The Hole System of Triangulated Shapes

by

Katharina Ölsböck

February, 2020

A thesis presented to the
Graduate School
of the
Institute of Science and Technology Austria, Klosterneuburg, Austria
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Institute of Science and Technology

The thesis of Katharina Ölsböck, titled *The Hole System of Triangulated Shapes*, is approved by:

Supervisor: Herbert Edelsbrunner, IST Austria, Klosterneuburg, Austria

Signature: _____

Committee Member: Bernd Bickel, IST Austria, Klosterneuburg, Austria

Signature: _____

Committee Member: Helmut Pottmann, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

Signature: _____

Defense Chair: Beatriz Vicoso, IST Austria, Klosterneuburg, Austria

Signature: _____

signed page is on file

IST Austria Thesis, ISSN: 2663-337X

I hereby declare that this thesis is my own work and that it does not contain other people's work without this being so stated; this thesis does not contain my previous work without this being stated, and the bibliography contains all the literature that I used in writing the dissertation.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee, and that this thesis has not been submitted for a higher degree to any other university or institution.

I certify that any republication of materials presented in this thesis has been approved by the relevant publishers and co-authors.

Signature: _____

Katharina Ölsböck

February, 2020

signed page is on file

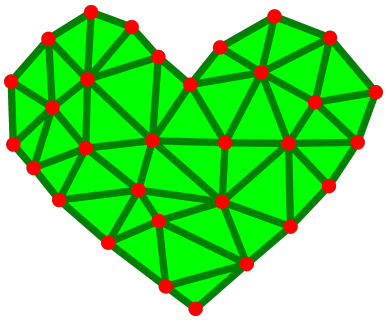
Abstract

Many methods for the reconstruction of shapes from sets of points produce *ordered simplicial complexes*, which are collections of vertices, edges, triangles, and their higher-dimensional analogues, called *simplices*, in which every simplex gets assigned a real value measuring its size. This thesis studies ordered simplicial complexes, with a focus on their *topology*, which reflects the connectedness of the represented shapes and the presence of holes. We are interested both in understanding better the structure of these complexes, as well as in developing algorithms for applications.

For the Delaunay triangulation, the most popular measure for a simplex is the radius of the smallest empty circumsphere. Based on it, we revisit Alpha and Wrap complexes and experimentally determine their probabilistic properties for random data. Also, we prove the existence of tri-partitions, propose algorithms to open and close holes, and extend the concepts from Euclidean to Bregman geometries.

Acknowledgments

Danksagung



List of Publications

The main part of this thesis is or will be published in the following articles: [21, 24, 23, 25]

- Herbert Edelsbrunner, Anton Nikitenko, Katharina Ölsböck, and Peter Synak. Poisson-Delaunay mosaics and related complexes experimentally. In *Proceedings of the Abel Symposium 2018*, to appear.
- Herbert Edelsbrunner and Katharina Ölsböck. Tri-partitions and bases of an ordered complex. *Discrete and Computational Geometry*, to appear.
- Herbert Edelsbrunner and Katharina Ölsböck. Holes and dependences in an ordered complex. *Computer Aided Geometric Design*, 73:1–15, 2019.
- Herbert Edelsbrunner, Katharina Ölsböck, and Hubert Wagner. Shape reconstruction in information space. Manuscript, IST Austria, 2019.

Table of Contents

Abstract v

Acknowledgments vii

List of Publications ix

Table of Contents xi

List of Figures xiii

List of Tables xvii

Notation xix

1 Introduction 1

2 Background 7

2.1 Discrete Geometry 7

2.2 Computational Topology 11

2.3 Discrete Morse Theory and Wrap Complex 15

3 Computation 19

3.1 Computing the Delaunay Radius Function 19

3.2 Computing the Wrap Complex 20

3.3 Dynamic Updates 23

3.4 Interval Reconstruction for Noisy Values 26

4 Stochastic Geometry and Topology **31**

4.1 Probabilistic and Computational Setting 31

4.2 Statistics for the Alpha Complex 33

4.3 Topological Statistics 38

4.4 Statistics for the Wrap Complex 38

5 Tri-Partition and Bases **45**

5.1 Example 45

5.2 Algorithms for Persistent Homology 47

5.3 Tri-Partition 51

5.4 Canonical Bases 53

5.5 Connection to Helmholtz-Hodge Theory 57

6 Manipulating Holes **59**

6.1 Changes to the Persistence Diagram 60

6.2 First Approach: Adding and Removing Basis Vectors 61

6.3 Dependence Structure 71

6.4 Second Approach: Recursive Operations 76

6.5 Statistics 82

7 Bregman Geometry **89**

7.1 Bregman Divergence 90

7.2 Bregman-Delaunay Triangulation 95

7.3 Rise Function 101

7.4 Fisher Metric 105

7.5 Computational Experiments 107

8 Conclusion **119**

Bibliography **127**

List of Figures

1.1 Convex hull, Delaunay triangulation, Alpha complex, Wrap complex, union of balls 2

2.1 Simplices and a simplicial complex 8
2.2 Voronoi diagram, Delaunay triangulation, empty circumcircles 9
2.3 Alpha complex 9
2.4 Weighted squared distance, weighted Delaunay triangulation 11
2.5 Examples of different kinds of chains and cochains 12
2.6 Persistence diagram 14
2.7 Wrap complex 17

3.1 Conflict zone 24
3.2 Dynamic update of the Wrap complex 26
3.3 Relaxed Wrap complex - Example 1 29
3.4 Relaxed Wrap complex - Example 2 29
3.5 Relaxed Wrap complex - Example 3 30

4.1 Periodic triangulation 32
4.2 Periodic Alpha complex 33
4.3 Estimated interval densities and variances in the Alpha complexes of a Poisson point process in $\mathbb{R}^2$ 36
4.4 Estimated interval densities and variances in the Alpha complexes of a Poisson point process in $\mathbb{R}^3$ 37

4.5 Estimated densities of critical simplices and experimentally observed Betti numbers in $\mathbb{R}^2$ . . . 39

4.6 Estimated densities of critical simplices and experimentally observed Betti numbers in $\mathbb{R}^3$ . . . 40

4.7 Estimated simplex and interval densities in the Wrap complexes of a Poisson point process in $\mathbb{R}^2$ . . . 42

4.8 Estimated simplex and interval densities in the Wrap complexes of a Poisson point process in $\mathbb{R}^3$ . . . 43

5.1 Tri-partition of the edge set of a 2-dimensional simplicial complex . . . 46

5.2 A reduced upper-triangular matrix . . . 48

5.3 Example of a canonical cycle and a canonical chain . . . 54

6.1 Manipulating a 1-dimensional hole . . . 60

6.2 Persistence diagram and hole manipulations . . . 61

6.3 Alpha complexes and persistence diagram of a point set in $\mathbb{R}^2$ . . . 63

6.4 Manipulating holes of Alpha complexes in $\mathbb{R}^2$ by adding and removing canonical basis vectors . . . 65

6.5 Alpha complexes and persistence diagram of Gramicidin A . . . 66

6.6 Manipulating holes in the Alpha complexes of Gramicidin A by adding and removing canonical basis vectors . . . 67

6.7 Manipulating holes of Wrap complexes in $\mathbb{R}^2$ by changing the set of critical simplices . . . 70

6.8 Wrap complexes of Gramicidin A . . . 71

6.9 Manipulating holes in the Wrap complexes of Gramicidin A by changing the set of critical simplices . . . 72

6.10 Swapping the adjacent columns i and j and the corresponding adjacent rows in R . . . 74

6.11 Manipulating holes of Alpha complexes in $\mathbb{R}^2$ with the recursive operations respecting the dependences between holes . . . 81

6.12 Locking and filling holes in the Alpha complexes of Gramicidin A with the recursive operations respecting the dependences between holes 83

6.13 Unlocking and unfilling holes in the Alpha complexes of Gramicidin A with the recursive operations respecting the dependences between holes 84

7.1 Geometric interpretation of the Bregman divergence 90

7.2 Geometric interpretation of the primal and the dual ball in Bregman geometry 91

7.3 Construction of the conjugate function with the polarity transform 92

7.4 Primal and dual Bregman-Voronoi diagram 96

7.5 Construction of the Voronoi diagram and the Delaunay triangulation by lifting the points 98

7.6 Construction of the weighted Voronoi diagram and the weighted Delaunay triangulation by lifting the points 98

7.7 Construction of the Bregman-Delaunay triangulation by lifting the points . 100

7.8 Illustration of the rise function ρ_F for lifted points in primal space 102

7.9 Illustration of the rise function ρ_F for lifted points in conjugate space . . . 103

7.10 Illustration of the rise function $\rho_{\overline{\omega}}$ for lifted points 104

7.11 Balls for the Fisher metric in Δ^2 and the corresponding Euclidean balls in $\mathbb{S}^2_+$ 106

7.12 Alpha complex of points in $(0, 2]^2$ in Shannon, conjugate Shannon, Fisher, Euclidean, and weighted Euclidean geometry 109

7.13 Wrap complex of points in $(0, 2]^2$ in Shannon, conjugate Shannon, Fisher, Euclidean, and weighted Euclidean geometry 110

7.14 Alpha complex of points in Δ^2 in Shannon, conjugate Shannon, Fisher, Euclidean, and weighted Euclidean geometry 112

7.15 Wrap complex of points in Δ^2 in Shannon, conjugate Shannon, Fisher, Euclidean, and weighted Euclidean geometry 113

7.16 Comparison of the Delaunay triangulations of points in $(0, 2]^2$ in Shannon, conjugate Shannon, Fisher, Euclidean, and weighted Euclidean geometry . 115

7.17 Comparison of the Delaunay triangulations of points in Δ^2 in Shannon, conjugate Shannon, Fisher, Euclidean, and weighted Euclidean geometry . 116

7.18 Delaunay triangulations color-coded by rise/radius function of points in $(0, 2]^2$ in Shannon, conjugate Shannon, Fisher, Euclidean, and weighted Euclidean geometry 117

7.19 Delaunay triangulations color-coded by rise/radius function of points in Δ^2 in Shannon, conjugate Shannon, Fisher, Euclidean, and weighted Euclidean geometry 118

8.1 Delaunay radii, Wrap radii, running times for dynamic insertion, relaxed Wrap complex 120

8.2 Numbers of simplices in the Alpha and Wrap complexes, Betti numbers . . 121

8.3 Partition of the edge set into a maximal 1-tree and a maximal 1-cotree . . 121

8.4 Filling the holes with small persistence with both approaches, resulting persistence diagram 122

8.5 Delaunay triangulation in Shannon, conjugate Shannon, and Fisher geometry, Wrap complex in Shannon geometry 123

List of Tables

4.1 Comparison of theoretical and experimental constants for numbers of intervals and simplices 35

4.2 Percentiles for the sizes of the lower sets of critical simplices in $\mathbb{R}^2$ and $\mathbb{R}^3$. 44

6.1 Average numbers of simplices in the Delaunay triangulation of a Poisson point process in $[0,1]^3$ 84

6.2 Comparison between the standard reduction algorithm and the exhaustive reduction algorithm for random points in $[0,1]^3$ 85

6.3 Average numbers of dependences of different types originating from a p-simplex 87

6.4 Average sizes of hole manipulating operations 88

7.1 Jaccard distances between Delaunay triangulations in Shannon, conjugate Shannon, Fisher, Euclidean, and weighted Euclidean geometry 111

Notation

$\mathbb{R}^d$	d-dimensional Euclidean space
$X \subseteq \mathbb{R}^d$	finite point set, point cloud
$\|x - y\|$	Euclidean distance of x, y
$B_r(x)$	closed ball of radius r centered at x
σ, τ	simplices
K	simplicial complex
$\dim \sigma$, $\dim K$	dimension of simplex, dimension of simplicial complex
$\mathrm{Vor}(x, X)$	Voronoi domain of x with respect to point set X
$\mathrm{Vor}(X)$	Voronoi diagram of X
$\mathrm{Del}(X)$	Delaunay triangulation of X
$\mathrm{Alpha}_r(X)$	Alpha complex of X for radius r
$\rho\colon X \to \mathbb{R}$	Delaunay radius function, radius of smallest empty circumsphere
w_x	weight of point x
$\pi_x(y)$	weighted squared distance from y to weighted point x
$\partial_p\colon \mathsf{C}_p \to \mathsf{C}_{p-1}$	p-th boundary map
$\mathsf{C}_p(K)$, $\mathsf{Z}_p(K)$, $\mathsf{B}_p(K)$	group of p-chains, p-cycles, p-boundaries of K
$\tilde{\mathsf{H}}_p(K)$	(reduced) p-th homology group of K
$\tilde{\beta}_p(K)$	(reduced) p-th Betti number of K
$\delta_p\colon \mathsf{C}^p \to \mathsf{C}^{p+1}$	p-th coboundary map
$\mathsf{C}^p(K)$, $\mathsf{Z}^p(K)$, $\mathsf{B}^p(K)$	group of p-cochains, p-cocycles, p-coboundaries of K
$\tilde{\mathsf{H}}^p(K)$	(reduced) p-th cohomology group of K

$[\sigma, \tau]$	interval of simplices with lower bound σ and upper bound τ
$\mathcal{G}$	interval graph
$\downarrow a$, $\downarrow \sigma$	lower set of interval a, lower set of critical simplex σ
$\mathrm{Wrap}_r(X)$, $\mathrm{Wrap}(X)$ ..	Wrap complex of X for radius r, for sufficiently large r
$\omega \colon X \to \mathbb{R} \cup \{\infty\}$	Wrap radius function
$\Omega \subseteq \mathbb{R}^d$, $\|\Omega\|$	Borel set, its measure
$\mathbb{P}(A)$	probability of event A
λ	density of a homogeneous Poisson point process
(ℓ, m)	interval type (dimension of lower and upper bound)
$\Gamma(\ell)$, $\gamma(m, s)$, $\tilde{\gamma}(m, s)$.	Gamma function, lower incomplete, regularized
ν_d	volume of the unit ball in $\mathbb{R}^d$
$c^d_{\ell,m}(r)$, $C^d_{\ell,m}$	expected number of intervals in Alpha complex, constant
$d^d_p(r)$, D^d_p	expected number of simplices in Alpha complex, constant
∂, Id	boundary matrix, identity matrix
$R = \partial U$, C	matrices after column reduction, book-keeping matrix
$Q = V\partial$, D	matrices after row reduction, book-keeping matrix
$\mathrm{low}(j)$, $\mathrm{left}(i)$	lowest one in column j, leftmost one in row i
$K_\ell = \{\sigma_1, \sigma_2, \ldots, \sigma_\ell\}$ ..	subcomplex of K
$\sqcup$	disjoint union of sets
$K^p = A_p \sqcup A^p \sqcup E_p$ ...	tri-partition of the p-simplices into a maximal p-tree, a maximal p-cotree, and a third set
n_p	number of p-dimensional simplices
n°_p, $n^\bullet_p$	number of birth simplices, death simplices
$\mathsf{z}_p(\sigma)$, $\mathsf{c}_p(\sigma)$	canonical cycle, canonical chain
$\mathsf{z}^p(\sigma)$, $\mathsf{c}^p(\sigma)$	canonical cocycle, canonical cochain
$\mathtt{Lock}_p$, $\mathtt{Fill}_p$	lock, fill operation for a p-hole
$\mathtt{Unlock}_p$, $\mathtt{Unfill}_p$	unlock, unfill operation for a p-hole
δ, BD, DB, BB, DD ...	forward dependences

$\partial, \mathsf{BD}^T, \mathsf{DB}^T, \mathsf{BB}^T, \mathsf{DD}^T$	backward dependences
$\mathcal{P}$	dependence structure
ι_0, ι	original ordering, permutation
$F\colon \Omega \to \mathbb{R}$	Legendre type function with domain $\Omega \subseteq \mathbb{R}^d$
∇F	gradient of F
$\langle x, y \rangle$	scalar product of x and y
$D_F(x\|y)$	Bregman divergence from x to y associated with F
$B_F(x, h)$, $B_F^*(x, h)$	primal, dual Bregman ball of size h centered at x
C^*	image under the polarity transform of a point $C \in \mathbb{R}^{d+1}$
	or an affine map $C\colon \mathbb{R}^d \to \mathbb{R}$
$F^*\colon \Omega^* \to \mathbb{R}$, $x^* \in \Omega^*$ ..	conjugate function of F, point in its domain
$\varpi\colon \mathbb{R}^d \to \mathbb{R}$, $D_\varpi(x\|y)$.	half the squared Euclidean norm, distance
$E\colon \mathbb{R}_+^d \to \mathbb{R}$, $D_E(x\|y)$.	Shannon entropy, Kullback-Leibler divergence
Δ^{d-1}	standard $(d-1)$-simplex
$E_\Delta\colon \Delta^{d-1} \to \mathbb{R}$	Shannon entropy restricted to standard simplex
$\mathrm{Vor}_F(x, X, \Omega)$	primal Voronoi domain associated with F of x
	with respect to point set X
$\mathrm{Vor}_F^*(x, X, \Omega)$	dual Voronoi domain
$\mathrm{Vor}_F(X, \Omega)$	(primal) Bregman-Voronoi diagram
$\mathrm{Del}_F(X, \Omega)$	Bregman-Delaunay triangulation
$\bar{X}$	lifted point set
$t_x\colon \mathbb{R}^d \to \mathbb{R}$	tangent hyperplane at x
$\rho_F\colon \mathrm{Del}_F(X, \Omega) \to \mathbb{R}$ ..	rise function for Bregman-Delaunay triangulation
$\iota\colon \mathbb{R}_+^d \to \mathbb{R}_+^d$	isometry between Fisher metric and Euclidean metric
$\mathbb{S}_+^{d-1}$	positive part of $(d-1)$-sphere with radius $\sqrt{2}$ centered
	at origin

Chapter 1

Introduction

What is the shape of a set of points, are there any holes? At first, it seems unreasonable to ask these questions. A set of points, or *point cloud*, is just a collection of 0-dimensional objects with no connections. But in many applications these points are samples of a higher-dimensional object, e.g. a surface in 3-dimensional space, and the goal is to reconstruct and analyze the shape of that object.

In order to recover the higher-dimensional structure, we need to connect the points. A very coarse representation is the *convex hull*, which is the collection of all points in between two or more points in the set. Many shape reconstruction techniques compute the *Delaunay triangulation*, which divides the *convex hull* into simpler pieces, called *simplices*, like triangles and tetrahedra, and choose a subset of them to represent the object. The *Alpha complex* and the *Wrap complex* are two examples. Figure 1.1 illustrates the convex hull, the Delaunay triangulation, an Alpha complex, and a Wrap complex for a 2-dimensional point set.

Having computed a higher-dimensional structure from the set of points, we can analyze its shape. From a geometrical point of view, this means studying all the properties that do not change after rotations, translations or uniform scaling of the object. However, this picture might be too detailed for many purposes. *Topology* takes a coarser view. Topological properties of an object are those that do not change even when the object is deformed without gluing or tearing. They reflect the connectedness of the object and the

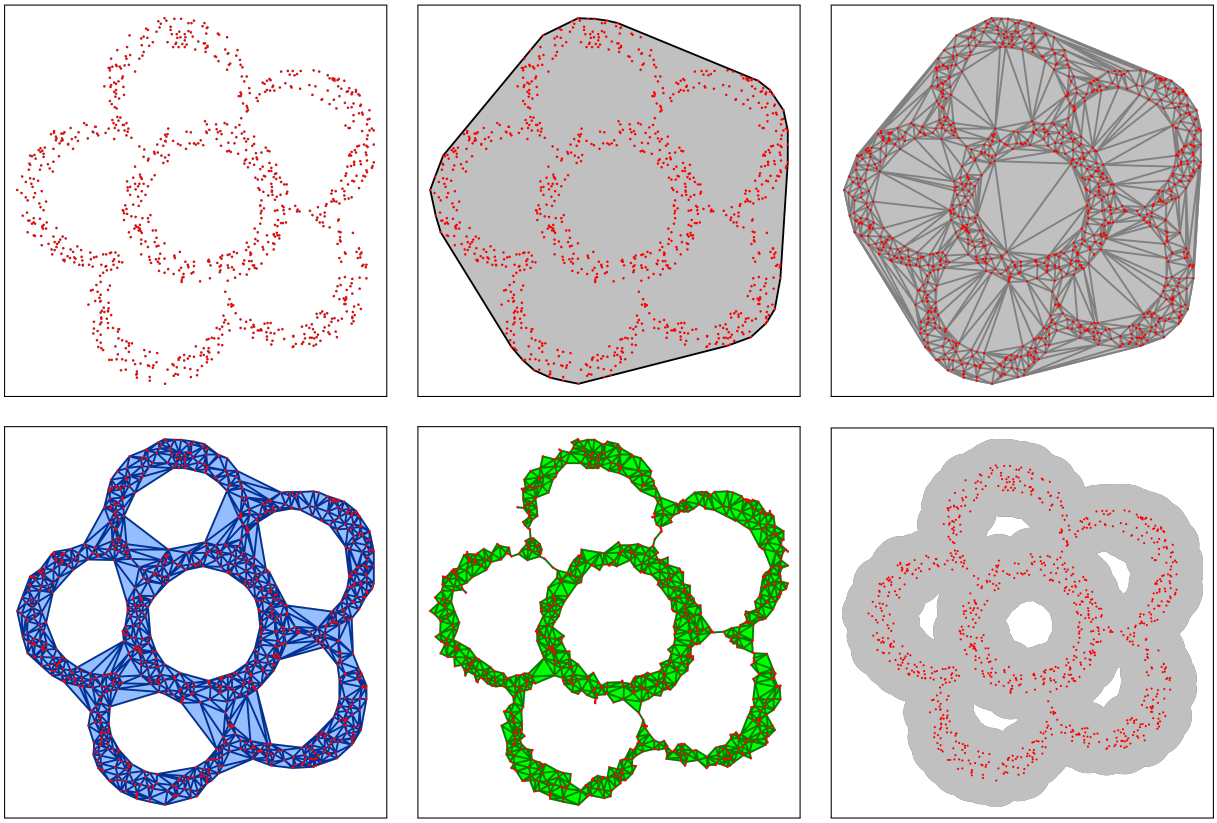


Figure 1.1: *First row:* A 2-dimensional point cloud, its convex hull, and its Delaunay triangulation. *Second row:* The Alpha complex, Wrap complex, and union of (2-dimensional) balls for the same radius.

presence of holes. Even when the points do not represent a solid object, but are samples from some kind of data, topology can help to retrieve useful information about the data.

This thesis extends prior work on shape reconstruction and analysis, with a focus on topology, in particular the hole system of the resulting shape. The Alpha and Wrap complexes are of particular interest in this context, as they have the same holes as the union of points blown up to balls of a given radius (see Figure 1.1). The *union of balls* is an important model in the study of biomolecules, where holes play an essential role for their functionality.

In this work, we propose algorithms to compute the Alpha and Wrap complexes, study their combinatorial and topological properties, analyze the dependence structure of holes, formulate ways to manipulate holes, and generalize the concepts to a non-Euclidean setting. Many of the results apply not only to Alpha and Wrap complexes but to *ordered complexes* in general, where some order is given on a collection of simplices.

The contributions of this thesis are of structural, algorithmic, and experimental nature. The structural contributions are the following:

- a tri-partition of the p-simplices of an ordered complex that can be used to construct canonical bases of the cycle, boundary, and homology groups,
- a dependence structure between the holes of an ordered complex,
- the extension of the notion of the Alpha and the Wrap complex to Bregman geometry.

Furthermore, we propose the following new algorithms, which we have implemented for points in $\mathbb{R}^2$ and $\mathbb{R}^3$: [1]

- an algorithm to compute the Wrap complexes of a point set for all $r \in \mathbb{R}$,
- an algorithm to update the Wrap complex locally after the insertion or deletion of a point,
- an algorithm to reconstruct the interval partition for a noisy generalized discrete Morse function, which can be used to compute the Wrap complex for noisy data or a relaxed version of it,
- operations to open and close holes in an ordered complex,
- algorithms to compute the Alpha and the Wrap complex in Bregman geometry.

The experimental contributions are in between the first two categories:

- statistics of combinatorial and topological properties of the Alpha and the Wrap complexes for Poisson distributed points,
- statistics about the sizes of the structures relevant for the hole manipulating operations, including the canonical bases and the dependence structure of holes.

[1] `wrap_2_3`, download from `https://git.ist.ac.at/katharina.oelsboeck/wrap_2_3-public.git` or `https://bitbucket.org/koelsboe/wrap_2_3-public.git`

Prior and Related Work. In recent years, a lot of work has been done on shape and in particular surface reconstruction from point clouds, because many applications, e.g. in medicine, geology, or engineering, require a digital representation of real-word objects. The point clouds are obtained with scanner devices or they are computed from multiple images. The state of the art in surface reconstruction is surveyed in [37], [4], and [9], where the last one focuses on methods based on the Delaunay triangulation.

The focus of our work is on the topology of the reconstructed shape. The Alpha complex has the same topology as the union of balls [17], which is frequently used for modeling molecules, and it has been shown to correctly reconstruct the topology of a surface, even in case of noisy sampling or sparse noise-free sampling [43]. A particular interest of this thesis is in the Wrap complex, which is a subset of the Alpha complex with the same topology [2]. It has been successfully used for surface reconstruction in practice, and geometric and topological guarantees for surface reconstruction in $\mathbb{R}^3$ were proved for a slightly modified version of it [40].

Topological principles cannot only be used for the reconstruction of shapes, but also for their analysis. For example, [38] analyzes the pore-geometry of materials with topological tools. In general, the field of *topological data analysis* provides tools, most prominently *persistent homology*, for understanding and quantifying global properties of data [20, 30].

Related to topology, discrete Morse theory [31] and its generalized version [32] shed a new light on simplicial complexes for which the function that assigns a radius value to the simplices fulfills certain properties. For points in general position, the radius function of the Delaunay triangulation is such a generalized discrete Morse function, and this insight gave rise to a new definition of the Wrap complex [2], which is much simpler than the original definition in [19] and is an important basis of this thesis.

Generalized discrete Morse theory also helped to reveal stochastic properties of the Alpha complex [22]. Since little has been known about the combinatorial properties of the Wrap complex of random point sets and it has proven difficult to compute the topological properties [41], we follow an experimental approach in this matter.

Turning our attention to general ordered simplicial complexes, the tri-partition of the p-simplices, which is introduced in this work, is a generalization of similar results for the edge set of a graph embedded in an orientable surface [6, 42]. It is related to the algebraic version of the Helmholtz-Hodge decomposition [16].

Regarding the extension of computational geometric concepts to non-Euclidean spaces, previous works [8, 28] have extended several concepts such as the Voronoi diagram and the Delaunay triangulation to the family of Bregman geometries. For many applications, the most interesting member of this family is the Shannon geometry, in which dissimilarity is measured by the Kullback-Leibler divergence [27]. We continue this work by studying Alpha and Wrap complexes in Bregman geometries.

Outline. Chapter 2 provides the necessary background for this thesis, including concepts from discrete geometry, computational topology, and discrete Morse theory. In Chapter 3 we explain how to compute the Wrap complex, and also present algorithms for a dynamic setting and for noisy data. Chapter 4 presents experimental results about Alpha and Wrap complexes on random point sets. Chapter 5 introduces the tri-partition of the p-simplices of an ordered simplicial complex and defines canonical bases for homology and cohomology groups. Chapter 6 analyzes the dependence structure of holes in an ordered simplicial complex and presents two approaches to opening and closing holes. Chapter 7 generalizes the Alpha and Wrap complexes to spaces in which dissimilarities are measured with a Bregman divergence.

Chapter 2

Background

This chapter provides the basic definitions and notation for the following work. In particular, it introduces concepts from discrete geometry, computational topology, and discrete Morse theory. For a more detailed introduction on these topics we refer to [18], [20], and [2], respectively.

2.1 Discrete Geometry

Let $X \subseteq \mathbb{R}^d$ be a finite set of points in d-dimensional Euclidean space, which we also call a *point cloud*. Write $\|x - y\|$ for the Euclidean distance of two points $x, y \in \mathbb{R}^d$ and $B_r(x) := \{y \in \mathbb{R}^d \mid \|x - y\| \leq r\}$ for the closed ball of radius $r \geq 0$ centered at $x \in \mathbb{R}^d$.

Simplicial Complex. In this work we only study simplicial complexes, which are well suited for computational purposes, although many of the results can be extended to more general cell complexes.

An *(abstract) p-simplex* is a set $\sigma \subseteq X$ of $p + 1$ points, where p is the *dimension* of the simplex. A 0-, 1-, 2-, or 3-simplex is also referred to as a *vertex*, *edge*, *triangle*, or *tetrahedron*, respectively. A simplex σ is the *face* of another simplex τ iff $\sigma \subseteq \tau$, and we call τ a *coface* of its faces. The $(p-1)$-dimensional faces of a p-simplex σ are called *facets*, and the $(p + 1)$-dimensional cofaces are called *cofacets*. Often, we identify an abstract

p-simplex, $\sigma = \{x_0, \ldots, x_p\}$, with its *geometric* counterpart in $\mathbb{R}^d$, which is the *convex hull* of the points in σ, $\operatorname{conv}\sigma := \{\sum_{i=0}^{p} \lambda_i x_i \mid \sum_{i=0}^{p} \lambda_i = 1 \text{ and } \lambda_i \geq 0 \ \forall i\}$.

An *(abstract) simplicial complex* K is a finite set of simplices, such that for every simplex $\sigma \in K$ all its faces are also elements of K. A geometric simplicial complex has the additional condition that the intersection of any two simplices is empty or a face of both. The *dimension* of K, $\dim K$, is the maximum dimension of any of its simplices.

A 0-, 1-, and 2-simplex, and a 2-dimensional simplicial complex are illustrated in Figure 2.1.

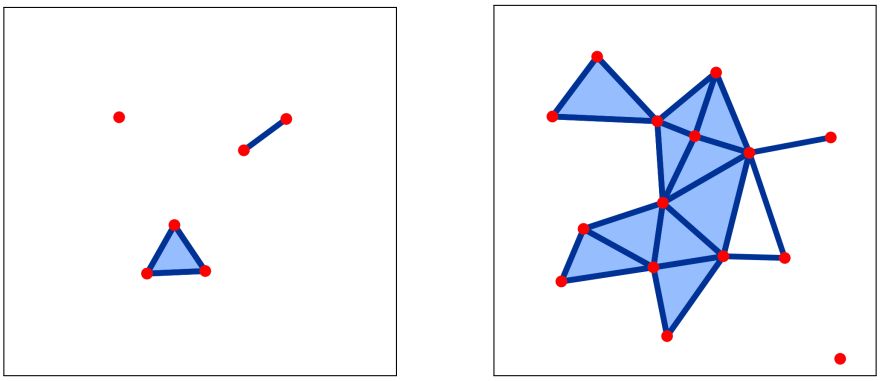


Figure 2.1: *Left:* A vertex, an edge, and a triangle. *Right:* A 2-dimensional simplicial complex.

General Position. In the following, we will assume that a given set of points is always in *general position*. For our purposes, this means that every subset $\sigma \subseteq X$ of at most $d+1$ points is affinely independent, and no point of $X \backslash \{\sigma\}$ lies on the $(\dim\sigma - 1)$-sphere defined by σ.

Voronoi Diagram and Delaunay Triangulation. The *Voronoi domain* of a point $x \in X$ with respect to X is defined as the set of all points $q \in \mathbb{R}^d$ that are at least as close to x as to any other point in X, $\operatorname{Vor}(x, X) := \{q \in \mathbb{R}^d \mid \|q - x\| \leq \|q - y\| \ \ \forall y \in X\}$. The collection of Voronoi domains of all points in X is the *Voronoi diagram* of X.

The *Delaunay triangulation* is dual to the Voronoi diagram. A set of points $\sigma \subseteq X$ is a simplex in the Delaunay triangulation $\operatorname{Del}(X)$ iff their Voronoi domains have a non-empty intersection. Equivalently, a d-simplex is in the Delaunay triangulation of X iff it has an *empty circumsphere*, which is a $(d-1)$-dimensional sphere such that all points of σ lie on the sphere and there are no points inside. With the assumption that the points

are in general position, it can be shown that the Delaunay triangulation can always be geometrically realized in $\mathbb{R}^d$.

The Voronoi diagram and the Delaunay triangulation are illustrated in Figure 2.2.

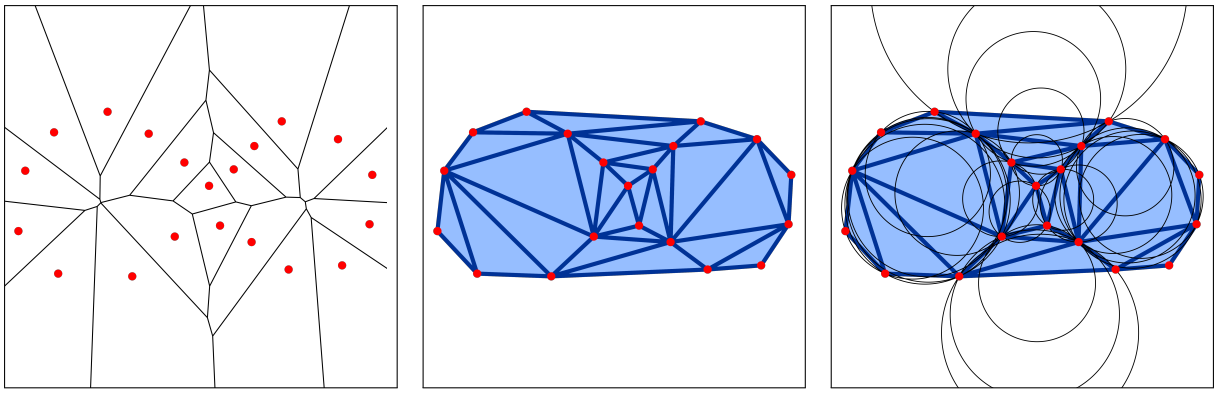


Figure 2.2: *Left:* Voronoi diagram. *Middle:* Delaunay triangulation. *Right:* Delaunay triangulation with empty circumcircles of triangles.

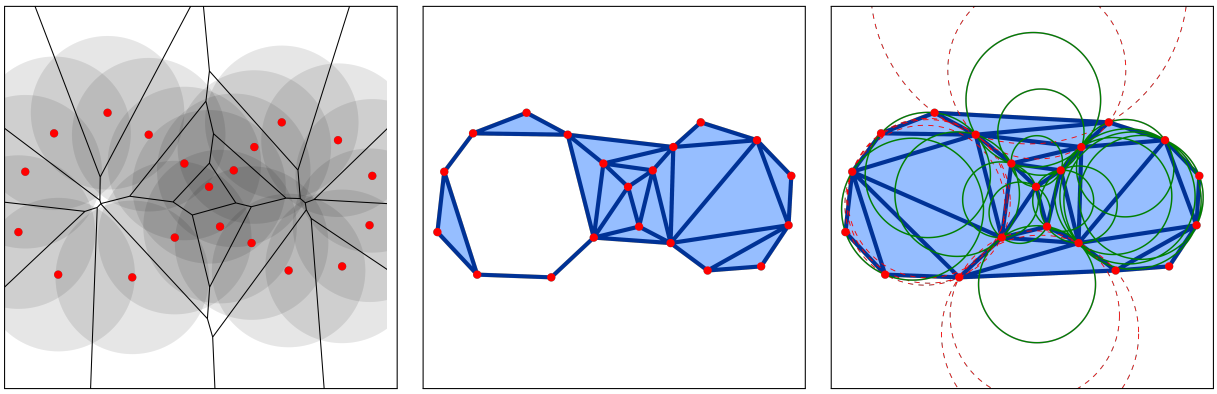


Figure 2.3: *Left:* Voronoi diagram with balls. *Middle:* Alpha complex. *Right:* Delaunay triangulation with the circumcircles of triangles in the Alpha complex drawn as solid green circles and the others as dashed red circles.

Alpha Complex. The *Alpha complex* of X for radius $r \geq 0$, $\mathrm{Alpha}_r(X)$, is a *subcomplex*, i.e., a subset that is also a simplicial complex, of the Delaunay triangulation of X. We intersect the Voronoi domains with closed balls of radius r centered at the points of X, and call them *Voronoi balls*, $\mathrm{Vor}_r(x, X) := \mathrm{Vor}(x, X) \cap B_r(x)$. The non-empty intersections of Voronoi balls give the simplices of the Alpha complex,

$$\mathrm{Alpha}_r(X) := \left\{ \sigma \subseteq X \mid \bigcap_{x \in \sigma} \mathrm{Vor}_r(x, X) \neq \emptyset \right\}.$$

Equivalently, $\text{Alpha}_r(X)$ contains all the simplices σ of the Delaunay triangulation of X for which the radius of the smallest empty circumsphere, which we call the *Delaunay radius* $\rho(\sigma)$, is smaller than or equal to r. Thus, all Alpha complexes are subcomplexes of the Delaunay triangulation, and $\text{Alpha}_r(X) \subseteq \text{Alpha}_s(X)$ for $r \leq s$.

The Alpha complex and its construction are illustrated in Figure 2.3.

Weighted Points. All concepts defined above can be extended to the case of *weighted points* in $\mathbb{R}^d$. Every point $x \in X$ is endowed with a weight $w_x \in \mathbb{R}$. The *weighted squared distance* or *power* of a point $q \in \mathbb{R}^d$ to x is defined as $\pi_x(q) := \|x - q\|^2 - w_x$. A point x with positive weight can be interpreted as the sphere with center x and squared radius w_x. For points q outside this sphere, the weighted squared distance to x is positive and equal to the squared length of a tangent line segment from q to the sphere (see the left panel of Figure 2.4). The weighted squared distance to x is zero for points on the sphere and negative for points inside the sphere.

Extending the previous definitions to weighted points, all occurrences of the (squared) Euclidean distance to a point in X are replaced by the weighted squared distance to a weighted point. For example, the *weighted Voronoi domain* of a weighted point x is the set of points $q \in \mathbb{R}^d$ with $\pi_x(q) \leq \pi_y(q)$ for all weighted points $y \in X$. For a sphere with radius r and center $z \in \mathbb{R}^d$, we say that a weighted point x lies on the sphere iff $\pi_x(z) = r^2$ and outside iff $\pi_x(z) > r^2$. Some domains in a weighted Voronoi diagram might be empty, and the corresponding point is not contained in the weighted Delaunay triangulation. As the weighted squared distance can be negative, we do not take any square roots, the weighted Delaunay radius function ρ^2 gives the weighted squared radii of the smallest empty circumspheres of the Delaunay simplices. A point set without weights can be considered as a special case of a weighted point set with all weights equal to 0.

The right panel of Figure 2.4 shows the weighted Delaunay triangulation for a point set with the same coordinates as in Figure 2.2, but with non-negative weights assigned.

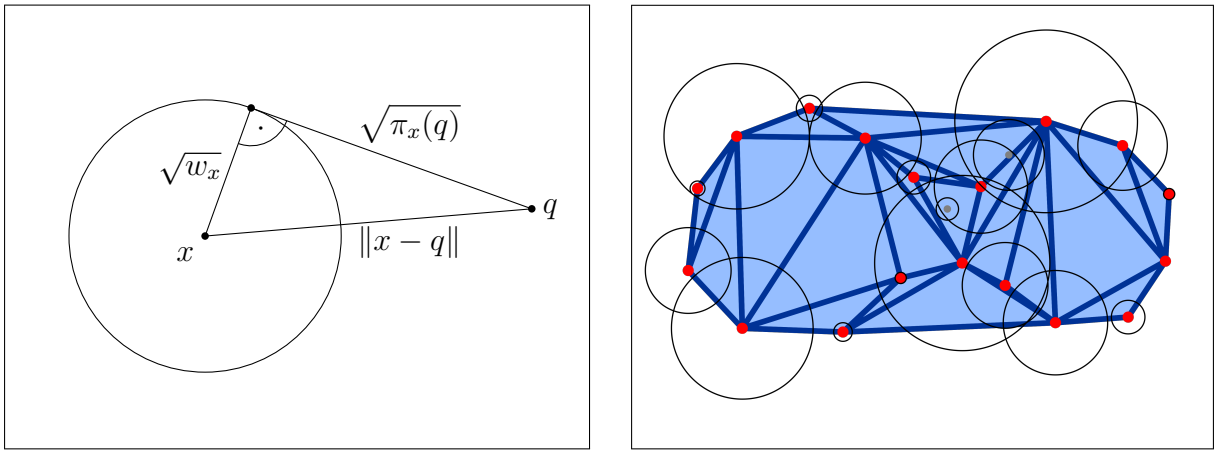


Figure 2.4: *Left:* Weighted squared distance to a point x with positive weight w_x. *Right:* Weighted Delaunay triangulation. We draw circles centered at the points with the squared radius equal to their weights. For visualization purposes, we have chosen all weights to be non-negative.

