\sigma_1} \rightarrow$ to h_k on the second tape and replace any occurrence of $\widetilde{v_{\sigma_1}}$ with v_{σ_1}. We then move over two on the first tape to read the face σ_j (such that $\sigma_1 \prec \sigma_j$) and write v_{σ_j}; on the second tape. We then enter some state r, in which we rewind the third tape-head to the leftmost position and go back to state s.

If $\sigma_1 \neq \alpha_i$ for all $1 \leq i \leq m$, then we will encounter a β on the third tape. Upon input β on the third tape, we enter a different state q_2. In q_2 we write $\leftarrow$ instead of $\rightarrow$ and replace every occurrence of v_{σ_1} with $\widetilde{v_{\sigma_1}}$. We continue in this fashion until the blank symbol β is encountered on the first tape, at which point we have visited every cell of the first tape and iterated over the third tape approximately $|K|$ times. Since an arrow $\leftarrow$ is flipped to $\rightarrow$ if and only if the face is in the discrete vector field, our string h_k on the second tape gives an encoding for $\mathcal{H}_K$.

Moreover, since we proceed in order of increasing dimension along k, we are able to consistently correct whether σ_n is represented as v_{σ_n} or $\widetilde{v_{\sigma_n}}$ on h_k until all adjacent edges have been considered. The set $\mathcal{M}_V$ is explicitly the concatenation of all substrings of h_k in the form $v_{\sigma_n} \rightarrow v_{\sigma_m}$; and the critical faces are explicitly those σ marked by a $\widetilde{v_\sigma}$.

By Theorem 2.6 above, $M_{\mathcal{H}_K}$ can be recognized by a single-tape Turing machine, say $M'_{\mathcal{H}_K}$, which therefore must also recognize the Hasse diagram of a given simplicial complex with edges flipped on $\mathcal{M}_V$. $\square$

Corollary 2.11. *The computation of the level set K_p is decidable for any simplicial complex K and $0 \leq p \leq \dim K$.*

Proof. We proceed over the alphabet $V_\Gamma \cup \tilde{V}_\Gamma \cup \{\leftarrow, \rightarrow, (,), ;, 0, *\}$. By Lemma 2.10 above, there is a Turing machine $M_{\mathcal{H}_K}$ which recognizes a string h_k representing our Hasse diagram in the form

$$h_k = v_{\sigma_1} \leftarrow v_{\sigma_j}; v_{\sigma_2} \rightarrow \widetilde{v_{\sigma_k}}; \ldots$$

(up to direction of arrows and placement of markers).

We now construct a second, two-tape Turing machine M_{K_p} that recognizes our level set K_p. By Lemma 2.3, the concatenation of Turing machines is again a Turing machine — thus, we may assume without loss of generality that M_{K_p} already recognizes an encoding of $\mathcal{H}_K$. Suppose our encoding h_k of $\mathcal{H}_K$ is placed leftmost on the first tape. In the starting state s we insert the string () before every $\leftarrow$, $\rightarrow$, and ;. Thus, M_{K_p}'s first tape is now of the form

$$h'_k = v_{\sigma_1}() \leftarrow v_{\sigma_j}(); v_{\sigma_2}() \rightarrow \widetilde{v_{\sigma_k}}(); \ldots$$

M_{K_p} then enters some state q_0. In state q_0 we replace $v_{\sigma_j}()$ with $v_{\sigma_j}(0)$ if σ_j is a vertex of K (that is, there exists no τ with $v_\tau \rightarrow v_{\sigma_j}$ or $v_\tau \leftarrow v_{\sigma_j}$). We then enter some state q_2. We construct the next-state function on δ in an iterative fashion: initially, we replace each $v_\sigma(0) \rightarrow v_\tau()$ (resp. $v_\sigma(0) \leftarrow v_\tau()$) with $v_\sigma(0) \rightarrow v_\tau(*)$ (resp. $v_\sigma(0) \leftarrow v_\tau(*)$).

Assume that on the i^{th} pass of q_2 we replace $v_{\sigma_j}()$ with $v_{\sigma_j}(** \cdots *)$ (i.e. the string containing only $*$ of length i) if and only if v_{σ_j} has depth i. Then on the $i+1^{th}$ pass we replace every $v_{\sigma_j}(** \cdots *) \leftarrow v_{\sigma_k}()$ (resp. $v_{\sigma_j}(** \cdots *) \rightarrow v_{\sigma_k}()$) with $v_{\sigma_j}(** \cdots *) \leftarrow v_{\sigma_k}(** \cdots **)$ (resp. $v_{\sigma_j}(** \cdots *) \rightarrow v_{\sigma_k}(** \cdots **)$) in a fashion such that markers $\tilde{v}$ are preserved. For the sake of notation, we may replace the string $** \cdots *$ of length i with the number i. Since the vertices are increasing along h'_k by depth (by Lemma 2.10 above), we need only modify h'_k after $v_{\sigma_k}(i+1)$. Thus, for each $v_{\sigma_k}(i+1) \in K_{i+1}$, we reiterate over h'_k and replace any instances of $v_{\sigma_k}(*) \leftarrow$ (resp. $v_{\sigma_k}(*) \rightarrow$) with $v_{\sigma_k}(i+1) \leftarrow$ (resp. $v_{\sigma_k}(i+1) \rightarrow$) in a fashion such that markers $\tilde{v}$ are preserved. Note that this will take exactly $|K_p|$ passes over h'_k. By construction, K_p is precisely the substring of h'_k in the form

$$v_{\sigma_k}(p) \leftarrow v_{\sigma_j}(p+1); \ldots; v_{\sigma_n}(p) \rightarrow \widetilde{v_{\sigma_m}}(p+1);$$

up to direction of arrows and placement of markers.

Thus, M_{K_p} enters a final state q_3 in which the first tape-head is placed leftmost along the tape. We then iterate over h'_k one last time. If $v_\sigma(p)$ (resp. $\widetilde{v_\sigma}(p)$) is under the first tape-head, then we write v_σ (resp. $\widetilde{v_\sigma}$;) on the second tape. If we encounter the blank symbol β in q_3, we enter our halting state h.

By Theorem 2.6 above we can find a single-tape Turing machine M'_{K_p} that recognizes the same language as M_{K_p}. By Lemma 2.3 above, the concatenation of $M'_{\mathcal{H}_K}$ and M'_{K_p} is again a Turing machine that recognizes an encoding of the level sets K_p on the Hasse diagram $\mathcal{H}_K$.

□

The reader should note that this proof is closely related to how the depth of a node is computed recursively, as often taught in elementary computer science courses. In particular, a set of roots are established to have depth 0 by construction. Then, the depth of an arbitrary node is defined to be one more than the minimum depth of its neighbors. For example, evaluating the depth of a tree can be simply computed as follows:

Algorithm 1: Depth

```
input  : Tree T, Node node
output: Depth of node
begin
    if node in T.Root then
        return 0;
    else
        return 1 + min(Depth(i) for i in node.children);
    end
end
```

As was the case in Chapter 1, the level sets of our Hasse diagram are of little use in pursuing Morse homology. However, the previous two proofs give us much flexibility in computing the Morse groups $\mathcal{M}_p$ on the Hasse diagram. This brings us to the first of several central results for this thesis:

Theorem 2.12. *The computation of the Morse basis $B_{\mathcal{M}_p}$ is decidable for any simplicial complex K, discrete vector field V on K and $0 \leq p \leq \dim K$.*

Proof. By Lemma 2.10 above, we are able to construct a Turing machine $M_{\mathcal{H}_K}$ which computes an encoding h_k of our Hasse diagram $\mathcal{H}_K$. By Corollary 2.11 above, we are able to construct a second Turing machine M_{K_p} which constructs an encoding of K_p — call it k_p — on its second tape whenever h_k is placed on its first tape. Continuing in this fashion, we construct a third two-tape Turing machine $M_{\mathcal{M}_p}$ which recognizes an encoding of $B_{\mathcal{M}_p}$ when k_p is placed on its first tape. Recall that the critical cells of K_p are represented in the encoding k_p as vertices v_σ which are marked as $\widetilde{v_\sigma}$. Thus, we need only pass over the first tape in one iteration: if an unmarked vertex v_σ; is under the first tape-head then we move to the right, otherwise we write $\widetilde{v_\sigma}$; onto the second tape-head and move to the right. Again utilizing Theorem 2.6 and Lemma 2.3 above, we simply construct a single-tape Turing machine $M'_{\mathcal{M}_p}$ that recognizes the same language as $M_{\mathcal{M}_p}$ and concatenate that to the Turing machines $M'_{\mathcal{H}_K}$ and M'_{K_p}. $\square$

The theorem above gives us a strong foundation for the two remaining theorems of this chapter. Clearly, we would not be able to show that the construction of our boundary map $\tilde{\partial}_p : \mathcal{M}_p \to \mathcal{M}_{p-1}$ is decidable if we did not know that the basis $B_{\mathcal{M}_p}$ is itself decidable. Since we know that $\tilde{\partial}_p$ is linear, it simply suffices to compute $\tilde{\partial}_p$ for all basis elements.

We will continue to make heavy use of the multi-tape Turing machine since orientation comes into consideration for the boundary map. Before we proceed with any other proofs, it is worth giving a brief outline on how we anticipate the boundary map computation to go. Adopting the notation from the proofs above, we write $v_\sigma \leftarrow v_\tau$ when $\sigma \prec \tau$ and $\{\sigma \prec \tau\}$ is not in the discrete vector field. On the other hand, we write $v_\sigma \to v_\tau$ when $\sigma \prec \tau$ and $\{\sigma \prec \tau\}$ is in the discrete vector field (since directional arrows are flipped along $\mathcal{M}_V$). In either case, the vertex representing a face of larger dimension is always on the right.

Instead of tackling the entire computation of $\tilde{\partial}_p$, we wish to show that certain subcomponents of our calculation are computable via a Turing machine — the first of such components is given by Algorithm 2 below. We must assume henceforth that our discrete vector field is a discrete gradient vector field of some Morse function. By Corollary 1.20 from Chapter 1, this will assure us that our Hasse diagram is in fact acyclic. The lack of cycles will prevent our recursive algorithm below from running into an endless loop. As one can clearly tell from the structure, we are searching for two cases (which cannot simultaneously occur since the Hasse diagram does not have edges between equi-dimensional faces):

Case 1: The last face of our current path μ is a $(p-1)$-face. In order to continue our path, we need some σ' such that $\mu \prec \sigma'$ and $\mu \prec \sigma'$ is in the discrete vector field. But then we must have that $v_\mu \to v_{\sigma'}$.

Case 2: The last face of our current path μ is a p-face. Since our path alternates between dimension p and $(p-1)$ we are clearly looking next for some $\sigma' \in K_{p-1}$. Then either $v_{\sigma'} \leftarrow v_\mu$ or $v_{\sigma'} \to v_\mu$. In the latter case, if $v_{\sigma'} \to v_\mu$ then $\{\sigma' \prec \mu\}$ is in the discrete vector field. But by definition of a discrete Morse function we cannot have some second $\{\sigma' \prec \mu'\}$ in the discrete vector field, so our path ends at σ'. But σ' is not critical so it does not affect the overall calculation of $\tilde{\partial}_p(\tau)$. Thus, we need only consider $v_{\sigma'} \leftarrow v_\mu$.

Algorithm 2: AppendPaths

```
input  : Path γ_σ, Set of paths Γ
begin
    μ := last face of γ_σ;
    if dim μ = p − 1 then
        for σ′ ∈ K_p with v_μ → v_σ′ do
            γ_σ′ = γ_σ ∪ {σ′};
            Append γ_σ′ to Γ;
            AppendPaths(γ_σ′);
        end
    else
        for σ′ ∈ K_{p−1} with v_σ′ ← v_μ do
            γ_σ′ = γ_σ ∪ {σ′};
            Append γ_σ′ to Γ;
            AppendPaths(γ_σ′);
        end
    end
end
```

As mentioned, if the Hasse diagram is acyclic then the function will eventually exhaust itself and Γ will contain all paths emanating from a maximal face of τ. Hence, the only remaining task is to select the relevant paths and sum up $m(\gamma)$ appropriately.

Algorithm 3: EvaluatePaths

input : Set of paths Γ

begin

CriticalSums := { $(\sigma, 0)$ **for** σ **in** $B_{\mathcal{M}_{p-1}}$ } ;

for γ ***in*** Γ **do**

μ := last face of γ;

L = len(γ);

Prod := 1;

if μ ***in*** $\mathcal{M}_{p-1}$ **then**

for i ***in*** $0 \ldots L-1$ **do**

Prod = Prod * Orient(τ_i, σ_{i+1});

end

CriticalSums[μ] = CriticalSums[μ] + Prod

end

end

end

In Algorithm 3 above, we simply iterate over all the paths that we found in Algorithm 2. Since Algorithm 2 collects paths regardless of whether they land on a critical simplex, we must check in our loop that the last face of a path is in fact a critical face in $B_{\mathcal{M}_p}$. If so, we proceed in the traditional manner of finding $m(\gamma)$. Each critical face is assigned its own sum, as given above by the array CriticalSums. Thus, if a given path ends on some critical cell μ then we add $m(\mu)$ to that particular sum.

The next goal is to prove that the algorithms above can be computed by a Turing machine. We in fact start with the most difficult algorithm to compute — Algorithm 2.

Lemma 2.13. *There exists a Turing machine which recognizes the set of all gradient paths emanating from a fixed face for any simplicial complex K and gradient vector field $V = -\nabla f$ on K.*

Proof. We may assume without loss of generality that K is connected (and thus the Hasse diagram $\mathcal{H}_K$ is connected as well). Fix some face $\sigma \in K$ and let v_σ be the associated vertex on the Hasse diagram $\mathcal{H}_K$. Let K^* denote the set of all faces of K. We wish to construct a 3-tape Turing machine M_Γ that will recognize all discrete gradient paths emanating from σ. Our first and second tape will contain strings over the alphabet $K^* \cup \underline{K^*}$, where $\underline{K^*} = \{\underline{\sigma} \mid \sigma \in K^*\}$ allows us to mark the last element of a path (since the $\widetilde{v_\sigma}$ is already used to mark critical faces).

Initially, the singleton $\underline{\sigma}$ is placed on the second tape under the second tape-head, since it is by definition the last element of a singleton path. We let γ denote the path under the second tape-head at any given moment (initially $\gamma = \{\underline{\sigma}\}$).

Lastly, by Corollary 2.11 we are able to construct a Turing machine $M_{\mathcal{H}_K}$ which constructs an encoding

$$h'_k = v_{\sigma_1}(0) \leftarrow v_{\sigma_j}(1); v_{\sigma_2}(0) \rightarrow \widetilde{v_{\sigma_k}}(1); \ldots$$

of our Hasse diagram with dimensions considered. Without loss of generality, we may assume that the tape alphabet on the third tape is precisely all combinations of $v_\sigma(k) \leftarrow v_\tau(k+1)$ (resp. $v_\sigma(k) \rightarrow v_\tau(k+1)$) in a manner that preserves markers $\widetilde{v_\sigma}$. Thus, we are able to write the string h'_k onto the third tape such that ; represents splitting of cells. We require 4 states: s, q_1, q_2, h.

In state s, we copy γ onto the first tape. The first tape-head then iterates over γ to the last face, call it $\underline{\mu}$. The third tape-head then iterates to find a character of the form

$$v_\mu(p-1) \leftarrow v_\tau(p), \quad v_\mu(p-1) \rightarrow v_{\sigma'}(p), \quad v_{\sigma'}(p-1) \leftarrow v_\mu(p), \quad v_{\sigma'}(p-1) \rightarrow v_\mu(p),$$

which must exist since the Hasse diagram is connected. In the last two cases, we enter state q_1.

In state q_1, we again iterate over the third tape looking specifically for a character of the form $v_\mu(p-1) \rightarrow v_{\sigma'}(p)$ for some $\sigma' \in K^*$. If we find such an element under the third tape-head, we write the path $\sigma \ldots \mu\underline{\sigma'}$ at the end of the second tape (note how we move the marker to the new last element). Once the blank character β is encountered on the third tape, we move the second tape-head left until the character $\underline{\mu}$ is encountered. Recall, this represents the last face of the path we just extended. We then move the second tape-head right one. If any such σ' were found in the previous step, then there would be a new path under the tape-head — thus, we set γ to be the new path under the tape-head and return to state s. Otherwise, we were unable to extend our path any further and there is a blank symbol β under the tape-head — in this case, we enter our halting state h.

In the case that we find the latter two symbols, we enter state q_2. In state q_2, we iterate over the third tape looking for a character of the form $v_{\sigma'}(p-1) \leftarrow v_\mu(p)$ for some $\sigma' \in K^*$ as above. If we find such an element, we do the exact same as in state q_1.