2.2 Computational Topology

We restrict ourselves to homology groups with $\mathbb{Z}/2\mathbb{Z}$-coefficients, in order to simplify the exposition.

Homology. Let K be a simplicial complex. Using modulo 2 coefficients, a *p-chain* is a (possibly empty) set of p-simplices of K. The *sum* of two p-chains is their symmetric difference. With this operation, the p-chains form a group, called the *chain group* $\mathsf{C}_p(K)$.

The *boundary* of a p-simplex σ is defined as the set of its $(p-1)$-dimensional faces, denoted by $\partial_p \sigma$. The *boundary map* $\partial_p \colon \mathsf{C}_p \to \mathsf{C}_{p-1}$ maps every p-chain to the sum of the boundaries of its simplices, which is the set of facets shared by an odd number of chain simplices.

A *p-cycle* is a p-chain with empty boundary. The set of p-cycles $\mathsf{Z}_p(K)$ forms a subgroup of the p-chains. A p-chain that is the boundary of a $(p+1)$-chain is called a *p-boundary*. Since $\partial_p \, \partial_{p+1} \, c = 0$ for every $(p+1)$-chain c, the set of p-boundaries $\mathsf{B}_p(K)$ is a subgroup of the p-cycles. A p-cycle in $\mathsf{B}_p(K)$ is sometimes referred to as *trivial*.

The *p-th homology group* $\mathsf{H}_p(K)$ is defined as the quotient of the groups of p-cycles and p-boundaries, $\mathsf{H}_p(K) := \mathsf{Z}_p(K)/\mathsf{B}_p(K)$. We refer to its rank as the *p-th Betti number* β_p.

The elements of $\mathsf{H}_p(K)$ are the *homology classes* of K, and we call two p-cycles *homologous* iff they belong to the same class, i.e., they only differ by a boundary.

In the following work, we use the convention that every simplicial complex contains the empty set, which is a simplex of dimension (-1) and is a face of every simplex. To indicate this, we decorate the symbols for homology groups and Betti numbers with a tilde, and call them *reduced homology groups* $\tilde{\mathsf{H}}_p(K)$ and *reduced Betti numbers* $\tilde{\beta}_p$. The main difference to conventional homology is that β_0 is the number of connected components, while $\tilde{\beta}_0$ is the number of gaps between connected components, which is generally 1 less than β_0, except when the complex is empty (consists of the (-1)-dimensional simplex only), in which case both vanish. In both cases, the homology classes of dimension 1, 2, and higher, correspond to tunnels, voids, and higher-dimensional *holes*, respectively.

Some chains, cycles, and boundaries of a simplicial complex are highlighted in the left panel of Figure 2.5.

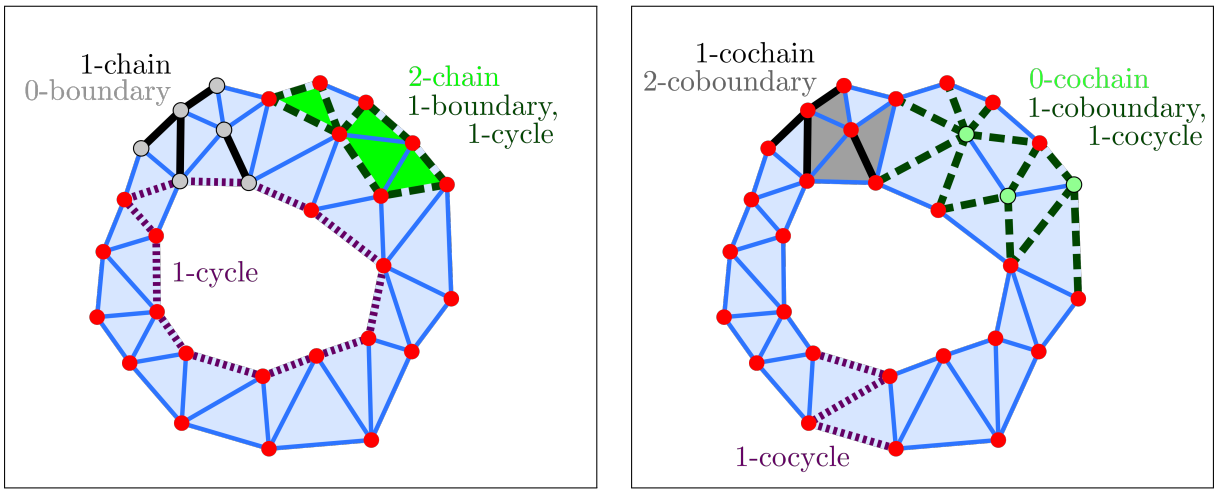


Figure 2.5: Examples of different kinds of chains and cochains in a 2-dimensional simplicial complex. *Left:* A 1-chain (black edges), a 2-chain (green triangles), and their respective boundaries (gray points and dashed, green edges). A 1-cycle (thinly dashed, purple edges). *Right:* A 0-cochain (light green points), a 1-cochain (black edges), and their respective coboundaries (dashed, green edges and gray triangles). A 1-cocycle (thinly dashed, purple edges).

Cohomology. In cohomology, we call a set of p-simplices a *p-cochain*. The *coboundary* of a p-simplex σ, denoted by $\delta_p \sigma$, is defined as the set of its $(p+1)$-dimensional cofaces. Thus, in contrast to homology, the *coboundary map* $\delta_p \colon \mathsf{C}^p \to \mathsf{C}^{p+1}$ goes up in dimension.

It maps a p-cochain to the sum of the coboundaries of its simplices, which is the set of cofacets shared by an odd number of cochain simplices.

A *p-cocycle* is a p-cochain with empty coboundary, and a *p-coboundary* is the coboundary of a $(p-1)$-cochain. For the corresponding cochain, cocycle, and coboundary groups, we have the same relationship as in homology: $\mathsf{C}^p(K) \supseteq \mathsf{Z}^p(K) \supseteq \mathsf{B}^p(K)$. Note that we write the dimension as superscript instead of subscript in cohomology in order to distinguish the symbols from their homological counterparts.

The *p-th (reduced) cohomology group* is defined as $\tilde{\mathsf{H}}^p(K) := \mathsf{Z}^p(K)/\mathsf{B}^p(K)$, i.e., two p-cocycles belong to the same cohomology class iff they differ by a p-coboundary. By a classic result, the ranks of homology and cohomology groups are the same for modulo 2 coefficients: $\tilde{\beta}_p = \tilde{\beta}^p$ [34, Section 3.3]. Some cochains, cocycles, and coboundaries of a simplicial complex are highlighted in the right panel of Figure 2.5.

Persistent Homology. When we study a finite sequence of nested simplicial complexes $\emptyset = K_0 \subseteq K_1 \subseteq \ldots \subseteq K_m = K$, called a *filtration* of K, we are interested in the homological features that persist across multiple scales, i.e., they exist in more than one complex of the filtration.

A filtration of K is usually given by means of a *monotonic* function (which is then also called a *filtration function*) $f\colon K \to \mathbb{R}$, which satisfies $f(\sigma) \leq f(\tau)$ if σ is a face of τ. Considering the sublevel sets $f^{-1}(-\infty, a]$ for $a \in \mathbb{R}$, we get a filtration of K. An example for this construction is the Delaunay triangulation of a point set, filtered by the Delaunay radius function, which gives the sequence of Alpha complexes.

For every $i \leq j$, the inclusion $K_i \subseteq K_j$ induces homomorphisms on the corresponding homology groups, $h_p^{i,j}\colon \tilde{\mathsf{H}}_p(K_i) \to \tilde{\mathsf{H}}_p(K_j)$. We define the *$p$-th persistent homology groups* as the images of these homomorphisms, $\tilde{\mathsf{H}}_p^{i,j} := \operatorname{im} h_p^{i,j}$, and the *$p$-th persistent Betti numbers* $\tilde{\beta}_p^{\,i,j}$ as the corresponding ranks.

Let γ be a homology class in $\tilde{\mathsf{H}}_p(K_i)$. It is said to be *born at K_i* iff $\gamma \notin \tilde{\mathsf{H}}_p^{i-1,i}$. In this case, it *dies entering K_j* iff $h_p^{i,j-1}(\gamma) \notin \tilde{\mathsf{H}}_p^{i-1,j-1}$ and $h_p^{i,j}(\gamma) \in \tilde{\mathsf{H}}_p^{i-1,j}$, i.e., it merges with an older class as we go from K_{j-1} to K_j. The *persistence* of a class γ is defined to be the difference of filtration values at birth and death of the class. We set the death value and the persistence of classes that never die to infinity. Intuitively, persistence measures how long the corresponding hole exists in the filtration.

When we are given a *monotonic ordering* of the simplices in K, where every simplex is preceded by its faces, we can identify the simplices that are "responsible" for the birth and death of a homology class. We get such a monotonic ordering by ordering the simplices by their filtration values, and when two simplices have the same value by their dimensions. Remaining ties are broken arbitrarily. For the monotonic ordering of simplices $\sigma_1, \sigma_2, \ldots, \sigma_m$ we get the corresponding filtration $\emptyset \subseteq K_1 \subseteq K_2 \subseteq \ldots \subseteq K_m$ with $K_\ell := \{\sigma_1, \sigma_2, \ldots, \sigma_\ell\}$. If a homology class γ is born at K_i and dies entering K_j, we say that σ_i *gives birth* to γ and that σ_j *gives death* to γ, and we call them the *birth* and the *death simplex* of the homology class.

We use the *persistence diagram* to visualize persistent homology. If a homology class is born at K_i and it dies entering K_j, the filtration values at birth and death, namely $f(\sigma_i)$ and $f(\sigma_j)$, are the coordinates of a point in the persistence diagram. The vertical distance of the point to the diagonal gives the corresponding persistence.

In Figure 2.6 we show the persistence diagram of the Delaunay triangulation of the point set of Figure 2.2, filtered by the Delaunay radius function, and the subcomplexes corresponding to the birth and the death of the 1-dimensional hole with highest persistence.

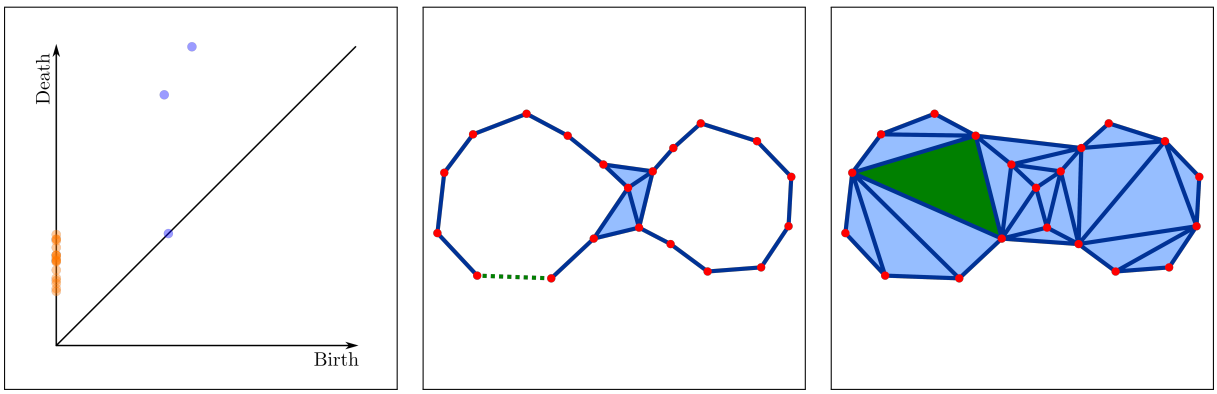


Figure 2.6: *Left:* Persistence diagram with the orange points on the death axis corresponding to homology classes of dimension 0 and the remaining blue points to classes of dimension 1. *Middle, Right:* Birth and death of homology class with highest persistence. The dashed green edge gives birth to the 1-dimensional class, the green triangle gives death to it.

2.3 Discrete Morse Theory and Wrap Complex

We use a generalization of the discrete Morse theory by Forman [31], which was first suggested by [32].

Homotopy Type. Let $\mathbb{X}, \mathbb{Y}$ be two topological spaces (e.g. subsets of the Euclidean space $\mathbb{R}^d$) and $f_1, f_2 \colon \mathbb{X} \to \mathbb{Y}$ two continuous maps. A continuous map $H \colon \mathbb{X} \times [0,1] \to \mathbb{Y}$ is a *homotopy* between f_1 and f_2 iff $H(x,0) = f_1(x)$ and $H(x,1) = f_2(x)$ for all $x \in \mathbb{X}$, and we write $f_1 \simeq f_2$.

We say that two topological spaces $\mathbb{X}$ and $\mathbb{Y}$ are *homotopy equivalent*, or have the same *homotopy type*, iff there are continuous maps $f \colon \mathbb{X} \to \mathbb{Y}$ and $g \colon \mathbb{Y} \to \mathbb{X}$ such that $g \circ f \simeq \mathrm{id}_{\mathbb{X}}$ and $f \circ g \simeq \mathrm{id}_{\mathbb{Y}}$.

If two topological spaces are homotopy equivalent, their homology groups are isomorphic [34].

Generalized Discrete Morse Theory. Let K be a simplicial complex. A subset of simplices of K of the form $[\sigma, \tau] := \{\nu \in K \mid \sigma \subseteq \nu \subseteq \tau\}$ is called an *interval*. A *generalized discrete vector field* V is a partition of K into intervals. Let f be a function $f \colon K \to \mathbb{R}$ that satisfies $f(\sigma) \leq f(\tau)$ whenever σ is a face of τ, where equality holds if and only if σ and τ belong to the same interval. Then we call f a *generalized discrete Morse function* and V its *generalized discrete gradient*. An interval is called *singular* iff it contains only one simplex, which we then call a *critical simplex* and its function value a *critical value*.

The critical simplices are of special importance, because they change the homotopy type of the complex when they are inserted or removed. On the other hand, the operation of removing a non-singular interval so that the result is still a simplicial complex (i.e. the interval simplices do not have cofaces which are not in the interval), which we call a *collapse*, does not change the homotopy type.

Wrap Complex. The Delaunay radius function $\rho \colon \mathrm{Del}(X) \to \mathbb{R}$ of a finite point set X in general position is a generalized discrete Morse function [2]. Therefore, we get a partition into intervals of simplices with the same Delaunay radius.

Let $\mathcal{G}$ be the directed graph whose nodes are the intervals of the Delaunay radius function, and with an arc from interval a to interval b iff there is a simplex in a which is a face of a simplex in b. This *interval graph* is the same as the graph of the face relation for simplices (which has an arc $\sigma \to \tau$ iff σ is a face of τ) where the nodes of simplices in the same interval were contracted to a single node.

The interval graph is acyclic, since the values of a generalized discrete Morse function strictly increase when going from one interval to another along an arc (the second interval contains a coface of a simplex in the first interval). We define the *lower set* of an interval a as the set of intervals from which it can be reached along a directed path in $\mathcal{G}$ and denote it by $\downarrow a$. The lower set of a critical simplex is the lower set of the corresponding singular interval.

The *Wrap complex* $\mathrm{Wrap}_r(X)$ of X for radius $r \in \mathbb{R}$ is defined as the set of simplices in the union of the lower sets of all critical simplices σ with $\rho(\sigma) \leq r$,

$$\mathrm{Wrap}_r(X) := \bigcup \bigcup_{\substack{\sigma \in \mathrm{Del}(X): \\ \sigma \text{ critical, } \rho(\sigma) \leq r}} \downarrow \sigma.$$

Equivalently, we get the Wrap complex $\mathrm{Wrap}_r(X)$ from the Alpha complex $\mathrm{Alpha}_r(X)$ by collapsing all non-singular intervals that can be collapsed.

Lemma 2.1 (Wrap Complex)**.** *An interval is in the lower set of a critical simplex in $Alpha_r(X)$ iff there is no sequence of collapses that removes it from $Alpha_r(X)$.*

Proof. First, we note that $\mathrm{Wrap}_r(X) \subseteq \mathrm{Alpha}_r(X)$ because the Delaunay radius of all the simplices in the Wrap complex cannot be larger than r. By definition, this is true for all the critical simplices, of which we consider their lower sets. Furthermore, the Delaunay radius function values of simplices in the lower set of an interval s need to be strictly smaller than $\rho(s)$, since ρ is a generalized discrete Morse function.

"$\Rightarrow$" We will show by induction that if there is a directed path of length $n \geq 2$ in $\mathcal{G}$ from interval a to the singular interval $s = \{\sigma\}$, then a cannot be collapsed. For $n = 2$, σ is the coface of a simplex in a. Thus, we cannot collapse a. Now, let there be a directed path of length $n + 1$ from a to s. Let b be the direct successor of a on this path. Then there is a directed path of length n from b to s, and by the inductive assumption there is no sequence of collapses that removes b. As there is a directed

arc from a to b, interval b contains the coface of a simplex in a, and we cannot collapse a as long as b is still in the complex.

"$\Leftarrow$" Now, let a be not in the lower set of any critical simplex. Consider the maximal directed paths in $\mathcal{G}$ starting at a. All intervals on these paths must be non-singular, otherwise a would be in the lower set of a critical simplex. We can collapse the last intervals on these paths. Their simplices do not have cofaces outside their own interval, otherwise there would be an arc from this interval which we could use to extend a maximal path. After this first round of collapses, all maximal paths are shorter by 1. We continue collapsing the intervals at the end of maximal directed paths starting at a, until we reach a itself and are free to collapse it.

□

This shows that $\text{Wrap}_r(X) \subseteq \text{Alpha}_r(X)$ and that the two complexes are homotopy equivalent, for all $r \in \mathbb{R}$. Often, we talk about the Wrap complex, $\text{Wrap}(X)$, without any parameter r, which corresponds to the Wrap complex for sufficiently large r.

Although the definition with collapses might be more intuitive, we prefer the first one for computing the Wrap complex. We can give an even more compact definition: The Wrap complex $\text{Wrap}_r(X)$ is the smallest subcomplex of $\text{Del}(X)$ that is a union of intervals and contains all critical simplices σ with $\rho(\sigma) \leq r$.

The first two definitions of the Wrap complex are illustrated in Figure 2.7.

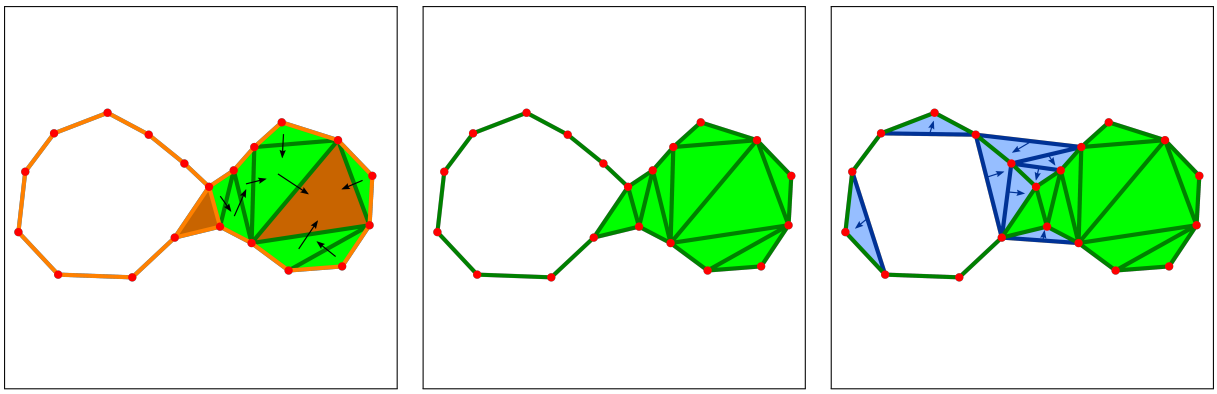


Figure 2.7: Wrap complex of the same point set and for the same radius as the Alpha complex in Figure 2.3. *Left:* The critical triangles and edges are highlighted in brown and orange. Arrows indicate the directed paths of $\mathcal{G}$ that give the lower sets of the critical triangles. *Right:* The blue triangle-edge pairs of the Alpha complex are collapsed.

Chapter 3

Computation

Having stated the definitions for the Alpha and the Wrap complex in the last chapter, we will now explain how to compute them efficiently. Furthermore, we will give algorithms for computing the Wrap complex in different settings: in a dynamic setting and for noisy input.

3.1 Computing the Delaunay Radius Function

Delaunay and Wrap Radius Function. We want to compute the Alpha and Wrap complexes of a point set X for all parameter values $r \in \mathbb{R}$ at once. So, what we actually want to compute are the Delaunay radius function ρ and the *Wrap radius function* ω on the set of simplices in $\mathrm{Del}(X)$. The Alpha and the Wrap complexes are the sublevel sets of these functions, $\mathrm{Alpha}_r(X) = \rho^{-1}(r)$ and $\mathrm{Wrap}_r(X) = \omega^{-1}(r)$, i.e., the functions give the parameter values for which the simplices first appear in the Alpha or the Wrap filtration. For those simplices $\sigma \in \mathrm{Del}(X)$ that do not belong to any Wrap complex, we set $\omega(\sigma) = \infty$.

Computing the Delaunay Triangulation. A natural way to start is to compute the Delaunay triangulation of X. We do not want to go into details here, since much work has already been done on this topic and many libraries provide efficient functions to compute the Delaunay triangulation. We just mention that in $\mathbb{R}^2$ the Delaunay triangulation

contains $\mathcal{O}(n)$ simplices and can be computed in $\mathcal{O}(n \log n)$ expected time [33], where n is the number of input points, and in $\mathbb{R}^d$ it can be computed in $\mathcal{O}(n \log n + n^{\lceil d/2 \rceil})$ expected time, where the second term is of the same order of magnitude as the maximum number of possible simplices [26]. The general strategy is to add the randomly sorted points one by one and perform local "flip" operations until the circumspheres of all maximal simplices are empty, which gives a valid Delaunay triangulation.

For our implementation, we use the CGAL library [10] to compute the Delaunay triangulation.

Computing the Delaunay Radius Function and its Intervals. We compute the Delaunay radius for every simplex, starting with the higher-dimensional ones. For a maximal simplex, the Delaunay radius is the radius of its unique circumsphere (which has to be empty). For a lower-dimensional simplex, we first compute the radius of its smallest circumsphere, then we check if any neighboring vertex, i.e., the other vertex of a cofacet, is inside (as we assume the points to be in general position, no neighboring vertex can lie exactly on the circumsphere). If this is the case, then the simplex has the same Delaunay radius as its cofacet with the smallest Delaunay radius, and they are in the same interval. From the last observation, we can deduce the intervals of the Delaunay radius function.

As every simplex has a constant number of facets and we touch each facet-simplex pair exactly once, computing the Delaunay radius function takes $\mathcal{O}(m)$ time, where m is the number of simplices in the Delaunay triangulation. The above described algorithm is given in pseudocode as Algorithm 1.

3.2 Computing the Wrap Complex

We compute the Wrap complex by using the first definition given in Section 2.3.

Computing the Interval Graph. We start by computing the interval graph $\mathcal{G}$ of the Delaunay radius function, see Algorithm 2. First, we get an explicit representation of the intervals from the information gathered in Algorithm 1. Then, we compute for each interval its predecessors in $\mathcal{G}$. The *predecessors* of a node a in a directed graph are all nodes b for which there is an arc $b \to a$. In our case, the predecessors of an interval are the

Algorithm 1 Computing Delaunay Radii and Intervals

max_interval_simplex(σ) := σ $\forall \sigma \in \mathrm{Del}(X)$ ▷ we compute the maximum simplex of the interval of each simplex

interval_faces(σ) := {} $\forall \sigma \in \mathrm{Del}(X)$

critical(σ) := `true` $\forall \sigma \in \mathrm{Del}(X)$

for $p \in \{d, \ldots, 0\}$ **do**

  for p-simplex $\sigma \in \mathrm{Del}(X)$ **do**

    c := `circumcenter`(σ); $r := \|c - \sigma[1]\|$

    for $\tau \in$ cofacets(σ) **do**

      $v := \tau \backslash \sigma$

      if $\|c - v\| < r$ **then**

        critical(σ) := `false`

    if critical(σ) **then**

      $\rho(\sigma) := r$

    else

      τ_0 := max_interval_simplex(argmin$_{\tau \in \mathrm{cofacets}(\sigma)}$ $\rho(\tau)$)

      $\rho(\sigma) := \rho(\tau_0)$

      max_interval_simplex(σ) := τ_0

      interval_faces(τ_0) := interval_faces(τ_0) $\cup$ {σ}

Algorithm 2 Computing Interval Graph

intervals := {} ▷ compute explicit representation of intervals

for $\sigma \in \mathrm{Del}(X)$ **do**

  if max_interval_simplex(σ) = σ **then**

    a := {σ} $\cup$ interval_faces(σ)

    max_simplex(a) := σ

    intervals := intervals $\cup$ {a}

    interval(τ) := a $\forall \tau \in a$

for $a \in$ intervals **do** ▷ compute predecessors in interval graph

  for $\tau \in$ faces(max_simplex(a))$\backslash a$ **do**

    predecessors(a) := predecessors(a) $\cup$ {interval(τ)}

intervals of those faces of the maximum simplex that are not themselves in the considered interval.

Algorithm 3 Computing Wrap Radii

```
ω(σ) := ∞  ∀σ ∈ Del(X)
critical_descendants(a) := {}  ∀a ∈ intervals        ▷ interval was found in lower sets of
                                                       these singular intervals
for a ∈ intervals do
    if |a| = 1 then                                  ▷ update radii in lower sets of critical simplices
        σ := max_simplex(a)
        ω(σ) := ρ(σ)
        for b ∈ predecessors(a) do
            updateWrapRadiiRecursively(a, b, ω(σ))

procedure updateWrapRadiiRecursively(a, b, r)
    if |b| > 1 and a ∉ critical_descendants(b) then    ▷ non-singular and not found yet
        critical_descendants(b) := critical_descendants(b) ∪ {a}
        if r < ω(b[1]) then
            ω(τ) := r  ∀τ ∈ b
            for c ∈ predecessors(b) do
                updateWrapRadiiRecursively(a, c, r)   ▷ recursively update in lower set
```

Computing the Wrap Radius Function. We compute the Wrap radii of the Delaunay simplices by traversing the lower sets of the critical simplices and updating the radii along the way. Initially, we set all Wrap radii to ∞. For every critical simplex, we set its Wrap radius to the Delaunay radius value (which is necessarily the same) and update the radius values in its lower set. Whenever an interval was found in the lower set of a critical simplex with a smaller radius than the current Wrap radius of its simplices, we update the Wrap radius. We recursively go to the predecessors of a lower set interval, but stop if we reach a singular interval (its radius value is necessary smaller) or if an interval was already found in the current lower set. The last condition is important to avoid traversing the same paths in $\mathcal{G}$ over and over again. For this purpose, we store for each non-singular

interval the set of critical simplices in the lower sets of which it was already found. We call them *critical descendants* of the interval. We do not include the critical simplices to which there is a directed path that contains other critical simplices, since they are not relevant for the radius computation.

When the algorithm stops, $\omega(\sigma)$ stores the Wrap radius of all simplices σ in the Wrap complex. For those simplices that are not contained in the Wrap filtration, the value is ∞. The details of the algorithm are given as Algorithm 3.

The algorithm takes $\mathcal{O}(m\,c)$ time, where m is the number of Delaunay simplices and c the maximum number of critical descendants of any non-singular interval. We could alternatively get a linear time algorithm by computing a topological sorting of the interval graph, going from top to bottom, using the Delaunay radius value for the critical simplices, and assigning to a non-singular interval the minimum value of its successors. We, however, prefer the first algorithm since c is usually bounded by a constant and the algorithm also computes the set of critical descendants of the non-singular intervals.

3.3 Dynamic Updates

So far we considered the case where we are given a fixed point set X and want to compute its Delaunay triangulation, Alpha complexes, and Wrap complexes. In many applications, however, the point set changes dynamically. Over time, new points get inserted into and some points get deleted from X. The naive strategy of recomputing everything from scratch seems rather wasteful. Therefore, we want to have a method that updates the complexes only locally around the inserted or deleted points.

Updating the Delaunay Triangulation. The insertion of a new point into the Delaunay triangulation affects a simplex only if the point is in its smallest empty circumsphere. We call these simplices the *conflict zone* of the point, see Figure 3.1. After the point insertion, all new simplices are incident to the new point. For the deletion of a point, it is exactly the other way round.

For the Delaunay triangulation, dynamic algorithms have already been proposed and implemented [15, 14, 7]. We use CGAL [10] to locate a new point in the current Delaunay triangulation and to get the simplices in and on the boundary of the conflict zone in the

case of a point addition, and to compute the new simplices in the conflict zone in the case of a point deletion. The new simplices for point addition and the old simplices for point deletion are the simplices incident to the new or old point, which we call the *star* of the point.

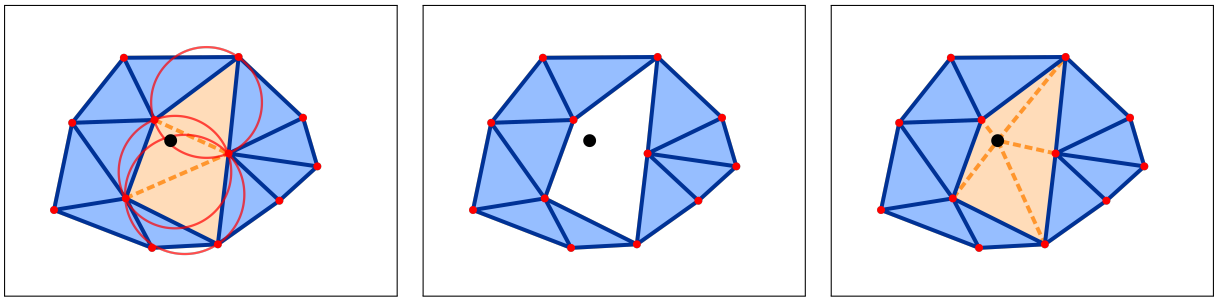


Figure 3.1: Conflict zone for insertion *(left to right)* and deletion (*right to left*) of a point.

Updating the Alpha Complex. After a dynamic update of the Delaunay triangulation, only the Delaunay radii in the interior and on the boundary of the conflict zone may change (both in the case of point insertion and deletion). We iterate through the corresponding simplices and recompute their Delaunay radii.

Updating the Wrap Complex. Unlike the Delaunay radii, the Wrap radii can change in a region that is bigger than the conflict zone. The changes in the conflict zone might affect simplices outside, namely those in the lower sets of simplices on the boundary of the conflict zone.

In order to allow efficient updates, every non-singular interval stores references to all its critical descendants. The critical descendant of smallest radius determines the Wrap radius of the interval, but it might change when simplices are deleted or newly created. To have all the critical descendants stored, we need to adapt the initial computation of the Wrap complex in Algorithm 3 slightly. We continue the recursive traversal of a lower set even when we reach a simplex for which the Wrap radius cannot be updated (its value is not larger than the current one), because we want to store the current critical simplex as a critical descendant in its full lower set (except nested lower sets of other critical simplices) even if it does not determine the Wrap radius of all the lower set simplices at the moment.

After recomputing the intervals inside and on the boundary of the conflict zone and updating the corresponding arcs in the interval graph, we perform the following steps to update the sets of critical descendants and thus also the Wrap radii, see the left panel of Figure 3.2: First, we remove the references to deleted critical descendants from the boundary intervals and their lower sets. Second, references to critical descendants outside of the conflict zone are added to the part of their lower sets that has changed (which reaches inside the conflict zone and maybe even beyond). Third, we add the new singular intervals as critical descendants to their lower sets. As in Algorithm 3, we always traverse the lower sets recursively, following the reversed arcs, and stop when we reach singular intervals, since they and their lower sets do not depend on the more distant critical descendant.

Finally, we get the new Wrap radius of all those simplices for which their set of critical descendants has changed, as the smallest radius of the updated critical descendants. For the other simplices, the Wrap radius remains unchanged.

Update times. In addition to the update times of the Delaunay triangulation, the running time for a dynamic update of the Wrap complex only depends on the size of the conflict zone and the lower sets of the intervals in the boundary of the conflict zone. If this region is small compared to the entire complex, we achieve much faster running times than for recomputing everything from scratch.

We use the Poisson point process (see Section 4.1) to generate random point sets of different sizes in $\mathbb{R}^2$ and $\mathbb{R}^3$ and compare the times for computing the Alpha and Wrap complexes from scratch and for updating them dynamically after inserting a single point. The results, averaged over 100 runs each, are depicted in the right panel of Figure 3.2. The local update algorithm clearly outperforms the full recomputation of the complexes. Its running time is however not constant: the region we need to update for the Wrap complex is not necessarily of constant size, and, more importantly, the location of a point in the Delaunay triangulation takes more than constant time.

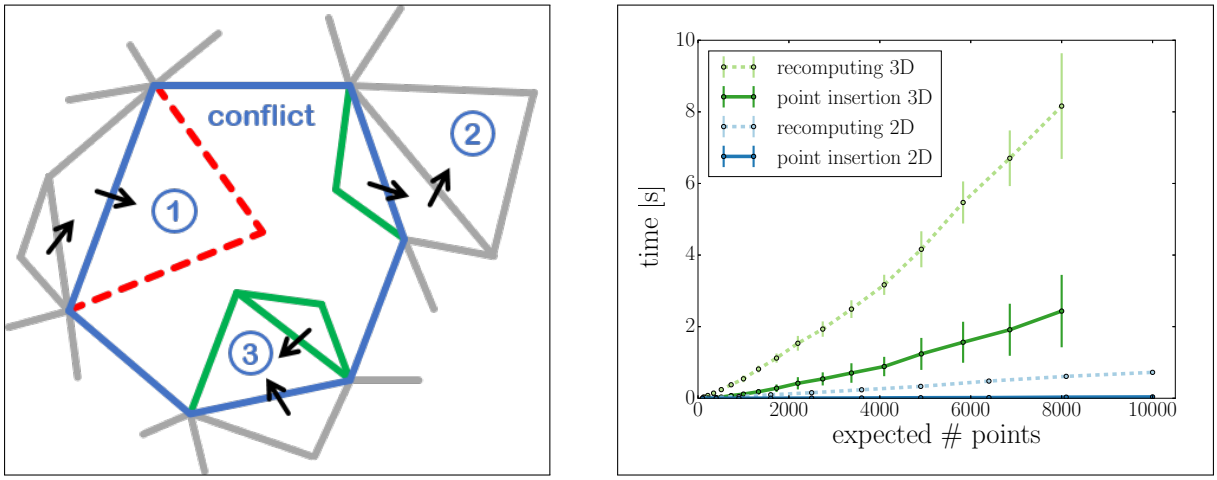


Figure 3.2: *Left:* Illustration of the 3 steps to update the Wrap complex locally. *Right:* Running times (with error bars) for random point sets of different size, averaged over 100 runs.

3.4 Interval Reconstruction for Noisy Values

Working with distance measures other than the Euclidean metric (see Chapter 7), it might be infeasible to compute the values of the Delaunay radius function exactly. Inexact function values can result in a non-monotonic ordering of the simplices. What is even more problematic, already very small perturbations of the function values would alter the interval partition, producing many additional critical simplices.

In order to overcome these issues, we propose an algorithm to reconstruct the intervals for a generalized discrete Morse function $f\colon K \to \mathbb{R}$ (for example, the Delaunay radius function) with noisy values. It gives a valid filtration of the simplicial complex K and a partition into intervals, which can be used to compute the Wrap complex. We will also show how to apply the same algorithm to compute a relaxed version of the Wrap complex.

Ensuring Monotonicity. First, we want to make sure that the given function values correspond to a monotonic ordering. Let $g\colon K \to \mathbb{R}$ be a noisy version of f. We define the *gap* of a set of simplices $A \subseteq K$ as the maximum difference of function values of related simplices:

$$\operatorname{gap} A := \max_{\sigma,\tau \in A,\, \sigma \subseteq \tau} (g(\tau) - g(\sigma)).$$

If g is monotonic, then all gaps are non-negative. Otherwise, we set $g(\sigma) := \min\{g(\sigma), g(\tau)\}$ for all $\sigma \subseteq \tau$. (Alternatively, we could set $g(\tau) := \max\{g(\sigma), g(\tau)\}$, depending on which

value is more reliable in the particular situation.) This makes g monotonic, while changing its values by at most the largest negative gap ε_0. The full algorithm is given as Algorithm 4.

Algorithm 4 Ensuring Monotonicity

```
ε_0 := 0
for p ∈ {d − 1, . . . , 0} do
    for p-simplex σ ∈ Del(X) do
        e := min_{τ∈cofaces(σ)}(g(τ) − g(σ))
        if e < 0 then
            g(σ) := g(σ) + e
            ε_0 := max {ε_0, −e}
```

Reconstructing Intervals. We will construct the final partition of intervals iteratively. We start with a trivial partition, where all the intervals are singular, and successively merge them to bigger intervals. An interval a is called *compatible* with the current partition V of K iff

(i) $a \notin V$ is a union of intervals in V, and

(ii) $g(\max(a)) \leq g(\tau)$ for every $\tau \notin a$ that has a face $\sigma \in a$, where $\max(a)$ is the maximal simplex in a.

We construct a priority queue of all possible intervals of K, which we consider in order of increasing gap. We add an interval a to V if it is compatible with V and its gap is not bigger than some fixed threshold value $\varepsilon \geq 0$. In this case, we set the function values of all simplices in a to the value of the maximal simplex of a (alternatively, we could use the value of the minimal simplex and alter condition (ii) accordingly). By condition (i), V is still a partition of K into intervals after adding a, and, by condition (ii), no arc in the interval graph is reversed so that it remains acyclic. The algorithm is stated as Algorithm 5.

Putting everything together, we use Algorithms 4 and 5 to update the values of a noisy generalized discrete Morse function g, so that the new values differ by at most $\varepsilon_0 + \varepsilon$ from the old ones.

Algorithm 5 Interval Reconstruction

priority queue $Q := \{\}$
$V := \bigcup_{\sigma \in K} \{[\sigma, \sigma]\}$
$\text{interval}_V(\sigma) := [\sigma, \sigma]$
for $\tau \in K$ **do**
  for $\sigma \in \text{faces}(\tau)$ **do**
    $Q.\texttt{insert}((g(\tau) - g(\sigma), [\sigma, \tau]))$
$e := 0$
while $Q \neq \emptyset$ **and** $e \leq \varepsilon$ **do**
  $(e, [\sigma, \tau]) := \min Q$; $Q := Q \setminus \min Q$
  if $e \leq \varepsilon$ **and** $\text{interval}_V(\upsilon) \subseteq [\sigma, \tau] \;\; \forall \upsilon \in [\sigma, \tau]$ **then** ▷ condition (i)
    if $g(\tau) \leq g(\varphi) \;\; \forall \varphi \notin [\sigma, \tau] : \exists \upsilon \in [\sigma, \tau], \upsilon \subseteq \varphi$ **then** ▷ condition (ii)
      $V := V \setminus \bigcup_{\upsilon \in [\sigma, \tau]} \text{interval}_V(\upsilon)$; $V := V \cup \{[\sigma, \tau]\}$
      $g(\upsilon) := g(\tau)$; $\text{interval}_V(\upsilon) := [\sigma, \tau] \;\; \forall \upsilon \in [\sigma, \tau]$

Variants. As the partition of intervals V evolves during the execution of the algorithm, intervals that were incompatible with V at some point might become compatible later. We could keep track of the intervals that were rejected due to condition (ii) and reinsert them into the priority queue if the value of the simplex that was responsible for their rejection changes.

Furthermore, we can make use of additional knowledge about the generalized discrete Morse function that g approximates. Consider, for example, the Delaunay radius function for an unweighted point set X. We know that all points are critical and have the value 0. So, we fix their values to 0 and do not consider non-singular intervals that contain a point. Every other simplex gets a positive value assigned.

Relaxed Wrap Complex. An alternative application for Algorithm 5 is to construct a relaxed version of the Wrap complex. In general, we can take a valid generalized discrete Morse function f as input and construct another generalized discrete Morse function g with $\|f - g\|_\infty \leq \varepsilon$ and possibly bigger intervals than f. For the relaxed Wrap complex, we start with the squared Delaunay radius function ρ^2, fix a threshold $\varepsilon \geq 0$, apply Algorithm 5, and compute the Wrap complex for the resulting interval partition V. We can interpret this *relaxed Wrap complex* as a modification of the original Wrap complex,

where we allow collapses of critical simplices that are "nearly non-critical", i.e., their Delaunay radius is very similar to the radius of a critical face. This is very useful for cases where we would like to have a smaller Wrap complex but the sequence of collapses halts at such a "nearly non-critical" simplex. A 2- and a 3-dimensional example are given in Figures 3.3 and 3.4. If one is unsure how to choose the threshold value, $\varepsilon = \infty$ is a viable choice, since intervals cannot grow arbitrarily and we can still only collapse a restricted part of the simplices, see Figure 3.5.