Since $V = -\nabla f$ is assumed to be a discrete gradient vector field, there are no cycles in the Hasse diagram by Corollary 1.20. If

$$\gamma_0 = \sigma \prec \mu_0 \succ \mu_1 \prec \mu_2 \ldots$$

is a discrete gradient path, then there will be elements of the form

$$v_\sigma(p-1) \rightarrow v_{\mu_0}(p) \ ; \ v_{\mu_1}(p-1) \leftarrow v_{\mu_0}(p) \quad ; \ v_{\mu_1}(p-1) \rightarrow v_{\mu_2}(p) \ ; \ \ldots$$

on the third tape. Thus,

$$\underline{\sigma}, \sigma\underline{\mu_0}, \sigma\mu_0\underline{\mu_1}, \sigma\mu_0\mu_1\underline{\mu_2}, \ldots$$

will all eventually be written onto the second tape. Therefore, the second tape of M_γ recognizes an encoding of all discrete gradient paths emanating from σ, call it Γ. By Theorem 2.6 we can find a single-tape Turing machine M'_Γ which recognizes the encoding for Γ. □

The reader should note that the second tape of M_Γ in the proof above is specifically used in a queue-like fashion; this is no coincidence, as a FIFO (first-in-first-out) data structure is commonly used for depth-first search. Thus, our algorithm can be broken down into recursively applying depth-first search until all nodes have been visited.
This brings us to the first example of this chapter:

Example 2.14. *Let K be the 1-skeleton of the tetrahedron Δ_3 (that is, Δ_3 with all 2-faces* *and the 3-face removed). Furthermore, let V be the following discrete gradient vector field, as pictured in blue in Figure 2.3:*

$$V = \{\{v_0 \prec e_2\}, \{v_1 \prec e_1\}, \{v_2 \prec e_5\}\}$$

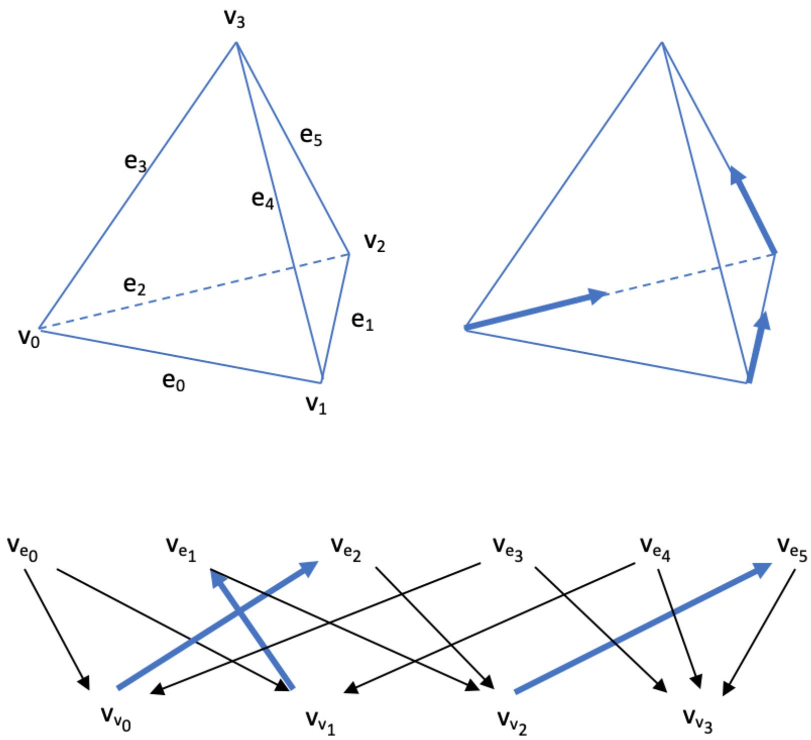


Figure 2.3: Skeleton of Δ_3 endowed with gradient vector field, along with Hasse diagram

It may be useful to the reader to note that adding the pair $\{v_3 \prec e_4\}$ to V would induce a cycle in the Hasse diagram $\mathcal{H}_K$ and, thus, no longer make V a discrete gradient vector field.

Following from Corollary 2.11, we are able to construct an encoding for $\mathcal{H}_K$ along with the depth of each node as follows:

$$h'_k = v_{v_0}(0) \leftarrow v_{e_0}(1); v_{v_0}(0) \rightarrow v_{e_2}(1); v_{v_0}(0) \leftarrow v_{e_3}(1); v_{v_1}(0) \leftarrow v_{e_0}(1); \ldots$$

Choose $\tau = e_0$. From Figure 2.3 we can see that e_0 has two maximal faces: v_0 and v_1. It is easy to see from $\mathcal{H}_K$ that there are only two distinct longest paths emanating from a *maximal face of τ:*

$$v_0 \prec e_2 \succ v_2 \prec e_5 \succ v_3$$

$$v_1 \prec e_1 \succ v_2 \prec e_5 \succ v_3$$

For this particular example, we choose $\sigma = v_0$ to be the maximal face of τ of interest. Following the proof of Lemma 2.13 above, we build a Turing machine M_Γ with the encoding h'_k on the third tape, the singleton maximal face σ on the second tape, and the first tape empty, as shown in Figure 2.4.

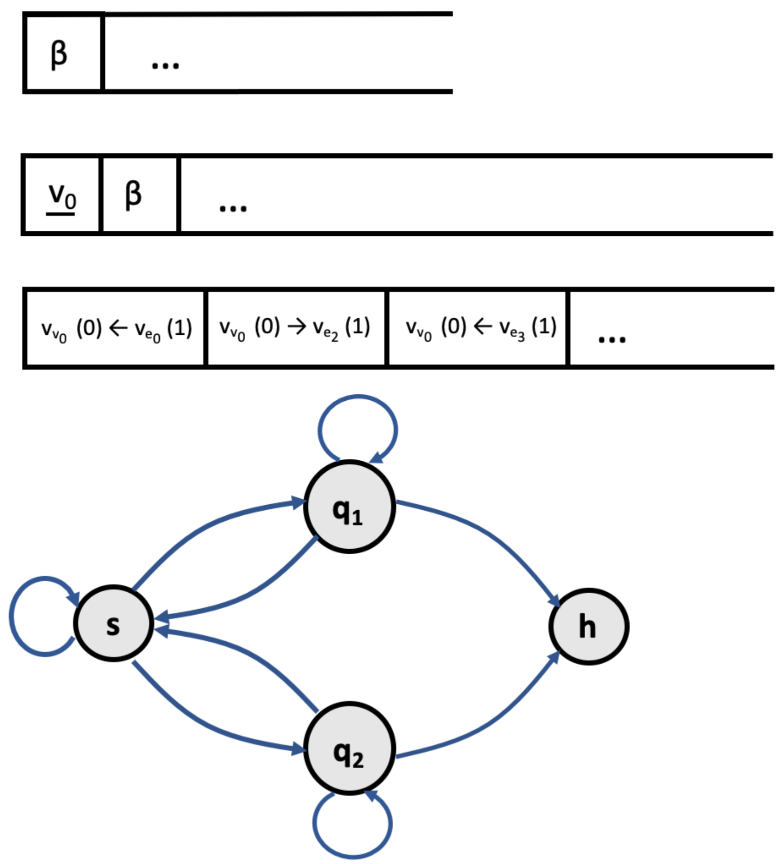


Figure 2.4: Turing machine M_Γ at initial stage

We then iterate over the third tape to find an element of the form

$$v_\sigma(0) \leftarrow v_\tau(1), \quad v_\sigma(0) \rightarrow v_{\sigma'}(1), \quad v_{\sigma'}(0) \leftarrow v_\sigma(1), \quad v_{\sigma'}(0) \rightarrow v_\sigma(1)$$

Since $\sigma = v_0$ and $\tau = e_0$, the first element on our third tape satisfies the first criterion so we enter state q_1. In state q_1, we again iterate over the third tape in search for a character of the form $v_\sigma(0) \to v_{\sigma'}(1)$ for some $\sigma' \in K_1$. Luckily, the second element on the third tape satisfies this condition, and thus we add $v_0\underline{e_2}$ to the end of the second tape. However, one can see from $\mathcal{H}_K$ that there is no other element of the form $v_{v_0}(0) \to v_{e_j}(1)$ for any $0 \leq j \leq 5$. Thus, we encounter a blank sybmol β and move the second tape-head to the right of $\underline{v_0}$ on the second tape. This, however, represents the beginning of our new path $\gamma = v_0\underline{e_2}$.

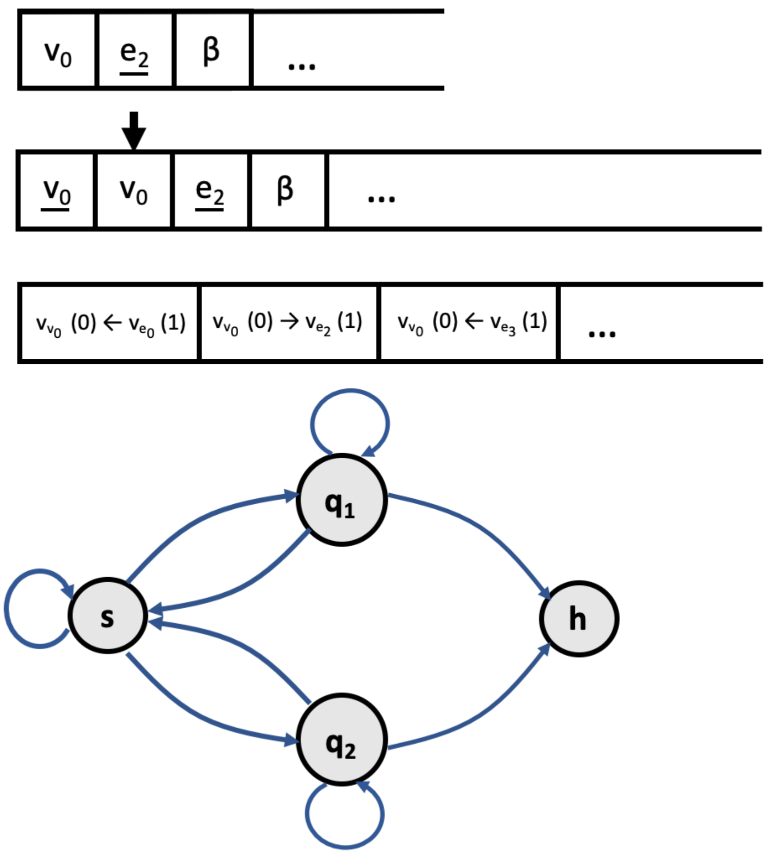


Figure 2.5: Turing machine M_Γ after returning to state s for first time

Again in state s, the first tape iterates over γ to find the last face $\underline{\mu} = \underline{e_2}$. The third tape-head then iterates over the third tape to find an element resembling one of the following:

$$v_{e_2}(0) \leftarrow v_\tau(1), \quad v_{e_2}(0) \rightarrow v_{\sigma'}(1), \quad v_{\sigma'}(0) \leftarrow v_{e_2}(1), \quad v_{\sigma'}(0) \rightarrow v_{e_2}(1).$$