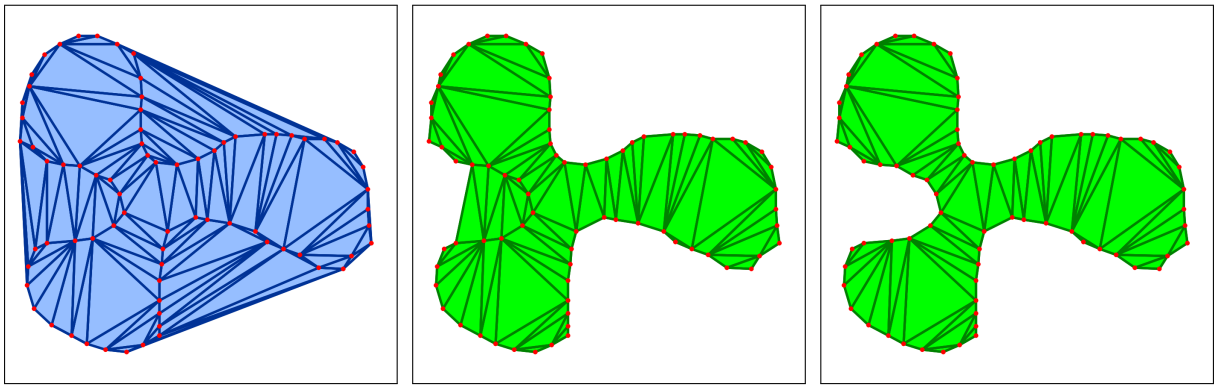


Figure 3.3: *Left:* Delaunay triangulation of the point set `kleeblatt` of CGAL [10]. *Middle:* Wrap complex. *Right:* Relaxed Wrap complex for any threshold $\varepsilon \geq 6$.

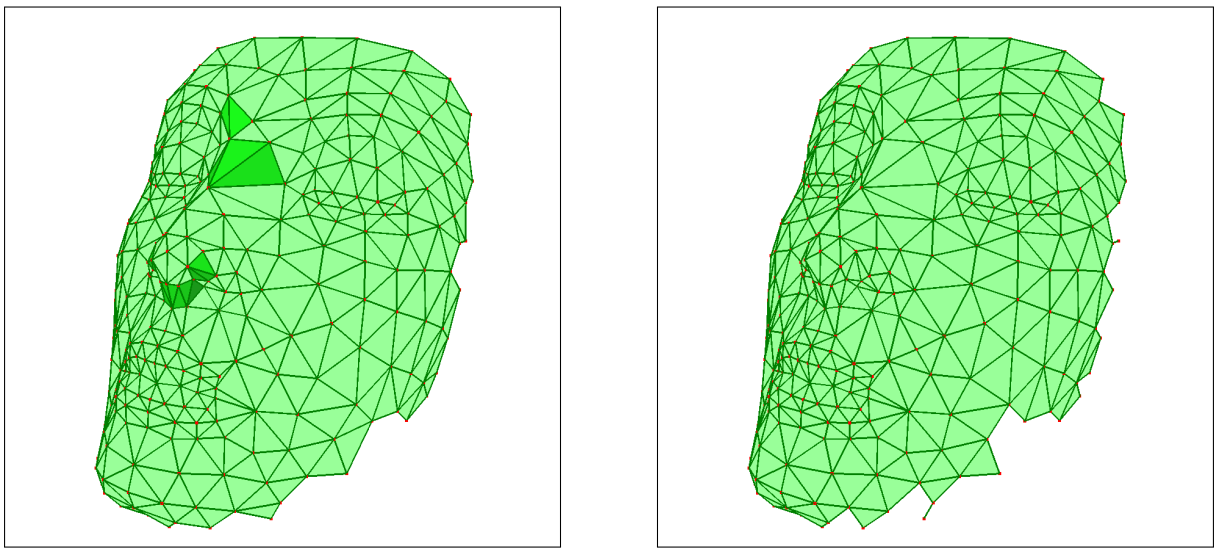


Figure 3.4: *Left:* Wrap complex for $r = \infty$ of the point set `nefertiti` of CGAL [10]. The darker triangles bound tetrahedra, while the pale ones do not have incident tetrahedra. *Right:* Relaxed Wrap complex for the threshold $\varepsilon = 0.01$. It does not contain a tetrahedron.

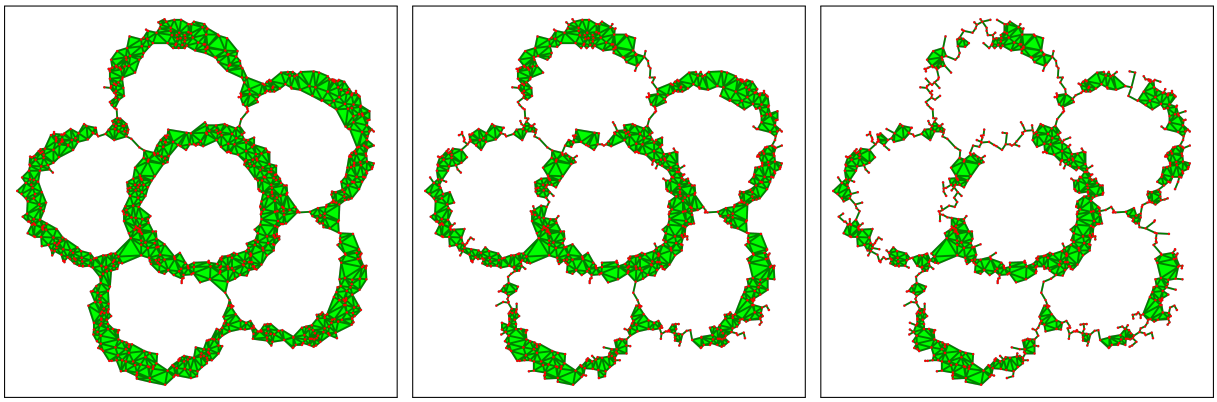


Figure 3.5: *Left:* Wrap complex of the same point set as in Figure 1.1 for $r = 50$. *Middle, Right:* Relaxed Wrap complexes with thresholds $\varepsilon = 10$ and $\varepsilon = \infty$.

Chapter 4

Stochastic Geometry and Topology

In this chapter, we present experimental results about Alpha and Wrap complexes on random point sets. We start by confirming previously known theoretical results [22], and continue by exploring cases where they are still missing. The results of this chapter were published as part of a survey paper [21].

4.1 Probabilistic and Computational Setting

First of all, we describe the model we use to generate the random point sets and the details of our computational experiments.

Poisson Point Process (PPP). A *(homogeneous) Poisson point process* X with density $\lambda > 0$ in $\mathbb{R}^d$ is defined by the following two properties:

(i) The number of points in a Borel set $\Omega \subseteq \mathbb{R}^d$ of measure $\|\Omega\|$ is Poisson distributed with parameter $\lambda\|\Omega\|$, i.e., $\mathbb{P}(|X \cap \Omega| = n) = \lambda^n \|\Omega\|^n e^{-\lambda\|\Omega\|}/n!$.

(ii) The numbers of points in any finite collection of pairwise disjoint Borel sets $\Omega_1, \ldots, \Omega_k$ are independent, i.e., $\mathbb{P}(|X \cap \Omega_i| = n_i, i = 1, \ldots, k) = \prod_{i=1}^{k} \lambda^{n_i} \|\Omega_i\|^{n_i} e^{-\lambda\|\Omega_i\|}/n_i!$.

The simulation of a Poisson point process is done in a bounded window W. First, we sample the number of points in the window from a Poisson distribution with parameter $\lambda\|W\|$. Then, we place the points uniformly and independently in W.

In our experiments, we use square windows of different sizes in $\mathbb{R}^2$ and cube windows in $\mathbb{R}^3$.

Periodic Triangulations. In order to avoid boundary effects, we impose periodic boundary conditions on our simulation windows. This means that we compute *periodic Delaunay triangulations* on a flat torus, for which we can use the CGAL library [10]. Intuitively, the point set in the simulation window is copied in all directions, we compute the triangulation on the bigger set of points and identify simplices with the same vertices (disregarding in which copy the points lie), see Figure 4.1.

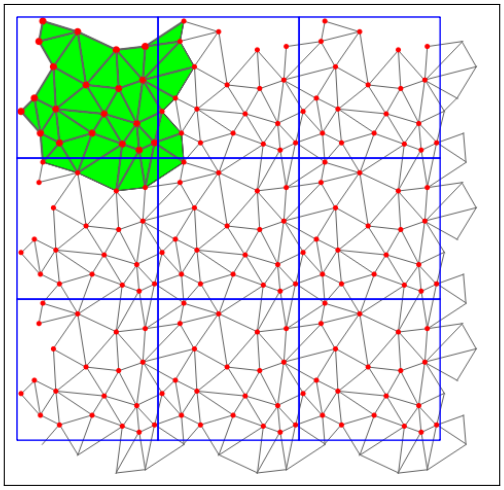


Figure 4.1: The point set in the central square was copied in all directions. The triangles of the periodic Delaunay triangulation are highlighted in green [10].

Experimental Setup. We have implemented the algorithms of Chapter 3 to compute the Alpha and Wrap complexes for (possibly periodic) point sets in $\mathbb{R}^2$ and $\mathbb{R}^3$. Our implementation is in C++, for parts of the computation we use the CGAL library [10].

We simulate the Poisson point process of density 1 in a square window of size 300×300 in $\mathbb{R}^2$ and in a cube window of size $60 \times 60 \times 60$ in $\mathbb{R}^3$. We compute the periodic Delaunay triangulation, Alpha, and Wrap complexes for the generated point sets. We get estimated values of various combinatorial and topological properties by averaging over 1000 runs in $\mathbb{R}^2$ and over 100 runs in $\mathbb{R}^3$. In order to get smooth results for the corresponding variances, we need a considerably higher number of runs. Therefore, we use a window of size 100×100 in $\mathbb{R}^2$ and a window of size $15 \times 15 \times 15$ in $\mathbb{R}^3$, and average over 10 000 runs

in both cases. Figure 4.2 shows examples for periodic Alpha complexes in $\mathbb{R}^2$ and $\mathbb{R}^3$ (for smaller windows).

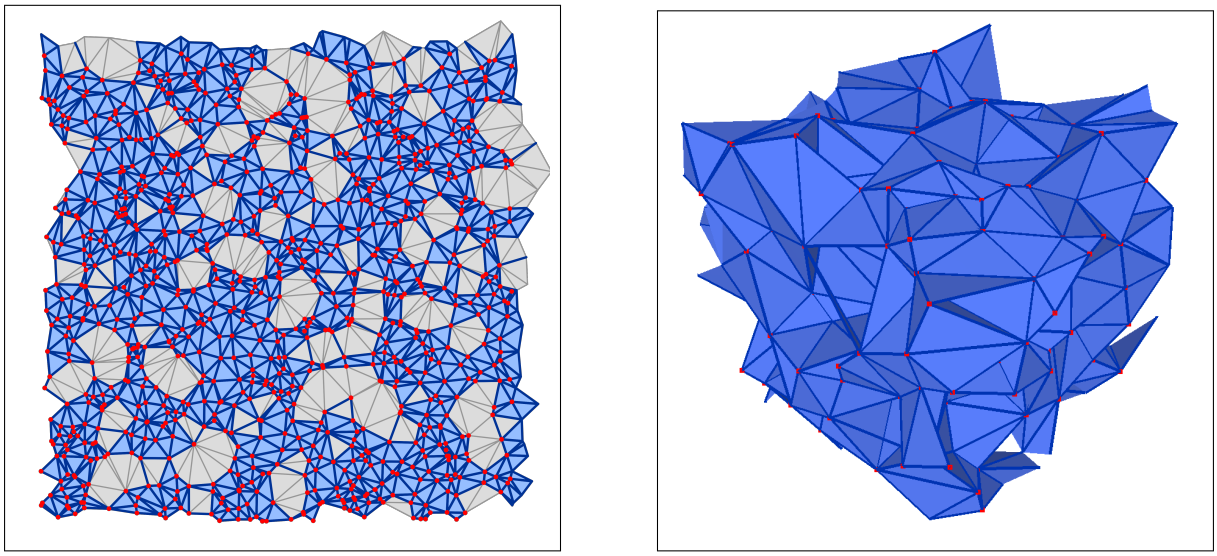


Figure 4.2: The periodic Alpha complex for $r = 1$ of random points generated by the Poisson point process of density 1 in the window $[0, 30]^2 \subseteq \mathbb{R}^2$ (*left*) and in the window $[0, 6]^3 \subseteq \mathbb{R}^3$ (*right*). For the 2-dimensional example, the Delaunay simplices that are not in the Alpha complex are shown in gray.

4.2 Statistics for the Alpha Complex

We study the expected numbers of intervals and simplices in the Alpha complex of a Poisson point process for radius r. We say an interval has *type* (ℓ, m) if ℓ is the dimension of its minimum simplex and m is the dimension of its maximum simplex.

Theoretical Results. It was proven in [22] that the number of intervals in the Alpha complex of a Poisson point process is Gamma distributed.

The *Gamma function* is defined as $\Gamma(m) := \int_0^\infty x^{m-1}e^{-x}\,dx$, and satisfies $\Gamma(m) = (m-1)!$ for any positive integer m. On the other hand, the *lower incomplete Gamma function* is $\gamma(m, s) := \int_0^s x^{m-1}e^{-x}\,dx$, and we call $\tilde{\gamma}(m, s) := \gamma(m, s)/\Gamma(m)$ its *regularized* version.

Let X be a homogeneous Poisson point process with density $\lambda > 0$ in $\mathbb{R}^d$. We write $c^d_{\ell,m}(r)$ for the expected number of intervals of type (ℓ, m) in $\mathrm{Alpha}_r(X)$ that lie in a

Borel set $\Omega \subseteq \mathbb{R}^d$. There exist constants $C^d_{\ell,m}$ such that

$$c^d_{\ell,m}(r) = \tilde{\gamma}(m, \lambda\, \nu_d\, r^d) \cdot C^d_{\ell,m} \cdot \lambda \|\Omega\| \tag{4.1}$$

for every $\ell > 0$ and every $r \geq 0$, where $\nu_d := \pi^{\frac{d}{2}}/\Gamma(\frac{d}{2}+1)$ is the volume of the unit ball in $\mathbb{R}^d$ [22]. To be precise, the expression "in Ω" here means that the center of the smallest empty circumsphere of the interval simplices lies in Ω. At the cost of weakening (4.1) by adding $o(\|\Omega\|)$ on the right-hand side, we can replace this condition by "lying inside Ω" or "intersecting Ω", of which we will make use in the numerical estimation.

An interval of type (ℓ, m) contains $\binom{m-\ell}{m-p}$ simplices of dimension p, and thus $2^{m-\ell}$ simplices in total. We use this fact to get an expression for the expected number of p-simplices in $\mathrm{Alpha}_r(X)$, which we denote by $d^d_p(r)$, from (4.1):

$$d^d_p(r) = \sum_{\ell=0}^{p} \sum_{m=p}^{d} \binom{m-\ell}{m-p} \tilde{\gamma}(m, \lambda\, \nu_d\, r^d) \cdot C^d_{\ell,m} \cdot \lambda \|\Omega\| \tag{4.2}$$

for every $p > 0$ and every $r \geq 0$. In analogy to the constants for the interval distributions, we define D^d_p to be the expected number of p-simplices per unit volume, which satisfies $d^d_p(\infty) = D^d_p\, \lambda\, \|\Omega\|$.

In [22], the constants $C^d_{\ell,m}$ and D^d_p were computed explicitly for dimensions $d \leq 4$.

Experimental Results for Constants. We run the experiments as described in the previous section. We compute the average numbers of intervals and simplices per unit volume in the Delaunay triangulation and compare them to the constants $C^d_{\ell,m}$ and D^d_p, see Table 4.1. We observe that the experimental values match the theoretically predicted ones with an accuracy of at least one position after the decimal point.

Experimental Results for Interval Densities. In addition to the constants, which give the expected numbers of intervals in the full Alpha complex (= the Delaunay triangulation), we estimate the densities of the intervals as functions of their Delaunay radius. For this purpose, we compute histograms with 250 uniform bins ranging from $r = 0$ to $r = 2.5$, normalize by the expected number of points and average over all trials.

In order to compare these estimated interval densities with the theoretical results, we fit regularized lower incomplete Gamma functions to the curves, using the `curve_fit`-function of Python 2.7's `scipy`-module, which is based on least squares optimization.

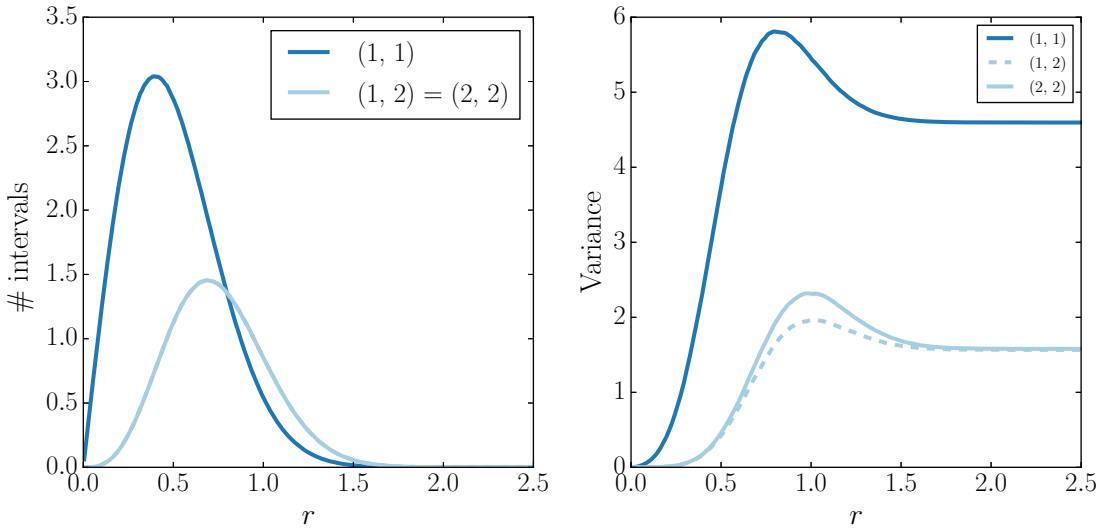
$d=2$	theor.	exp.
$C^2_{0,0}$	1	1.00
$C^2_{1,1}$	2	2.00
$C^2_{1,2}$	1	1.00
$C^2_{2,2}$	1	1.00
D^2_0	1	1.00
D^2_1	3	3.00
D^2_2	2	2.00

$d=3$	theoretical	exp.
$C^3_{0,0}$	$1 = 1.00$	1.00
$C^3_{1,1}$	$4 = 4.00$	4.00
$C^3_{1,2}$	$\frac{9}{16}\pi^2 - 3 \approx 2.55$	2.55
$C^3_{2,2}$	$\frac{3}{16}\pi^2 + 3 \approx 4.85$	4.85
$C^3_{1,3}$	$\frac{69}{560}\pi^2 \approx 1.22$	1.22
$C^3_{2,3}$	$\frac{3}{8}\pi^2 \approx 3.70$	3.70
$C^3_{3,3}$	$\frac{3}{16}\pi^2 \approx 1.85$	1.85
D^3_0	$1 = 1.00$	1.00
D^3_1	$\frac{24}{35}\pi^2 + 1 \approx 7.77$	7.77
D^3_2	$\frac{48}{35}\pi^2 \approx 13.54$	13.53
D^3_3	$\frac{24}{35}\pi^2 \approx 6.77$	6.77

Table 4.1: Comparison of theoretical [22] and experimentally estimated values for $C^d_{\ell,m}$ and D^d_p in $d = 2$ (*left*) and $d = 3$ (*right*) dimensions.

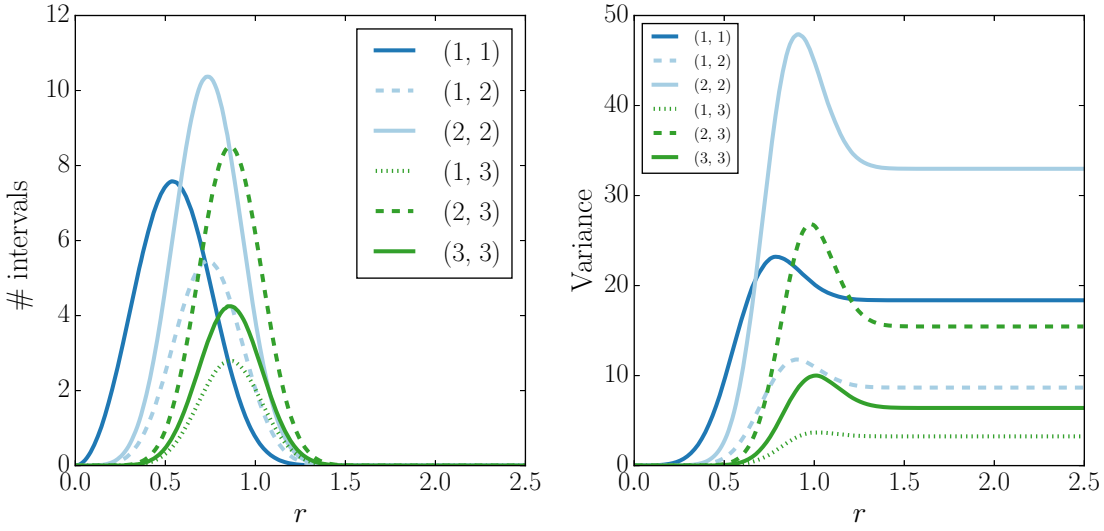
The density plots and the results for the curve fitting are shown in Figure 4.3 for $\mathbb{R}^2$ and in Figure 4.4 for $\mathbb{R}^3$. It is not surprising that the estimated curves are very close to the theoretically predicted ones. However, it is useful to compute the error values, as they give a calibration for the error of fitting curves to other densities, which are not known theoretically yet (see Section 4.4). We compute the fitting errors as the square root of the mean squared deviation of the fitted curve to the density samples, which estimates their L^2-difference.

Furthermore, we compute the variances of the estimated cumulative distribution functions, which are not known theoretically. We get almost the same curves for different window sizes, which suggests that the variances are proportional to the expected number of points. As predicted by [39], the expected number of intervals of type $(1, 2)$ (non-critical = obtuse triangles) and intervals of type $(2, 2)$ (critical = acute triangles) are the same in $\mathbb{R}^2$. On the other hand, we observe that the variance for intervals of type $(1, 2)$ is consistently smaller than the one for intervals of type $(2, 2)$.



interval	distribution	experimental fit	fitting error
$(1,1)$	$C^2_{1,1} \cdot \tilde{\gamma}(1, \pi r^2)$	$2.00 \cdot \tilde{\gamma}(1.00, 3.14r^2)$	1.37×10^{-5}
$(1,2)$	$C^2_{1,2} \cdot \tilde{\gamma}(2, \pi r^2)$	$1.00 \cdot \tilde{\gamma}(2.00, 3.14r^2)$	1.33×10^{-5}
$(2,2)$	$C^2_{2,2} \cdot \tilde{\gamma}(2, \pi r^2)$	$1.00 \cdot \tilde{\gamma}(2.00, 3.14r^2)$	1.51×10^{-5}

Figure 4.3: Estimated interval densities and variances in the Alpha complexes of a Poisson point process in $\mathbb{R}^2$. *Left:* Estimated interval densities as functions of the Delaunay radius, normalized by the expected number of points. *Right:* Variances of the corresponding cumulative distributions. *Bottom:* Results for fitting regularized lower incomplete Gamma functions to the estimated densities.



interval	distribution	experimental fit	fitting error
$(1,1)$	$C^3_{1,1} \cdot \tilde{\gamma}(1, \frac{4}{3}\pi r^3)$	$4.00 \cdot \tilde{\gamma}(1.00, 4.19r^3)$	6.96×10^{-5}
$(1,2)$	$C^3_{1,2} \cdot \tilde{\gamma}(2, \frac{4}{3}\pi r^3)$	$2.55 \cdot \tilde{\gamma}(2.00, 4.19r^3)$	4.01×10^{-5}
$(2,2)$	$C^3_{2,2} \cdot \tilde{\gamma}(2, \frac{4}{3}\pi r^3)$	$4.85 \cdot \tilde{\gamma}(2.00, 4.19r^3)$	6.68×10^{-5}
$(1,3)$	$C^3_{1,3} \cdot \tilde{\gamma}(3, \frac{4}{3}\pi r^3)$	$1.22 \cdot \tilde{\gamma}(3.00, 4.19r^3)$	3.23×10^{-5}
$(2,3)$	$C^3_{2,3} \cdot \tilde{\gamma}(3, \frac{4}{3}\pi r^3)$	$3.70 \cdot \tilde{\gamma}(3.00, 4.19r^3)$	5.50×10^{-5}
$(3,3)$	$C^3_{3,3} \cdot \tilde{\gamma}(3, \frac{4}{3}\pi r^3)$	$1.85 \cdot \tilde{\gamma}(3.00, 4.19r^3)$	3.13×10^{-5}

Figure 4.4: Estimated interval densities and variances in the Alpha complexes of a Poisson point process in $\mathbb{R}^3$. *Left:* Estimated interval densities as functions of the Delaunay radius, normalized by the expected number of points. *Right:* Variances of the corresponding cumulative distributions. *Bottom:* Results for fitting regularized lower incomplete Gamma functions to the estimated densities.

4.3 Topological Statistics

We change our focus from purely combinatorial statistics to those that are relevant for the topology of the Alpha (and Wrap) complexes. We only give the experimental results, as no analytic expressions are known yet.

Critical Simplices. For the critical simplices (whose densities were already given since they coincide with the singular intervals), we now differentiate between birth and death simplices, which we label by $\circ$ and $\bullet$, respectively. The densities and variances are displayed in Figure 4.5 for $\mathbb{R}^2$ and in Figure 4.6 for $\mathbb{R}^3$. Although the density curves are visually similar to the interval densities, they are not Gamma distributed. The fitting error is never less than 10^{-3}. Even a linear combination of up to $d = 2$ or $d = 3$ regularized lower incomplete Gamma functions does not give an error of less than 10^{-4}. The only exception are the death-giving d-simplices, since there are no birth-giving ones (or rather, because of the periodic boundary conditions, there is exactly 1).

Betti Numbers. The p-th Betti number is the rank of the p-dimensional homology group. We get it by subtracting the number of death-giving critical $(p+1)$-simplices (corresponding to p-dimensional homology classes that have already died) from the number of birth-giving critical p-simplices (corresponding to p-dimensional homology classes that were born so far). The Betti numbers as functions of the radius and the corresponding variances are shown in Figure 4.5 for $\mathbb{R}^2$ and in Figure 4.6 for $\mathbb{R}^3$.

4.4 Statistics for the Wrap Complex

Finally, we collect and analyze statistical data about the Wrap complexes of a Poisson point process.

Interval and Simplex Densities. As the Alpha and Wrap complexes have the same critical simplices, the densities for the singular intervals are the same. The estimated densities for the non-singular intervals are however different. We try fitting linear combinations of up to d regularized lower incomplete Gamma functions to the estimated interval

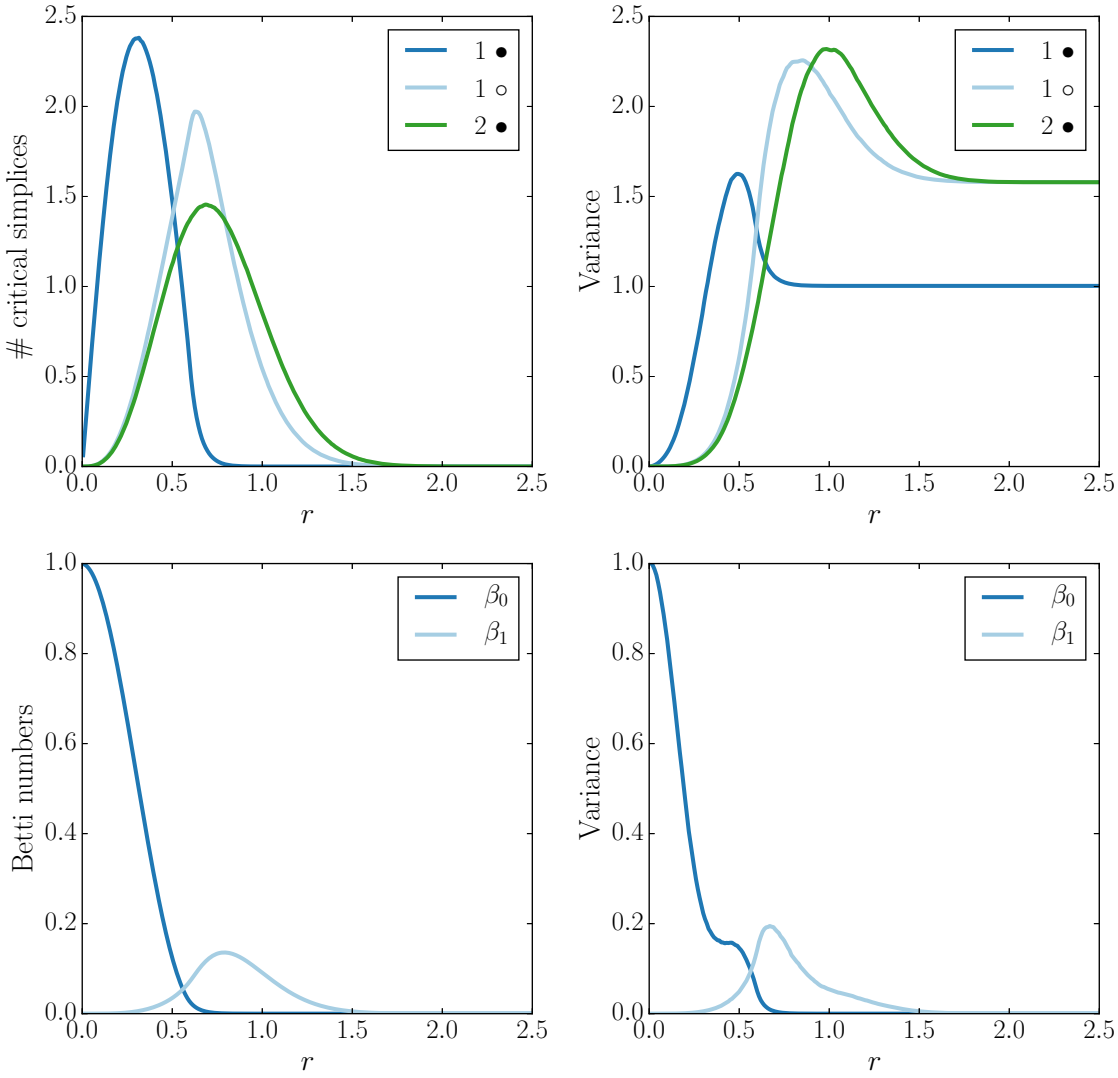


Figure 4.5: *First row:* Estimated densities (*left*) of birth- (∘) and death-giving (•) critical simplices and the corresponding cumulative variances (*right*) in the Alpha complexes of a Poisson point process in $\mathbb{R}^2$. *Second row:* Experimentally observed Betti numbers (*left*) of the Alpha complexes and the corresponding variances (*right*).

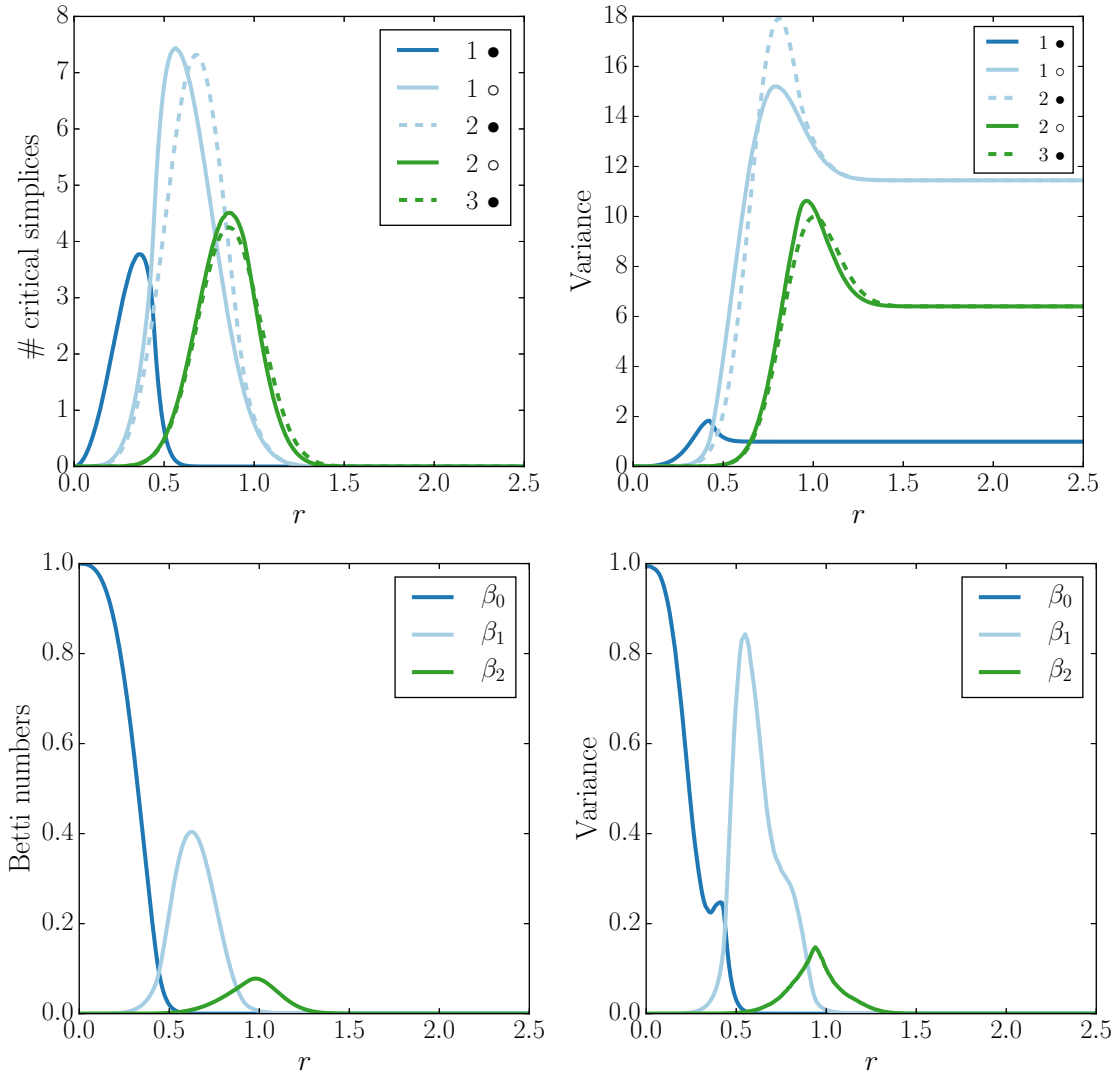


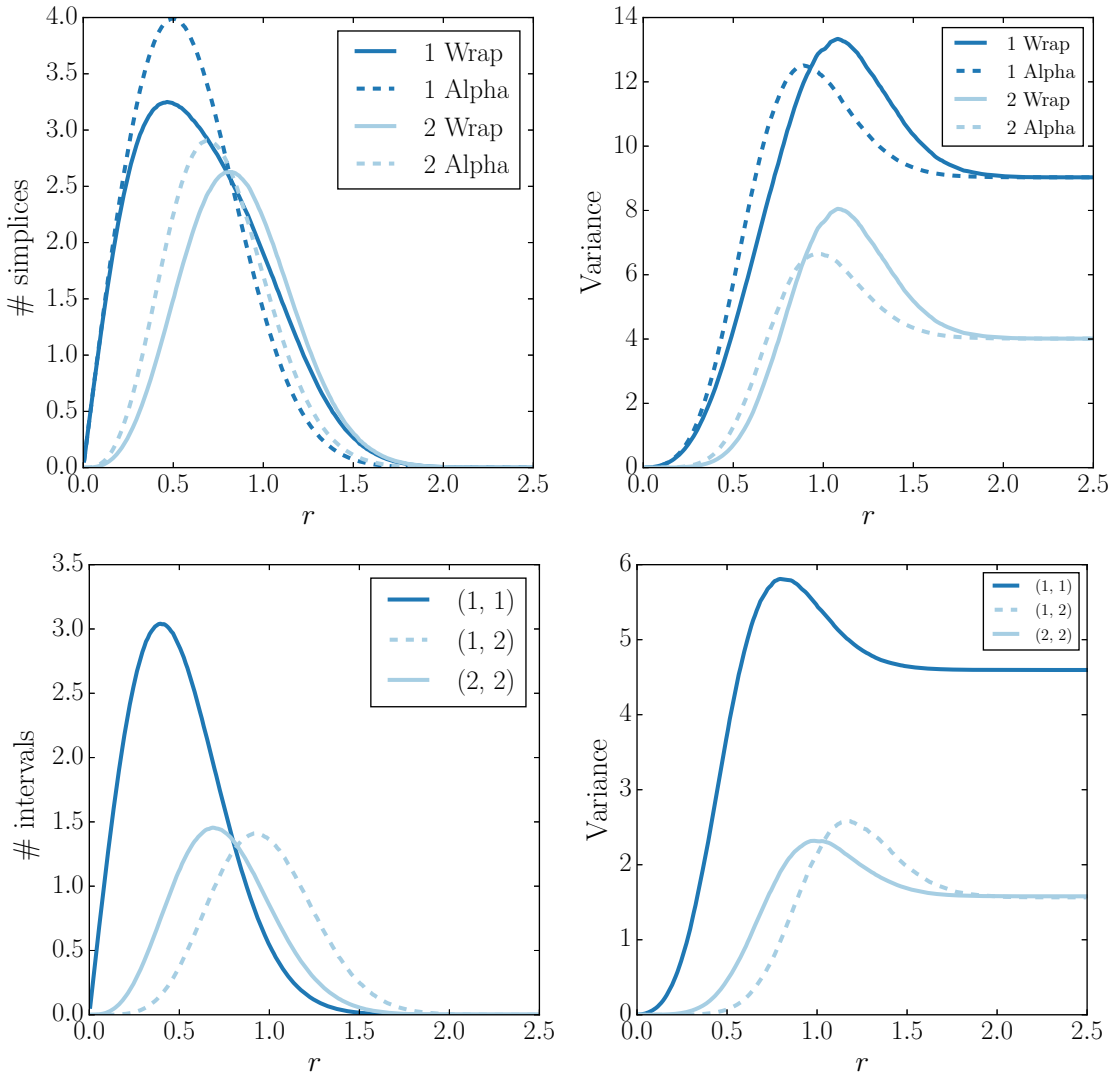
Figure 4.6: *First row:* Estimated densities (*left*) of birth- (∘) and death-giving (•) critical simplices and the corresponding cumulative variances (*right*) in the Alpha complexes of a Poisson point process in $\mathbb{R}^3$. *Second row:* Experimentally observed Betti numbers (*left*) of the Alpha complexes and the corresponding variances (*right*).

densities in the Wrap complexes in $\mathbb{R}^d$, for $d = 2, 3$. In all cases, we get results with fitting errors of less than 10^{-4}, which are comparable to the errors for the singular intervals. This suggests that the intervals of the Wrap complexes might follow such a distribution. The normalized density plots and the results for the curve fitting are shown in Figure 4.7 for $\mathbb{R}^2$ and in Figure 4.8 for $\mathbb{R}^3$.

Fitting linear combinations of Gamma distributions was inspired by the known theoretical distributions of simplices in the Alpha complexes (see Section 4.2). If the densities of intervals in the Wrap complexes also follow such distributions, the same must hold true for the densities of simplices in the Wrap complexes. In the first row of Figures 4.7 and 4.8 the estimated densities of simplices in the Wrap and Alpha complexes are directly compared. Since a Wrap complex is always contained in the Alpha complex of the same radius, the simplex densities in the Wrap complexes are shifted to the right as compared to the same densities in the Alpha complexes.