Since the second element of our third tape satisfies the fourth criterion above, we enter state q_2. As described in Lemma 2.13, while in state q_2 we reiterate over the third tape to look for an element specifically of the form $v_{\sigma'}(0) \leftarrow v_{e_2}(1)$ for some $\sigma' \in K_0$. As one can observe from $\mathcal{H}_K$, the only choice of σ' is v_2. Hence, we write $v_0 e_2 \underline{v_2}$ to the end of the second tape and continue searching for some $v_{\sigma'}(0) \leftarrow v_{e_2}(1)$. Since there is no other σ' satisfying $v_{\sigma'}(0) \leftarrow v_{e_2}(1)$, we hit a blank symbol β and return to state s.

We continue in this fashion until our second tape has

$$\underline{v_0} v_0 \underline{e_2} v_0 e_2 \underline{v_2} v_0 e_2 v_2 \underline{e_5} v_0 e_2 v_2 e_5 \underline{v_3}$$

On our last iteration, when $\gamma = v_0 e_2 v_2 e_5 \underline{v_3}$ is placed on the first tape, we enter state q_1 to try to find an element of the form $v_{v_3}(0) \rightarrow v_{\sigma'}(1)$. As one can tell from $\mathcal{H}_K$, no such string occurs on the third tape and thus we encounter a blank symbol β without putting anything on the tape. We then go back to state s, go to our marked vertex $\underline{v_3}$ and move the second tape-head right one. However, since we did not append any lists in our last step the second tape-head reads a blank symbol β — thus, we enter our halting state. The reader can check that the second tape contains all possible discrete gradient paths emanating from $\sigma = v_0$.

Before we can prove that the computation of $\tilde{\partial}_p(\tau)$ is decidable, it remains to show that Algorithm 4 is decidable.

Lemma 2.15. *There exists a Turing machine which computes*

$$\sum_{\sigma \in B_{\mathcal{M}_{p-1}}} \sigma \sum_{\gamma \in \Gamma(\tau, \sigma)} m(\gamma)$$

for any simplicial complex K and discrete gradient vector field $V = -\nabla f$ with $\tau \in K_p$ (for $1 \leq p \leq \dim K$).

Proof. Let K be a simplicial complex and $V = -\nabla f$ be a discrete gradient vector field on K. Fix some $1 \leq p \leq \dim K$ and $\tau \in K_p$. In order to keep track of our sums, we first wish to find an upper bound on the number of paths emanating from τ (since each $m(\gamma)$ is at most 1). Let $k_p = |K_p|$. From a given p-face, there are exactly $\binom{p+1}{p}$ maximal faces (by definition of a simplex). However, there can only be one edge in $\mathcal{M}_V$ emanating from a $(p-1)$-face σ since $\mathcal{M}_V$ contains non-adjacent edges. Suppose that edge goes to some τ' with $\sigma \prec \tau'$ (and $\{\sigma \prec \tau'\} \in V$ by construction). Since returning to σ would induce a cycle and there are no cycles on discrete gradient vector fields, there must be at most $k_{p-1}-1$ faces to return to. Continuing in this fashion, we have that the number of discrete paths emanating from σ is bounded by

$$M = \binom{p+1}{p}(k_p - 1)!$$

We wish to construct a Turing machine M_Σ which will recognize the sum of all multiplicities of paths emanating from τ and terminating at a critical $(p-1)$-face.

Since M above is finite, we let $\mathcal{M}_p \cup \{-M, \dots, 0, \dots, M\}$ be the alphabet for the third tape. Suppose $\mathcal{M}_p = \{\alpha_1, \dots, \alpha_{m_p}\}$. Then our third tape will initially contain the string

$$\alpha_1 0 \alpha_2 0 \dots \alpha_{m_p} 0$$

By Lemma 2.13 above, we are able to construct a Turing machine M_Γ which produces a string Γ of all paths emanating from τ with the last face of each path marked by $\underline{\mu}$. Originally, this is over the alphabet $K^* \cup \underline{K^*}$ where K^* denotes the set of all faces of K. We add the sets $(K^*)^+ \cup \underline{(K^*)^+}$ and $(K^*)^- \cup \underline{(K^*)^-}$ in order to account for positive and negative orientation. Initially, we let the first and second tapes of our Turing machine M_Σ be Γ.

We need 8 states for our Turing machine M_Σ: $s, s', h, \Sigma^+, \Sigma^-, r, q_1, q_2$. In the starting state s, the second tape-head iterates over its tape marking each σ with either σ^+ or σ^- depending on its orientation. Once a blank symbol β is encountered, we enter state s' and the second tape-head returns to the left-most position.

In state s', if there is a blank symbol under the first and second tape-heads, we enter state h and halt. Otherwise, we look at the cell under the first tape-head. If it is a critical cell (that is, $\underline{\mu}$ is marked by a $\underline{\tilde{\mu}}$), we enter state q_1. Otherwise we enter state r.

Whenever M_Γ enters state r, we know that the first and second tape-head are over the same element, which is the last face $\underline{\mu}$ of some path γ. In state r, we move the first tape-head to the right until it finds the next occurrence of a last face $\underline{\mu'}$ of the next path γ', and move the second tape-head right by one (so that it is at the first element of γ'). Thus, r allows us to iterate over both copies of Γ in a manner such that we traverse one path at a time. After moving the second tape-head right one, M_Σ transitions from state r to state s' again.

Our state q_1 represents an overall positive product for $m(\gamma)$ whereas state q_2 represents an overall negative product for $m(\gamma)$. Thus in state q_1, if the first and second tape-heads are not in the same position (i.e. we have not traversed our entire path yet) and a face of the form σ^+ is under the second tape-head, we remain in state q_1. Otherwise, a face of the form σ^- is under the second tape-head and we enter state q_2. State q_2 is almost identical in that we remain in state q_2 if we read σ^+ and transition back to q_1 if we read σ^-. Lastly, if the first and second tape-heads are in the same position (meaning we have iterated over a full path) we either transition from state q_1 or q_2 to Σ^+ or Σ^- (depending on the orientation of our last element $\underline{\mu}$).

If our machine is in Σ^+, we read the cell $\underline{\mu}$ under the first tape-head (which we know is critical since we initially entered state q_1) and increment the cell directly to the right of μ on the third tape by 1. Similarly, if we are in Σ^-, we reduce such a cell by -1.

Clearly M_Σ must halt since the first and second tape-heads only move rightwards after state s and, therefore, must eventually encounter a blank symbol β on both tapes. Moreover, since the multiplicity of every path in Γ is accounted for (as we do not jump indices on the second tape after state s), we must have that our third tape is of the form

$$\alpha_1 \sum_{\gamma\in\Gamma(\sigma,\alpha_1)} m(\gamma)\ldots\alpha_{m_p}\sum_{\gamma\in\Gamma(\sigma,\alpha_{m_p})} m(\gamma)$$

Since each $m(\gamma)$ is either 1 or -1, we have that $\sum_{\gamma\in\Gamma(\sigma,\alpha_i)} m(\gamma)\in\{-M,\ldots,0,\ldots,M\}$. As in previous proofs, it follows from Theorem 2.6 and Lemma 2.3 that we are able to construct a single-tape Turing machine M'_Σ which recognizes

$$\sum_{\nu\in B_{\mathcal{M}_{p-1}}}\nu\sum_{\gamma\in\Gamma(\sigma,\nu)} m(\gamma)$$

□

The reader should note that the mechanism in M_Σ above which recognizes $m(\gamma)$ for any given path (i.e. states q_1 and q_2) is really just the finite-state machine which computes the binary exclusive-OR function (also known as XOR). This is due to the fact that multiplying by 1 or -1 is the same as adding by 0 or 1 over $\mathbb{Z}_2$, respectively.