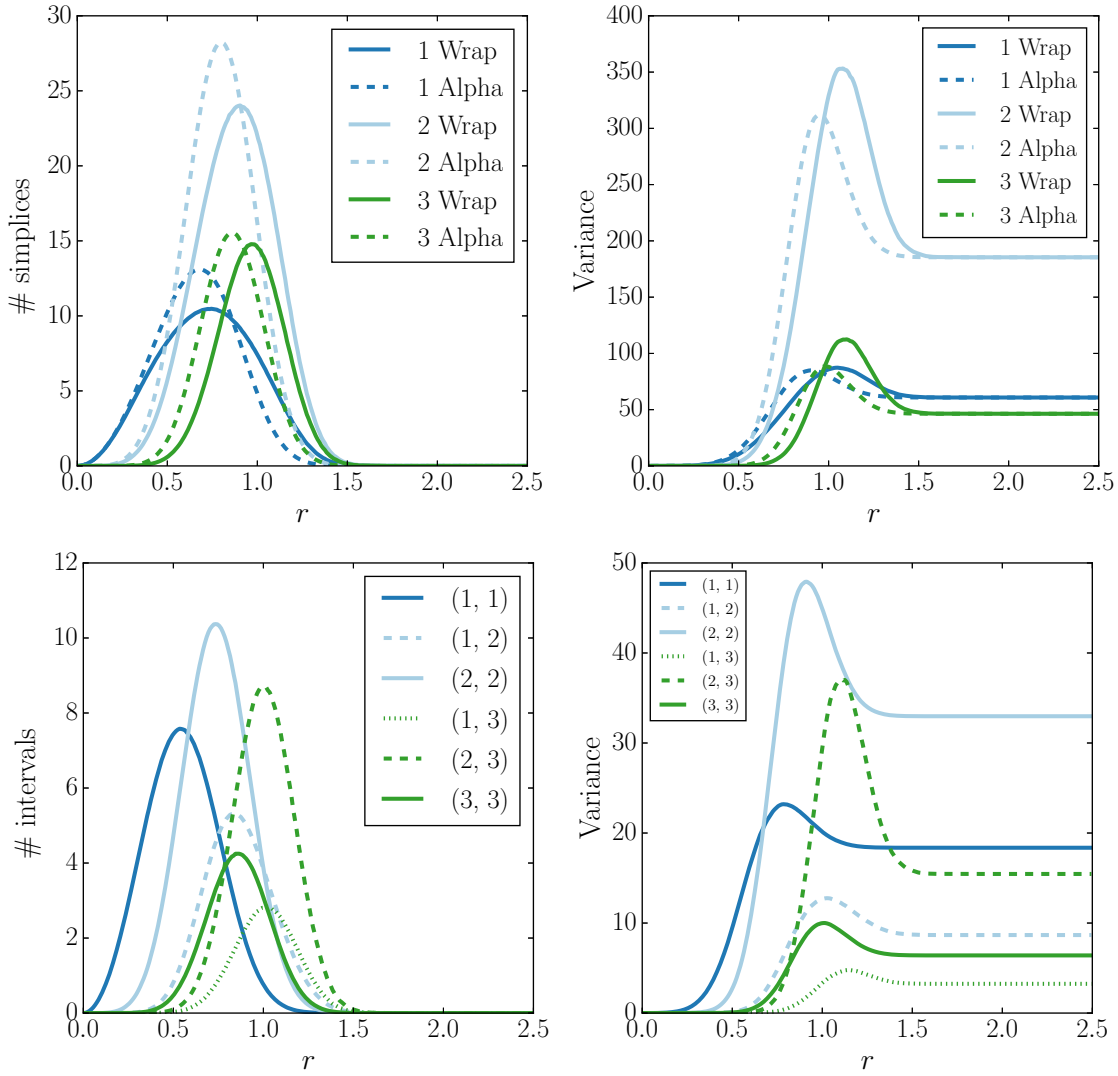
Lower Sets. The lower sets of critical simplices play an important role in the computation of the Wrap complexes. When we increase the radius of the Wrap complex, we have to add the critical simplices with values in the additional range of radii, plus the simplices in their lower sets to the complex. In the case of dynamic updates (see Section 3.3), the running time depends on the size of the lower sets of the simplices in the boundary of the conflict zone. The lower set sizes are also relevant for our computational experiments with periodic boundary conditions. If the average diameter of a lower set would scale linearly with the size of the simulation window, our results would suffer from boundary effects.

Fortunately, our computational experiments show that the lower sets are typically small for points generated by the Poisson point process. We quantify the sizes of the lower sets with two different measures: their number of simplices and their *diameter*, which is defined as the maximum distance between any two points in the set. In Table 4.2, we represent the results as percentiles. The kth-*percentile* is the value below which $k\%$ of the observations fall. For example, the third and fourth entry in the second row of the table say that the lower sets of 95% of the critical triangles in $\mathbb{R}^2$ contain at most 16 simplices and have a diameter of at most 2.39. The first row gives the maximum number of simplices and diameters that we have observed in the experiments. For the numbers of simplices in



interval	experimental fit	fitting error
$(1,1)$	$2.00 \cdot \tilde{\gamma}(1.00, 3.14r^2)$	1.37×10^{-5}
$(1,2)$	$0.62 \cdot \tilde{\gamma}(3.01, 3.41r^2) + 0.38 \cdot \tilde{\gamma}(4.07, 3.29r^2)$	2.41×10^{-5}
$(2,2)$	$1.00 \cdot \tilde{\gamma}(2.00, 3.14r^2)$	1.51×10^{-5}

Figure 4.7: Estimated simplex and interval densities in the Wrap complexes of a Poisson point process in $\mathbb{R}^2$. *First row:* Comparison of estimated simplex densities (*left*) and corresponding cumulative variances (*right*) in the Wrap and Alpha complexes. *Second row:* Estimated interval densities as functions of the Wrap radius (*left*) and corresponding cumulative variances (*right*). The table *below* shows the results for fitting linear combinations of Gamma functions to the estimated interval densities.



interval	experimental fit	fitting error
$(1,1)$	$4.00 \cdot \tilde{\gamma}(1.00, 4.19r^3)$	6.96×10^{-5}
$(1,2)$	$1.78 \cdot \tilde{\gamma}(2.99, 5.28r^3) + 0.77 \cdot \tilde{\gamma}(4.48, 3.95r^2)$	7.52×10^{-5}
$(2,2)$	$4.85 \cdot \tilde{\gamma}(2.00, 4.19r^3)$	6.68×10^{-5}
$(1,3)$	$0.90 \cdot \tilde{\gamma}(4.10, 3.96r^3) + 0.31 \cdot \tilde{\gamma}(6.15, 4.40r^2)$	6.89×10^{-5}
$(2,3)$	$2.94 \cdot \tilde{\gamma}(4.98, 4.23r^2) + 0.56 \cdot \tilde{\gamma}(4.11, 6.19r^3) + 0.20 \cdot \tilde{\gamma}(6.34, 7.31r^2)$	8.28×10^{-5}
$(3,3)$	$1.85 \cdot \tilde{\gamma}(3.00, 4.19r^3)$	3.13×10^{-5}

Figure 4.8: Estimated simplex and interval densities in the Wrap complexes of a Poisson point process in $\mathbb{R}^3$. *First row:* Comparison of estimated simplex densities (*left*) and corresponding cumulative variances (*right*) in the Wrap and Alpha complexes. *Second row:* Estimated interval densities as functions of the Wrap radius (*left*) and corresponding cumulative variances (*right*). The table *below* shows the results for fitting linear combinations of Gamma functions to the estimated interval densities.

	$d=2$				$d=3$					
%	$p=1$		$p=2$		$p=1$		$p=2$		$p=3$	
100	1	5.13	93	5.17	1	3.37	479	3.37	835	3.56
95	1	1.95	16	2.39	1	1.79	19	2.03	94	2.20
90	1	1.71	13	2.17	1	1.64	13	1.89	69	2.08
75	1	1.33	7	1.80	1	1.38	7	1.67	42	1.88
50	1	0.94	4	1.42	1	1.10	4	1.43	25	1.65
25	1	0.61	4	1.07	1	0.82	4	1.18	16	1.42

Table 4.2: Percentiles for the sizes of the lower sets of critical simplices in $\mathbb{R}^2$ and $\mathbb{R}^3$. The double-columns contain the percentiles for the number of simplices (*left*, disregarding (-1)- and 0-simplices) and the diameter (*right*) of the lower sets of critical p-simplices.

the table we have only counted 1-, 2-, and 3-simplices, since lower-dimensional simplices are always critical for unweighted points.

Chapter 5

Tri-Partition and Bases

A *spanning tree* of a graph is a connected subgraph that does not contain any cycle (i.e. it is a *tree*) and connects all the vertices. It is a well-known result that a connected graph G embedded on the sphere can be split into a spanning tree and a *dual tree* (where the set of vertices corresponds to the faces of G and there is an edge connecting two vertices if the corresponding faces are adjacent in G). For graphs embedded on a general surface with positive genus, we get a tri-partition of the edge set into the edges of a spanning tree, the edges of a dual spanning tree, and $2g$ additional edges, where g is the genus of the surface [6, 42].

We want to generalize this result to the set of p-dimensional simplices of a simplicial complex, for every dimension p. This will enable us to construct canonical bases of the cycle, boundary, and homology groups as well as their counterparts in cohomology, which can help study homology from a more geometrical point of view and are also useful for discovering and manipulating hole systems (see Chapter 6).

The results of this chapter were published in [24].

5.1 Example

We consider the 2-dimensional simplicial complex in Figure 5.1. It contains $n_0 = 30$ vertices, and thus every spanning tree contains $n_0 - 1 = 29$ edges. We highlight the edges

of one such spanning tree in black.

The thin gray edges and their vertices constitute a *spanning cotree*, i.e., a subgraph that includes all vertices but does not contain any cocycle. The corresponding dual tree is drawn in orange in the right panel of Figure 5.1. We think of the edges dangling to the outside as connected to an additional dual vertex representing the "outside" of the complex. Later on, we allow trees and cotrees to be disconnected (which are then usually called *forests*), so that our results also hold for disconnected graphs and simplicial complexes.

In our current example, the spanning tree cannot go completely around the central hole, otherwise it would contain a cycle. Likewise, the cotree cannot go from the inner to the outer boundary. Thus, the dashed green edge cannot be contained in either of the two. Adding it to the tree, we get a cycle that goes around the central hole, adding it to the cotree, we get a cocycle that connects the inner and the outer boundary. In general, we get one leftover edge for each hole in the complex.

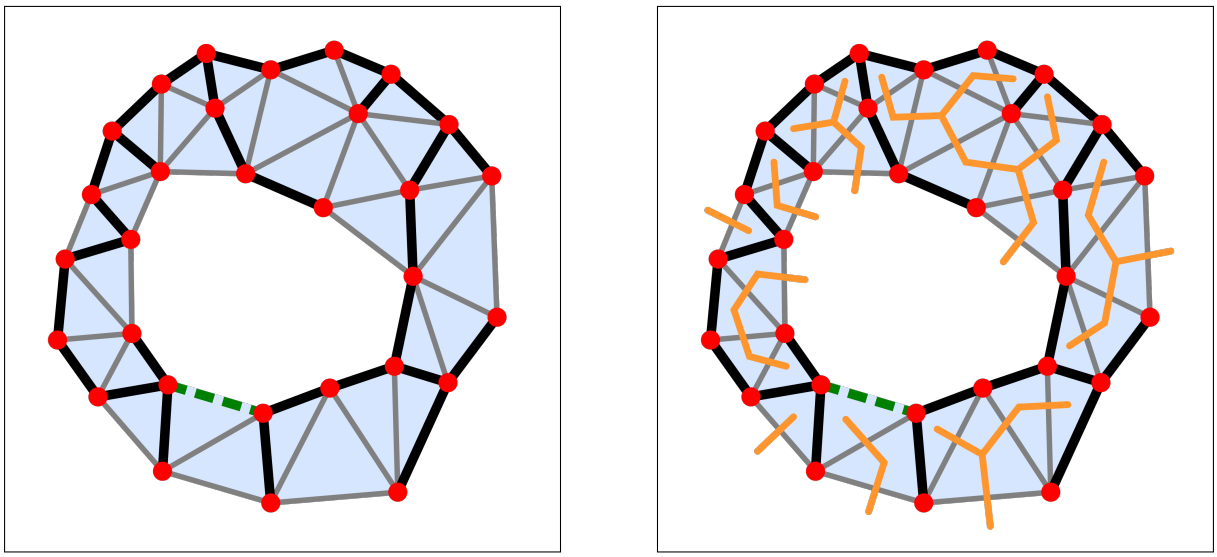


Figure 5.1: Tri-partition of the edge set of a 2-dimensional simplicial complex representing an annulus. The edge set is split into the edges of a spanning tree (thick black edges), the edges of a spanning cotree (thin gray edges), and a leftover edge (dashed green). The tree that is dual to the cotree is drawn in orange in the *right panel*.

5.2 Algorithms for Persistent Homology

An important tool for establishing the tri-partition are the matrix reduction algorithms for computing persistent homology [20], which we will introduce in this section.

Boundary Matrix. To represent a simplicial complex in the computer, we use a monotonic ordering of the simplices, $\sigma_1, \cdots, \sigma_m$, (which we get from a filtration function or we use an arbitrary monotonic ordering if we do not have a filtration) and encode the face relation in matrix form. The *boundary matrix* ∂ is defined by

$$\partial[i,j] := \begin{cases} 1, & \text{if } \sigma_i \subseteq \sigma_j \text{ and } \dim \sigma_i = \dim \sigma_j - 1, \\ 0, & \text{otherwise,} \end{cases}$$

for $i, j \in \{1, \ldots, m\}$. Its columns store the facets of the corresponding simplices, and its rows store the cofacets. Since the simplices are monotonically ordered, the boundary matrix is upper-triangular.

Column Reduction Algorithms. If we are only interested in the homology of the full simplicial complex, we use Gaussian elimination (with modulo 2 arithmetic) on the boundary matrix. For a filtered simplicial complex, we want to compute persistent homology and therefore we restrict ourselves to left-to-right column additions.

The *lowest one* of column j in a matrix is defined as the row index of the lowest non-zero entry in the column and denoted by $\text{low}(j)$. We set $\text{low}(j) := -\infty$ if the column is zero. We call a matrix R *(left-to-right) reduced* if $\text{low}(j) \neq \text{low}(k)$ for all non-zero columns $j \neq k$. Figure 5.2 gives an example of a reduced matrix.

We initialize the matrices R, U, and C to the boundary matrix, the identity matrix, and the zero matrix, respectively. We reduce R using left-to-right column additions while maintaining the relation $R = \partial U$ and using C for book-keeping purposes. The *standard column reduction algorithm* [20] reduces one column after the other until the lowest ones of all non-zero columns of R are unique, see Algorithm 6. Although the gathered homological information, which is encoded in the lowest ones of R, will always be the same no matter how the algorithm is implemented, the matrices R and U are generally not unique and depend on the chosen order of column additions.

$$
\begin{array}{c}
 \\
low(j) \\
low(i) \\
 \\
 \\

\end{array}
\begin{array}{c}
\begin{array}{ccccc} & & & i & j \end{array} \\
\left[\begin{array}{ccccc}
0 & 0 & 0 & * & * \\
 & 0 & 0 & * & \mathbf{1} \\
 & & 0 & \mathbf{1} & 0 \\
 & & & 0 & 0 \\
 & & & & 0
\end{array}\right]
\end{array}
$$

Figure 5.2: A reduced upper-triangular matrix with labels for the lowest ones of the non-zero columns.

Algorithm 6 Standard Column Reduction Algorithm

$R := \partial$; $U := \mathrm{Id}$; $C := 0$

for $j \in \{1, \ldots, m\}$ **do**

  while $\exists \ell < j$ with $\mathrm{low}(\ell) = \mathrm{low}(j) > -\infty$ **do**

    $R[., j] := R[., j] + R[., \ell]$

    $U[., j] := U[., j] + U[., \ell]$

    $C[\ell, j] := C[\ell, j] + 1$

Therefore, we prefer the *exhaustive column reduction algorithm*, which was already employed in [29]. It tries to remove non-zero entries in column j even after $\mathrm{low}(j)$ has been established, see Algorithm 7. The resulting matrices R and U do not depend on the specific implementation of the algorithm, they can be uniquely defined in terms of their algebraic structure (which we will discuss later).

Algorithm 7 Exhaustive Column Reduction Algorithm

$R := \partial$; $U := \mathrm{Id}$; $C := 0$

for $j \in \{1, \ldots, m\}$ **do**

  while $\exists \ell < j$ with $\mathrm{low}(\ell) > -\infty$ and $R[\mathrm{low}(\ell), j] \neq 0$ **do**

    $R[., j] := R[., j] + R[., \ell]$

    $U[., j] := U[., j] + U[., \ell]$

    $C[\ell, j] := C[\ell, j] + 1$

The two algorithms might compute different matrices R, but they are both reduced and have the same zero columns and the same lowest ones [11]. We relate this information with the persistent homology groups of the filtered simplicial complex. In both algorithms, R is

reduced from left to right. We interpret this as adding one simplex after the other to the simplicial complex. The addition of a p-simplex σ_j to the current subcomplex can have one of two possible outcomes [13]:

- column j of R is reduced to zero, in which case $U[.,j]$ contains a p-cycle and $\tilde{\beta}_p$ increases by 1, or
- column j of R remains non-zero, in which case $R[.,j]$ contains a p-boundary and $\tilde{\beta}_{p-1}$ decreases by 1.

In the first case, σ_j gives birth to a p-dimensional homology class, in the second it gives death to a $(p-1)$-dimensional homology class, namely the one represented by the $(p-1)$-cycle stored in column $U[.,\mathrm{low}(j)]$ [20]. The pairs $(\mathrm{low}(j), j)$ for the non-zero columns j correspond exactly to the finite points in the persistence diagram. We call the corresponding pairs of simplices *birth-death pairs*. For a zero column i, where i does not appear as the lowest one of any column, σ_i gives birth to a homology class that never dies (which we call *essential*).

Relative Cohomology. When computing persistent homology, there is a duality between (absolute) homology and relative cohomology [12]. Nearly the same algorithms can be used to compute both. Therefore, we will consider relative cohomology for filtered simplicial complexes in the following. The persistence diagrams for absolute and relative cohomology are the same, except for a dimension shift for the finite points [12].

We define relative cohomology for a pair (K, L), in which L is a subcomplex of K. The *relative p-cochains* are the p-cochains in $K \backslash L$. Since L is a subcomplex, their coboundaries are also in $K \setminus L$. Therefore, the *relative p-cocycles* are just the p-cocycles in $K \setminus L$, and the *relative p-coboundaries* are the p-coboundaries in $K \setminus L$. As in absolute cohomology, we get $\mathsf{B}^p(K,L) \subseteq \mathsf{Z}^p(K,L) \subseteq \mathsf{C}^p(K,L)$ for the corresponding groups and define the *p-th (reduced) relative cohomology group* as $\tilde{\mathsf{H}}^p(K,L) := \mathsf{Z}^p(K,L)/\mathsf{B}^p(K,L)$.

Row Reduction Algorithms. To compute relative persistent cohomology of a filtered simplicial complex, $\emptyset = K_0 \subseteq K_1 \subseteq \ldots \subseteq K_m = K$, where $K_\ell := \{\sigma_1, \sigma_2, \ldots, \sigma_\ell\}$, we consider relative cohomology for the sequence of pairs

$$(K, K = K_m), (K, K_{m-1}), \ldots, (K, K_0 = \emptyset).$$

In practice, this means we initially disregard all simplices, and in every step we consider one additional simplex for relative cohomology, in reversed order.

For the computation, we again start with the boundary matrix, but instead of the columns we rather consider the rows, as they contain the cofacets of the simplices. Since we consider the simplices in reversed order, we start at the bottom of the matrix and perform bottom-to-top row additions to reduce the matrix. Here, we call the matrix *(bottom-to-top) reduced* if the indices of the leftmost non-zero entries of the non-zero rows i, which we denote by $\text{left}(i)$, are unique. We set $\text{left}(i) = \infty$ if row i is zero.

We initialize Q, V, and D to the boundary matrix, the identity matrix, and the zero matrix, respectively. In analogy to column reduction, we reduce Q with bottom-to-top row additions while maintaining the relation $Q = V\partial$ and using D for book-keeping purposes. The *exhaustive row reduction algorithm* is given as Algorithm 8.

Algorithm 8 Exhaustive Row Reduction Algorithm

$Q := \partial$; $V := \text{Id}$; $D := 0$
for $i \in \{m, \dots, 1\}$ **do**
  while $\exists \ell > i$ with $\text{left}(\ell) < \infty$ and $Q[i, \text{left}(\ell)] \neq 0$ **do**
    $Q[i, .] := Q[i, .] + Q[\ell, .]$
    $V[i, .] := V[i, .] + V[\ell, .]$
    $D[i, \ell] := D[i, \ell] + 1$

Again, we get the "standard" version of the algorithm if we replace $Q[i, \text{left}(\ell)] \neq 0$ by $\text{left}(\ell) = \text{left}(i) < \infty$, stopping the reduction of a row as soon as we have established the leftmost one. No matter which of the two algorithms we choose and how they are specifically implemented, we always get the same zero rows and the same leftmost ones for the non-zero rows.

Zero rows of the reduced matrix correspond to birth and non-zero rows correspond to death simplices for relative persistent cohomology. The pairs $(\text{left}(i), i)$ for the non-zero rows i in Q correspond to the birth and death simplices of relative cohomology classes with finite persistence. They are closely related to the birth-death pairs in R. The following holds:

Lemma 5.1. *$low(j) = i$ in R $\Leftrightarrow$ $left(i) = j$ in Q.*

Proof. This is a direct consequence of the Pairing Uniqueness Lemma in [11], which states that $\text{low}(j) = i$ in R iff $r_\partial(i, j) = 1$, where we define

$$r_\partial(i, j) := \text{rank}\, \partial_i^j - \text{rank}\, \partial_{i+1}^j + \text{rank}\, \partial_{i+1}^{j-1} - \text{rank}\, \partial_i^{j-1}.$$

This implies that $\text{low}(j) = i$ depends on ∂ but not on R. Reflecting ∂ across its minor diagonal, and exchanging the two indices, we get $\text{left}(i) = j$ in Q iff $r_\partial(i, j) = 1$. The claimed equivalence follows. □

5.3 Tri-Partition

In this section, we present the tri-partition of the p-simplices of a simplicial complex, and we give a constructive proof for it. The result also holds for more general cell complexes, see [24]. We restrict our exposition to simplicial complexes, in order to be consistent with the rest of the work.

Trees and Cotrees. Let K be a simplicial complex. We define a *p-tree* as a subset of the p-simplices, $A_p \subseteq K^p$, that does not contain a non-empty p-cycle. We do not require connectedness. A p-tree is *maximal* if it is not properly contained in another p-tree of the same complex. Note the difference between a spanning tree and the maximal 1-tree of a graph: a spanning tree is necessarily connected and it also contains the vertices of the edges. As a consequence, a disconnected graph has a maximal 1-tree but no spanning tree. Similarly, we define a p-cotree as a subset of the p-simplices, $A^p \subseteq K^p$, that does not contain a non-empty p-cocycle, and call it *maximal* if it is not properly contained in another p-cotree.

Statement and Proof. We state the theorem about the tri-partition of K^p and construct such a tri-partition for every fixed ordering of the p-simplices. The ordering of the other simplices does not matter as long as the full ordering of K is monotonic.

Theorem 5.2 (Tri-Partition)**.** *Let K be a simplicial complex. For every dimension p, there exists a tri-partition of the p-simplices, $K^p = A_p \sqcup A^p \sqcup E_p$, such that A_p is a maximal p-tree, A^p is a maximal p-cotree, and $|E_p| = \tilde{\beta}_p$. There is a unique tri-partition for every monotonic ordering of K.*

Proof. We construct the tri-partition for a fixed monotonic ordering of K in three steps. First, we construct A_p, then we construct A^p, and finally we prove that $E_p = (K^p \setminus A_p) \setminus A^p$ has cardinality $\tilde{\beta}_p$.

We sort the rows and columns of the boundary matrix by the given monotonic ordering $K = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ and use the exhaustive column reduction algorithm stated as Algorithm 7. After reduction, column j remains non-zero if it is independent of the preceding columns. Thus, the corresponding simplex σ_j together with the set of simplices corresponding to earlier non-zero columns do not contain a cycle. We add σ_j to A_p, where $p = \dim \sigma_j$. In the case that column j gets reduced to zero, it was a combination of previous columns. The corresponding simplex gives birth to a cycle and we do not add it to A_p. In the end, A_p is a maximal p-tree. It contains exactly the p-dimensional death simplices of K.

For the second step, we use the exhaustive row reduction algorithm stated as Algorithm 8. After reduction, row i is zero if it was a combination of the succeeding rows and the corresponding simplex σ_i gives birth to a cocycle. Otherwise, it remains non-zero and was independent of the succeeding rows. In that case we add σ_i to A^p, where $p = \dim \sigma_i$. In the end, A^p is a maximal p-cotree that contains exactly the p-dimensional simplices that give death to relative cohomology classes.

We still have to show that $A_p \cap A^p = \emptyset$. Assume that $\sigma_i \in A^p$. Then, $\dim \sigma_i = p$ and $\text{left}(i) < \infty$. Writing $j = \text{left}(i)$, by Lemma 5.1 this is equivalent to $i = \text{low}(j)$. It follows that σ_i gives birth to a homology class, hence $\sigma_i \notin A_p$. With the symmetric argument we get that no simplex of A_p is contained in A^p.

Finally, we prove that $E_p = (K^p \setminus A_p) \setminus A^p$ has cardinality $\tilde{\beta}_p$. If a p-simplex σ is contained in E_p, then the corresponding column in the boundary matrix is zero after column reduction and the corresponding row is zero after row reduction. So, σ gives birth to a p-cycle, but by Lemma 5.1 there is no corresponding death simplex. There are $\tilde{\beta}_p$ independent essential p-cycles, thus $|E_p| = \tilde{\beta}_p$. □

There is a unique tri-partition for every monotonic ordering, but different monotonic orderings do not necessarily have different tri-partitions. Considering the collection of all tri-partitions generated by monotonic orderings of K, it is an interesting fact that we

get the maximal sets of three *matroids*[1] [24]: the collection of sets A_p, of sets A^p, and of sets E_p.

Although the tri-partition depends on the monotonic ordering, the cardinalities of the three parts do not. Write n_p° and $n_p^\bullet$ for the number of p-simplices that give birth and death, respectively. Then the cardinalities of the three parts are $n_p^\bullet$, $(n_p^\circ - \tilde{\beta}_p)$, and $\tilde{\beta}_p$. Their sum is the total number of p-simplices, $n_p = n_p^\circ + n_p^\bullet$, and for the reduced Betti numbers we have $\tilde{\beta}_p = n_p^\circ - n_{p+1}^\bullet$. Therefore, we can express both the n_p° and the $n_p^\bullet$ in terms of the n_p and the $\tilde{\beta}_p$, which are invariants that do not depend on the ordering of the simplices or the particular reduction algorithm.

5.4 Canonical Bases

With the exhaustive reduction algorithms we can compute more than just the tri-partitions of a simplicial complex, we also get canonical bases in homology and cohomology. They are *canonical* in the sense that they are defined in terms of their algebraic properties and do not depend on the algorithm that computes them.

Canonical Cycles and Chains. Write $K^p = A_p \sqcup A^p \sqcup E_p$ for the tri-partition of the p-simplices of a simplicial complex K with a fixed monotonic ordering. As A_p is a maximal p-tree, for every $\sigma_i \in A^p \sqcup E_p$ there is a unique p-cycle $\mathsf{z}_p(\sigma_i) \subseteq A_p \sqcup \{\sigma_i\}$, which we call the *canonical p-cycle* of σ_i. Symmetrically, we get the *canonical p-cocycle* $\mathsf{z}^p(\sigma_i) \subseteq A^p \sqcup \{\sigma_i\}$ for every $\sigma_i \in A_p \sqcup E_p$.

For the $\sigma_j \in A_p$, there is a unique sum of canonical and non-trivial $(p-1)$-cycles in K_{j-1} that is rendered trivial by the addition of σ_j. Denoting this $(p-1)$-cycle by z, there is a unique p-chain $\mathsf{c}_p(\sigma_j) \subseteq A_p$ with $\partial \mathsf{c}_p(\sigma_j) = \mathsf{z}$, which we refer to as the *canonical p-chain* of σ_j. Symmetrically, we define the *canonical p-cochain* $\mathsf{c}^p(\sigma_j)$ for every $\sigma_i \in A^p$.

An example for a canonical cycle and a canonical chain is shown in Figure 5.3.

We show that each cycle is the unique sum of non-trivial canonical cycles. This will become useful in proving Theorems 5.4 and 5.6 of this section.

[1] Let B be a finite set and $\mathcal{C}$ a collection of subsets of B. We call $(\mathcal{C}, B)$ a *matroid* if $\emptyset \in \mathcal{C}$, $\mathcal{C}$ is closed under taking subsets, and the following property holds: $C, D \in \mathcal{C}$ with $|C| < |D|$ implies that there exists $d \in D$ such that $C \cup \{d\} \in \mathcal{C}$.

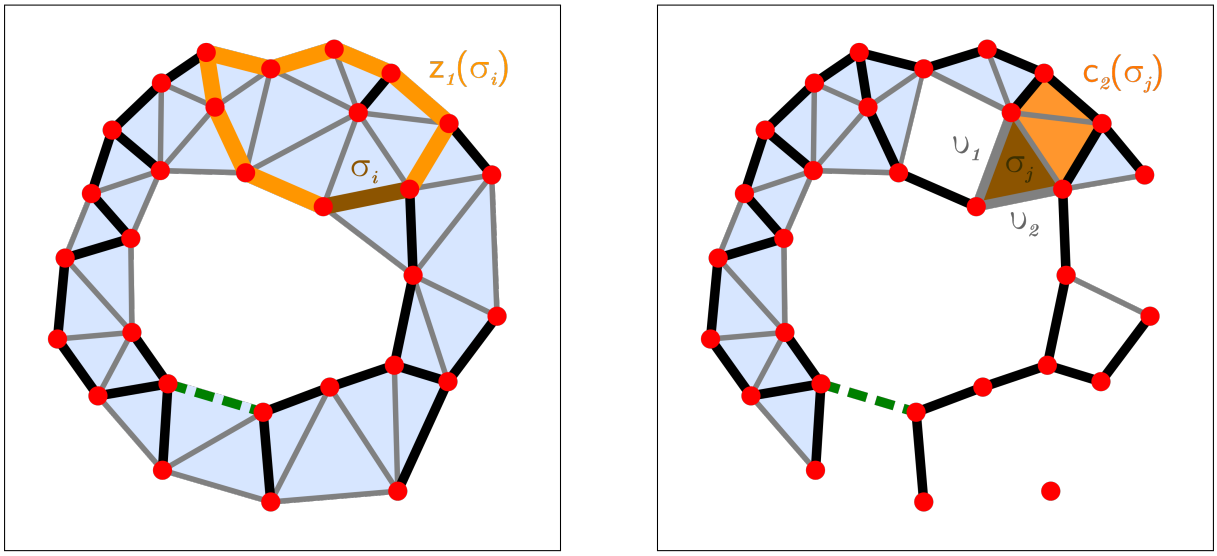


Figure 5.3: Example of a canonical cycle and a canonical chain in the simplicial complex of Figure 5.1, filtered by the Delaunay radius function. The colors for the tri-partition of the edge set are the same as in Figure 5.1. *Left:* The canonical 1-cycle $\mathsf{z}_1(\sigma_i)$ of σ_i is highlighted in orange, it includes σ_i which is drawn in brown. *Right:* Only the simplices of the subcomplex $K_j = \{\sigma_1, \sigma_2, \ldots, \sigma_j\}$ are shown. The canonical 2-chain $\mathsf{c}_2(\sigma_j)$ of σ_j is colored in orange, it includes σ_j which is drawn in brown. The addition of σ_j renders the sum of the canonical 1-cycles of v_1 and v_2 trivial. It is the boundary of $\mathsf{c}_2(\sigma_j)$.

Lemma 5.3. *For every p-cycle z there is a unique set of simplices $B \subseteq A^p \sqcup E_p$ with non-trivial canonical cycles such that $\mathsf{z} = \sum_{\sigma \in B} \mathsf{z}_p(\sigma)$.*

Proof. Let B be the simplices in z that are contained in $A^p \sqcup E_p$. The sum of their canonical cycles $\sum_{\sigma \in B} \mathsf{z}_p(\sigma)$ is the same as z. If they were different, then $\mathsf{z} + \sum_{\sigma \in B} \mathsf{z}_p(\sigma)$ would be a non-trivial p-cycle contained in A_p, which is a contradiction. For any other sum of non-trivial canonical cycles, the set of simplices in $A^p \sqcup E_p$ is not the same as B, so this representation of z is unique. □

Canonical Cycles and Chains Stored in Matrices. No additional work has to be done for computing the canonical cycles and chains, we get them directly after exhaustive matrix reduction.

Theorem 5.4 (Canonical Cycles and Chains in Matrices)**.** *The columns of matrix U, computed by exhaustive column reduction of $R = \partial U$, store the canonical cycles and*

chains. And the rows of matrix V, computed by exhaustive row reduction of $Q = V\partial$, store the canonical cocycles and cochains.

$$U[.,j] = \begin{cases} \mathsf{z}_p(\sigma_j), & \text{if } \sigma_j \in A^p \sqcup E_p, \\ \mathsf{c}_p(\sigma_j), & \text{if } \sigma_j \in A_p. \end{cases}$$

$$V[i,.] = \begin{cases} \mathsf{z}^p(\sigma_i), & \text{if } \sigma_i \in A_p \sqcup E_p, \\ \mathsf{c}^p(\sigma_i), & \text{if } \sigma_i \in A^p. \end{cases}$$

First, we prove a lemma about the off-diagonal entries of the matrices after reduction.

Lemma 5.5. *Let $R = \partial U$ and $Q = V\partial$ be the matrices after exhaustive reduction. For every $i \neq j$ there is a dimension p such that $U[i,j] = 1$ implies $\sigma_i \in A_p$, i.e., all non-zero off-diagonal entries in U belong to rows of simplices in A_p. Furthermore, $V[i,j] = 1$ implies $\sigma_j \in A^p$.*

Proof. We prove the claim for matrix U, the argument for V is symmetrical. At the start of the reduction algorithm, U is the identity matrix and the statement is trivially true. A column ℓ is added to column j only if $\ell < j$ and the column $R[.,\ell]$ is non-zero, which implies that $\sigma_\ell \in A_p$. We inductively assume that the off-diagonal non-zero entries in $U[.,\ell]$ are in rows corresponding to simplices in A_p and see that the column operation maintains this statement for all columns of U. □

We are now ready to prove Theorem 5.4.

Proof of Theorem 5.4. We only prove the claims about the canonical cycles and chains, the proof for their cohomological counterparts is symmetrical.

First, let $\sigma_j \in A^p \sqcup E_p$. After the reduction of column j, $R[.,j]$ is zero and $U[.,j]$ stores a cycle. By Lemma 5.5, this cycle is a subset of $A_p \sqcup \{\sigma_j\}$. The canonical cycle $\mathsf{z}_p(\sigma_j)$ is the only such cycle, which implies that it is stored in $U[.,j]$.

Now, consider the case that $\sigma_j \in A_p$. Then, all the non-zero entries of $U[.,j]$ are in rows of simplices in A_p, including σ_j. We write $\mathsf{c} \subseteq A_p$ for this chain and z for its boundary, which is stored in $R[.,j]$. In the proof of Lemma 5.3 we have seen that $\mathsf{z} = \sum_{\sigma \in (A^{p-1} \sqcup E_{p-1}) \cap \mathsf{z}} \mathsf{z}_{p-1}(\sigma)$ and this is a unique representation as a sum of non-trivial canonical cycles. Since the column reduction algorithm is exhaustive, every canonical $(p-1)$-cycle in the sum is born before σ_j and dies after σ_j. By construction, z goes from non-trivial to trivial when we add σ_j. This implies that $U[.,j]$ stores $\mathsf{c}_p(\sigma_j)$. □

Canonical Bases. By means of the canonical cycles and chains, we get bases for the cycle, the boundary, and the homology groups, as well as their cohomological counterparts.

Theorem 5.6 (Canonical Bases)**.** *Let K be a simplicial complex with a fixed monotonic ordering and let $K^p = A_p \sqcup A^p \sqcup E_p$ be the corresponding tri-partition. Then*

(i) $\{\mathsf{z}_p(\sigma) \mid \sigma \in A^p \sqcup E_p\}$ *is a basis of* $\mathsf{Z}_p(K)$.

(ii) $\{\mathsf{z}_p(\sigma) \mid \sigma \in E_p\}$ *generates a basis of* $\tilde{\mathsf{H}}_p(K)$.

(iii) $\{\partial\mathsf{c}_p(\sigma) \mid \sigma \in A_p\}$ *is a basis of* $\mathsf{B}_{p-1}(K)$.

(iv) $\{\mathsf{z}^p(\sigma) \mid \sigma \in A_p \sqcup E_p\}$ *is a basis of* $\mathsf{Z}^p(K)$.

(v) $\{\mathsf{z}^p(\sigma) \mid \sigma \in E_p\}$ *generates a basis of* $\tilde{\mathsf{H}}^p(K)$.

(vi) $\{\delta\mathsf{c}^p(\sigma) \mid \sigma \in A^p\}$ *is a basis of* $\mathsf{B}^{p+1}(K)$.

Proof. We only prove (i)-(iii), the rest follows by symmetry.

By Lemma 5.3, every cycle has a unique representation as a sum of canonical cycles, so $\{\mathsf{z}_p(\sigma) \mid \sigma \in A^p \sqcup E_p\}$ is indeed a basis of $\mathsf{Z}_p(K)$.

By construction, $\mathsf{z}_p(\sigma)$ generates an essential class if $\sigma \in E_p$. On the other hand, if $\sigma \in A^p$, then the corresponding canonical cycle is trivial in K or it is homologous to a sum of cycles defined by simplices in E_p. Since $\tilde{\mathsf{H}}_p(K)$ requires $\tilde{\beta}_p$ generators and there are exactly $\tilde{\beta}_p$ simplices in E_p, all of their canonical cycles must be generators of $\tilde{\mathsf{H}}_p(K)$.

Finally, we show that the boundaries of the canonical chains of the simplices in A_p form a basis of $\mathsf{B}_{p-1}(K)$. These cycles are independent, because otherwise there would be a non-empty sum of canonical chains with empty boundary, but all the chains are contained in A_p, which does not contain a cycle. The size of the set $\{\partial\mathsf{c}_p(\sigma) \mid \sigma \in A_p\}$ is the same as the number of p-dimensional death simplices $n_p^\bullet$. By definition, the rank of $\mathsf{B}_{p-1}(K)$ is equal to the rank of $\mathsf{Z}_{p-1}(K)$ minus the rank of $\tilde{\mathsf{H}}_{p-1}(K)$. The first is the same as the number of p-dimensional birth simplices n_p°, and the latter is the $(p-1)$-th reduced Betti number, which can be expressed as a difference between the number of $(p-1)$-dimensional birth simplices and the number of p-dimensional death simplices: $n_{p-1}^\circ - \tilde{\beta}_{p-1} = n_{p-1}^\circ - (n_{p-1}^\circ - n_p^\bullet) = n_p^\bullet$. This is the same as the number of elements in $\{\partial\mathsf{c}_p(\sigma) \mid \sigma \in A_p\}$, so we conclude that they form a basis of $\mathsf{B}_{p-1}(K)$. □

5.5 Connection to Helmholtz-Hodge Theory

The tri-partition of the p-simplices of a simplicial complex into a p-tree, a p-cotree, and a third set of size $\tilde{\beta}_p$ can be viewed as a discrete analogue of the Helmholtz-Hodge decomposition of a smooth vector field into a rotation-free component, a divergence-free component, and a harmonic component whose dimension is the rank of the homology group [35, 36].

An algebraic version of the composition states that

$$\mathsf{C}_p \simeq \mathsf{B}_p \oplus \mathsf{B}^p \oplus \tilde{\mathsf{H}}_p, \tag{5.1}$$

that is, every p-chain can be written as a sum of a p-boundary, a p-coboundary, and a *harmonic* p-chain (which is both a p-cycle and a p-cocycle) [16]. The tri-partition of Theorem 5.2 implies that every p-simplex either gives death to a $(p-1)$-cycle, gives birth to a p-cycle that later dies (which is equivalent to giving death to a $(p+1)$-cocycle in relative cohomology), or gives birth to an essential p-cycle. We can formulate this also as $\mathsf{C}_p \simeq \mathsf{B}_{p-1} \oplus \mathsf{B}^{p+1} \oplus \tilde{\mathsf{H}}_p$. Note that $\mathsf{B}_{p-1} \simeq \mathsf{B}^p$ and $\mathsf{B}^{p+1} \simeq \mathsf{B}_p$, from which we get the standard form (5.1). This holds because the rank of B_p is the number of pairs $i = \mathrm{low}(j)$ with $\dim \sigma_i = p$ in the column reduced matrix R. Similarly, the rank of B^{p+1} is the number of pairs $j = \mathrm{left}(i)$ with $\dim \sigma_j = p+1$ in the row reduced matrix Q. By Lemma 5.1, the two ranks are equal and the two vector spaces are isomorphic.

Chapter 6

Manipulating Holes

The canonical bases for the cycle, cocycle, boundary, and coboundary groups presented in the last chapter provide us with a means of manipulating holes in a filtered simplicial complex. Given a fixed subcomplex, we will define algorithms for opening and closing holes of any dimension.