We are now ready for the second central result of this thesis:

Theorem 2.16. *There is a Turing machine which recognizes the Morse boundary map $\tilde{\partial}_p$ for any simplicial complex K, discrete gradient vector field $V = -\nabla f$ and $0 \leq p \leq \dim K$.*

Proof. By Theorem 2.12, we are able to construct Turing machines $M_{\mathcal{M}_p}$ and $M_{\mathcal{M}_{p-1}}$ which recognize $B_{\mathcal{M}_p} = \{\alpha_1, \ldots, \alpha_{m_p}\}$ and $B_{\mathcal{M}_{p-1}} = \{\beta_1, \ldots, \beta_{m_{p-1}}\}$. Fix some $1 \leq i \leq m_p$. By Corollary 2.11, we are able to also construct a Turing machine M_{K_p} which devises an encoding

$$h'_k = v_\sigma(0) \leftarrow v_\tau(1); \ldots$$

for our Hasse diagram $\mathcal{H}_K$.

By Lemma 2.13, we are able to construct a Turing machine M_{Γ_i} which computes all discrete gradient paths emanating from α_i. By Lemma 2.15 above, we are further able to construct a Turing machine M_{Σ_i} which recognizes the sum

$$\sum_{\beta_i \in B_{\mathcal{M}_{p-1}}} \beta_i \sum_{\gamma \in \Gamma(\tau, \beta_i)} m(\gamma)$$

However, this is precisely $\tilde{\partial}(\alpha_i)$. By allowing $1 \leq i \leq m_p$ to vary, we come up with m_p Turing machines which collectively recognize $\tilde{\partial}_p$. By Corollary 2.4, the finite concatenation of such Turing machines is again a Turing machine. By Theorem 2.6, we are able to construct a single-tape Turing machine which recognizes an encoding of $\tilde{\partial}_p$. □

The reader should note that by no means is a Turing machine ever the most efficient method of computation — this is evident in the proof above by the large bound M on the incidence number along a path. As previously stated, Turing machines simply offer an answer to which computations are always feasible versus which computations may never halt. It will later be the goal of Chapter 3 to identify an asymptotic upper bound for Morse homology.

Returing to the question of whether Morse homology is always computationally feasible, Theorem 2.16 above gives us all the machinery needed to provide a rigorous answer. Recall from Corollary 1.36 in order to find our homology groups $\tilde{H}_i(K;\mathbb{Z})$ it suffices to find the matrices D_{p+1} and D_p representing our boundary maps and compute their Smith normal forms.

Fortunately, an algorithm which computes the Smith normal form of an integer matrix was proved to be P-complete in [9]. Clearly our matrix representation of $\tilde{\partial}_p$ is an integer matrix since the multiplicity of a path μ is only defined over the integers. This brings us to the last theorem of this chapter.

Theorem 2.17. *There exists a Turing machine which recognizes*

$$\tilde{H}_i(K;\mathbb{Z})$$

for any simplicial complex K (endowed with a discrete gradient vector field V) and $0 \leq i \leq \dim K$.

Proof. By Theorem 2.16 above, there is a Turing machine M_∂ which recognizes the boundary maps $\tilde{\partial}_p$ and $\tilde{\partial}_{p+1}$. We may assume without loss of generality that they are already represented in a linearized matrix form — that is, an index $0 \leq n \leq m_p m_{p-1}$ on the tape of M_∂ would correspond to an element $[D_p]_{(i,j)}$ in the matrix representation D_p of $\tilde{\partial}_p$ with $i = \lfloor n/m_{p-1} \rfloor$ and $j = n \mod m_{p-1}$.

From [9] we know that there exists a Turing machine that runs in polynomial time (with respect to length of input) that recognizes the Smith normal forms D'_p and D'_{p+1} of D_p and D_{p+1} respectively (where D_p and D_{p+1} are the matrix representations of $\tilde{\partial}_p$ and $\tilde{\partial}_{p+1}$). Moreover, by Theorem 11.5 in [13] we know that the torsion elements of $\tilde{H}_i$ are precisely the diagonal elements of D'_{p+1} which are greater than 1. However, since all other entries of D'_{p+1} are 0 by definition of Smith normal form, it follows that we need only find the entries of D'_{p+1} which are not 0 or 1 — this is clearly feasible in linear time with a Turing machine. To find the Betti number β_i of $\tilde{H}_i(K;\mathbb{Z})$, we construct a Turing machine M_β which iterates over the columns of D'_{p+1} (we would technically need some kind of marker $\tilde{\sigma}$ to distinguish the first element of a column since we only know the product of the number of rows and number of columns) - since our matrix D_{p+1} has dimesnsion $m_{p+1} \times m_p$, this would give us precisely m_p. Since D'_{p+1} is of the form

$$M' = \begin{pmatrix} \mathbf{T} & 0 \\ 0 & 0 \end{pmatrix}, \qquad \mathbf{T} = \begin{pmatrix} t_1 & & 0 \\ & \ddots & \\ 0 & & t_r \end{pmatrix}$$

we know that $\operatorname{rank}(D'_{p+1})$ is simply the number of torsion coefficients, which we already know. Thus, we would merely have to iterate over D'_p once to count the number of non-zero columns in D'_p, ultimately giving us

$$\beta_i = m_p - \operatorname{rank}(D'_p) - \operatorname{rank}(D'_{p+1})$$

Since our homology group $\tilde{H}_i(K, \mathbb{Z})$ can be completely described in terms of the Betti number and torsion coefficients, we are done. $\square$

Chapter 3

Computational Complexity of Morse Homology

3.1 Preliminaries

In the previous chapter we focused on the question of whether certain tasks relating to Morse homology are always computationally feasible. Knowing that this is the case (by Theorem 2.17), a natural follow up question is to ask how much work is required to compute such a task. More specifically, if we know that our computation takes T steps on an input of size N, we wish to know how many steps our computation takes on inputs of sizes $2N, 3N, 4N$, etc. Such questions are part of a much larger field of study known as asymptotic analysis and computational complexity — in this chapter, however, we will restrict our focus to the asymptotic analysis of Morse homology.

Since the Morse boundary map is generally much more tedious to compute than the simplicial boundary map for a fixed simplex, one would hope that Morse homology has the benefit of computational speed-up in most cases; otherwise, there would never be any reason to compute Morse homology instead of simplicial homology. From our background in Chapter 1, an easy way to optimize Morse homology is to make the groups $\mathcal{M}_p$ much smaller than K_p — in particular, we seek discrete Morse functions which minimize our critical cells. Therefore, we hope to identify a class of simplicial complexes which minimize critical cells and thus admit a computational speedup for Morse homology.

3.2 Limit Behavior

In the fields of mathematics and computer science, it is not always necessary to find explicit formulae for the complexity of a function, but rather to find the most accurate bounds for such a function. For example, if we are given some unknown function $g(x)$ and we know that eventually (that is, for sufficiently large x) our function satisfies $f(x) \leq g(x) \leq h(x)$ for two other well-known functions $f(x)$ and $h(x)$, then we can make inferences on the behavior of g at infinity based on the well-known behavior of f and h at infinity. This brings us to the primary concept of this chapter:

Definition 3.1 (Big-O)**.** *Let $f : \mathbb{R} \to \mathbb{R}$ be a real-valued function and $g : \mathbb{R} \to [0, \infty)$ a non-negative function. We write $f(x) = O(g(x))$ if there exists some $M > 0$ and x_0 such that*

$$|f(x)| \leq Mg(x)$$

for all $x \geq x_0$.

In the vast majority of cases, we need only concern ourselves with a select group of functions $g(x)$:

$$O(1), \quad O(x^k), \quad O(2^x), \quad O(\log x), \quad O(x^k \log x), \quad O(x!)$$

where $k \in \mathbb{N}$. Indeed, given a polynomial

$$p(x) = a_k x^k + \dots a_1 x + a_0$$

with $k > 0$, we know that for $x > 1$ we have $x^k > x^{k-1} > \cdots > x > 1$. Thus, for $x > 1$

$$p(x) < |a_k|x^k + |a_{k-1}|x^k + \dots |a_1|x^k + a_0 x^k = x^k \sum_{i=0}^{k} |a_i|$$

so that $p(x) = O(x^k)$. Hence, we are often allowed to predict the behavior of polynomials by looking at their leading coefficients.

A significant drawback of Big-O notation is that upper bounds are not unique for our target function $f(x)$. For example, suppose we have some function $f(x)$ with $f(x) = O(1)$. Then there exist some $M \geq 0$ and x_0 such that $|f(x)| \leq M$ for $x \geq x_0$. But then we must also have

$$|f(x)| \leq Mx \leq Mx^2 \leq \cdots \leq Mx^k \leq \dots$$

for $x \geq \max\{x_0, 1\}$. Thus, if a function is $O(1)$ then it is also $O(x^n)$ for all $n \in \mathbb{N}$ regardless of whether it looks anything like x^n at infinity. Therefore, the objective of asymptotic analysis is to make our bounding function $g(x)$ inside $O(g(x))$ as small as possible.