Actually, there are four different operations to manipulate a given hole and it depends on the current subcomplex which of them are applicable. We can

- *lock:* complete the cycle around the hole if it was not born yet (advance its birth),
- *fill:* add simplices to fill the hole if it is part of the current subcomplex (advance its death),
- *unfill:* remove simplices to open the hole if it has already died (delay its death), or
- *unlock:* destroy the cycle around the hole if it is part of the current subcomplex (delay its birth).

We have these operations for holes of any dimension, which we indicate by writing $\texttt{Lock}_p$, $\texttt{Fill}_p$, $\texttt{Unfill}_p$, and $\texttt{Unlock}_p$ for the operations on a p-dimensional hole. Figure 6.1 illustrates the four operations for a 1-dimensional hole.

We can realize the first two operations by adding basis vectors in homology, and the second two by removing basis vectors in cohomology. Dependences between the holes

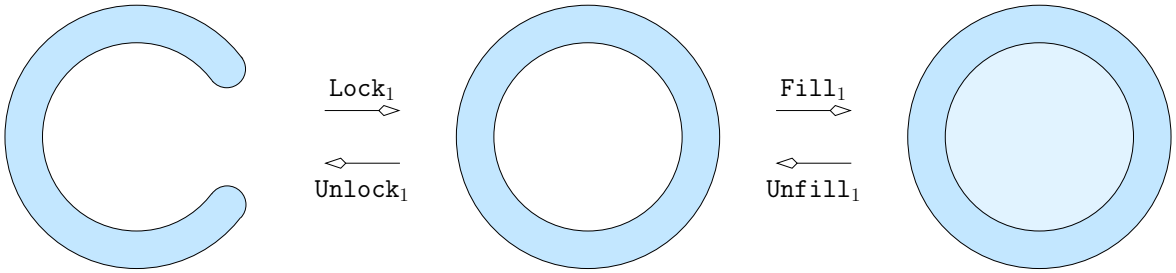


Figure 6.1: Manipulating a 1-dimensional hole. A p-cycle can be created by adding a p-dimensional piece (*lock*) or by removing a $(p+1)$-dimensional piece (*unfill*). Symmetrically, a p-cycle can be destroyed by removing a p-dimensional piece (*unlock*) or by adding a $(p+1)$-dimensional piece (*fill*).

lead to side effects of these operations. Therefore, we also propose an alternative way of manipulating holes respecting the dependence structure.

In the following chapter, we analyze the dependence structure of holes, present two approaches to hole manipulation and illustrate them with examples in 2 and 3 dimensions. The results of this chapter were partly published in [23].

6.1 Changes to the Persistence Diagram

We use the persistence diagram to visualize the effects of the hole manipulating operations. Furthermore, it is a useful guide to select the holes for manipulation.

Subcomplex Quadrants. The persistence diagram stores the birth and death times for all the holes in a filtered simplicial complex. For a given filtration value, the points in the upper-left quadrant anchored on the diagonal at that value correspond to the homology classes in the corresponding subcomplex, see the left panel of Figure 6.2. The horizontal border of the quadrant is open, while the vertical border is closed: it includes points with a birth value at most as big as the filtration value and a death value strictly bigger than the filtration value (the death value can also be ∞).

Moving Points. The hole manipulating operations can be interpreted as moving a point in the persistence diagram. The lock and the unlock operations advance or delay the birth of a hole, which means a movement to the left or the right. Filling and unfilling

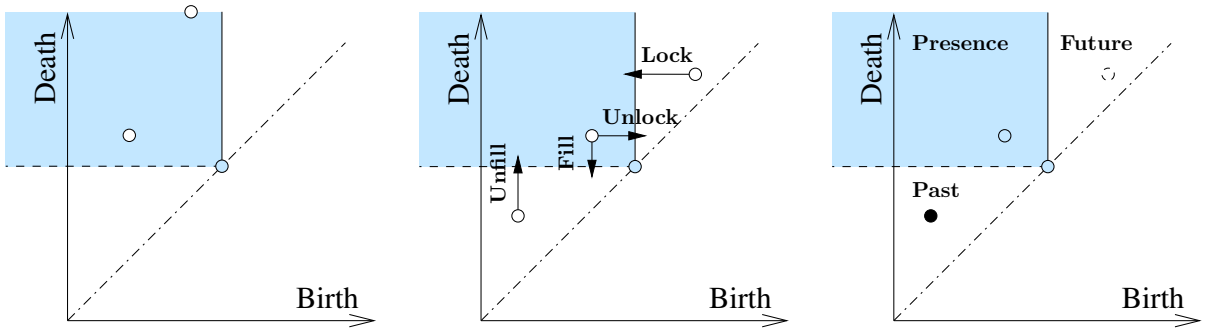


Figure 6.2: Persistence diagram and hole manipulations. *Left:* The upper-left quadrant anchored on the diagonal highlights the points that correspond to the homology of a subcomplex. *Middle:* The arrows indicate the movement of points in the persistence diagram corresponding to the four hole manipulating operations. *Right:* The points in the persistence diagram are decorated with status information corresponding to an unmodified subcomplex. We draw points in the past, presence, and future with a filled, empty, and dashed circle, respectively.

correspond to advancing or delaying the death of a hole, which means a downward or an upward movement. This is illustrated in the second panel of Figure 6.2.

Persistence Diagram with Status Information. We however choose a different view in the following. We do not want the persistence diagram to change with the hole manipulations. Instead, we assign a status to each point (and the corresponding hole) and only modify the status information with the operations. A point $(b, d) \in \mathbb{R} \times (\mathbb{R} \cup \{\infty\})$ in the persistence diagram gets assigned one of the following three *status*, depending on the filtration value r of the current subcomplex: the point is in the *future* iff $r < b$, it is in the *presence* iff $b \leq r < d$, and it is in the *past* iff $d \leq r$. The right panel of Figure 6.2 illustrates the regions of points with the same status in the persistence diagram, assuming the subcomplex has not been modified. The hole manipulating operations will change the status of the points.

6.2 First Approach: Adding and Removing Basis Vectors

Our first approach to opening and closing holes is to add and remove basis vectors in homology and cohomology.

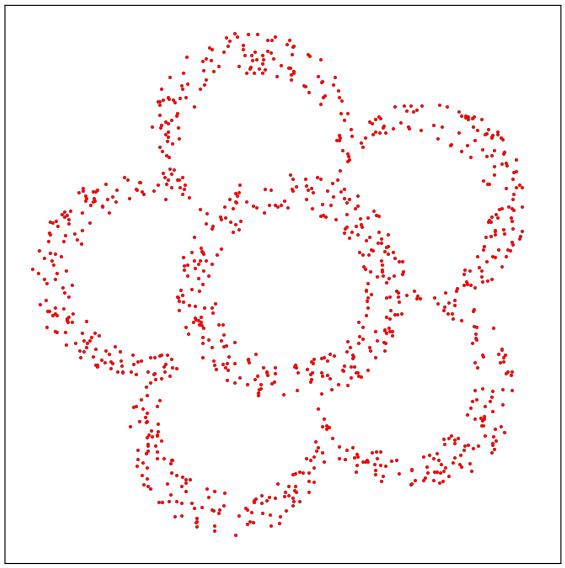
Operations. We realize the operations by assigning new filtration values to some of the simplices. If we want to add a simplex to the current subcomplex for value r, the new value has to be smaller than or equal to r. On the other hand, if we want to remove a simplex, the new value has to be strictly bigger than r. At the same time, we have to take care that the ordering of the simplices stays monotonic.

We formulate the four operations in the following. Let K be a simplicial complex with filtration function $f\colon K \to \mathbb{R}$, we consider the subcomplex for value r, $\{\sigma \in K \mid f(\sigma) \leq r\}$. Consider a p-dimensional hole with birth simplex σ and death simplex τ (or none if the hole is essential), corresponding to the point $(f(\sigma), f(\tau))$ (or $(f(\sigma), \infty)$) in the persistence diagram.

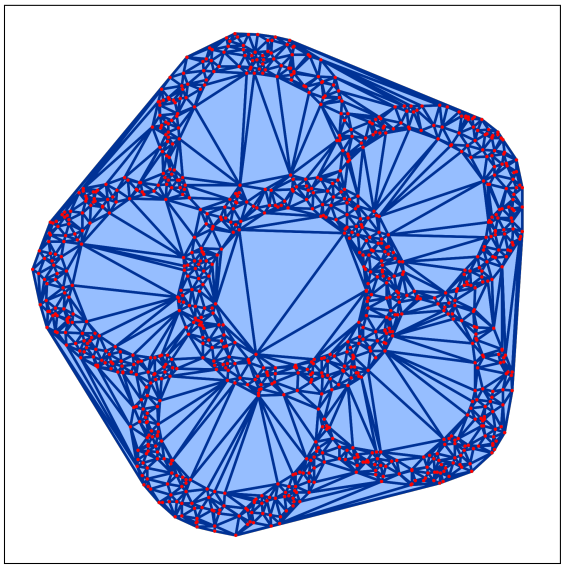
- *Locking:* We assume that the hole is in the future: $r < f(\sigma)$. We add the simplices of the canonical cycle $\mathsf{z}_p(\sigma)$ and all their faces to the subcomplex.
- *Filling:* We assume that the hole is in the presence: $f(\sigma) \leq r < f(\tau)$. We add the simplices of the canonical chain $\mathsf{c}_{p+1}(\tau)$ and all their faces to the subcomplex.
- *Unfilling:* We assume that the hole is in the past: $r \geq f(\tau)$. We remove the simplices of the canonical cocycle $\mathsf{z}^{p+1}(\tau)$ and all their cofaces from the subcomplex.
- *Unlocking:* We assume that the hole is in the presence: $f(\sigma) \leq r < f(\tau)$. Depending on whether the hole is essential or not, we remove the simplices of the canonical cocycle $\mathsf{z}^p(\sigma)$ or the canonical cochain $\mathsf{c}^p(\sigma)$ and all their cofaces from the subcomplex.

Example for the Alpha Complex in $\mathbb{R}^2$. We illustrate the results of the hole manipulating operations on Alpha complexes of a 2-dimensional point set consisting of 1000 points sampled from the drawing of a flower.

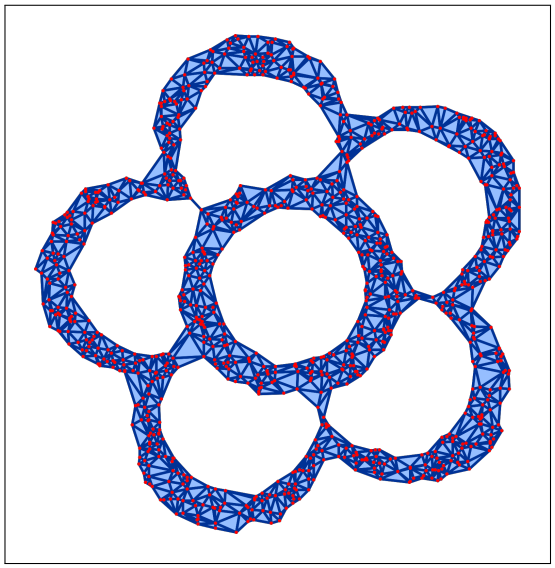
Figure 6.3 shows the point set, its Delaunay triangulation, the Alpha complex for $r = 25.0$, and the persistence diagram with the corresponding quadrant. The diagram guides the application of the hole manipulating operations. Points that represent 0-dimensional homology classes are drawn in orange, and those for 1-dimensional homology classes in blue. We see that there are six 1-dimensional holes in the Alpha complex for $r = 25.0$, they correspond to the five petals and the center of the flower. The central hole is the most persistent one.



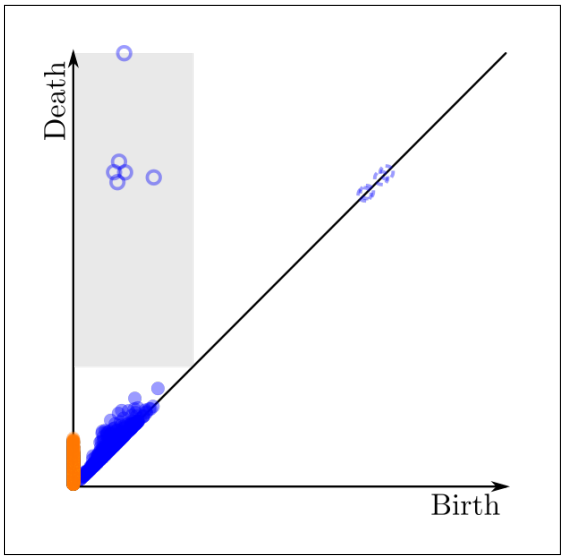
(a) Point set.



(b) Delaunay triangulation.



(c) Alpha complex for $r = 25.0$.



(d) Persistence diagram.

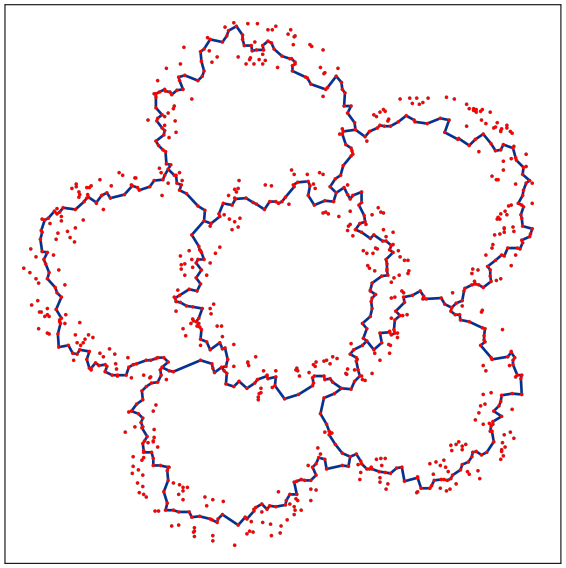
Figure 6.3: (a) A point set in $\mathbb{R}^2$, (b) its Delaunay triangulation, and the (c) Alpha complex for $r = 25.0$. The highlighted quadrant in the (d) persistence diagram contains the points corresponding to holes in the Alpha complex.

We give one example for each of the four operations in Figure 6.4. Locking the six most persistent 1-holes in the Alpha complex for $r = 0.0$ gives a 1-dimensional complex with six loops, see panel (a). If we think of the data as a noisy sample of a line drawing, this could serve as a reconstruction while preserving the homotopy type. The Alpha complex for $r = 25.0$ has the same number of holes, but contains many more edges and also triangles. We remove the central 1-hole in this complex by filling and unlocking, see panels (b) and (c). Note that as a side effect the unlocking operation also unlocks two of the five petals, in order to connect the inner hole with the outside. For unfilling the central 1-hole in the Delaunay triangulation, it is sufficient to remove a single triangle, see panel (d).

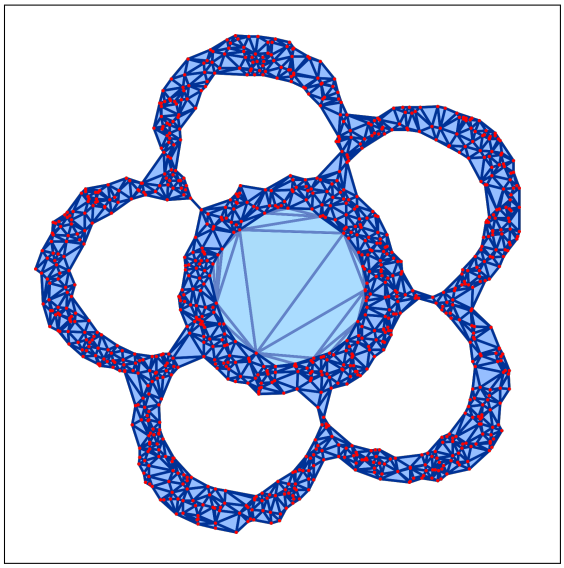
Example for the Alpha Complex in $\mathbb{R}^3$. As an example in $\mathbb{R}^3$, we study the effects of the hole manipulating operations on the Alpha complexes of a protein, namely Gramicidin A. This is a small protein that is embedded in the cell membrane and acts as an ion channel, which explains the tunnel alongside the structure. We get the coordinates of its atom centers from the Protein Data Bank (PDB ID: 1NT5) [5] and construct Alpha complexes based on this point set.

Figure 6.5 shows the molecular structure of the protein [5], the point set, the Alpha complexes for various values of r, and the persistence diagram with the corresponding quadrants. We observe that one point is significantly more persistent than the others, it corresponds to the ion channel of the protein. The tunnel is born at $r = 0.93$ and it is the only hole of the Alpha complex at $r = 2.35$. In the quadrant for $r = 0.93$, we seem to see only two additional points, but they are of higher multiplicity and correspond to the many pentagons and hexagons in the Alpha complex.

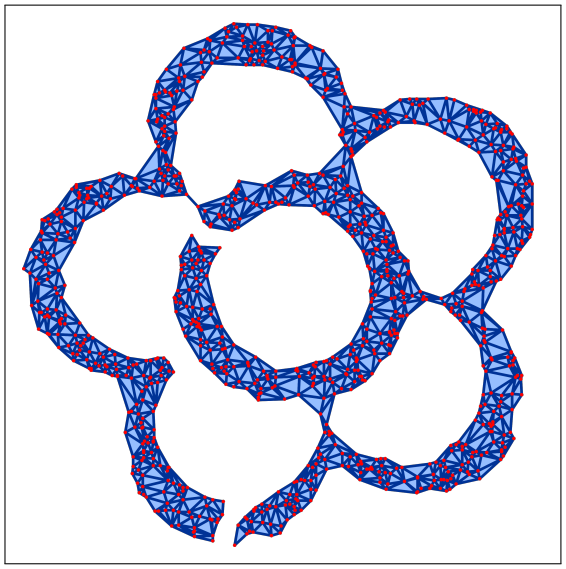
We use the operations to manipulate the 1- and 2-hole of highest persistence, see Figure 6.6. Locking at $r = 0.0$ reveals the canonical cycles of the holes. The 1-cycle goes around the tunnel, while the 2-cycle encloses most of its volume, which implies that the most narrow parts of the tunnel are located near its ends. Filling the 1-hole at its birth time $r = 0.93$ results in almost the same surface, except that it remains open at one end. Unlocking the 1-hole at $r = 2.35$ gives a surprising result: instead of slicing open the cylinder with a straight cut along one side, we see a spiral cut that leaves a spiraling tube revealing the helix structure of the protein. Indeed, between contiguous 360° turns of the helix the connections are weaker and the distances larger than along the helix, so cutting there



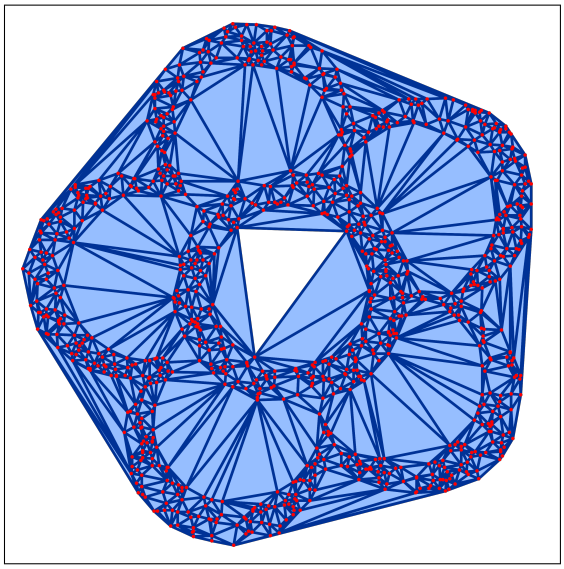
(a) $\texttt{Lock}_1$ at $r = 0.0$.



(b) $\texttt{Fill}_1$ at $r = 25.0$.

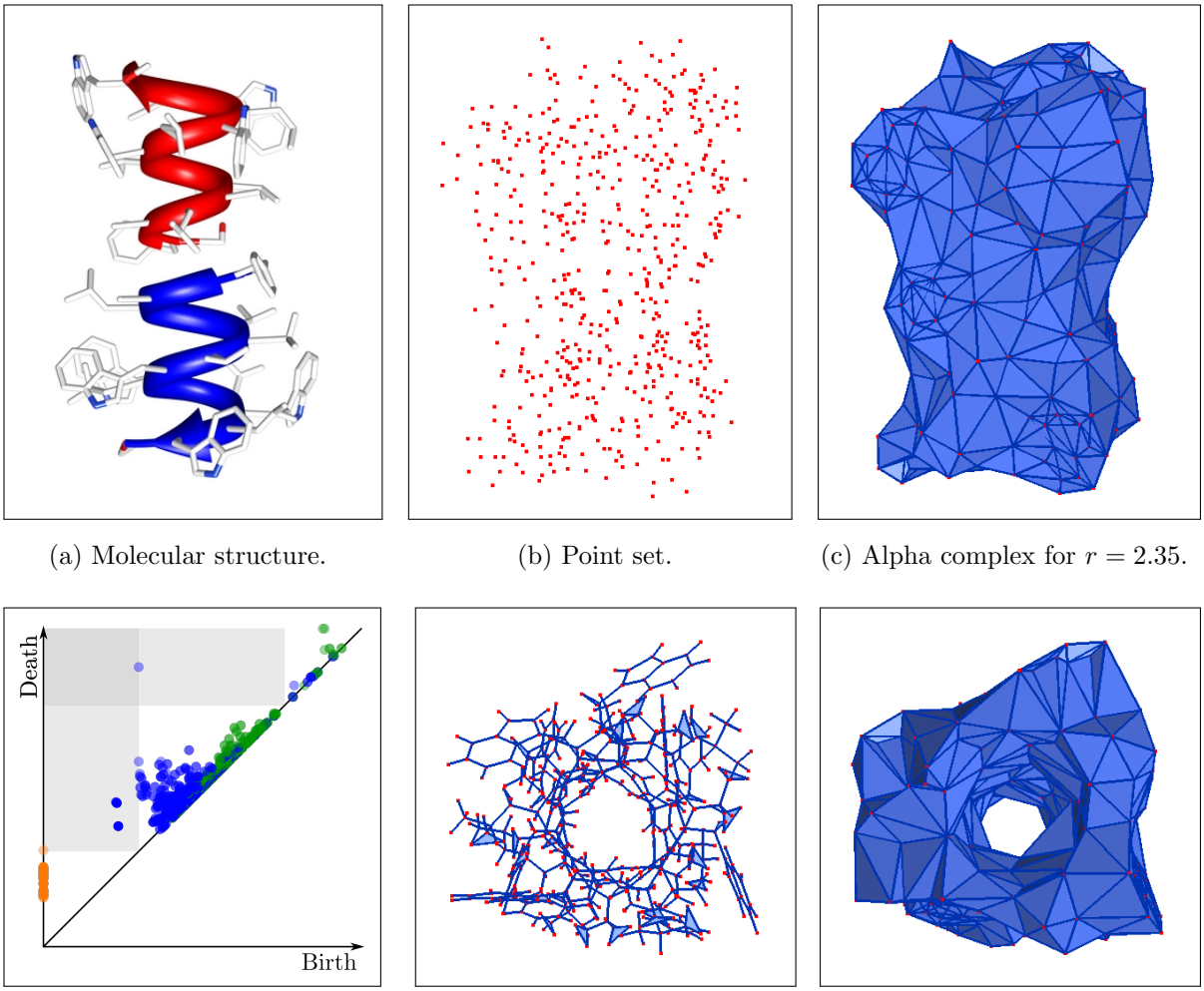


(c) $\texttt{Unlock}_1$ at $r = 25.0$.



(d) $\texttt{Unfill}_1$ at $r = \infty$.

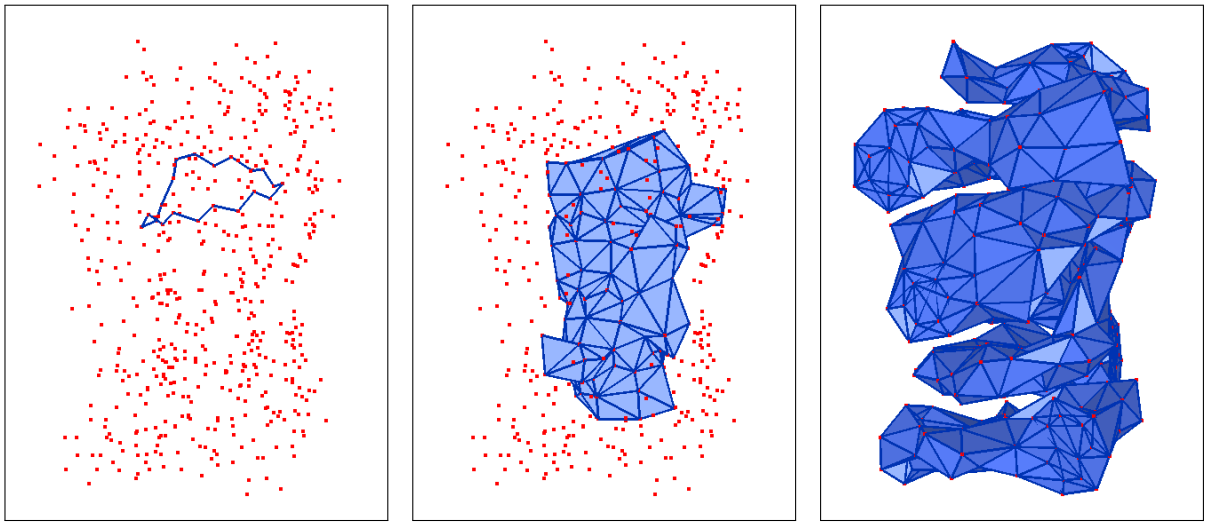
Figure 6.4: We manipulate holes in the Alpha complexes of Figure 6.3 by adding and removing canonical basis vectors. We (a) lock the six most persistent 1-holes at $r = 0.0$, (b) fill and (c) unlock the 1-hole of highest persistence at $r = 25.0$, and unfill the same hole at $r = \infty$.



(a) Molecular structure. (b) Point set. (c) Alpha complex for $r = 2.35$.

(d) Persistence diagram. (e) Alpha complex for $r = 0.93$. (f) Alpha complex for $r = 2.35$.

Figure 6.5: From the Protein Data Bank we get the (a) molecular structure of Gramicidin A [5] and its (b) atom centers. We show the Alpha complexes of this 3-dimensional point set for (e) $r = 0.93$ and (c),(f) $r = 2.35$ in side or top view. Triangles with no incident tetrahedra are drawn in pale blue, while the others are drawn in darker blue, with the shade depending on their normal vector. In the (d) persistence diagram the orange, blue, and green points correspond to 0-, 1-, and 2-holes, respectively. The quadrants for the two subcomplexes are highlighted.



(a) $\texttt{Lock}_1$ at $r = 0.0$. (b) $\texttt{Lock}_2$ at $r = 0.0$. (c) $\texttt{Unlock}_1$ at $r = 2.35$.

(d) $\texttt{Fill}_1$ at $r = 0.93$. (e) $\texttt{Fill}_1$ at $r = 0.93$. (f) $\texttt{Unfill}_1$ at $r = 3.10$.

Figure 6.6: We manipulate holes in the Alpha complexes of Gramicidin A by adding and removing canonical basis vectors. We lock the most persistent (a) 1- and (b) 2-hole at $r = 0.0$. All the other operations target the most persistent 1-hole. We (d),(e) fill it at its birth time $r = 0.93$, (c) unlock it at $r = 2.35$, and (f) unfill it at $r = 3.10$.

is the action of least effort. Finally, we unfill the 1-hole at $r = 3.10$ by removing the corresponding canonical cocycle. This results in a narrow tunnel that passes through the length of the protein, although it is barely visible.

Modifications for the Wrap Complex. For the Wrap complex, we modify the operations formulated above slightly, so that we make use of the special role of the critical simplices in the Wrap complex. Every Wrap complex is the union of lower sets of critical simplices. Instead of modifying all the simplices in the complex directly, we only want to change the set of critical simplices and maintain the Wrap complex as the union of their lower sets.

When manipulating a p-dimensional hole with birth simplex σ and death simplex τ (if the latter exists) in a Wrap complex for a fixed radius, we change the current set of critical simplices in the following way:

- *Locking:* For all the simplices in the canonical cycle $\mathsf{z}_p(\sigma)$, add them to the current set of critical simplices if they are critical, or add their critical descendant of smallest radius otherwise.

- *Filling:* For all the simplices in the canonical chain $\mathsf{c}_{p+1}(\tau)$, add them to the current set of critical simplices if they are critical, or add their critical descendant of smallest radius otherwise.

- *Unfilling:* For all the simplices in the canonical cocycle $\mathsf{z}^{p+1}(\tau)$, remove them from the current set of critical simplices if they are critical, and remove all their critical descendants in any case.

- *Unlocking:* For all the simplices in the canonical cochain $\mathsf{c}^p(\sigma)$ (or the canonical cocycle $\mathsf{z}^p(\sigma)$ if the hole is essential), remove them from the current set of critical simplices if they are critical, and remove all their critical descendants in any case.

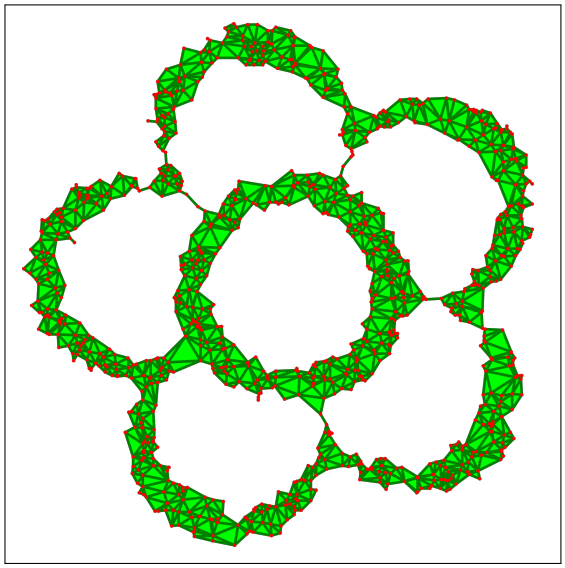
We get the modified Wrap complex as the union of lower sets of the updated set of critical simplices.

Example for the Wrap Complex in $\mathbb{R}^2$. We want to contrast the results we have seen before for manipulating holes in Alpha complexes with the ones we get for the corresponding Wrap complexes. We start with the 2-dimensional point set of Figure 6.3. The Wrap complexes for $r = 25.0$ and $r = \infty$ are displayed in the upper panels of Figure 6.7. They are subsets of the corresponding Alpha complexes. Recall that we get them from the Alpha complexes by collapsing non-singular intervals from the outside. Since the Alpha and Wrap complexes are homotopy equivalent, the persistence diagram is the same.

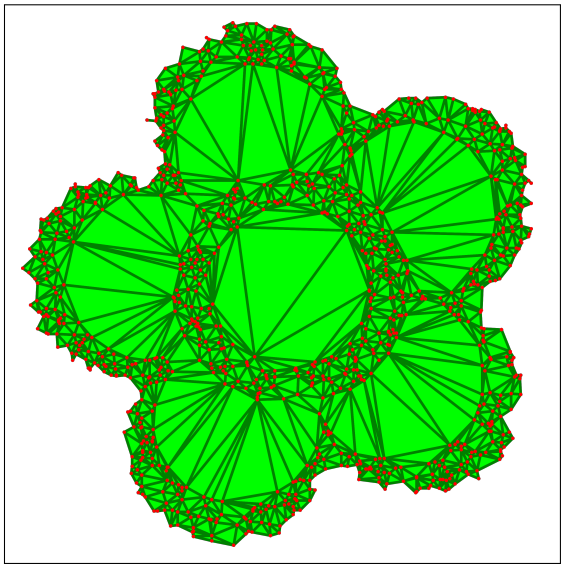
The lock and the fill operation applied to the 1-holes of highest persistence just add the corresponding canonical cycles or chains to the complex. The results are as in the upper panels of Figure 6.4. We can see differences, however, for the other two operations. The results are shown in the lower panels of Figure 6.7. When we unlock the most persistent 1-hole, not only the corresponding canonical chain is removed but also some non-critical simplices that were in the lower set of the canonical chain. For the unfill operation, the difference to the result for the Alpha complex is striking. In addition to the single simplex of the canonical cocycle, many simplices in its lower set get removed. A big hole in the center of the complex is opened.

Example for the Wrap Complex in $\mathbb{R}^3$. Wrap complexes for our 3-dimensional example, which is the point set representing the atoms of Gramicidin A, are displayed in Figure 6.8. Again, due to collapses of non-singular intervals, they contain less simplices than the Alpha complexes for the same radii shown in Figure 6.5.

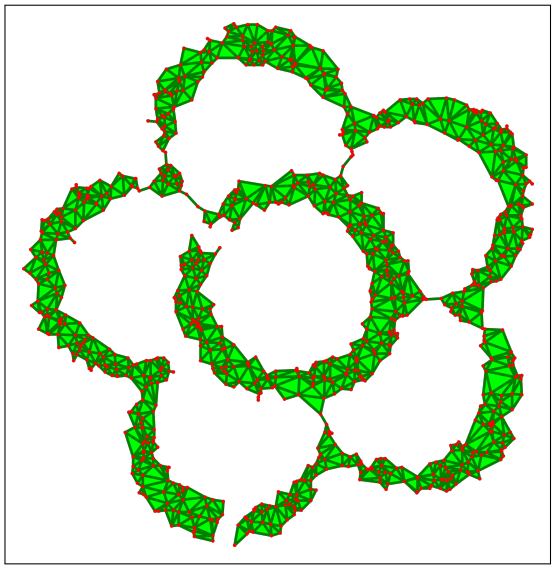
We apply the modified hole operations to the same holes and for the same subcomplexes as for the Alpha complexes and discuss the differences. Figure 6.9 shows some of the manipulated Wrap complexes. While the result for locking the most persistent 1-hole is the same, we get additional simplices when locking the most persistent 2-hole at $r = 0.0$. These simplices, which also include tetrahedra, are in the lower set of critical simplices in the canonical cycle. The reason why we only get differences between locking in an Alpha complex and locking in the corresponding Wrap complex in the case of a 2-hole in $\mathbb{R}^3$, is that in $\mathbb{R}^3$ there are intervals of size 4 with a tetrahedron as upper bound and an edge as lower bound, and as a result an interval of a tetrahedron can be in the lower set of a critical triangle. Similarly, we get additional simplices including tetrahedra when filling



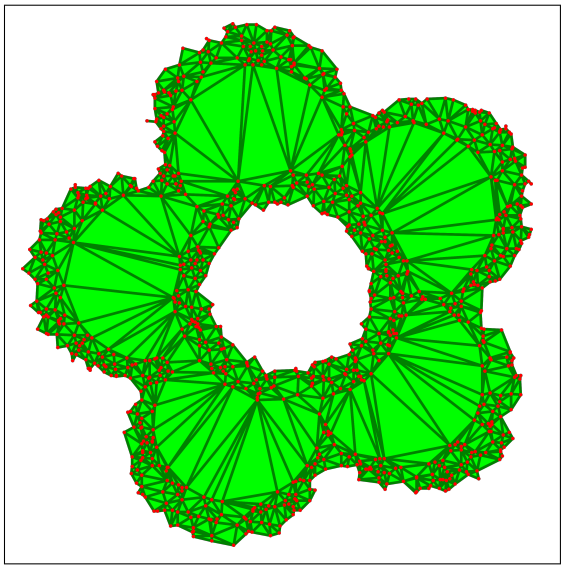
(a) Wrap complex for $r = 25.0$.



(b) Wrap complex for $r = \infty$.

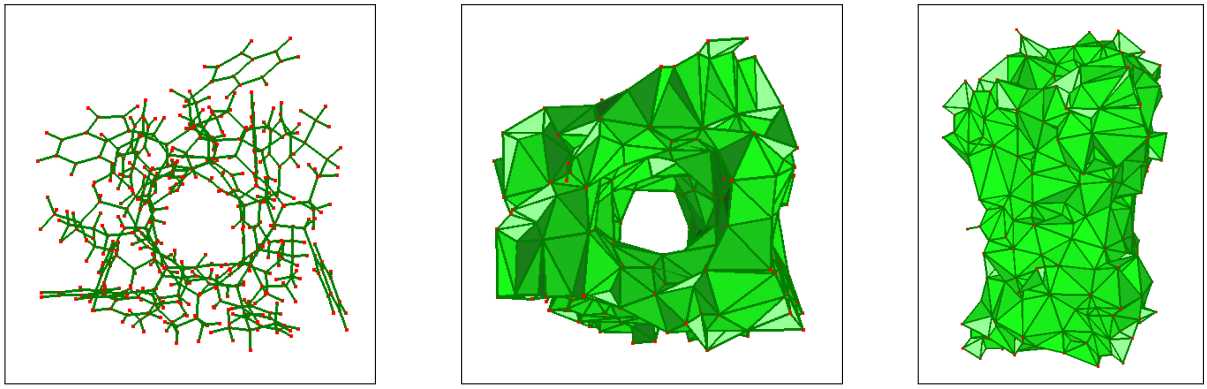


(c) $\texttt{Unlock}_1$ at $r = 25.0$.



(d) $\texttt{Unfill}_1$ at $r = \infty$.

Figure 6.7: We manipulate holes in the Wrap complexes corresponding to the Alpha complexes of Figure 6.3 by changing the set of critical simplices based on the addition and removal of canonical basis vectors. We (c) unlock the 1-hole of highest persistence at $r = 25.0$ and (d) unfill the same hole at $r = \infty$.



(a) Wrap complex for $r = 0.93$. (b) Wrap complex for $r = 2.35$. (c) Side view.

Figure 6.8: Wrap complexes of Gramicidin A, corresponding to the Alpha complexes of Figure 6.5. Triangles with no incident tetrahedra are drawn in pale green, while the others are drawn in darker green, with the shade depending on their normal vector.

the most persistent 1-hole in the Wrap complex for $r = 0.93$.

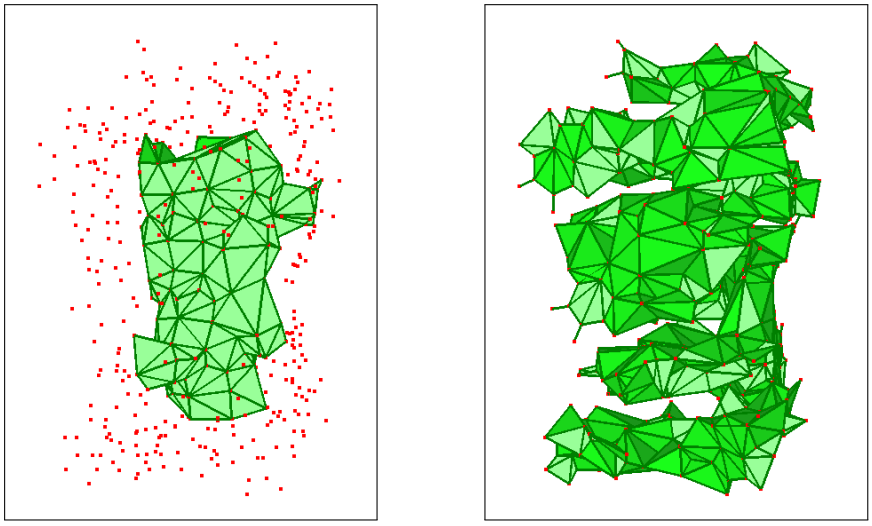
For unlocking and unfilling we get similar results as for the Alpha complexes, except that more simplices get removed because of additional collapses of non-singular intervals. Unlocking the 1-hole at $r = 2.35$ again reveals the spiraling backbone of the protein, and unfilling the 1-hole at $r = 3.10$ carves the tunnel open but the view is still obstructed by some critical simplices that are not removed.

6.3 Dependence Structure

Although the operations formulated in Section 6.2 manipulate the targeted holes as desired, they have one shortcoming: there are side effects on other holes which are hard to control or predict.

Holes of other or even the same dimension might be affected by an operation. There is no easy way to predict the change in the Betti numbers of the complex. As an example, consider unlocking the central 1-hole in the 2-dimensional Alpha complex for $r = 25.0$ in Figure 6.4 (c). Removing its canonical cochain not only destroys the cycle around the central 1-hole, but also disconnects two other cycles. The first Betti number is reduced by 3.

In order to understand such side effects better, we study the dependences between holes in a filtered complex. Afterwards, we will propose an alternative formulation of the hole



(a) $\texttt{Lock}_2$ at $r = 0.0$.

(b) $\texttt{Unlock}_1$ at $r = 2.35$.

(c) $\texttt{Fill}_1$ at $r = 0.93$.

(d) $\texttt{Unfill}_1$ at $r = 3.10$.

Figure 6.9: We manipulate holes in the Wrap complexes of Gramicidin A by changing the set of critical simplices based on the addition and removal of canonical basis vectors. We (a) lock the most persistent 2-hole at $r = 0.0$. All the other operations target the most persistent 1-hole. We (b) unlock it at $r = 2.35$, (c) fill it at $r = 0.93$, and (d) unfill it at $r = 3.10$.

manipulating operations, which respects the dependence structure of holes.