Though there are numerous other concepts in asymptotic analysis such as big-Omega notation, big-Theta notation, and little-o notation, we will solely focus on the concept of big-O notations in the remaining theorems regarding Morse homology.

3.3 Asymptotic Analysis of Morse Homology

As done in previous sections, we approach the problem of Morse homology by considering the primary components of its algorithm. In Chapter 2, we constructed Algorithms 2 and 3 to help find a Turing machine which would recognize our Morse homology. As it turns out, both of these algorithms are relatively crude in terms of their computational complexity — Algorithm 2 in particular gives an exponential estimation for what is at most a polynomial time computation.

The reader should note that the lion's share of work in computing the complexity of Morse homology is to find an accurate bound on the complexity of computing

$$c_{\tau,\sigma} = \sum_{\gamma \in \Gamma(\tau,\sigma)} m(\gamma)$$

To begin, we wish to find an upper bound on the number of discrete paths emanating from a critical face τ. As before, let K be some simplicial complex endowed with a discrete gradient vector field V. For each $0 \leq p \leq \dim K$, we let $V^{(p)}$ denote the set

$$V^{(p)} = \{\{\sigma \prec \tau\} \in V \mid \dim \tau = p\}$$

Thus, when computing $\tilde{\partial}_p(\tau)$ we restrict our attention solely to paths that travel from τ to a critical $(p-1)$-face using only the vectors in $V^{(p)}$ and no other vector in V. As one might expect, we use $\mathcal{M}_{V^{(p)}}$ to denote the partial matching on the Hasse diagram restricted to $V^{(p)}$. Lastly, we use $\mathcal{H}_K^{(p)}$ to denote the sub-graph of the Hasse diagram restricted to p and $(p-1)$ faces. We henceforth refer to the graph $G = \mathcal{H}_K^{(p)}$ with edges flipped along $\mathcal{M}_{V^{(p)}}$ as the p^{th} section of the modified Hasse diagram.

Theorem 3.2. *Let K be a simplicial complex and $-\nabla f$ be a discrete gradient vector field. Fix some $0 \leq p \leq \dim K$ and let $G = (E, V) = \mathcal{H}_K^{(p)}$ be the p^{th} section of the modified Hasse diagram. Then we can compute a matrix representation of $\tilde{\partial}_p$ in $O(m_p(|V| + |E|))$ time.*

Proof. To begin, we first wish to reduce the problem to finding the sum of multiplicities of paths between a critical p-simplex $\tau \in \mathcal{M}_p$ and critical $(p-1)$-simplex $\sigma \in \mathcal{M}_{p-1}$. Let $c_\tau : V \to \mathbb{Z}$ denote the sum of multiplicities of paths from τ to a vertex $v \in V$. Now clearly we have that $c_\tau(\tau) = 1$. Suppose u is a $(p-1)$-simplex and the only co-faces of u are some a and b. Then any path going to u must also go through a or b. However, both $c_\tau(a)$ and $c_\tau(b)$ are just sums of multiplicities

$$c_\tau(a) = m(\gamma_{a,1}) + m(\gamma_{a,2}) + \cdots + m(\gamma_{a,n})$$
$$c_\tau(b) = m(\gamma_{b1}) + m(\gamma_{b2}) + \cdots + m(\gamma_{b,m})$$

suppose we let $\gamma_{au,i}$ denote our extension of the path $\gamma_{a,i}$ to u. Then it follows that

$$m(\gamma_{au,i}) = \langle \partial a, u \rangle m(\gamma_{a,i})$$

by our definition of multiplicity. Since each path which reaches u must go through a or b, it follows that $c_\tau(u) = \langle \partial a, u \rangle c_\tau(a) + \langle \partial b, u \rangle c_\tau(b)$. Generalizing this concept, we come up with the recursive formula

$$c_\tau(u) = \begin{cases} 1 & u = \tau \\ \sum_{(v,u)\in E} \langle \partial v, u\rangle c_\tau(v) & \dim(u) = p-1 \\ \langle \partial u, v\rangle c_\tau(v) & \{v \prec u\} \in (-\nabla f)^{(p)} \end{cases}$$

where the last case follows from the fact that there's at most one vector in the discrete vector field emanating from a $(p-1)$-simplex (however, the last case does not affect the complexity). If we approach this in a manner similar to modified depth-first search, then we may cache (i.e. store) the results as we proceed along the graph. Thus, we evaluate each edge and vertex at most once by the time we compute all of c_τ, giving us a complexity of $O(|V|+|E|)$. Since our results are cached, accessing the $c_\tau(\sigma)$ takes at most constant time, and thus the computation

$$\tilde{\partial}_p(\tau) = \sum_{\sigma\in\mathcal{M}_{p-1}} c_\tau(\sigma)\sigma$$

is $O(|E|+|V|+m_{p-1})$ — however, we clearly have $|V| > m_{p-1}$ so this bound reduces back to $O(|E|+|V|)$. Lastly, our cached information does not apply for other p-faces τ' so we must again compute $c_{\tau'}$ using the methodology above for all $\tau' \in \mathcal{M}_p$. Since we follow the notation of [10] by using m_p to denote the number of critical p-cells, this gives an overall bound of computing $\tilde{\partial}_p$ in $O(m_p(|E|+|V|))$ time. □

Again, the reader should note the similarity of this algorithm to a modified depth-first search. It is often the case that recursive formulae computed over graphs can be dynamically sped up by storing each calculation in case it is used later.

The above calculation may be slightly modified since we know that we are dealing with simplicial complexes. Recall that a p-simplex σ is simply a set of $p+1$ vertices with the property that any subset of those vertices is again a face of σ. Thus, a $(p-1)$-dimensional face of σ is simply a choice of p vertices out of our original $p+1$ vertices. It follows that the number of $(p-1)$-faces of a given p simplex is exactly $\binom{p+1}{p}$. Thus, the total number of edges on the p^{th} section of the Hasse diagram $\mathcal{H}_K^{(p)}$ is really $|E| = \binom{p+1}{p}|K_p|$. Since

$|V| = |K_p| + |K_{p-1}|$, we have that

$$O(m_p(|V| + |E|)) = O(m_p(|K_p| + |K_{p-1}|))$$

on our simplicial complexes K.

All that is left in order to compute the Morse homology of a simplicial complex K is to find the complexity of computing the Smith normal form of a matrix. In numerical analysis, authors often distinguish between sparse matrices and dense matrices (the former is simply a matrix with the majority of entries equal to 0 and the latter is a matrix that is not sparse) when considering the complexity of linear systems. That said, computing the complexity of Smith normal form is no different — we would expect that far fewer operations are required if we know many of the matrix entries are 0.

One difficulty of providing an accurate bound on the complexity of matrix reduction algorithms is that the true lower bound on fast matrix multiplication is not currently known. We chose to follow the notation of [17] in constructing a variable θ such that two $n \times n$ matrices over a commutative ring can be multiplied in $O(n^\theta)$ time. As [17] states, standard matrix multiplication has $\theta = 3$ whereas Strassen's divide-and-conquer algorithm [18] admits $\theta = \log_2 7$. Special cases arise over finite fields in papers such as [16] which prove that the number of gates needed has order $\theta = 2$. For the case of dense Smith normal form, we now provide the following lemma as proven by [17]:

Lemma 3.3. *Let $A \in \mathbb{Z}^{n \times m}$. Then the Smith normal form UAV of A can be recovered in*

$$O(nmr^{\theta-1} \log(nm) \log \|A\| + nmr \log(nm) B(\log \|A\|))$$

operations, where

$$B(k) = O(k(\log k)^2(\log \log k))$$

$r = \text{rank}(A)$*, $U \in \mathbb{Z}^{n \times n}$ is unimodular and $V \in \mathbb{Z}^{m \times m}$.*

The need for terms such as $B(k)$ above is due to the fact that Storjohann frequently accommodates for bounds on integer bit-wise representations in our matrix. For example, in

[17] he states that a 1000×1000 matrix of integers with entries restricted to two decimals may lead to intermediate results with entries up to 3500 decimals. Thus, the $B(k)$ serves to bound the bit-wise representation of entries in terms of $\|A\|$. In general, one can see that the leading term of the equation above is quite similar to that of Gaussian elimination's complexity. Indeed, Smith normal form can be thought of as the generalization of Gaussian elimination over groups, and thus many of the standard row elimination methods are used or slightly adapted (since inversion is no longer well-defined). Many of the remaining logarithmic terms, as mentioned above, arise due to the representation of our matrices.

For the remainder of this chapter, we will refer to the poly-logarithmic term inside the bound above as $S_{dense}(A)$ for an integer matrix A. Since all other terms (i.e. n, m and r) are defined in terms of A, $S_{dense}(A)$ gives a well-defined bound on the number of operations to recover the Smith normal form of a dense integer matrix.

For sparse matrices, the complexity of computing the Smith normal form of an integer matrix $A \in \mathbb{Z}^{n \times m}$ has been proven to be

$$O(n^2 m \log \|A\| + n^3 \log^2 \|A\|)$$

by Giesbrecht in [6]. The proof of this bound will be omitted since it requires a significant amount of prerequisite material not covered in this thesis. We will simply refer to this polynomial as S_{sparse} for the remainder of the chapter.

Ultimately, the reader should note that both computations take the number of rows and columns into account when evaluating the Smith normal form. Now if we compare Morse homology to singular homology, it is worth noting that the matrix representation of our simplicial boundary map ∂_p takes the form of a $|K_p| \times |K_{p-1}|$ integer matrix, while the matrix representation of $\tilde{\partial}_p$ takes the form of an $m_p \times m_{p-1}$ integer matrix. In particular, the matrix representation of our simplicial boundary map ∂_p is a matrix whose entries are either 1 or -1. Moreover, that matrix is sparse whenever $|K_p| > 2\binom{p+1}{p}$ for all $0 \leq p \leq \dim K$ (since, for a fixed $\tau \in K_p$, there will be more $\sigma \in K_{p-1}$ faces with a zero coefficient in the sum $\partial_p(\tau)$ than with coefficient of 1 or -1).

It should be evident to the reader at this point that, for a given simplicial complex K, the computation of Morse homology on K is far more efficient than the simplicial homology on K whenever $m_p \ll |K_p|$. Since the critical faces of a discrete gradient vector field V correspond to the unmatched vertices in $\mathcal{H}_K \backslash \mathcal{M}_V$, one wishes to find a maximal matching $\mathcal{M}'$ on $\mathcal{H}_K$ that does not induce any cycles. This problem of finding a maximal acyclic matching $\mathcal{M}'$ is a well-known problem in the field of algebraic combinatorics, commonly referred to as the Maximal Morse Matching Problem (MMMP). In 2006, the Maximal Morse Matching Problem was shown to be NP-hard by Joswig and Pfetsch [8] in the following context:

Definition 3.4 (NP-hard). *Let L be a decision problem (that is, a representation of a formal language). We say that L is NP-hard if, for every H in NP, there exists a polynomial reduction of H to L.*

One immediate takeaway of a maximal matching $\mathcal{M}'$ is that the number of paths going into any p face will likely be large. For example, when $V = 0$ there is at most one path going from a fixed $(p+1)$-face to a fixed p-face. However, as one begins to add more pairs $\{\alpha \prec \beta\}$ to V, we can see that it is now possible for numerous paths to go from a fixed $(p+1)$-face to a fixed p-face. Therefore, there is no guarantee that our matrix representing $\tilde{\partial}_{p+1}$ will be sparse, so we assume without loss of generality that it is dense. Conversely, we may assume that our simplicial complex K is sufficiently large so that the matrix representation of ∂_{p+1} is sparse.

Now if we take $k = \max\{|K_{p-1}|, |K_p|\}$ and $m = \max\{m_{p-1}, m_p\}$, we have that

$$S_{\text{dense}}(\tilde{D}_{p+1}) \sim O(m^4 \log m)$$
$$S_{\text{sparse}}(D_{p+1}) \sim O(k^3 \log k)$$

This indicates that the computation of simplicial homology will asymptotically outgrow the computation of Morse homology for the class of simplicial complexes which satisfy $m = O(k^{3/4})$. Since we have that the number of critical cells on K is equal to r if and only if the matching $\mathcal{M}$ on $\mathcal{H}_K$ is of size

$$|\mathcal{M}| = \frac{|K| - r}{2}$$

we define the class of Morse-ideal simplicial complexes as follows:

Definition 3.5 (Morse-Ideal)**.** *We say that a simplicial complex K is Morse-ideal if there exists a matching $\mathcal{M}$ on $\mathcal{H}_K$ with*

$$|\mathcal{M}| \geq \frac{|K| - |K|^{3/4}}{2}$$

We define the class $M_{\mathcal{I}}$ to be the set of all Morse-ideal simplicial complexes.

It follows that a Morse-ideal simplicial complex K is a simplicial complex with a discrete Morse function that has at most $|K|^{3/4}$ critical faces. For a fixed simplicial complex, we clearly have that $|K|$ is fixed — thus, its does not make sense to say that the complexity of computing $\tilde{H}_p(K;\mathbb{Z})$ is any better than the complexity of computing $H_p^{\text{simp}}(K;\mathbb{Z})$. However, we are interested in how close the class $M_{\mathcal{I}}$ is to the overall class of simplicial complexes.

In fact, if we introduce the notion of barycentric subdivision, we have that every simplicial complex K is homotopy equivalent to a simplicial complex K' in $M_{\mathcal{I}}$. This leads us to the following definition:

Definition 3.6 (Barycentric Subdivision)**.** *Let K be a simplicial complex. For each simplex $\Delta_n \subset K$ with vertices $v_1, \ldots, v_{n+1}$, we define the barycenter b_{Δ_n} of Δ_n to be the point*

$$\frac{1}{n+1}(v_1 + \cdots + v_{n+1})$$

We define the simplicial complex $B(K)$ to be the largest simplicial complex with the property that for every strictly increasing sequence $\sigma_0 \subset \cdots \subset \sigma_k$ of simplices in K, there is a simplex $\widetilde{\Delta_k} \in B(K)$ with vertices $b_{\sigma_0}, \ldots, b_{\sigma_k}$. The simplicial complex $B(K)$ is known as the barycentric subdivision of K.

Example 3.7. *Let $B^i(K)$ denote the iteration of the operator B exactly i times on the simplicial complex K. If we consider our simplicial complex to simply be the 2-simplex Δ_2, we have the following barycentric decompositions:*

In general, homotopy theory is extensive and requires background in topology (which will not be covered by this thesis). For the interested reader, an in-depth proof of the

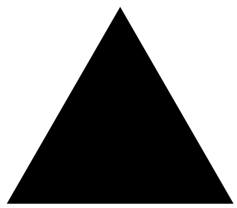


Figure 3.1: Original simplex Δ_2

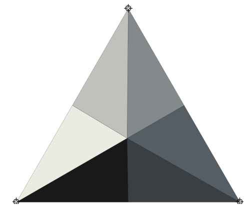


Figure 3.2: $B(\Delta_2)$

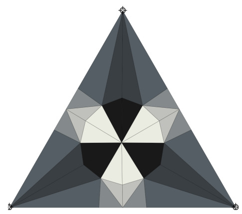


Figure 3.3: $B^2(\Delta_2)$

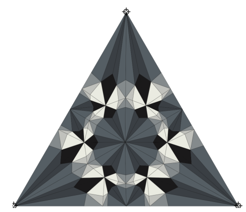


Figure 3.4: $B^3(\Delta_2)$

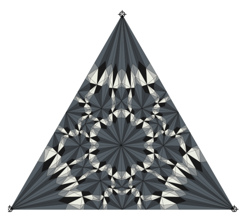


Figure 3.5: $B^4(\Delta_2)$

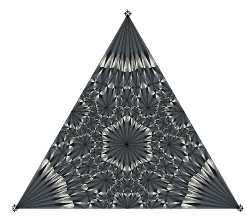


Figure 3.6: $B^5(\Delta_2)$

following lemma can be found in Proposition 2.21 in Allen Hatcher's *Algebraic Topology* [7].

Lemma 3.8. *If one extends the barycentric subdivision operator B linearly so that it is a morphism of chains $B : C_*(K;\mathbb{Z}) \to C_*(K;\mathbb{Z})$ for every simplicial complex K, then B is chain homotopic to the identity map.*

A direct consequence of the lemma above is that for any $N \in \mathbb{N}$ and simplicial complex K, we can find a simplicial complex K' with at least N faces (of any dimension less then or equal to $\dim K$) that has the same topological properties as K. From our previous chapters, we know that Morse homology is a topological invariant — therefore, one may consider how the barycentric subdivision affects our Morse-theoretic properties (i.e. discrete gradient vector field, critical cells, etc). Fortunately, Zhukova proved the following result in 2016 [20]:

Lemma 3.9. *Let K be a simplicial complex and $f : K \to \mathbb{R}$ a discrete Morse function on K. Then there exists a discrete Morse function $B_f : B(K) \to \mathbb{R}$ on the barycentric* *subdivision $B(K)$ satisfying the following conditions:*

(i) Every critical p-face of f contains exactly one critical p-face of B_f.

(ii) There is a bijection between the critical p-faces of K and critical p-faces of $B(K)$.

(iii) There is a bijection between the gradient paths on K and the gradient paths on. $B(K)$

Again, the proof will be omitted for the sake of brevity; however, the results of [20] answer our question of whether discrete Morse properties are invariant under barycentric subdivision. In particular, Lemma 3.8 and Lemma 3.9 give us the following theorem:

Theorem 3.10. *Let K be a simplicial complex and $f : K \to \mathbb{R}$ a discrete Morse function on K. Then there exists a simplicial complex K' with $K' \in M_{\mathcal{I}}$ satisfying the following conditions:*

(i) K' is homotopy equivalent to K.