Matrix Reduction and Dependences. Let K be a simplicial complex with a monotonic ordering of the simplices $\{\sigma_1, \cdots, \sigma_m\}$. In Section 5.2, we explained how to use matrix reduction algorithms for computing persistent homology. Algorithms 6 and 7 both use left-to-right column additions to reduce the boundary matrix of K. The resulting matrix R contains the same information about persistent homology in both cases, but in general depends on the algorithm that was used for reduction.

We consider a fixed reduced matrix R and the corresponding book-keeping matrix C,

which contains a 1 at position (i, j) if column i was added to column j, and use them as a starting point to construct five maps that capture different types of dependences between homology classes. The goal is to distinguish the ordered pairs of simplices in the monotonic ordering that are necessary to keep R reduced from the others, which can be swapped without affecting R. Writing (b_i, d_i) for the simplex indices of the birth-death pair to which σ_i belongs, we introduce maps $\mathsf{X}\colon \{1, \dots, m\} \to 2^{\{1,\dots,m\}}$ with $\mathsf{X} \in \{\delta, \mathsf{BD}, \mathsf{DB}, \mathsf{BB}, \mathsf{DD}\}$:

$$\begin{aligned}
\delta(i) &:= \{\, j \mid \sigma_i \subseteq \sigma_j \text{ and } \dim \sigma_j = \dim \sigma_i + 1\},\\
\mathsf{BD}(i) &:= \{\, j \mid \mathrm{low}(j) = i\},\\
\mathsf{DB}(i) &:= \{\, j \mid i = d_i < j = b_j \text{ and } (C[i, j] = 1 \text{ or } R[i, d_j] = 1)\},\\
\mathsf{BB}(i) &:= \{\, j \mid i = b_i < j = b_j \text{ and } R[i, d_j] = 1\},\\
\mathsf{DD}(i) &:= \{\, j \mid i = d_i < j = d_j \text{ and } C[i, j] = 1\}.
\end{aligned}$$

Each maps the index of a simplex to the indices of a subset of simplices that appear later in the monotonic ordering, with the property that these ordered pairs should be maintained. The coboundary map δ maps a simplex to its cofacets. BD maps a birth simplex of a non-essential homology class to the death simplex of this class. The other three maps encode dependences between simplices of the same dimension that result from the reduction process. They record each column addition ($C[i, j] = 1$) and each non-zero entry above the lowest one in each non-zero column ($R[i, d_j] = 1$). We get three maps by differentiating between birth and death simplices in the domain and the codomain. In the reverse direction, we define $\mathsf{X}^T(j) := \{\, i \mid j \in \mathsf{X}(i)\}$ for each X. For example, the inverse of δ, $\delta^T = \partial$, maps a simplex to its facets.

Partial Order. We summarize all five maps in a partial order $\mathcal{P}$, which we refer to as the *dependence structure* of the holes in the filtered complex, or more precisely of the reduced matrix R. Specifically, $\mathcal{P}$ is the transitive closure of the collection of pairs $i < j$ such that $j \in \mathsf{X}(i)$ for at least one $\mathsf{X} \in \{\delta, \mathsf{BD}, \mathsf{DB}, \mathsf{BB}, \mathsf{DD}\}$.

We write $\iota_0 : 1 < \cdots < m$ for the original order of simplex indices. For a permutation ι of ι_0, let $R(\iota)$ be the matrix R after reordering the columns and rows according to ι. We prove, among other things, that $R(\iota)$ is left-to-right reduced for all linear extensions ι of $\mathcal{P}$.

Theorem 6.1 (Dependence Structure)**.** *Let K be a simplicial complex with a fixed monotonic ordering and let R be a left-to-right reduced version of its boundary matrix. The corresponding dependence structure $\mathcal{P} \subseteq \{1, \ldots, m\}^2$ is the smallest partial order on $\{1, \ldots, m\}$ with linear extension $\iota_0 : 1 < \cdots < m$ such that every linear extension ι of $\mathcal{P}$ satisfies the following:*

- *ι corresponds to a monotonic ordering of the cells,*
- *$R(\iota)$ is left-to-right reduced, and*
- *the persistence pairs defined by $R(\iota)$ are the same as those of $R(\iota_0)$.*

Proof. First, we prove that the dependence structure $\mathcal{P}$ is sufficient to satisfy the three claimed properties. Given two orderings ι_0 and ι, an *inversion* is a pair of indices (i, j) that are ordered differently in ι_0 and in ι. There is a finite sequence of *elementary transpositions*, which swap two adjacent items, that transforms ι_0 to ι without introducing new inversions during the process. None of these transpositions violate the monotonicity of the ordering, since $\sigma_i \subseteq \sigma_j$ implies that (i, j) is not an inversion. An elementary transposition of an inversion (i, j) in the process translates to swapping the two corresponding columns and the two corresponding rows in the boundary matrix. In the following, we argue that swapping these columns and rows in the reduced version of the boundary matrix preserves the claimed properties. We distinguish between four cases depending on whether σ_i and σ_j give birth or death, they are illustrated in Figure 6.10.

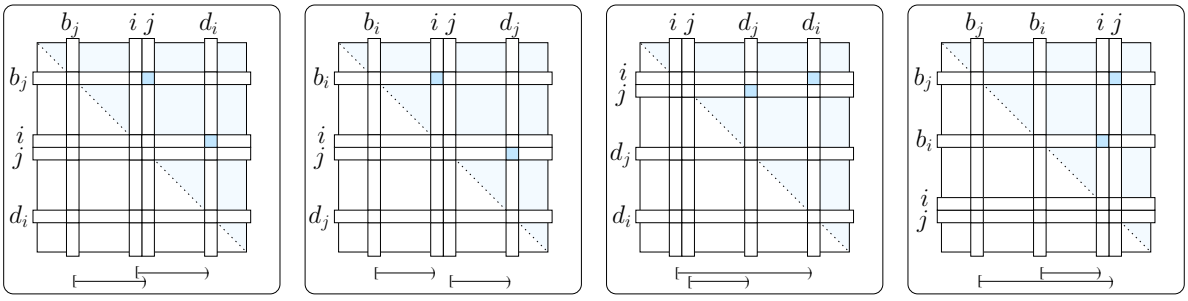


Figure 6.10: Swapping the adjacent columns i and j and the corresponding adjacent rows in R. The panels show the four different cases (birth-death, death-birth, birth-birth, death-death) with the lowest ones of the affected columns highlighted.

Case BD: σ_i gives birth and σ_j gives death. Since both orderings are extensions of $\mathcal{P}$, we have $j \notin \mathsf{BD}(i)$, which implies that $\mathrm{low}(j) \neq i$. The matrix remains left-to-right reduced

and the persistence pairs stay the same after swapping the columns because column i is zero, and after swapping the rows because row j does not contain any lowest one.

Case DB: σ_i gives death and σ_j gives birth. As $j \notin \mathsf{DB}(i)$ implies $C[i,j] = 0$, column i had no role in reducing column j to zero, so we can swap the columns without changing R. To also swap the corresponding rows, we need that the entry in row i and column d_j is 0. This is guaranteed because $j \notin \mathsf{DB}(i)$ also implies $R[i,d_j] = 0$.

Case BB: both σ_i and σ_j give birth. We can easily swap the columns because both are zero. For the rows, we need that the entry in row i and column d_j is 0, which is guaranteed by $j \notin \mathsf{BB}(i)$. Only the case $d_j < d_i$ is illustrated in the third panel of Figure 6.10, but the reasoning for the case $d_i < d_j$ is the same.

Case DD: both σ_i and σ_j give death. Since $j \notin \mathsf{DD}(i)$ implies that column i was not added to column j during reduction, we can swap the two columns while maintaining a left-to-right reduced matrix and without changing any persistence pairs. This holds both in the case $b_j < b_i$, which is illustrated in the fourth panel of Figure 6.10, and in the case $b_i < b_j$, which is not shown. Swapping the rows causes no complications because they do not contain the lowest ones of any columns.

We now prove that $\mathcal{P}$ is necessary to satisfy the three claimed properties, by which we mean that every properly contained partial order has linear extensions that violate at least one of the properties. Let $\mathcal{P}_1 \subseteq \mathcal{P}$ be properly contained, then there exists a pair $(i,j) \in \mathcal{P} \setminus \mathcal{P}_1$ that is not derived by transitivity from other pairs in $\mathcal{P}$. Hence, $j \in \mathsf{X}(i)$ for at least one $\mathsf{X} \in \{\delta, \mathsf{BD}, \mathsf{DB}, \mathsf{BB}, \mathsf{DD}\}$. Let ι_1 be a linear extension of $\mathcal{P}_1$ with $j < i$. Then, every sequence of elementary transpositions that changes ι_0 to ι_1 contains one that swaps i with j. We can assume that before the transposition of i with j all claimed properties are satisfied. After the transposition, at least one of the properties is violated: If $j \in \delta(i)$ then the ordering is no longer monotonic, if $j \in \mathsf{BD}(i)$ then the pairing changes, if $C[i,j] = i$ then the matrix reduction requires a right-to-left column addition, and if $R[i,d_j] = 1$, then the persistence pairs change. Thus, the linear extensions of $\mathcal{P}_1$ violate at least one of the three required properties. □

6.4 Second Approach: Recursive Operations

We propose a formulation of the hole manipulating operations alternative to adding and removing canonical basis vectors, which respects the dependence structure of holes. All the complexes computed by the operations are in the filtrations defined by the linear extensions of the partial order introduced above. Thus, the persistence pairing is not changed by any of the operations. We can manipulate the hole system without side effects, but with less flexibility.

Operations. We formulate the operations in a recursive way. The manipulation of one hole entails the manipulation of other holes, which are dependent. In order to change the status of the targeted holes (and the corresponding points in the persistence diagram) in the desired way, the operations add or remove the corresponding birth or death simplices to or from the complex. Given a point B in the persistence diagram, we write b_B and d_B for the indices of the birth and the death simplex (if the latter exists). We write x_B if it is not clear which of the two it is. Let B correspond to a p-dimensional hole and let $M = K_\ell$ be the current subcomplex. We define the lock and fill operations as follows.

1 $\texttt{Lock}_p(b_B)$: **if** B in future **then**
2 **for** $x_A \in \partial(b_B)$ **do if** $x_A = b_A$ **then** $M = \texttt{Lock}_{p-1}(b_A)$
3 **elseif** $x_A = d_A$ **then** $M = \texttt{Fill}_{p-2}(d_A)$;
4 **for** $d_A \in \mathsf{DB}^T(b_B)$ **do** $M = \texttt{Fill}_{p-1}(d_A)$;
5 **for** $b_A \in \mathsf{BB}^T(b_B)$ **do** $M = \texttt{Lock}_p(b_A)$;
6 **return** $M \cup \{\sigma_{b_B}\}$ with B in presence.

1 $\texttt{Fill}_p(d_B)$: **if** B not in past **then**
2 **for** $x_A \in \partial(d_B)$ **do if** $x_A = b_A$ **then** $M = \texttt{Lock}_p(b_A)$
3 **elseif** $x_A = d_A$ **then** $M = \texttt{Fill}_{p-1}(d_A)$;
4 **for** $d_A \in \mathsf{DD}^T(d_B)$ **do** $M = \texttt{Fill}_p(d_A)$;
5 **if** B in future **then** $M = \texttt{Lock}_p(b_B)$;
6 **return** $M \cup \{\sigma_{d_B}\}$ with B in past.

The lock operation moves B from the future to the presence by adding the birth simplex σ_{b_B} (line 6). Recursively, the operation adds all dependent simplices that may not appear

after this simplex in the ordering, as specified by the maps δ, DB^T, and BB^T. This includes the faces of the simplex (lines 2 and 3), and the canonical cycle $\mathsf{z}_p(\sigma_{b_B})$ stored in column b_B of U, but also other dependent cells (lines 4 and 5). Similarly, the fill operation moves B from the presence to the past by adding the death simplex σ_{d_B} (line 6) and by recursively adding other simplices as necessary. In the event that B is not yet in the presence, it first moves B from the future to the presence. To prove that the two operations avoid infinite loops, we note that each recursive call decreases the parameter, which is the index of the birth or the death simplex.

The unlock and unfill operations delay the birth or the death of a hole by removing simplices. We formulate these operations for a point A in the persistence diagram corresponding to a p-dimensional hole with birth-death pair $(\sigma_{b_A}, \sigma_{d_A})$.

1 $\texttt{Unlock}_p(b_A)$: **if** A not in future **then**
2 **for** $x_B \in \delta(b_A)$ **do if** $x_B = b_B$ **then** $M = \texttt{Unlock}_{p+1}(b_B)$
3 **elseif** $x_B = d_B$ **then** $M = \texttt{Unfill}_p(d_B)$;
4 **for** $b_B \in \mathsf{BB}(b_A)$ **do** $M = \texttt{Unlock}_p(b_B)$;
5 **if** A in past **then** $M = \texttt{Unfill}_p(d_A)$;
6 **return** $M \setminus \{\sigma_{b_A}\}$ with A in future.

1 $\texttt{Unfill}_p(d_A)$: **if** A in past **then**
2 **for** $x_B \in \delta(d_A)$ **do if** $x_B = b_B$ **then** $M = \texttt{Unlock}_{p+2}(b_B)$
3 **elseif** $x_B = d_B$ **then** $M = \texttt{Unfill}_{p+1}(d_B)$;
4 **for** $b_B \in \mathsf{DB}(d_A)$ **do** $M = \texttt{Unlock}_{p+1}(b_B)$;
5 **for** $d_B \in \mathsf{DD}(d_A)$ **do** $M = \texttt{Unfill}_p(d_B)$;
6 **return** $M \setminus \{\sigma_{d_A}\}$ with A in presence.

The unlock and the unfill operations are symmetric to the first two. They avoid infinite loops, since each recursive call increases the parameter.

Consistency of Status. Observe that a complex K_ℓ in the filtration of K contains exactly those simplices that give birth and death to holes in the past plus those that give birth to holes in the presence. This relation is maintained by the four operations, i.e., the adapted subcomplex M consists of all simplices of K that are the birth or the death simplex of a persistence point in the past or the birth simplex of a point in the presence.

Furthermore, the operations maintain what we call the consistency of a status, which we prove in the following. The status of points in the persistence diagram together define the status of a subcomplex. We call a status *consistent* with the dependence structure $\mathcal{P}$ iff the following conditions are satisfied for any two points A and B in the persistence diagram:

$$(b_A, d_B) \in \mathcal{P} \Longrightarrow [A \text{ is in the future } \Rightarrow B \text{ is not in the past}]; \tag{6.1}$$

$$(d_A, b_B) \in \mathcal{P} \Longrightarrow [A \text{ is not in the past } \Rightarrow B \text{ is in the future}]; \tag{6.2}$$

$$(b_A, b_B) \in \mathcal{P} \Longrightarrow [A \text{ is in the future } \Rightarrow B \text{ is in the future}]; \tag{6.3}$$

$$(d_A, d_B) \in \mathcal{P} \Longrightarrow [A \text{ is not in the past } \Rightarrow B \text{ is not in the past}]. \tag{6.4}$$

The status defined for subcomplex K_ℓ is indeed consistent with $\mathcal{P}$. This is preserved by the hole manipulating operations.

Lemma 6.2 (Consistency of Status)**.** *Let K be a filtered simplicial complex, K_ℓ a complex in the filtration, and $\mathcal{P}$ the corresponding dependence structure. If the subcomplex M is obtained by executing a finite sequence of hole manipulating operations starting with K_ℓ, then the status of M is consistent with $\mathcal{P}$.*

Proof. We prove the claim by induction. Assuming the conditions (6.1) to (6.4) for consistency are satisfied before an operation, we show that they are also satisfied after the operation. There are four operations to consider, but because the arguments are almost the same in all cases, we focus on locking.

When we lock the hole corresponding to point B, its status changes from the future to the presence. We need to check if the conditions (6.2) and (6.3) are still satisfied. For the first one, points A with $(d_A, b_B) \in \mathcal{P}$ have to move to the past. For the second one, points A with $(b_A, b_B) \in \mathcal{P}$ have to move out of the future. The operation does exactly that: lines 2 and 3 adjust the faces, line 4 adjusts the points with a relation captured by (6.2), and line 5 adjusts the points with a relation captured by (6.3). Together, lines 2 to 5 capture all points with dependences expressed in $\mathcal{P}$. □

On the other hand, the consistency of the status of a subcomplex M implies the existence of a linear extension of the partial order such that M belongs to the filtration of this monotonic ordering, as we show in the following.

Lemma 6.3 (Existence of Linear Extension)**.** *Let K be a filtered simplicial complex and $\mathcal{P}$ the corresponding dependence structure. For every subcomplex M of K whose status is consistent with $\mathcal{P}$, there exists a linear extension ι of $\mathcal{P}$ and an index k such that M is the k-th complex in the filtration defined by ι.*

Proof. Recall that $M \subseteq K$ contains exactly the birth and death simplices of points in the past and the birth simplices of points in the presence. Hence $N = K \setminus M$ contains the birth and death simplices of points in the future and the death simplices of points in the presence. We claim that there is a linear extension $\iota = \iota_M \iota_N$ of $\mathcal{P}$ such that $i \in \iota_M$ iff $\sigma_i \in M$ and $j \in \iota_N$ iff $\sigma_j \in N$. To prove this, it is sufficient to show that all pairs in $\mathcal{P}$ that go between M and N go in fact from M to N. To get a contradiction, assume there is a pair $(j, i) \in \mathcal{P}$ with $j = x_A \in \iota_N$ and $i = x_B \in \iota_M$.

Case $j = b_A$, $i = d_B$: Since every simplex in N is the birth or death simplex of a point in the future or the death simplex of a point in the presence, A must be in the future. Similarly, since every simplex in M is the birth or death simplex of a point in the past or the birth simplex of a point in the presence, B must be in the past. But this contradicts condition (6.1) for the consistency of a status.

Case $j = d_A$, $i = b_B$: A cannot be in the past and B cannot be in the future, which contradicts (6.2).

Case $j = b_A$, $i = b_B$: A must be in the future and B cannot be in the future, which contradicts (6.3).

Case $j = d_A$, $i = d_B$: A cannot be in the past and B must be in the past, which contradicts (6.4).

We conclude that $\iota = \iota_M \iota_N$ is a linear extension of $\mathcal{P}$. Setting $k = |M|$, M is the k-th complex in the corresponding filtration. □

By Lemmas 6.2 and 6.3, every subcomplex that can be constructed by a sequence of hole manipulating operations belongs to the filtration of a linear extension of $\mathcal{P}$. Conversely, given a complex M in the filtration of a linear extension of $\mathcal{P}$, it is possible to design a sequence of operations that constructs M from a complex in the original filtration of K. Hence, the dependence structure describes precisely what can and what cannot be constructed with the hole manipulating operations defined in this section.

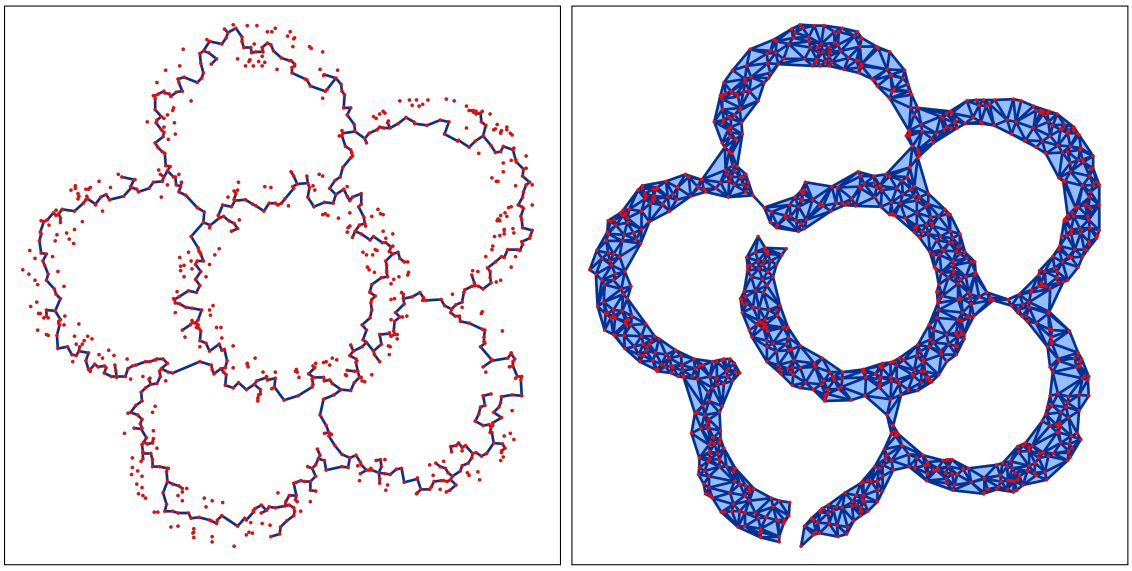
Example for the Alpha Complex in $\mathbb{R}^2$. We want to compare the results of the hole manipulating operations defined in this section with those of the alternative operations of Section 6.2. First, we consider manipulations of the Alpha complexes of the 2-dimensional point set of Figure 6.3.

Locking the six most persistent 1-holes in the Alpha complex for $r = 0.0$ again gives a 1-dimensional complex with six loops. The recursive operation defined in this section, however, adds extra edges attached to the canonical cycles, compare panel (a) of Figure 6.4 and panel (a) of Figure 6.11. These are the death edges of dependent 0-holes, and they are forced into the subcomplex to preserve the dependence structure. For the other three operations, we get the same results as in Figure 6.4. In these examples, the simplices corresponding to dependent holes are exactly the ones of the canonical basis vector added or removed by the operation. In contrast to the operations of Section 6.2, the recursive operations however guarantee that the dependence structure is preserved.

We have a closer lock at the unlock operation applied to the 1-hole of highest persistence in the Alpha complex for $r = 25.0$, see panel (b) of Figure 6.11. The corresponding persistence diagram before and after the operation is shown in the second row of the figure. In panel (d) we highlight the point corresponding to the targeted 1-hole in black and show the dependent points that are relevant for unlocking. To preserve the consistency of the status, the operation moves all of these points into the future, unless they are already there. This includes two other 1-holes that were in the presence and are thus recursively unlocked.

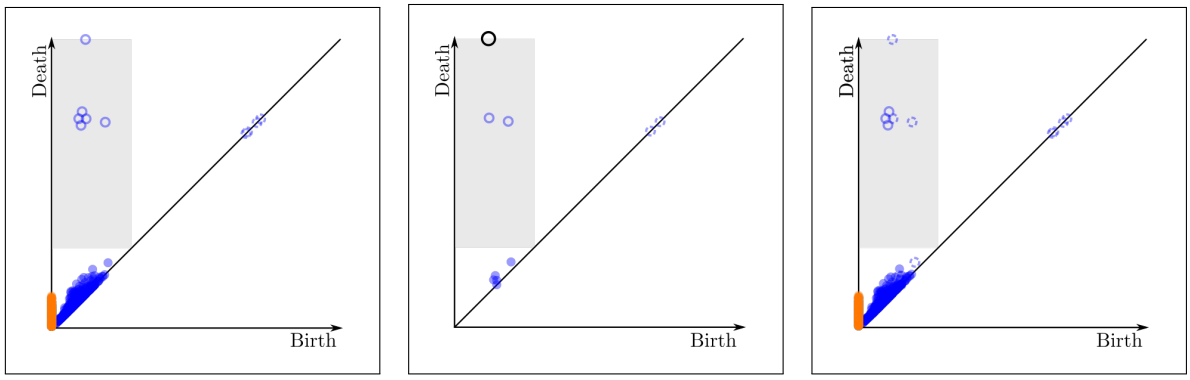
Example for the Alpha Complex in $\mathbb{R}^3$. We now consider the 3-dimensional example of Figure 6.5, which represents a model for the protein Gramicidin A, and apply the same operations as in Figure 6.6. The results are displayed in Figures 6.12 and 6.13.

In addition to views of the resulting complexes, they show for each operation the points in the persistence diagram corresponding to dependent holes. The status of the points in the unmodified subcomplex can be inferred from the highlighted quadrant: points inside, below, or to the right of the quadrant are in the presence, past, or future, respectively. The status of the points after the operation is encoded in their drawing style: unfilled, filled, or dashed circles for points in the presence, past, or future. Finally, the colors of the points give their dimensions: orange, blue, or green for dimensions 0, 1, or 2, except



(a) $\texttt{Lock}_1$ at $r = 0.0$.

(b) $\texttt{Unlock}_1$ at $r = 25.0$.



(c) Persistence diagram.

(d) Dependences for $\texttt{Unlock}_1$.

(e) Result for $\texttt{Unlock}_1$.

Figure 6.11: We manipulate holes in the Alpha complexes of Figure 6.3 with the recursive operations respecting the dependences between holes. We (a) lock the six most persistent 1-holes at $r = 0.0$ and (b) unlock the 1-hole of highest persistence at $r = 25.0$. The lower panels show the changes to the status of the points in the persistence diagram for unlocking. Points in the past, presence, and future are drawn with a filled, empty, and dashed circle, respectively. We show (c) the original persistence diagram for $r = 25.0$, (d) the relevant dependences for unlocking the hole corresponding to the point highlighted in black, and the (e) resulting persistence diagram.

for the targeted point, which is highlighted in black.

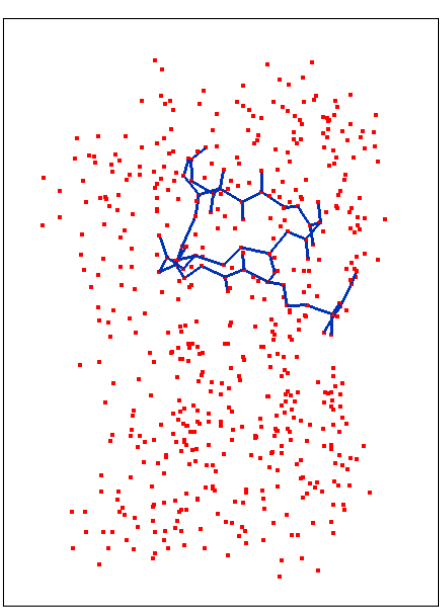
As in the 2-dimensional example, the recursive lock operation adds more simplices than are contained in the canonical cycle of the targeted hole, see the first row of Figure 6.12. The targeted hole moves from the future to the presence, while the dependent holes move to the past. Filling the most persistent 1-hole, moves this and all dependent holes to the past, see the second row of Figure 6.12. Comparing to Figure 6.6, more simplices are added than are contained in the canonical chain.

The results for unlocking and unfilling, which are shown in Figure 6.13, are quite different to those obtained with the alternative hole manipulating operations of Figure 6.6. The major difference is that both for unlocking and unfilling several dependent holes are moved to the presence. The advantage of the new formulation of the operations is that these unintended side effects can be predicted precisely. If the dependence of another hole is due to its birth simplex, this hole will be unlocked and thus moves to the future. If the dependence is due to the death simplex, the hole will be unfilled and moves to the presence. The reason why we get these side effects for unlocking and unfilling but not for the other two operations, is that the dependence structure is defined with the column reduced matrices for persistent homology. It encodes the canonical bases for homology but not those for relative cohomology. Thus, the recursive unlock and unfill operations do not remove canonical cocycles and cochains but approximate versions of them. One way to approach this problem would be to formulate another dependence structure for relative cohomology and use it for recursive unlocking and unfilling. We do however not pursue this idea further in this work.

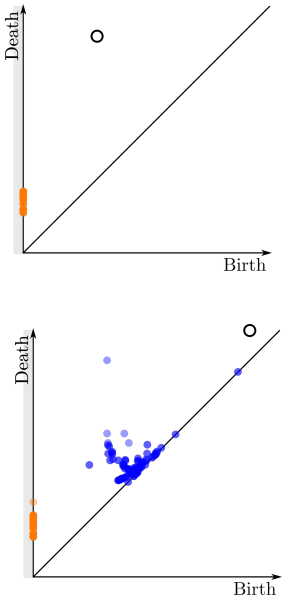
6.5 Statistics

In this section, we present statistics about the sizes of the structures relevant for the hole manipulating operations. We have implemented the matrix reduction algorithms of Section 5.2 and both approaches to hole manipulation, which are formulated in Sections 6.2 and 6.4, for point sets in $\mathbb{R}^2$ and $\mathbb{R}^3$. Our implementation is in C++.

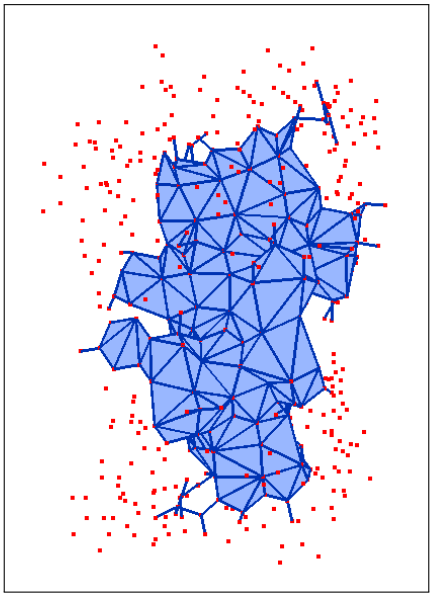
We generate random point sets in $\mathbb{R}^3$ according to the Poisson point process. We perform Experiments I, II, and III for an expected number of 125, 512, and 1000 points in $[0,1]^3$, respectively, averaging the results over 100 runs each.



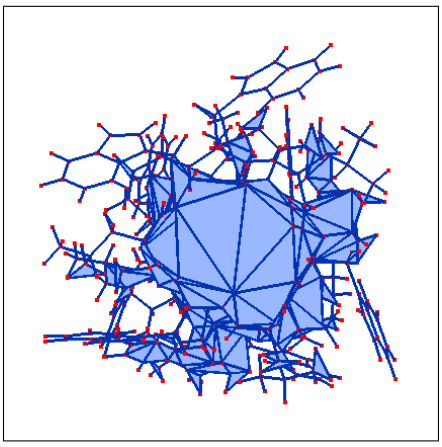
(a) $\texttt{Lock}_1$ at $r = 0.0$.



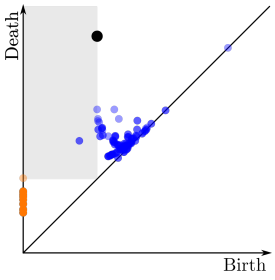
(b) Dependences.



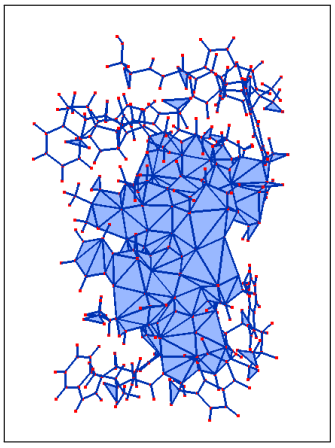
(c) $\texttt{Lock}_2$ at $r = 0.0$.



(d) $\texttt{Fill}_1$ at $r = 0.93$, top view.

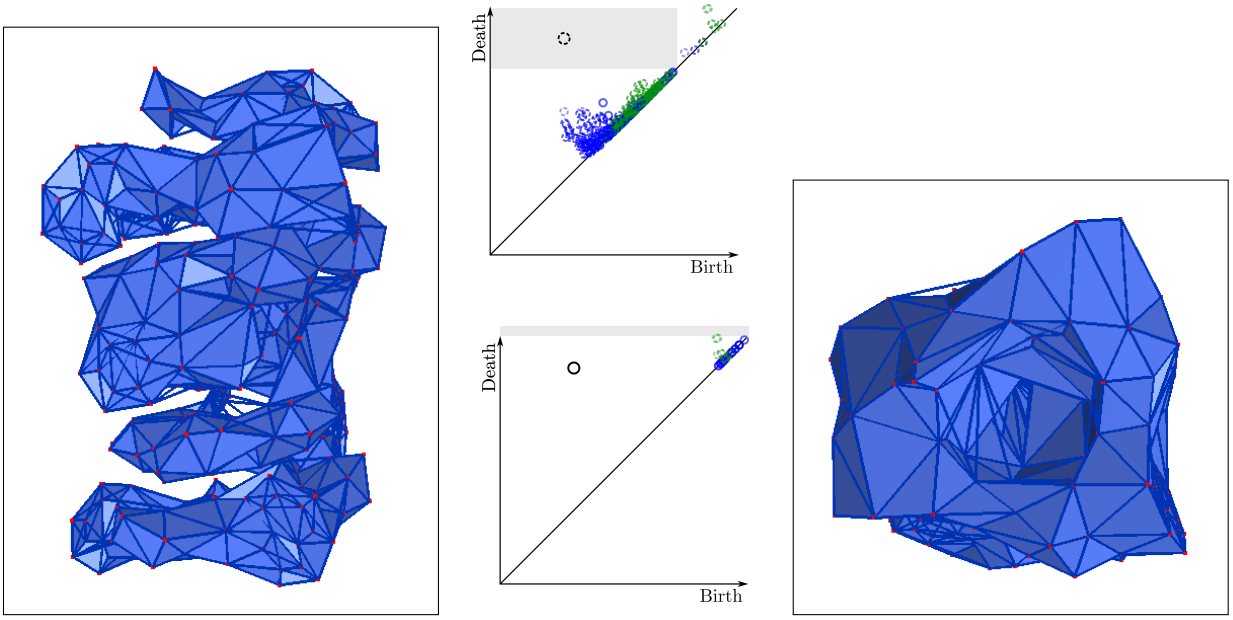


(e) Dependences.



(f) $\texttt{Fill}_1$ at $r = 0.93$.

Figure 6.12: We lock and fill holes in the Alpha complexes of Gramicidin A with the recursive operations respecting the dependences between holes. We lock the (a) 1- and (c) 2-hole of highest persistence at $r = 0.0$, and we (d),(f) fill the 1-hole of highest persistence at $r = 0.93$. The middle column shows the persistence points of the dependent holes for the operations in the corresponding rows.



(a) $\texttt{Unlock}_1$ at $r = 2.35$. (b) Dependences. (c) $\texttt{Unfill}_1$ at $r = 3.10$.

Figure 6.13: We unlock and unfill holes in the Alpha complexes of Gramicidin A with the recursive operations respecting the dependences between holes. We (a) unlock the 1-hole of highest persistence at $r = 2.35$ and (c) unfill the same hole at $r = 3.10$. Panel (b) shows the persistence points of the dependent holes for the two operations.

Numbers of Simplices. We begin with the size of the simplicial complex, which in our case is the Delaunay triangulation of the points, see Table 6.1. Since all the reduced Betti numbers are 0 for the Delaunay triangulation, the numbers of simplices determine the sizes of the components of the tri-partition discussed in Chapter 5.

	$p = -1$	0	1	2	3	total
Experiment I	1.0	124.1	817.7	1 357.6	663.1	2 963.5
Experiment II	1.0	510.8	3 699.5	6 323.5	3 133.9	13 668.7
Experiment III	1.0	998.4	7 397.6	12 730.7	6 330.5	27 458.3

Table 6.1: Average numbers of simplices in the Delaunay triangulation of a Poisson point process in $[0, 1]^3$.

Densities of Matrices. We compute the dependence structure defined in Section 6.3 with the exhaustive column reduction algorithm of Section 5.2. We compare the densities of the computed matrices to those obtained with the standard reduction algorithm in Table 6.2. The table also shows the same numbers for row reduction.

	Experiment I		Experiment II		Experiment III	
	std	exh	std	exh	std	exh
density of R	0.061	0.159	0.014	0.046	0.007	0.026
density of U	0.203	0.336	0.065	0.124	0.039	0.078
density of C	0.118	0.076	0.032	0.018	0.017	0.009
density of Q	0.093	0.135	0.022	0.036	0.011	0.019
density of V	0.203	0.308	0.065	0.107	0.039	0.065
density of D	0.142	0.044	0.043	0.010	0.024	0.005

Table 6.2: Comparison between the standard reduction algorithm and the exhaustive reduction algorithm for random points in $[0, 1]^3$. We quantify the *density* of a matrix as the percentage of non-zero elements. *Upper half:* Matrices computed by column reduction. *Lower half:* Matrices computed by row reduction.

The exhaustive column reduction algorithm produces denser matrices R and U, but performs fewer column additions. Comparing the density of C for the standard and the exhaustive algorithm, we see that the latter uses only about half the number of column additions. Perhaps this is because the extra time invested in further reducing early columns pays off later, when these columns are used to reduce later columns. The difference between standard and exhaustive reduction is even more pronounced when we work with rows rather than with columns.

The densities of the matrices have a direct influence on the sizes of the canonical basis vectors and the number of pairs that make up the dependence structure. Matrix U stores the canonical cycles and chains that are used for locking and filling, and C stores the same information in hierarchical form. Additional pairs in the dependence structure are due to off-diagonal entries in matrix R. The canonical cocycles and cochains that are used for unlocking and unfilling with the first approach are stored in V. The second approach to manipulating holes uses the information in the dependence structure. Since the dependence structure is built from the column and not the row reduced matrices, canonical cocycles and cochains are replaced by approximate versions of them, which are functionally equivalent.

Number of Dependences. The total number of dependences is roughly an order of magnitude larger than the number of simplices. Specifically, we have about 26 015,

145 409, and 317 663 dependences on average in Experiments I, II, and III. A comparison with Table 6.1 shows that this is about ten times the total number of simplices in the Delaunay triangulation. We provide detailed statistics in Table 6.3, which differentiates between types of dependences and dimensions. More precisely, for each type, dimension, and experiment, the table gives the average number of pairs of the given type that originate from a simplex of the given dimension. Some of these numbers do not depend on the sampled points, such as the dependences of type BD, for which there is exactly one per simplex, and the dependences of type ∂, as every p-simplex has exactly $p+1$ facets. Since pairs go from left to right and lower-dimensional simplices tend to precede higher-dimensional simplices in the filtration, it is not surprising that the average out-degree in the forward direction is higher for lower dimensions and in the backward direction is higher for higher dimensions. Note that there is exactly one vertex that gives death (it is paired with the empty simplex), and this vertex accumulates the largest number of dependences of type DB.

We observe that the numbers barely change between the experiments, which suggests that they are primarily local. The numbers we get for the Gramicidin A data (not shown) are very similar to those in Table 6.3.

Size of Operations. We consider the average numbers of simplices affected by the different hole manipulating operations, for both approaches. The first approach adds or removes canonical cycles, chains, cocycles, or cochains to or from the subcomplex. The upper half of Table 6.4 gives their average sizes. The second approach recursively adds or removes the dependent simplices of a birth or death simplex. Rather than just the pairs originating from the simplex to which the operation is applied, we need to consider the entire emanating paths in the dependence structure.

Consider for example the lock operation applied to σ_i. The recursive algorithm also adds the simplices of the canonical cycle of σ_i to the subcomplex, but rather than fetching these simplices directly from the matrix U, which stores the canonical cycle in column i, the algorithm finds the simplices by following paths in the dependence structure. Generally, these paths contain more simplices than just those in the canonical cycle. Filling works the same, but with canonical chains. The situation is not entirely symmetric for the other two operations. In the recursive versions of unlocking and unfilling, the removal of the

	Experiment I				Experiment II				Experiment III			
	0	1	2	3	0	1	2	3	0	1	2	3
δ birth	13.2	4.8	1.9	-	14.5	4.9	2.0	-	14.8	4.9	2.0	-
BD	1.0	1.0	1.0	-	1.0	1.0	1.0	-	1.0	1.0	1.0	-
BB	0.9	0.2	0.0	-	0.9	0.3	0.1	-	0.9	0.3	0.1	-
δ death	13.2	6.2	2.0	0.0	13.8	6.5	2.0	0.0	14.8	6.6	2.0	0.0
DB	123.1	55.4	10.2	0.0	509.8	87.8	13.7	0.0	997.4	104.8	15.3	0.0
DD	0.0	1.2	1.4	0.9	0.0	1.3	1.5	1.0	0.0	1.3	1.6	1.0
total	16.0	14.5	8.4	0.9	17.4	18.5	10.2	1.0	17.7	20.6	11.0	1.0
∂ birth	1.0	2.0	3.0	-	1.0	2.0	3.0	-	1.0	2.0	3.0	-
DB^T	1.0	9.8	10.7	-	1.0	14.0	14.0	-	1.0	16.3	15.5	-
BB^T	0.9	0.2	0.0	-	0.9	0.3	0.1	-	0.9	0.3	0.1	-
∂ death	1.0	2.0	3.0	4.0	1.0	2.0	3.0	4.0	1.0	2.0	3.0	4.0
BD^T	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
DD^T	0.0	1.2	1.4	0.9	0.0	1.3	1.5	1.0	0.0	1.3	1.6	1.0
total	2.8	10.8	9.5	5.9	2.9	14.6	11.2	6.0	2.9	16.7	12.0	6.0

Table 6.3: Average numbers of dependences of different types originating from a p-simplex. *Upper half:* To capture the pairs relevant for unlocking and unfilling, we count the dependences in forward direction, distinguishing the ones that originate from birth-giving and from death-giving simplices. *Lower half:* To capture the pairs relevant for locking and filling, we count the dependences in backward direction, distinguishing again the ones that originate from birth-giving and from death-giving simplices.