(ii) There is a one-to-one correspondence between critical faces of K and critical faces of K'.

(iii) There is a one-to-one correspondence between gradient paths on K and gradient paths on K'.

Proof. We first show that the number of p-faces of $B^i(\Delta_n)$ is bounded below by 2^{i-1} for all $0 \leq p \leq n$ by induction on $n \in \mathbb{N}$. Without loss of generality, we may assume that $K = \Delta_n$ for some $n \geq 0$ (as otherwise, we may look at each simplex σ not contained in a larger $\tau \in K$ with $\sigma \prec \tau$).

For the base case, we consider Δ_1. It is easy to check that the number of edges of $B^i(\Delta_1)$ is twice the number of edges of $B^{i-1}(\Delta_1)$ and the number of vertices of $B^i(\Delta_1)$ is the number of vertices of $B^{i-1}(\Delta_1)$ plus the number of edges of $B^{i-1}(\Delta_1)$. If we let $B^0(\Delta_1) = \Delta_1$, we have that the number of edges of $B^i(\Delta_i)$ is exactly 2^i (since there is only one edge of Δ_1). Since the number of vertices of $B^i(\Delta_1)$ is defined in terms of the number of edges of $B^{i-1}(\Delta_1)$, which is exactly 2^{i-1}, we have that the number of 0-faces and 1-faces of $B^i(\Delta_1)$ is bounded below by 2^{i-1}.

Next, fix $n > 1$ and assume that the number of p-faces of $B^i(\Delta_k)$ is bounded below by 2^{i-1} for all $0 \le p \le k$ and $0 \le k \le n$. Consider the $(n+1)$-simplex Δ_{n+1}. There are exactly $\binom{n+2}{n+1} = n+2$ simplices of dimension n in Δ_{n+1} — label these simplices as $\Delta_{n,1}, \dots, \Delta_{n,n+2}$. By inductive hypothesis, the number of p-faces of $B^i(\Delta_{n,j})$ is bounded below by 2^{i-1} for all $0 \le p \le n$ and $1 \le j \le n+2$. Since $n > 1$, this implies that there are strictly greater than 2 maximal faces of Δ_{n+1}. Therefore, the number of p-faces of $B^i(\Delta_{n+1})$ is bounded below by 2^{i-1} for all $0 \le p \le n$. The rationale showing that the number of $(n+1)$-simplices of $B^i(\Delta_{n+1})$ is the similar to the argument above.

Let m be the number of critical cells on K (with respect to our discrete Morse function f). Since the number of p-cells of $B^i(K)$ is bounded below by 2^{i-1}, we can find some N such that the $m \le \left(2^{N-1}\right)^{3/4}$ and thus $m \le |B^N(K)|^{3/4}$. By Lemma 3.8 above, we have that $B^N(K)$ is homotopy equivalent to K. Furthermore, by applying Lemma 3.9 exactly N times we have that there exists a Morse function $B_{f,N} : B^N(K) \to \mathbb{R}$ that satisfies properties (ii) and (iii). By construction, $B_{f,N}$ only has m critical cells, so the matching $\mathcal{M}_B$ corresponding to $-\nabla B_{f,N}$ satisfies:

$$|\mathcal{M}_B| = \frac{|B^N(K)| - m}{2} \ge \frac{|B^N(K)| - |B^N(K)|^{3/4}}{2}$$

Therefore, $B^N(K) \in M_{\mathcal{I}}$. □

Ultimately, Theorem 3.10 above tells us a few important facts about our class $M_{\mathcal{I}}$ of Morse-ideal simplicial complexes. First off, our class $M_{\mathcal{I}}$ is non-empty due to the example pictured in Figure 3.7. By our theorem above, we have that $B^n(\Delta_3) \in M_{\mathcal{I}}$ as well for all

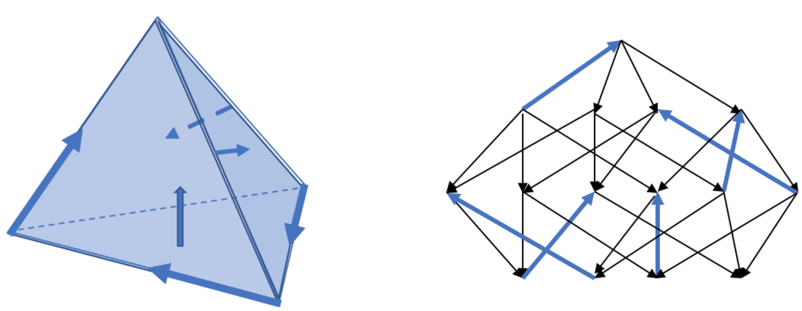


Figure 3.7: Maximal matching on Δ_3

$n \in \mathbb{N}$ — therefore, we are able to conclude that $M_{\mathcal{I}}$ is infinite.

One caveat of Theorem 3.10 is that the complexity of computing the Morse homology of $B^i(K)$ becomes exponentially harder as i becomes large. That is, we constructed the class $M_{\mathcal{I}}$ to describe simplicial complexes for which the computation of Morse homology is more efficient than the computation of simplicial homology. However, that does not mean that given a simplicial complex $K \notin M_{\mathcal{I}}$, it is more efficient to compute the Morse homology of $B^N(K)$ than the simplicial homology of K.

Despite the shortcoming mentioned above, it can be easily shown that many low-dimensional simplicial complexes lie inside our class $M_{\mathcal{I}}$ (e.g. dunce cap, real projective plane). However, one question which remains is whether the class of Morse-ideal simplicial complexes, when restricted to higher dimensions, consists of only barycentric subdivisions. Unfortunately, the contents of this thesis will not shed any more light on the nature of our class $M_{\mathcal{I}}$ due to a lack of constraints. Fortunately, the results of Rathod et al. [14] show that the matchings required for $M_{\mathcal{I}}$ are approximable in polynomial time. Indeed, if one were to restrict focus to, say, manifolds, then our picture of the Morse-deal complexes may expand significantly. Ultimately, we conclude this thesis by leaving the structure of $M_{\mathcal{I}}$ as an open problem for readers to explore.

Bibliography

[1] Thomas Banchoff. Critical Points and Curvature for Embedded Polyhedral Surfaces. *The American Mathematical Monthly*, 77:475–485, 1970.

[2] Jorgen Bang-Jensen and Gregory Gutin. *Digraphs, Theory, Algorithms and Application.* Springer-Verlag, 2007.

[3] B. Jack Copeland. The Church-Turing Thesis. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Metaphysics Research Lab, Stanford University, spring 2019 edition, 2019.

[4] Robin Forman. Morse Theory for Cell Complexes. *Advances in Mathematics*, 134:90–145, 1998.

[5] Robin Forman. A User's Guide to Discrete Morse Theory. *Séminaire Lotharingien de Combinatoire*, 48:35, 2002.

[6] M. Giesbrecht. Fast Computation of the Smith Form of a Sparse Integer Matrix. *computational complexity*, 10(1):41–69, Nov 2001.

[7] Allen Hatcher. *Algebraic topology.* Cambridge Univ. Press, Cambridge, 2000.

[8] Michael Joswig and Marc E. Pfetsch. Computing Optimal Morse Matchings. *SIAM Journal on discrete mathematics*, 20:11–25, 2006.

[9] Ravindran Kannan and Achim Bachem. Polynomial Algorithms for Computing the Smith and Hermite Normal Forms of an Integer Matrix. *SIAM Journal on Computing*, 8:499–507, 1979.

[10] Kevin P. Knudson. *Morse Theory Smooth and Discrete.* World Scientific, 2015.

[11] John Milnor. Morse Theory. *The Annals of Mathematical Studies*, 51, 1963.

[12] John Milnor. On Rational Maps with Two Critical Points. *Experimental Mathematics*, 9:480–522, 2000.

[13] J.R. Munkres. *Elements of Algebraic Topology.* Avalon Publishing, 1996.

[14] Abhishek Rathod, Talha Bin Masood, and Vijay Natarajan. Approximation algorithms for max-morse matching. *CoRR*, abs/1604.04707, 2016.

[15] John E. Savage. *Models of Computation: Exploring the Power of Computing.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1997.

[16] Amir Shpilka. Lower Bounds for Matrix Product. *CoRR*, cs.CC/0201001, 2002.

[17] Arne Storjohann. *Algorithms for Matrix Canonical Forms.* PhD thesis, University of Waterloo, 2000.

[18] Volker Strassen. Gaussian Elimination is not Optimal. *Numer. Math.*, 13(4):354–356, August 1969.

[19] John Henry C. Whitehead. On C1-Complexes. *Annals of Mathematics*, 41:809–824, 1940.

[20] Alena Zhukova. Discrete Morse Theory for the Barycentric Subdivision. *Journal of Mathematical Sciences*, 232(4), 2018.