	Experiment I			Experiment II			Experiment III		
	$p=0$	1	2	$p=0$	1	2	$p=0$	1	2
p-cycle	2.0	10.9	12.1	2.0	17.6	17.3	2.0	21.9	20.1
$(p+1)$-chain	4.5	12.1	7.0	6.6	25.9	10.8	8.3	35.7	12.9
p-cochain	5.1	8.6	6.8	6.8	15.5	11.1	7.3	20.3	13.7
$(p+1)$-cocycle	56.6	11.6	1.0	105.1	17.1	1.0	135.2	19.9	1.0
recursive $\texttt{Lock}_p$	4.0	50.2	139.0	4.5	112.6	382.2	5.0	176.0	633.2
recursive $\texttt{Fill}_p$	15.1	88.1	157.1	24.5	234.2	414.6	34.4	382.8	677.2
recursive $\texttt{Unlock}_p$	707.7	78.9	7.1	2 093.6	222.0	11.0	3 570.2	362.7	13.0
recursive $\texttt{Unfill}_p$	686.7	77.1	6.0	2 065.8	218.5	9.8	3 536.4	357.8	11.9

Table 6.4: Average sizes of hole manipulating operations. *Upper half:* Average numbers of simplices of a canonical cycle, chain, cochain, cocycle. *Lower half:* Average numbers of dependences for recursive locking, filling, unlocking, unfilling. We disregard the status and count every dependent simplex. The order of rows in the two halves is parallel, comparing the two different approaches to hole manipulation.

canonical cocycle and cochain is replaced by functionally equivalent deletions of simplices. This asymmetry arises because the dependence structure is based on the column reduced matrices rather than the row reduced ones.

The lower half of Table 6.3 gives the average numbers of simplices along the paths in the dependence structure relevant for the different hole manipulating operations. We count all simplices that are possibly affected by an operation, neglecting the status information. Thus, the numbers give an upper bound on the average numbers of status changes per operation, and they are likely to be rather loose. We observe an anomaly in Experiment III, in which the average size of a canonical 2-cochain is 13.7, while the average number of dependent 2-simplices for unlocking is only 13.0. This happens because the recursive unlock operation is based on the column reduced matrices instead of the row reduced ones, where the canonical cochains are stored. In this example, the recursive unlock operation finds a faster way to unlock the 2-cycles than by removing the canonical 2-cochains.

Chapter 7

Bregman Geometry

A general assumption in the previous chapters is that our input data is in Euclidean space, i.e., the distance between two points is measured by the Euclidean metric. For many applications, especially those that work with high-dimensional data, other measures of dissimilarity are more suitable and achieve better results. One example are discrete probability distributions, which can represent images, text documents, or other kinds of data, and can be viewed as points in the standard simplex of appropriate dimension. It is common to use the Kullback-Leibler divergence, which has its foundations in information theory, as a dissimilarity measure between such probability distributions, even though it does not fulfill the properties of a metric. However, it fits into the more general framework of Bregman divergences, which we will study in this chapter.

Our goal is to generalize constructions from computational geometry and topology such as the Delaunay triangulation and the Wrap complex to spaces where distance is measured with a Bregman divergence. This work was initiated in [8] and [28], which already studied the Delaunay triangulation and its radius function in Bregman geometry. We will complement these results by explaining how the Bregman-Delaunay triangulation can be computed as a weighted Euclidean Delaunay triangulation and by also studying Bregman-Alpha and Bregman-Wrap complexes.

The results of this chapter can also be found in [25], which we will submit for publication.

7.1 Bregman Divergence

First, we provide some background on Bregman divergences. We follow [3] in the details of the definition.

Bregman Divergence. Let $\Omega \subseteq \mathbb{R}^d$ be a non-empty open convex set. A function $F\colon \Omega \to \mathbb{R}$ is of *Legendre type* iff it is differentiable, strictly convex, and the length of the gradient ∇F goes to infinity whenever we approach the boundary of Ω. The third condition guarantees a conjugate function of the same kind (see below), and it is void if the boundary of Ω is empty, which is the case for $\Omega = \mathbb{R}^d$.

For points $x, y \in \Omega$, the *Bregman divergence* from x to y associated with F is the difference between F and the best affine approximation of F at y, both evaluated at x:

$$D_F(x\|y) := F(x) - [F(y) + \langle \nabla F(y), x - y\rangle], \tag{7.1}$$

where $\nabla F = (\frac{\partial F}{\partial x_1}, \ldots, \frac{\partial F}{\partial x_d})^T$ denotes the gradient operator and $\langle x, y\rangle = \sum_{i=1}^{d} x_i y_i$ the scalar product. The definition is illustrated in Figure 7.1. We draw the tangent hyperplane of F at y, the Bregman divergence from x to y is the vertical distance between the hyperplane and the function graph at x.

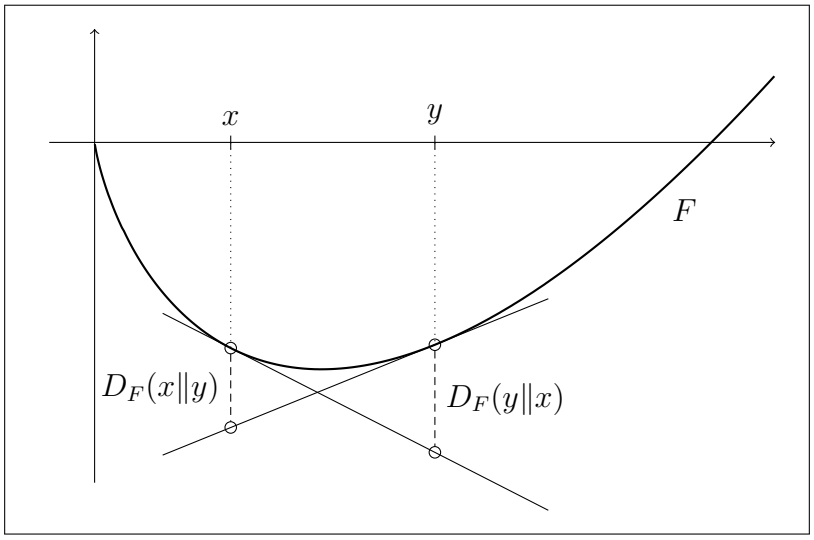


Figure 7.1: Geometric interpretation of the Bregman divergence from x to y and from y to x associated with F. In this example, F is the Shannon entropy in $\mathbb{R}^1$.

Note that the Bregman divergence is not necessarily symmetric: $D_F(x\|y) \neq D_F(y\|x)$ in general. Also, it violates the triangle inequality. Only the first axiom of a metric is

satisfied by the Bregman divergence: $D_F(x\|y) \geq 0$ with equality iff $x = y$. Nonetheless, it is a useful measure for dissimilarity.

Bregman Balls. Due to asymmetry, we can define two balls of size $h \geq 0$ centered at a point $x \in \Omega$. The *primal ball* $B_F(x, h)$ contains all points y for which the divergence from x to y is at most h, and the *dual ball* $B_F^*(x, h)$ contains all y for which the divergence from y to x is at most h:

$$B_F(x, h) := \{y \in \Omega \mid D_F(x\|y) \leq h\}, \tag{7.2}$$

$$B_F^*(x, h) := \{y \in \Omega \mid D_F(y\|x) \leq h\}. \tag{7.3}$$

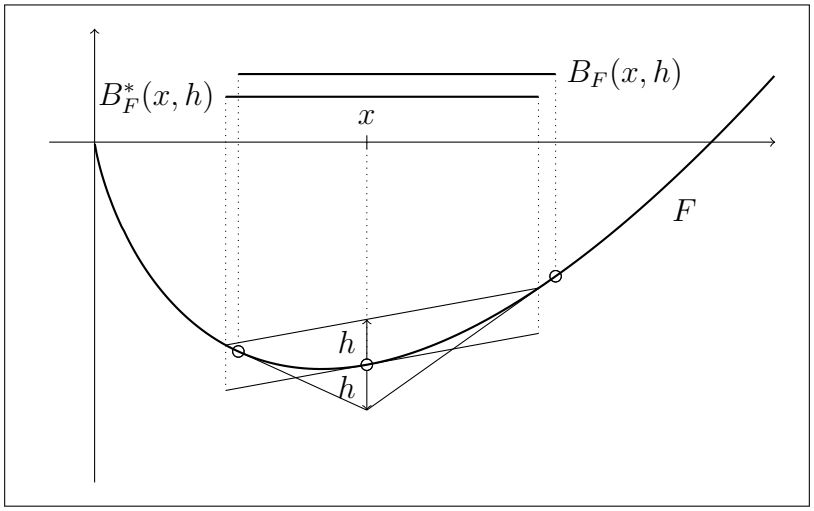


Figure 7.2: Geometric interpretation of the primal and the dual ball in Bregman geometry. In this example, F is the Shannon entropy in $\mathbb{R}^1$.

Figure 7.2 illustrates a geometric construction for the balls in Bregman geometry. For the primal ball, we cast light onto the graph of F from the point $(x, F(x) - h) \in \mathbb{R}^d \times \mathbb{R}$, and project the illuminated part of the graph onto $\mathbb{R}^d$. The primal ball is not necessarily convex. For the dual ball, we vertically shift the tangent hyperplane at x upward by h, and project the part of the graph on or below this shifted hyperplane onto $\mathbb{R}^d$. The dual ball is always convex [28].

Conjugate Function. We use polarity to relate F to another function F^*, which we call the conjugate function. The *polarity transform* maps a point $C = (c, \gamma) \in \mathbb{R}^d \times \mathbb{R}$ to the affine map $C^*\colon \mathbb{R}^d \to \mathbb{R}$ defined by $C^*(x) = \langle c, x\rangle - \gamma$, and it maps C^* back to

$(C^*)^* = C$. The *conjugate function*, $F^*\colon \Omega^* \to \mathbb{R}$, is defined such that polarity maps the points of the graph of F to the tangent affine maps of the graph of F^*, and it maps the points of the graph of F^* to the tangent affine maps of the graph of F, see Figure 7.3.

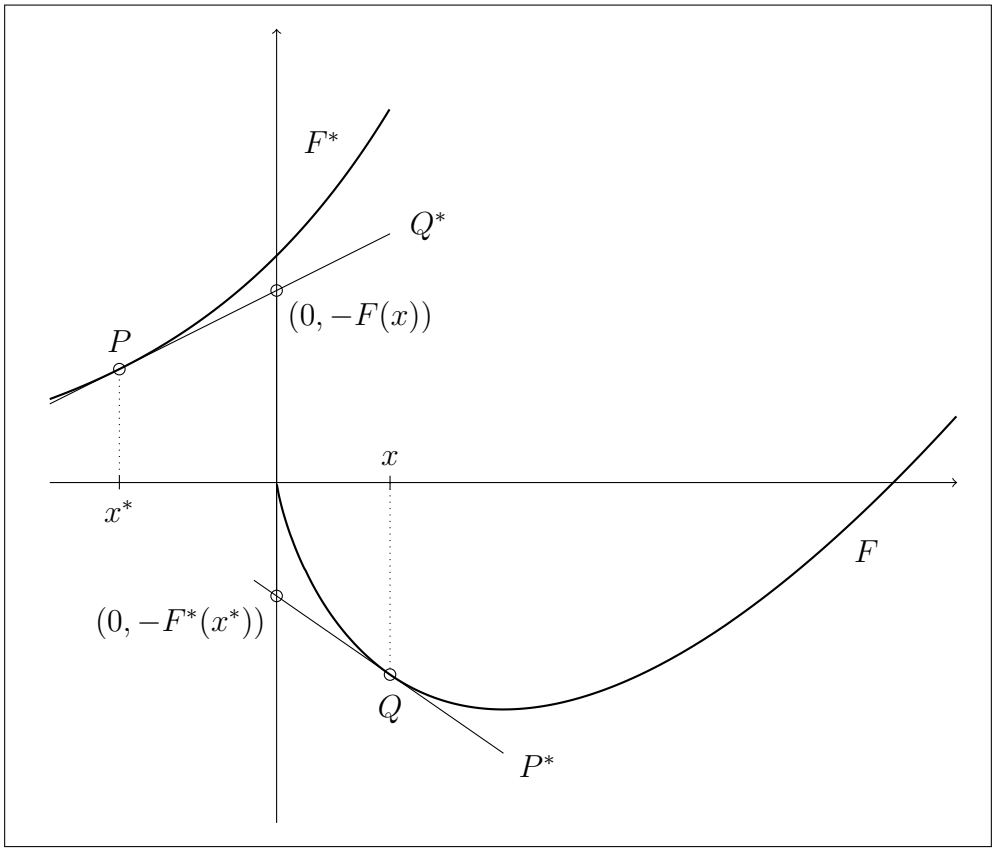


Figure 7.3: Construction of the conjugate function F^* with the polarity transform. It maps $P = (x^*, F^*(x^*))$ to the tangent of F at x, and it maps $Q = (x, F(x))$ to the tangent of F^* at x^*. In this example, F is the Shannon entropy in $\mathbb{R}^1$.

Specifically, polarity maps $P = (x^*, F^*(x^*))$ to $P^*(a) = \langle x^*, a\rangle - F^*(x^*)$, which should be the tangent affine map of F at the corresponding point x. Evaluating at x gives

$$\langle x^*, x\rangle - F^*(x^*) = F(x) \quad \Leftrightarrow \quad \langle x^*, x\rangle - F(x) = F^*(x^*), \tag{7.4}$$

and since it should be tangent we get

$$x^* = \nabla F(x), \tag{7.5}$$

and thus $\Omega^* = \nabla F(\Omega)$. Also, $Q = (x, F(x))$ is mapped to $Q^*(a) = \langle x, a\rangle - F(x)$ by polarity, which should be the tangent affine map of F^* at x^*. Hence,

$$x = \nabla F^*(x^*). \tag{7.6}$$

If F is of Legendre type, then so is F^*. The associated Bregman divergences are symmetric [28]:

$$D_F(x\|y) = D_{F^*}(y^*\|x^*). \tag{7.7}$$

Example: Half the Squared Euclidean Norm. Let $\varpi\colon \mathbb{R}^d \to \mathbb{R}$ be the function defined by $\varpi(x) := \frac{1}{2}\|x\|^2$, i.e., it is half the squared Euclidean norm. Its gradient is $\nabla\varpi(x) = x$, and its conjugate function ϖ^* is the same as ϖ. It follows that for this example the associated Bregman divergence is symmetric. It indeed corresponds to half the squared Euclidean distance: $D_\varpi(x\|y) = \frac{1}{2}\|x-y\|^2$.

Example: Shannon Entropy and Kullback-Leibler Divergence. The *Shannon entropy* of a discrete probability distribution is $-\sum_{i=1}^d x_i \ln x_i$. We adapt it slightly by changing the sign and subtracting the sum of the x_i, defining $E(x) := \sum_{i=1}^d [x_i \ln x_i - x_i]$ over the positive orthant $\mathbb{R}^d_+$. This function is of Legendre type and the associated divergence is

$$D_E(x\|y) = \sum_{i=1}^d [x_i \ln x_i - x_i \ln y_i - x_i + y_i], \tag{7.8}$$

which is known as *Kullback-Leibler divergence* or *relative entropy* and plays an important role in information theory. The gradient is $\nabla E(x) = (\ln x_1, \ldots, \ln x_d)^T$ and for the conjugate function we get $E^*(y) = \sum_{i=1}^d \exp y_i$ for $y \in \mathbb{R}^d$. In Figures 7.1, 7.2, and 7.3 we have in fact used the Shannon entropy in $\mathbb{R}^1$ as an example of a Legendre type function F.

For applications, the restriction of the Shannon entropy to the standard simplex is of special interest. We define the *standard* $(d-1)$*-simplex* as the subset of points $x = (x_1, \ldots, x_d)^T$ in $\mathbb{R}^d_+$, for which $x_1 + \cdots + x_d = 1$, and denote it by Δ^{d-1}. We consider the open standard simplex instead of the closed one (where coordinates can have the value 0), because we use it as the domain of a Legendre type function. We write $E_\Delta\colon \Delta^{d-1} \to \mathbb{R}$ for the restriction of the Shannon entropy to the standard simplex.

Every $x \in \Delta^{d-1}$ can be interpreted as a discrete probability distribution on d disjoint events. Then, $-E_\Delta(x)$ is the expected efficiency to optimally encode a sample from this distribution. And the Kullback-Leibler divergence from x to y is the expected loss in coding efficiency (the number of extra bits required) if we use the code optimized for y to encode a sample from x.

We get the gradient of the restricted Shannon entropy as the projected gradient of the unrestricted Shannon entropy onto the hyperplane of the simplex. Projecting to the parallel hyperplane that goes through the origin instead, we get an expression in terms of the barycentric coordinates in Δ^{d-1}. Also for the conjugate we get a function that is different from the unrestricted case, which we state both in terms of the original and the conjugate coordinates $x^* = \nabla E_\Delta(x)$.

Lemma 7.1 (Gradient and Conjugate of Shannon Entropy Restricted to Δ^{d-1})**.**

$$\nabla E_\Delta(x) = \begin{bmatrix} \ln x_1 \\ \vdots \\ \ln x_d \end{bmatrix} - \tfrac{1}{d} \sum_{i=1}^{d} \ln x_i \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \tag{7.9}$$

$$E^*_\Delta(x^*) = E^*_\Delta(\nabla E_\Delta(x)) = 1 - \tfrac{1}{d} \sum\nolimits_{i=1}^{d} \ln x_i \tag{7.10}$$

$$= 1 + \ln \sum\nolimits_{i=1}^{d} \exp x_i^* \tag{7.11}$$

Proof. For the gradient, we project the vector $v := \nabla E(x) = (\ln x_1, \ldots, \ln x_d)^T$ onto the hyperplane with normal vector $n = (1, \ldots, 1)$ that goes through the origin:

$$v - \frac{\langle v, n \rangle\, n}{\|n\|^2} = \begin{bmatrix} \ln x_1 \\ \vdots \\ \ln x_d \end{bmatrix} - \frac{\sum_{i=1}^{d} \ln x_i}{d} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

To compute the conjugate function, we use (7.4) and (7.5):

$$\begin{aligned} E^*_\Delta(x^*) = E^*_\Delta(\nabla E_\Delta(x)) &= \langle \nabla E_\Delta(x), x \rangle - E_\Delta(x) \\ &= \sum_{i=1}^{d} x_i \ln x_i - \tfrac{1}{d} \sum_{i=1}^{d} \ln x_i \sum_{i=1}^{d} x_i - \sum_{i=1}^{d} [x_i \ln x_i - x_i] \\ &= -\tfrac{1}{d} \sum_{i=1}^{d} \ln x_i + 1. \end{aligned}$$

We get from the second to the third line by using the fact that $\sum_{i=1}^{d} x_i = 1$ in Δ^{d-1}.

Now, we want to express this in terms of the conjugate coordinates. We observe that the

inverse of ∇E_Δ is $(\nabla E_\Delta)^{-1}(x^*) = \exp x^* / \sum_{i=1}^d \exp x_i^*$:

$$\begin{aligned}
\nabla E_\Delta((\nabla E_\Delta)^{-1}(x^*)) &= \ln \frac{\exp x^*}{\sum_{i=1}^d \exp x_i^*} - \tfrac{1}{d} \sum_{i=1}^d \ln \frac{\exp x_i^*}{\sum_{j=1}^d \exp x_j^*} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \\
&= x^* + (-\ln \sum_{i=1}^d \exp x_i^* - \tfrac{1}{d} \sum_{i=i}^d x_i^* + \ln \sum_{i=1}^d \exp x_i^*) \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \\
&= x^*.
\end{aligned}$$

In the above derivation we have used the fact that $\sum_{i=1}^d x_i^* = 0$ for points x^* in a hyperplane through the origin. Finally, we express E_Δ^* in terms of $x^* = \nabla E_\Delta(x)$:

$$\begin{aligned}
E_\Delta^*(x^*) &= 1 - \tfrac{1}{d} \sum_{i=1}^d \ln[(\nabla E_\Delta)^{-1}(x^*)_i] \\
&= 1 - \tfrac{1}{d} \sum_{i=1}^d [\ln \exp x_i^* - \ln \sum\nolimits_{j=1}^d \exp x_j^*] \\
&= 1 - \tfrac{1}{d} \sum_{i=1}^d x_i^* + \ln \sum_{i=1}^d \exp x_i^* \\
&= 1 + \ln \sum_{i=1}^d \exp x_i^*.
\end{aligned}$$

□

7.2 Bregman-Delaunay Triangulation

We extend Voronoi diagrams and Delaunay triangulations to Bregman geometry, as already done in [8].

Bregman-Voronoi Diagrams. Let $\Omega \subseteq \mathbb{R}^d$ be a non-empty open convex set, F a Legendre type function with domain Ω, and $X \subseteq \Omega$ a finite point set. Since the Bregman divergence associated with F is asymmetric, we can define two kinds of Voronoi domains in Bregman geometry. The *primal Voronoi domain* $\mathrm{Vor}_F(x, X, \Omega)$ contains all points for which x minimizes the Bregman divergence from that point, the *dual Voronoi domain*

$\mathrm{Vor}_F^*(x, X, \Omega)$ contains all points for which x minimizes the Bregman divergence to that point:

$$\mathrm{Vor}_F(x, X, \Omega) := \{q \in \Omega \mid D_F(x\|q) \leq D_F(y\|q) \quad \forall y \in X\},$$
$$\mathrm{Vor}_F^*(x, X, \Omega) := \{q \in \Omega \mid D_F(q\|x) \leq D_F(q\|y) \quad \forall y \in X\}.$$

We define the *primal* or *dual Bregman-Voronoi diagram* of X as the collection of primal or dual Voronoi domains of all points in X, denoted as $\mathrm{Vor}_F(X, \Omega)$ or $\mathrm{Vor}_F^*(X, \Omega)$. The dual Voronoi domains are always convex polyhedra, while the primal ones can have curved borders [8]. It follows from (7.7) that the primal Bregman-Voronoi diagram of X associated with F is the same as the dual Bregman-Voronoi diagram of X^* associated with F^*, but drawn on the original point set X.

Figure 7.4 shows the primal and the dual Bregman-Voronoi diagram associated with the Shannon entropy of 100 random points in $(0, 2]^2$, and compares them to the Euclidean Voronoi diagram, which is the same as the Bregman-Voronoi diagrams associated with ϖ.

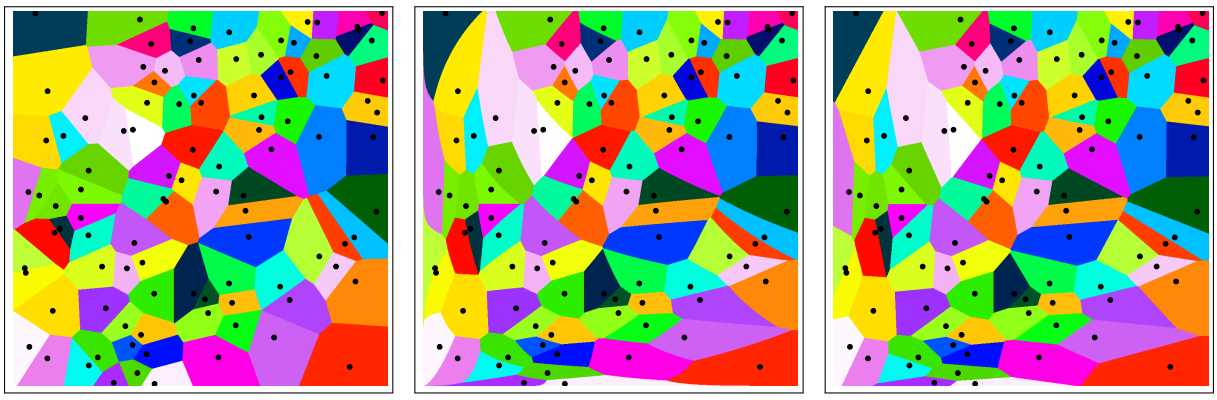


Figure 7.4: *Left:* Euclidean Voronoi diagram of 100 random points. *Middle:* Primal Bregman-Voronoi diagram associated with Shannon entropy. *Right:* Dual Bregman-Voronoi diagram associated with Shannon entropy.

Lifting for Euclidean Voronoi Diagram and Delaunay Triangulation. We will show that computing the Bregman-Delaunay triangulation of a given set of points is equivalent to computing the Euclidean Delaunay triangulation of some weighted point set.

We start by explaining how we can construct the Voronoi diagram and the Delaunay triangulation of an unweighted point set in Euclidean space, $X \subseteq \mathbb{R}^d$, as the projection

of a polyhedron in $\mathbb{R}^{d+1}$. We consider the collection of hyperplanes tangent to the graph of ϖ at a point $x \in X$. A point $a \in \mathbb{R}^d$ is in the Voronoi domain of a point x iff the value at a of the tangent hyperplane at x is greater than or equal to the values of the tangent hyperplanes at other points of X.

Lemma 7.2 (Lifted Voronoi Diagram)**.** *Let $X \subseteq \mathbb{R}^d$ be a finite point set, $\varpi\colon \mathbb{R}^d \to \mathbb{R}$ half the squared Euclidean norm, and $t_x\colon \mathbb{R}^d \to \mathbb{R}$ the tangent hyperplanes of ϖ at $x \in X$. For every $a \in \mathbb{R}^d$,*

$$a \in \mathrm{Vor}(x, X) \quad \Leftrightarrow \quad t_x(a) \geq t_y(a)\ \forall y \in X.$$

Proof. For the tangent hyperplane of ϖ at x, we have

$$t_x(a) = \varpi(x) + \langle \nabla\varpi(x), a - x\rangle = \tfrac{1}{2}\|x\|^2 + \langle x, a - x\rangle.$$

The following derivation proves the claim.

$$\begin{aligned}
\tfrac{1}{2}\|x - a\|^2 &\leq \tfrac{1}{2}\|y - a\|^2 && \Leftrightarrow\\
\tfrac{1}{2}\|x\|^2 + \tfrac{1}{2}\|a\|^2 - \langle x, a\rangle &\leq \tfrac{1}{2}\|y\|^2 + \tfrac{1}{2}\|a\|^2 - \langle y, a\rangle && \Leftrightarrow\\
-\tfrac{1}{2}\|x\|^2 + \langle x, a\rangle &\geq -\tfrac{1}{2}\|y\|^2 + \langle y, a\rangle && \Leftrightarrow\\
\tfrac{1}{2}\|x\|^2 + \langle x, a - x\rangle &\geq \tfrac{1}{2}\|y\|^2 + \langle y, a - y\rangle && \Leftrightarrow\\
t_x(a) &\geq t_y(a).
\end{aligned}$$

□

We thus get the Voronoi diagram of X as the projection of the boundary complex of the graph of $\max_{x\in X} t_x$ onto $\mathbb{R}^d$, see the left panel of Figure 7.5. Since the Delaunay triangulation is dual to the Voronoi diagram, we get it by projecting the boundary complex of the dual polyhedron, which is the lower convex hull of the lifted points $\{(x, \varpi(x)) \mid x \in X\} \subseteq \mathbb{R}^{d+1}$ [18], see the right panel of Figure 7.5.

If the points are endowed with weights, we only need to adapt the construction slightly by vertically shifting the lifted points and their tangent hyperplanes according to the weights:

$$\begin{aligned}
\tfrac{1}{2}(\|x - a\|^2 - w_x) &\leq \tfrac{1}{2}(\|y - a\|^2 - w_y) && \Leftrightarrow\\
\tfrac{1}{2}\|x\|^2 + \langle x, a - x\rangle + \tfrac{1}{2}w_x &\geq \tfrac{1}{2}\|y\|^2 + \langle y, a - y\rangle + \tfrac{1}{2}w_y && \Leftrightarrow\\
t_x(a) + \tfrac{1}{2}w_x &\geq t_y(a) + \tfrac{1}{2}w_y.
\end{aligned}$$

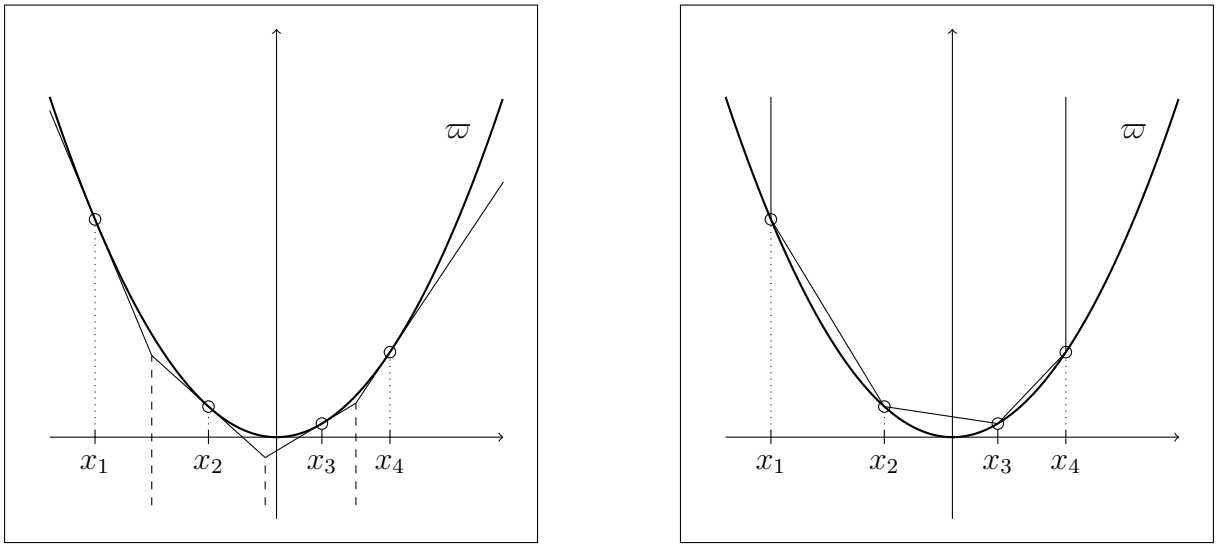


Figure 7.5: Construction of the Voronoi diagram and the Delaunay triangulation in $\mathbb{R}^1$ by lifting the points to the paraboloid ϖ. *Left:* The Voronoi diagram is the projected boundary complex of the polyhedron that is the upper hull of the tangent hyperplanes. *Right:* The Delaunay triangulation is the projected lower convex hull of the lifted points.

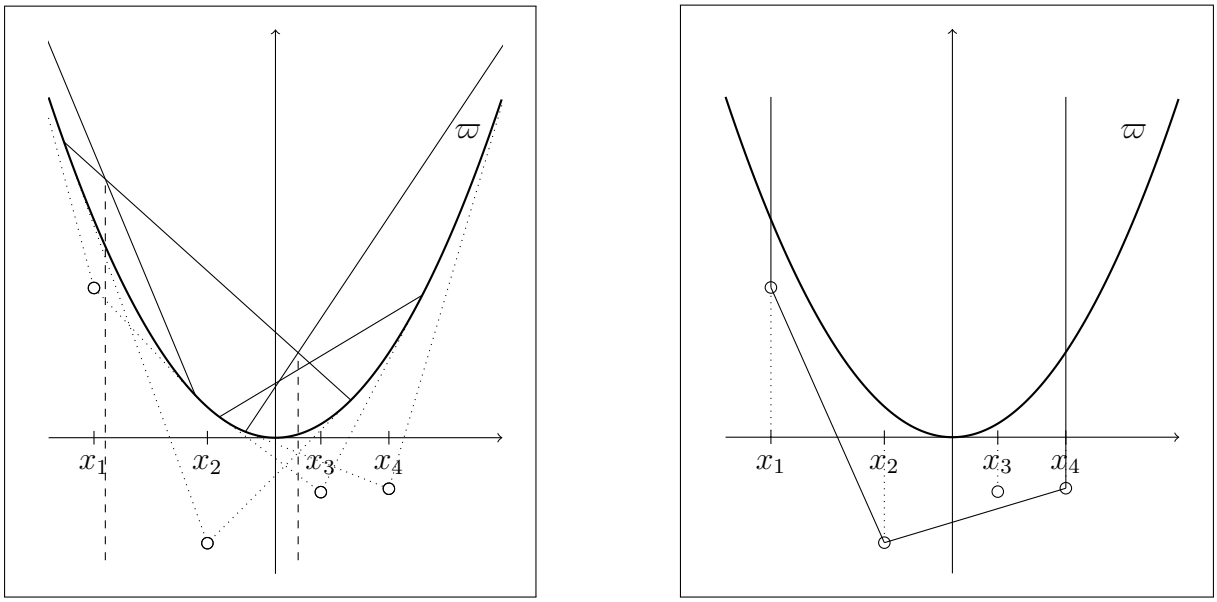


Figure 7.6: Construction of the weighted Voronoi diagram and the weighted Delaunay triangulation in $\mathbb{R}^1$ by lifting the points to the paraboloid ϖ and vertically shifting according to their weights. *Left:* The Voronoi diagram is the projected boundary complex of the polyhedron that is the upper hull of the shifted tangent hyperplanes. *Right:* The Delaunay triangulation is the projected lower convex hull of the lifted points.

The construction of the weighted Voronoi diagram and the weighted Delaunay triangulation is illustrated in Figure 7.6. Note that the tangent hyperplane of a point x is shifted by $(+\frac{1}{2}\, w_x)$, while the lifted point is shifted by $(-\frac{1}{2}\, w_x)$. From the lifted point $(x, \varpi(x) - \frac{1}{2} w_x)$, exactly the part of the graph of ϖ is visible, which lies below the shifted tangent hyperplane corresponding to x.

Lifting for Bregman-Delaunay Triangulation. We go back to Bregman geometry. Let $\Omega \subseteq \mathbb{R}^d$ be a non-empty open convex set, F a Legendre type function with domain Ω, and $X \subseteq \Omega$ a finite point set. As in Euclidean geometry, the *(primal) Bregman-Delaunay triangulation* $\mathrm{Del}_F(X, \Omega)$ of X is dual to the (primal) Bregman-Voronoi diagram of X. A set of points $\sigma \subseteq X$ is a simplex in the Bregman-Delaunay triangulation iff their (primal) Voronoi domains have a non-empty intersection. Equivalently, a d-simplex is in the Bregman-Delaunay triangulation iff it has an empty dual circumsphere, i.e., a dual Bregman ball whose boundary passes through the vertices of the simplex and whose interior is empty.

As illustrated in Figure 7.2, we get the boundary of the dual Bregman ball $B^*_F(x, h)$ as the projection of the intersection of the graph of F and the tangent hyperplane at x vertically shifted by $+h$. The part of the graph of F which is lower than the shifted tangent hyperplane corresponds to the interior of the dual Bregman ball.

Just as in Euclidean geometry, we can construct the Bregman-Delaunay triangulation as the projected lower convex hull of lifted points. Indeed, the Euclidean Delaunay triangulation is a special case of a Bregman-Delaunay triangulation and the function ϖ, which we have used for lifting before, is exactly the corresponding Legendre type function. Now, for a general Bregman-Delaunay triangulation associated with the function F, we lift every point $x \in \Omega$ to the point $(x, F(x)) \in \mathbb{R}^{d+1}$, and construct the convex hull of the lifted point set, see Figure 7.7. If a subset of points $\sigma \subseteq X$ is the set of vertices of a lower convex hull face, then the supporting hyperplane corresponds to a dual Bregman ball with the vertices on the boundary and no points of X inside, i.e., its boundary is an empty dual circumsphere. Thus, we get the simplices of the Bregman-Delaunay triangulation by projecting the faces of the lower convex hull onto Ω. This construction also shows that the Bregman-Delaunay triangulation has a straight-line embedding in Ω.

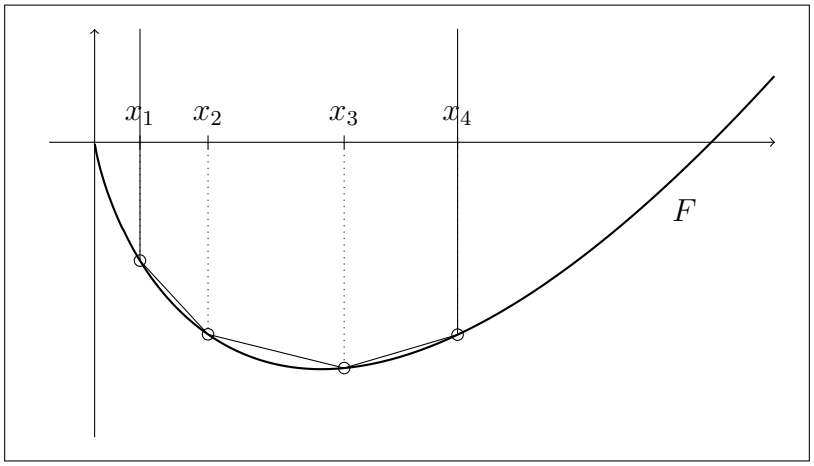


Figure 7.7: Construction of the Bregman-Delaunay triangulation associated with F by lifting the points to the graph of F and projecting the lower convex hull of the lifted points onto Ω. In this example, F is the Shannon entropy in $\mathbb{R}^1$.

Computing the Bregman-Delaunay Triangulation. Finally, we want to compute the Bregman-Delaunay triangulation by computing an equivalent weighted Euclidean Delaunay triangulation. We have explained above that both can be constructed by lifting the points and computing the convex hull of the lifted points. We choose the weights for the Euclidean Delaunay triangulation in such a way that the lifted points are the same as for the Bregman-Delaunay triangulation. In particular, we set $w_x = 2\varpi(x) - 2F(x)$ for every point $x \in X$. The corresponding lifted points $\bar{X}$ are

$$(x, \varpi(x) - \tfrac{1}{2}w_x) = (x, F(x)) \quad \text{with } x \in X,$$

which are the same as the lifted points for computing the Bregman-Delaunay triangulation. For this construction, we assume that the weighted point set is in general position.

Note that the weighted Delaunay triangulation $\mathrm{Del}(\bar{X})$ is not exactly the same as the Bregman-Delaunay triangulation $\mathrm{Del}_F(X, \Omega)$, which we want to compute. It can contain additional simplices, since it is constructed in the full Euclidean space $\mathbb{R}^d$, as opposed to the Bregman-Delaunay triangulation, where the domain $\Omega \subseteq \mathbb{R}^d$ can be smaller. In the next section, we will explain how to compute circumcenters for simplices in the Bregman-Delaunay triangulation. If we are not able to find a circumcenter within the domain Ω, the simplex will be removed from the triangulation.

7.3 Rise Function

We want to generalize the Delaunay radius function from Euclidean geometry to Bregman-Delaunay triangulations. For this purpose, we make use of the construction with lifted points presented above.

In Primal Space. Let $\Omega \subseteq \mathbb{R}^d$ be a non-empty open convex set, F a Legendre type function with domain Ω, $X \subseteq \Omega$ a finite point set, $\mathrm{Vor}_F(X, \Omega)$ its primary Bregman-Voronoi diagram, and $\mathrm{Del}_F(X, \Omega)$ its Bregman-Delaunay triangulation. Each simplex in the Bregman-Delaunay triangulation corresponds to a collection of Voronoi domains whose intersection is not empty. As in the Euclidean case, we intersect the Voronoi domains with primal Bregman balls of size h centered at the points of X, and call them *Voronoi balls*. We associate to each simplex σ the smallest value h for which the corresponding Voronoi balls have a non-empty intersection, and denote it by $\rho_F(\sigma)$.

As illustrated in Figure 7.2, we get the primal Bregman ball of size h centered at a point $x \in \Omega$ by casting light from the point $(x, F(x) - h)$ onto the graph of F and projecting the illuminated part of the graph onto $\mathbb{R}^d$. Now, if we start from the points X lifted to the graph of F and raise the graph of F by h, the "cones of light" cast from the lifted points onto the raised graph correspond to primal balls of size h. When the cones cast from the points of σ and clipped to within their corresponding Voronoi domains have a point in common for the first time, the amount h we have raised the graph of F is the value $\rho_F(\sigma)$, see Figure 7.8. To avoid confusion with the Euclidean case, we call this value the *rise function value* of σ.

Equivalently, $\rho_F(\sigma)$ is the size of the smallest dual Bregman ball with the points of σ on the boundary and no points of X inside. We call the center of this ball the *circumcenter* of σ. As illustrated in Figure 7.2, a dual Bregman ball corresponds to the part of the graph of F below a shifted tangent hyperplane. Thus, $\rho_F(\sigma)$ is also the minimum amount we have to raise the graph of F so it has a tangent hyperplane that passes through all points $(x, F(x))$ with $x \in \sigma$, while all other lifted points lie on or above the hyperplane, see again Figure 7.8.

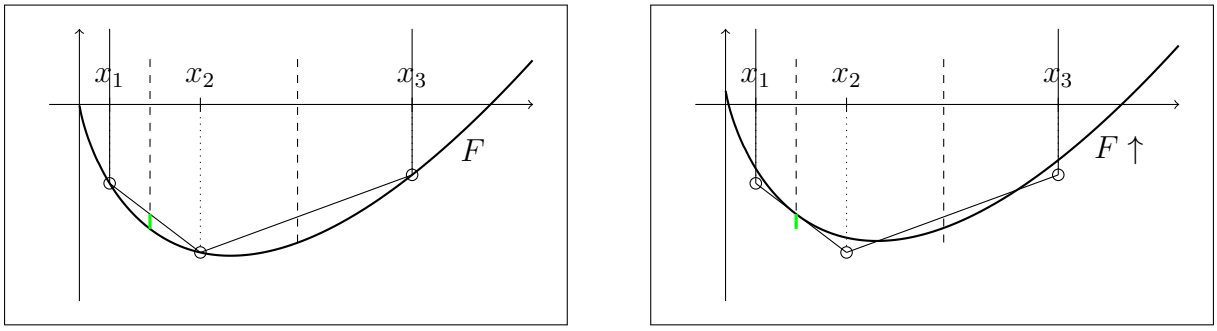


Figure 7.8: Illustration of the rise function ρ_F for lifted points in primal space. We lift the points of X onto the graph of F, and grow primal balls around the points by raising the graph of F. When the primal balls centered at the points of a simplex σ intersect for the first time at an intersection of the corresponding Voronoi domains, we get the value $\rho_F(\sigma)$ as the amount we have raised F. Here, we highlight the rise function value for the edge x_1x_2 in green. Equivalently, there is a hyperplane tangent to the raised graph that passes through all the lifted points of σ, while all other lifted points lie on or above the hyperplane. In this example, F is the Shannon entropy in $\mathbb{R}^1$.

In Conjugate Space. We propose an alternative characterization of the rise function ρ_F in conjugate space, which gives a compact formula. Due to (7.7), a Voronoi ball of size h in primal space, which is the intersection of a Voronoi domain with a primal Bregman ball of size h, corresponds to the intersection of the image of this Voronoi domain in conjugate space with a dual Bregman ball of size h. Then, the value $\rho_F(\sigma)$ is the smallest h for which the dual Voronoi balls of size h centered at the $x^* = \nabla F(x)$ for $x \in \sigma$ have a non-empty intersection.

For every point x^* lifted to $(x^*, F^*(x^*))$, we have a hyperplane that is tangent to the graph of F^* at x^*, which we denote by t_{F^*,x^*}. When we lift the points x^* to the graph of F^* and also draw the upper envelope of the tangent hyperplanes, $\max_{x\in X} t_{F^*,x^*}$, then growing dual balls centered at the x^* corresponds to lowering the graph of F^*, as illustrated in Figure 7.9. Thus, $\rho_F(\sigma)$ is the infimum amount we have to lower the graph of F^* until it intersects the graph of $\max_{x\in X} t_{F^*,x^*}$ at a point vertically above the image of the intersection of Voronoi domains corresponding to the points of σ. Hence, we get the following expression for the rise function value of σ:

$$\rho_F(\sigma) = \inf_{a \in \bigcap_{q\in\sigma} \mathrm{Vor}_F(q,X,\Omega)} [F^*(a^*) - \max_{x\in X} t_{F^*,x^*}(a^*)]. \tag{7.12}$$

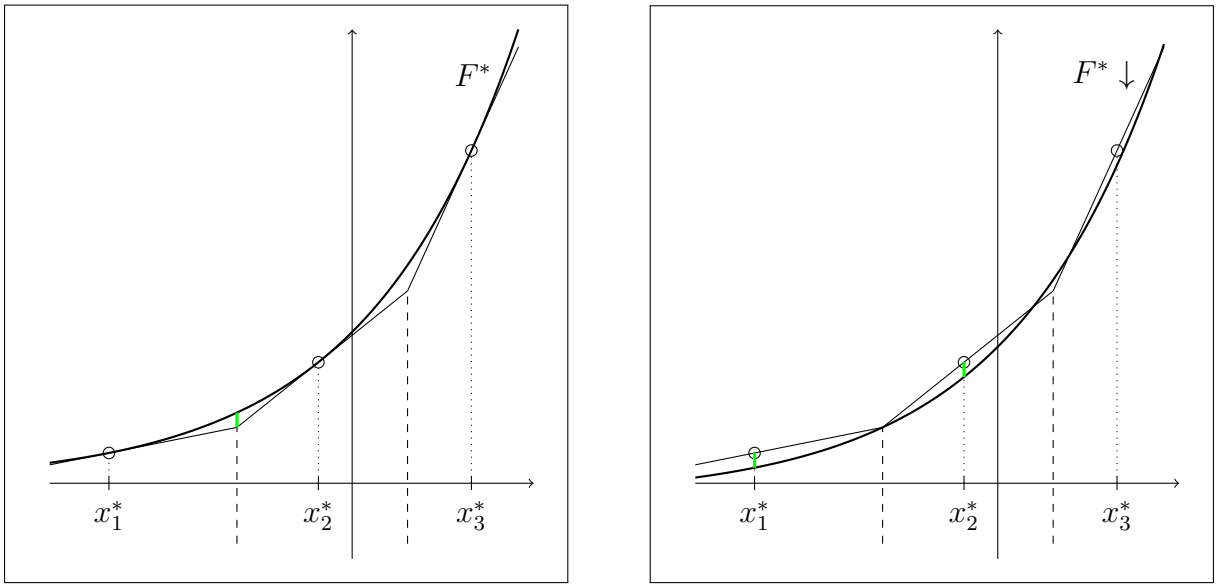


Figure 7.9: Illustration of the rise function ρ_F for lifted points in conjugate space. We lift the points in conjugate space, $X^* = \{\nabla F(x) \mid x \in X\}$, onto the graph of F^*, and grow dual balls around them by lowering the graph of F. The rise function value of a simplex σ is the infimum amount we have to lower the graph of F^* until it intersects the upper envelope of the tangent hyperplanes at a point vertically above the intersection of Voronoi domains in conjugate space. Here, we highlight the rise function value for the edge x_1x_2 in green. It is the same as in Figure 7.8.

Rise Function of ϖ. We illustrate the rise function for the case $F = \varpi$ and show its relation to the Euclidean Delaunay radius function. Recall that $\varpi = \varpi^*$, so we can draw the construction of the rise function in primal space and in conjugate space in a single picture, see Figure 7.10. Primal and dual Bregman balls are the same in this case. And since the Bregman divergence associated with ϖ is the same as half the squared Euclidean distance, the Bregman ball of size $h \geq 0$ centered at a point x is the same as the Euclidean ball of radius $\sqrt{2h}$ centered at the same point. In order to avoid any confusion arising from this correspondence, we prefer to use the term *rise function* instead of *radius function* in Bregman geometry.

Computation. For computation, we prefer the characterization of the rise function in primal space, as illustrated in Figure 7.8. For a p-simplex σ, the affine hull of the points $\bar{\sigma} := \{(x, F(x)) \mid x \in \sigma\}$ is a p-dimensional plane. The point (q, h) in the plane that maximizes the vertical distance to the graph of F, $h - F(q)$, is the center of the smallest

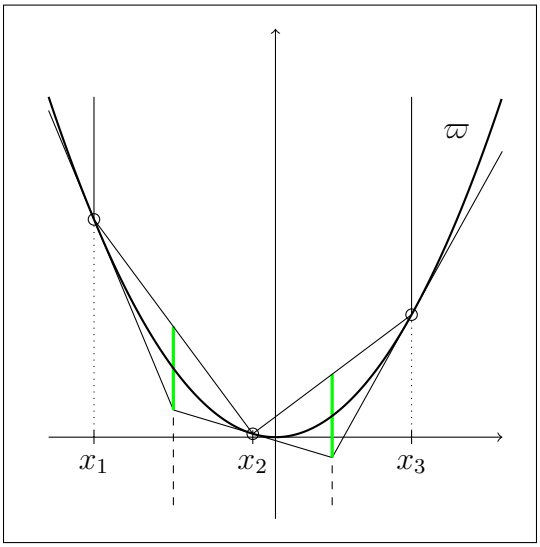


Figure 7.10: Illustration of the rise function ρ_ϖ for lifted points. In this case, the primal and the conjugate space are the same, and also primal and dual Bregman balls are equal. Therefore, we can draw the construction in primal and in conjugate space in the same picture and see that the resulting rise function values (highlighted in green) are the same. They are equal to half the squared Euclidean Delaunay radii of the simplices.

dual Bregman ball with the points of σ on the boundary, which we call a *dual circumball* of σ, and $h - F(q)$ is the size of this ball [28].

To get the rise function value of a simplex, we need the additional property that the dual circumball is empty. For a simplex σ of dimension d, this is always the case for the smallest dual circumball. Thus, we can compute $\rho_F(\sigma)$ by maximizing $h - F(q)$ over the affine hull of $\bar{\sigma}$ intersected with $\Omega \times \mathbb{R}$, which is a convex optimization problem that can be solved using numerical methods [28]. Also, the point q found by optimization is the circumcenter of σ.

For the lower-dimensional simplices, we have to check whether the smallest dual circumball is empty. This can be easily done by checking whether none of the vertices of the cofacets of the simplex lie inside. If this is the case, the size of the smallest dual circumball gives the rise function value. If not, we get it as the smallest rise function value of the cofacets, because the rise function is a generalized discrete Morse function [28].

Bregman-Alpha and Bregman-Wrap Complexes. Having assigned a function value to all simplices of the Bregman-Delaunay triangulation, we define the Alpha complex as

in the Euclidean case: The *Bregman-Alpha complex* of X for size h is the set of all Delaunay simplices σ with $\rho_F(\sigma) \leq h$. Also, we can define the Bregman-Wrap complex just as described for the Euclidean case in Section 2.3, because ρ_F is a generalized discrete Morse function.

7.4 Fisher Metric

A Bregman divergence associated with a twice differentiable Legendre type function induces a metric that integrates infinitesimal steps along shortest paths, and for Bregman divergences that decompose into a term per coordinate there exists an isometry between this path metric and the Euclidean metric [1, 27]. As an example, we consider the metric induced by the Kullback-Leibler divergence, which is known as the *Fisher metric*, and we will later compare the corresponding Delaunay triangulation with the Bregman-Delaunay triangulation associated with the Shannon entropy.

Induced Path Metric. For a twice differentiable Legendre type function $F\colon \Omega \to \mathbb{R}^d$ the *Hessian matrix* of second derivatives is well defined at every $x \in \Omega$:

$$H_F(x) := \left[\frac{\partial^2 F}{\partial x_i \partial x_j}(x)\right]_{1\leq i,j\leq d}.$$

Since the matrix is symmetric, we get a scalar product at x defined by $\langle a, b\rangle_x := \frac{1}{2}a^T H_F(x) b$ for $a, b \in \Omega$. The corresponding metric measures the distance between a and b as the length of the shortest path from a to b: $\gamma\colon [0,1] \to \Omega$ with $\gamma(0) = a$ and $\gamma(1) = b$, where the length of γ is $\int_0^1 \sqrt{\langle \dot{\gamma}(t), \dot{\gamma}(t)\rangle_{\gamma(t)}}\mathrm{d}t$. Note that a restriction of the domain Ω makes the set of paths smaller. Thus, the distance between two points in the restricted set can be larger than the distance of the same points in the unrestricted set, while the value of the Bregman divergence does not change.

Fisher Metric in $\mathbb{R}^d_+$. There exists an isometry between the Euclidean metric and the path metric induced by a Bregman divergence that decomposes into a term per coordinate [1, 27]. We present the isometry for the Kullback-Leibler divergence, which induces the *Fisher metric* in $\mathbb{R}^d_+$. It is the mapping $\iota\colon \mathbb{R}^d_+ \to \mathbb{R}^d_+$ defined by

$$\iota(x) = (\sqrt{2x_1}, \ldots, \sqrt{2x_d})^T. \tag{7.13}$$

The distance between two points $x, y \in \mathbb{R}^d_+$ under the Fisher metric satisfies $\|x - y\|_{\text{Fisher}} = \|\iota(x) - \iota(y)\|$.

Fisher Metric in Δ^{d-1}. Restricting the Fisher metric to the standard simplex Δ^{d-1}, we get an isometry to the Euclidean metric in the positive part of the $(d-1)$-sphere with radius $\sqrt{2}$ centered at the origin in $\mathbb{R}^d$, which we denote by $\mathbb{S}^{d-1}_+$. The distance between two points $x, y \in \Delta^{d-1}$ is the length of the arc between $\iota(x)$ and $\iota(y)$ on the sphere. It is generally longer than the straight line segment connecting $\iota(x)$ and $\iota(y)$ in $\mathbb{R}^d_+$. Figure 7.11 illustrates some balls for the Fisher metric in the standard 2-simplex and the corresponding balls for the Euclidean metric in $\mathbb{S}^2_+$.

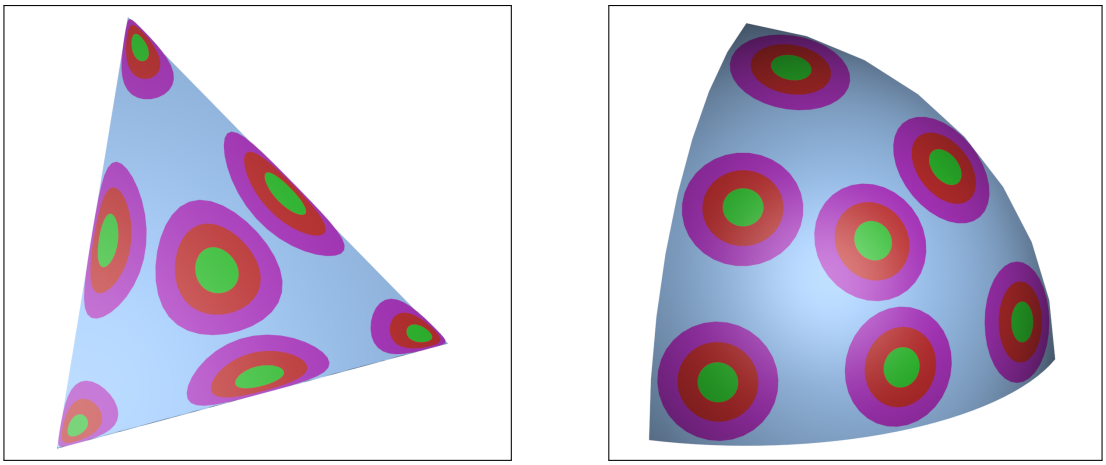


Figure 7.11: *Left:* Balls for the Fisher metric in Δ^2. *Right:* Corresponding balls for the Euclidean metric in $\mathbb{S}^2_+$. [27]

Computing Delaunay Triangulations in Fisher Space. We can easily construct Delaunay triangulations in Fisher space by means of the isometry $\iota\colon \mathbb{R}^d_+ \to \mathbb{R}^d_+$ from the Fisher metric to the Euclidean metric. Given a set of points $X \subseteq \mathbb{R}^d_+$, we compute the Euclidean Delaunay triangulation of $\iota(X)$, remove those simplices for which the center of the smallest empty circumsphere lies outside of $\mathbb{R}^d_+$, and draw the resulting complex on the original points X. The radius for the simplices in Fisher space is the same as the radius of the corresponding simplices in Euclidean space.

For points in Δ^{d-1}, we need to compute the Delaunay triangulation of $\iota(X)$ under the geodesic distance on $\mathbb{S}^{d-1}_+$. This can be done by constructing the convex hull of $\iota(X) \cup \{0\}$ in $\mathbb{R}^d$ and removing the faces incident to 0 [44]. Again, we also remove simplices for

which the circumcenter lies outside of $\mathbb{S}_+^{d-1}$. We get the circumcenters of the simplices in $\mathbb{S}_+^{d-1}$, which we also use for computing the radii of the simplices, by computing the circumcenters of the simplex vertices in $\mathbb{R}^d$ and centrally projecting them onto $\mathbb{S}_+^{d-1}$.

7.5 Computational Experiments

We illustrate the Bregman-Alpha and the Bregman-Wrap complex for the Shannon function with examples in $\mathbb{R}_+^2$ and Δ^2, and compare them to the corresponding complexes in conjugate space, Fisher geometry, and Euclidean geometry.

Example in $\mathbb{R}_+^2$. We sample a set of 1000 points uniformly distributed according to the Fisher metric in $(0,2]^2 \subseteq \mathbb{R}_+^2$. We do this by sampling 1000 points uniformly according to the Euclidean metric in $(0, \sqrt{2 \cdot 2} = 2]^2$, and use the inverse of the isometry ι to map them to Fisher space: $\iota^{-1}(x_1, x_2) = \frac{1}{2}(x_1^2, x_2^2)$.

The upper panel of Figure 7.12 displays the Bregman-Alpha complex in Shannon geometry for the threshold $h = 0.004$. We get a fairly uniform arrangement of simplices and holes in the complex, since the Kullback-Leibler divergence infinitesimally agrees with the squared Fisher metric, which we have used for sampling the points. The closer we get to the left or the lower side of the domain, the denser the points get and the more anisotropically aligned with the sides the edges and triangles get.

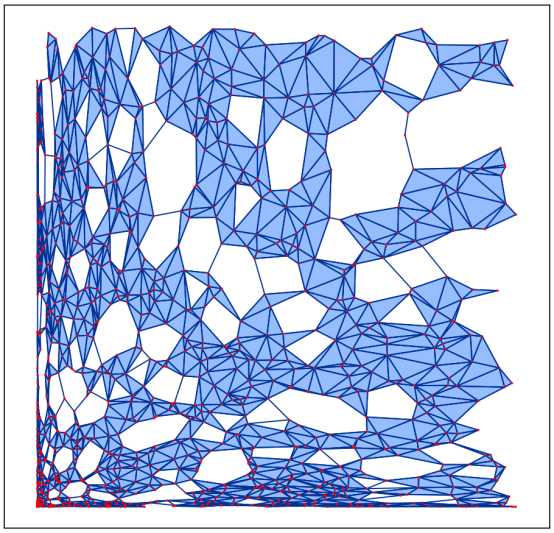
The other panels of Figure 7.12 show the corresponding Alpha complex in conjugate Shannon geometry, in Fisher geometry, in Euclidean geometry, and in weighted Euclidean geometry. The primal Delaunay triangulation in conjugate Shannon geometry is the same as the one corresponding to the dual Voronoi diagram in primal space, except that the latter misses some of the simplices close to the boundary of $\mathbb{R}_+^2$, since the corresponding dual Voronoi domains might not intersect inside $\mathbb{R}_+^2$, see the Voronoi diagram in the right panel of Figure 7.4. We mean subcomplexes of this clipped version of the Delaunay triangulation, when we talk about Alpha and Wrap complexes in conjugate Shannon geometry. Also, for the Delaunay triangulation in Euclidean and in Fisher geometry, we remove simplices for which the corresponding Voronoi domains do not intersect inside $\mathbb{R}_+^2$, i.e., the center of the smallest empty circumsphere lies outside.

The Bregman-Alpha complex in conjugate Shannon geometry is very similar to the one in Shannon geometry, since primal and dual Bregman balls associated with the Shannon entropy are similar. Not surprisingly, there is also a striking similarity between the Bregman-Alpha complex in Shannon geometry and the Alpha complex in Fisher geometry. In contrast, the Alpha complex in Euclidean geometry is very different. Because of the uneven distribution of points (measured by the Euclidean metric), the density of the complex decreases along the diagonal. The complex in weighted Euclidean geometry was constructed with the same weighted points as used for the computation of the Bregman-Alpha complex, but then the usual squared Delaunay radii for weighted points were assigned to the simplices instead of the rise function values of Bregman geometry. We observe that the squared Delaunay radii increase toward the boundary of the domain.

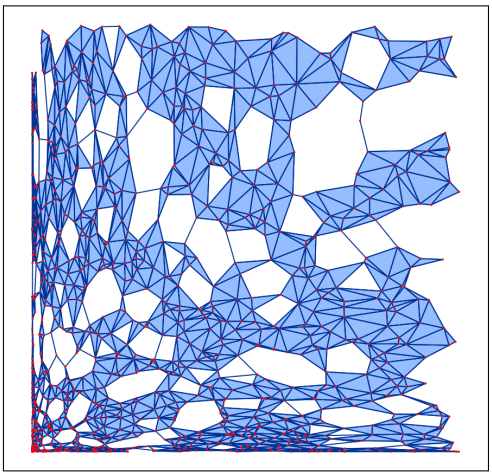
Figure 7.13 shows the Wrap complexes corresponding to the Alpha complexes of Figure 7.12. Most of them only differ by a few collapses from the corresponding Alpha complex. The exception is the Wrap complex in weighted Euclidean geometry, which consists of a single vertex near the upper right corner of $(0, 2]^2$. This reflects a simple relation between the Shannon entropy and the squared Euclidean norm: $2\varpi(x) - E(x)$ is monotonically increasing along the diagonal. This translates into a discrete gradient that introduces a flow with a single critical simplex, namely the vertex near the upper right corner.

Example in Δ^2. We repeat the above experiment within the standard triangle Δ^2, because this is the scenario which is relevant for information-theoretic applications. Every point in Δ^2 can be interpreted as a probability distribution on three disjoint events. To sample a set of 1000 points uniformly at random according to the Fisher metric in Δ^2, we use the isometry to the Euclidean metric and uniformly sample in $\mathbb{S}^2_+$ according to the geodesic distance.

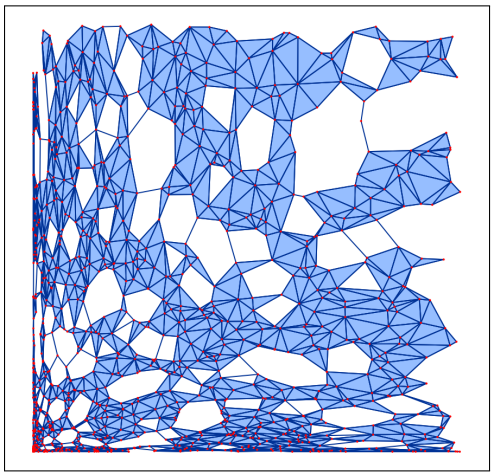
The Bregman-Alpha complex associated with the Shannon entropy for the threshold $h = 0.0025$, which is shown in the upper panel of Figure 7.14, exhibits a random distribution of simplices across the domain. This is again explained by the fact that the Kullback-Leibler divergence agrees with the squared Fisher metric in the infinitesimal regime. Also, as in the above example, the Alpha complex in Shannon geometry is similar to those in conjugate Shannon geometry and in Fisher geometry, while the ones in Euclidean



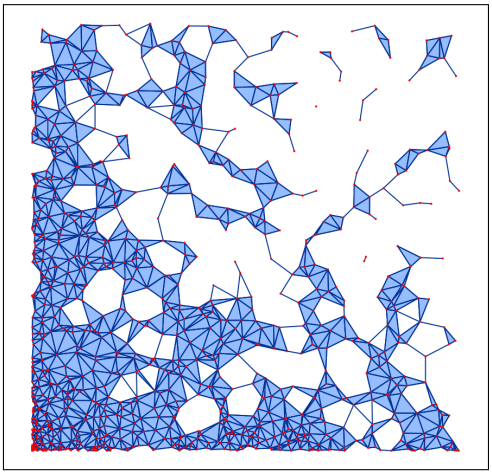
(a) Shannon.



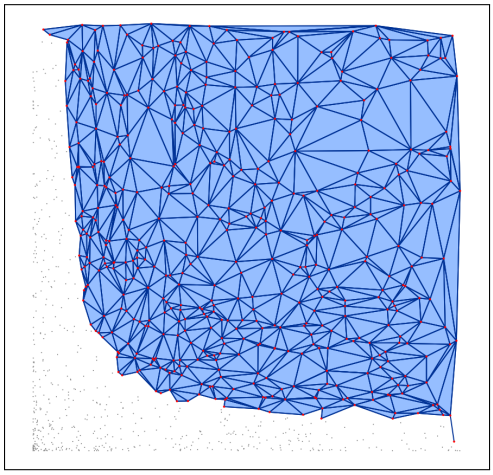
(b) Conjugate Shannon.



(c) Fisher.

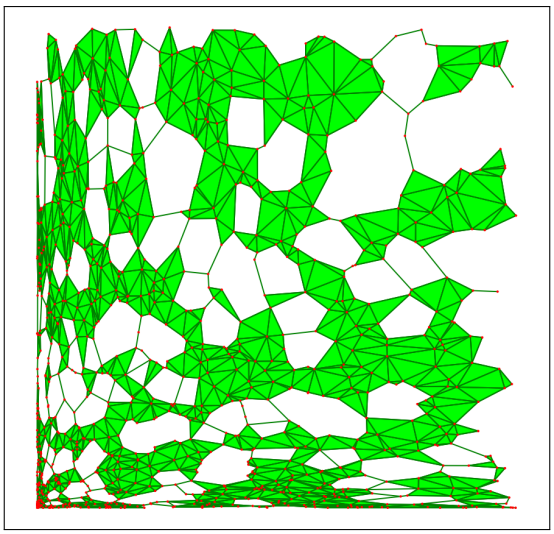


(d) Euclidean.

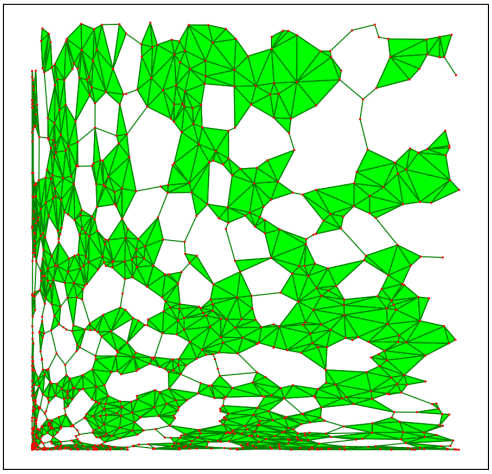


(e) Weighted Euclidean.

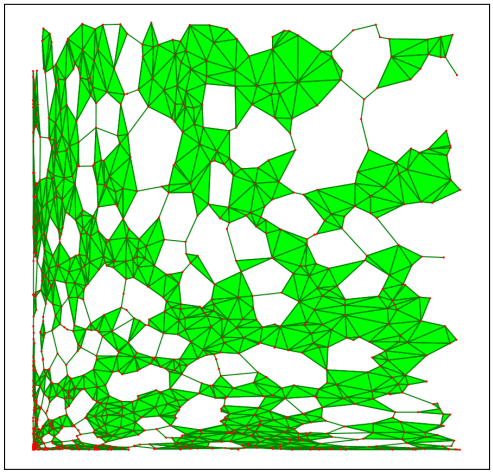
Figure 7.12: The Alpha complex of a set of 1000 points uniformly distributed according to the Fisher metric in $(0, 2]^2$ for $h = r^2 = 0.004$ in (a) Shannon, (b) conjugate Shannon, (c) Fisher, (d) Euclidean, and (e) weighted Euclidean geometry.



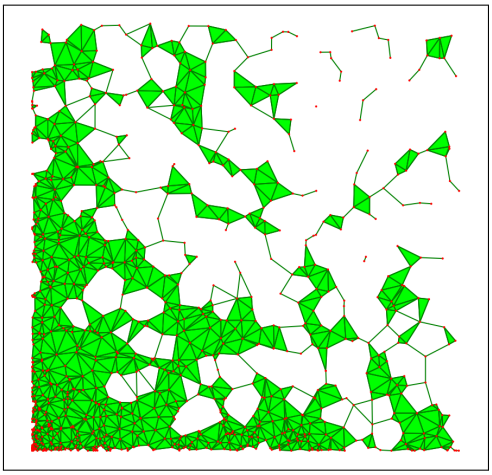
(a) Shannon.



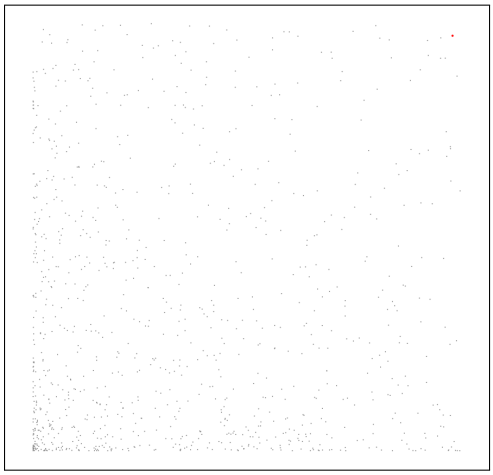
(b) Conjugate Shannon.



(c) Fisher.



(d) Euclidean.



(e) Weighted Euclidean.

Figure 7.13: The Wrap complexes of the same points and for the same threshold as in Figure 7.12.

geometry and weighted Euclidean geometry are quite different, see panels (b) to (e) of Figure 7.14. To interpret the result for weighted Euclidean geometry, we observe that the difference between the squared Euclidean norm and the Shannon entropy has a minimum at the center and no other critical points in the interior of the triangular domain. The simplices close to the boundary get assigned larger squared radius values and are therefore removed from the Alpha complex. More drastically, in the corresponding Wrap complex, which is displayed in panel (e) of Figure 7.15, all simplices except for a single critical edge near the center are collapsed.

Difference Between Delaunay Triangulations. We further study the differences between the Alpha and Wrap complexes in various geometries. In addition to the visual comparison above, we also want to quantify the differences. We start by comparing the underlying Delaunay triangulations. Letting D and E be two Delaunay triangulations of the same point set X, we compare them by counting the common simplices:

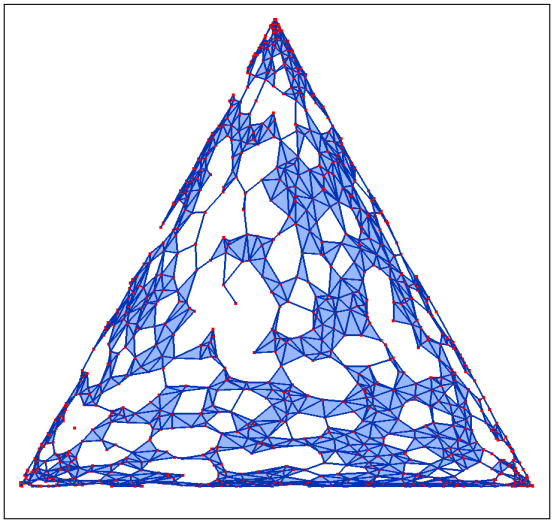
$$J(D, E) = 1 - \frac{|D \cap E|}{|D| + |E| - |D \cap E|}, \tag{7.14}$$

which is sometimes referred to as the *Jaccard distance* between the two sets. It is normalized so that $J = 0$ iff $D = E$ and $J = 1$ iff D and E share no simplices at all. We write D_S, D_C, D_F, D_E, D_W for the Delaunay triangulations of our example point set in $(0, 2]^2$ in Shannon, conjugate Shannon, Fisher, Euclidean, and weighted Euclidean geometry. And we write E_S, E_C, E_F, E_E, E_W for the Delaunay triangulations of the example point set in Δ^2.

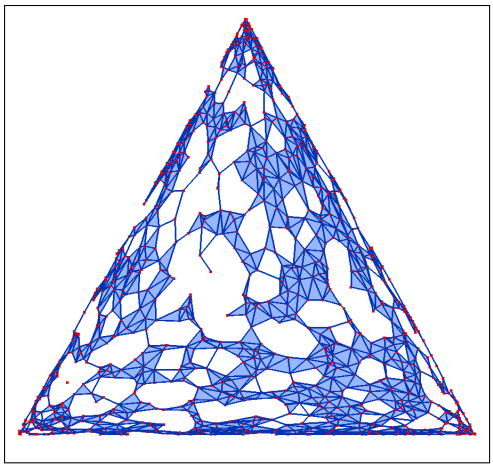
J	D_S	D_C	D_F	D_E	D_W
D_S	0.00	0.06	0.04	0.48	0.00
D_C		0.00	0.02	0.47	0.06
D_F			0.00	0.47	0.04
D_E				0.00	0.48
D_W					0.00

J	E_S	E_C	E_F	E_E	E_W
E_S	0.00	0.10	0.06	0.52	0.00
E_C		0.00	0.04	0.51	0.10
E_F			0.00	0.51	0.06
E_E				0.00	0.52
E_W					0.00

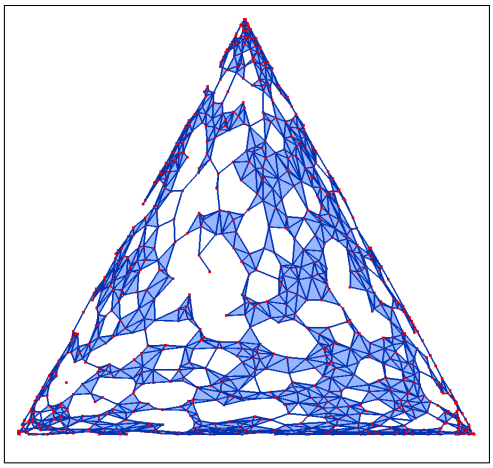
Table 7.1: The Jaccard distances between the Delaunay triangulations in Shannon, conjugate Shannon, Fisher, Euclidean, and weighted Euclidean geometry for the example point set in $(0, 2]^2$ on the *left* and in Δ^2 on the *right*.



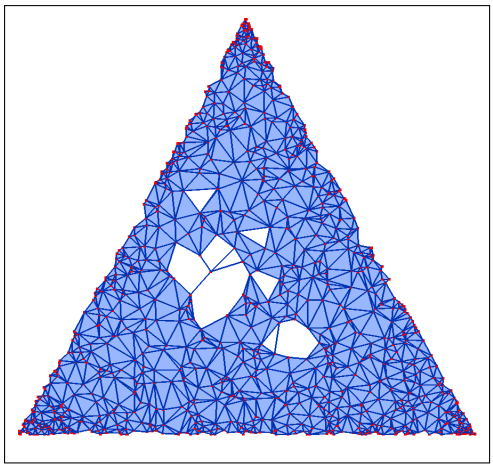
(a) Shannon.



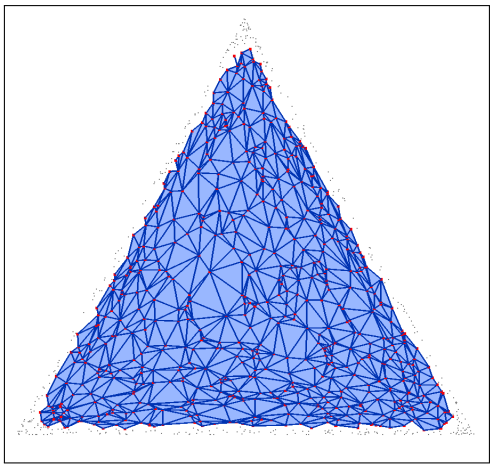
(b) Conjugate Shannon.



(c) Fisher.

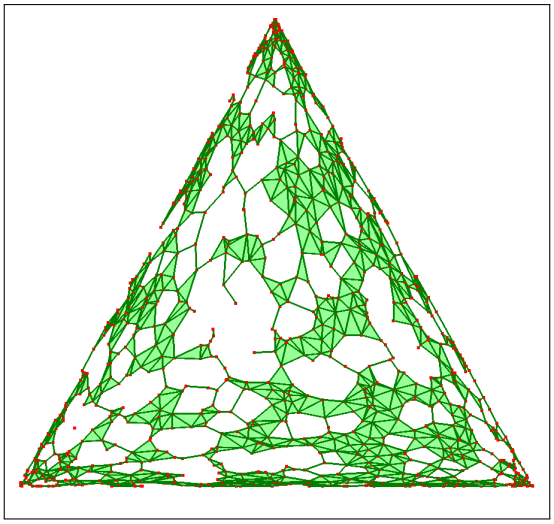


(d) Euclidean.

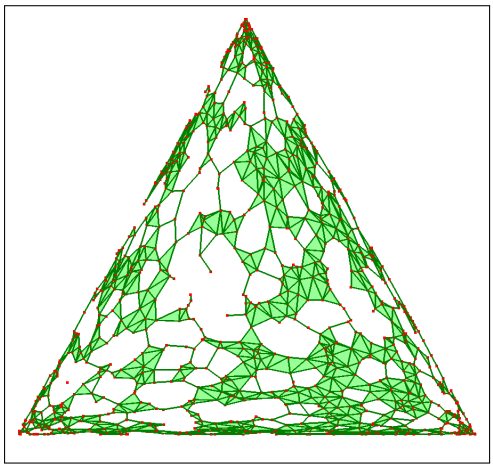


(e) Weighted Euclidean.

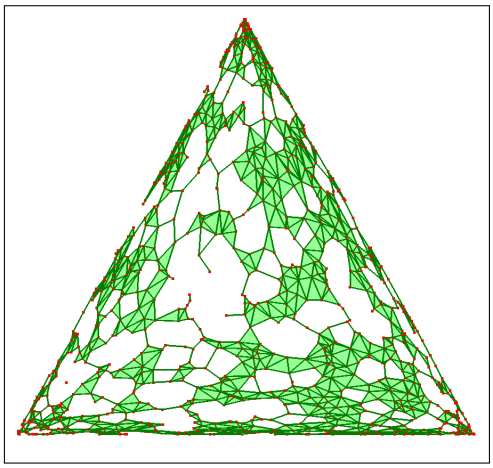
Figure 7.14: The Alpha complex of a set of 1000 points uniformly distributed according to the Fisher metric in Δ^2 for $h = r^2 = 0.0025$ in (a) Shannon, (b) conjugate Shannon, (c) Fisher, (d) Euclidean, and (e) weighted Euclidean geometry.



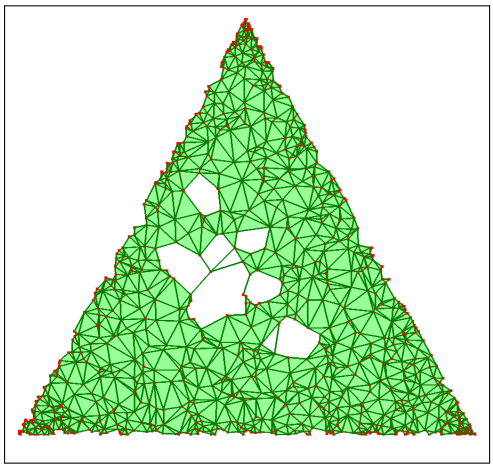
(a) Shannon.



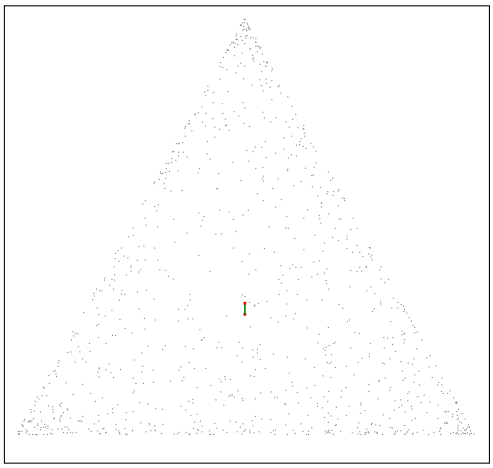
(b) Conjugate Shannon.



(c) Fisher.



(d) Euclidean.



(e) Weighted Euclidean.

Figure 7.15: The Wrap complexes of the same points and for the same threshold as in Figure 7.14.

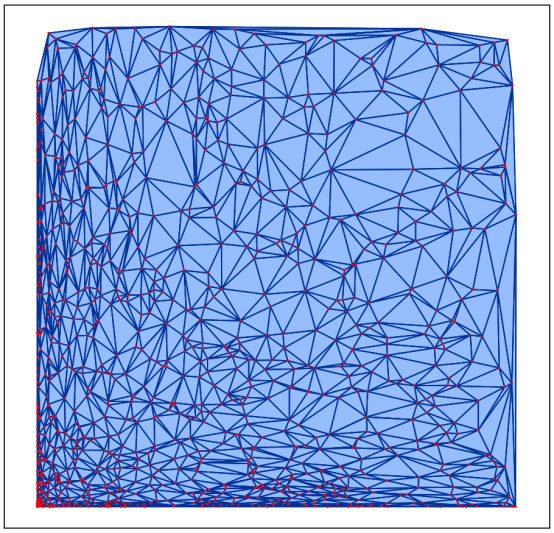
All Delaunay triangulations are different except the ones in Shannon geometry and weighted Euclidean geometry, which follows from the construction. For the others, the triangulations in conjugate Shannon geometry and in Fisher geometry are most similar to each other. The Delaunay triangulation in Euclidean geometry is most dissimilar to the others. We highlight the differences between the Delaunay triangulations in Figures 7.16 and 7.17.

Difference Between Rise/Radius Functions. We start by comparing the rise or squared radius functions in the different geometries visually, by color-coding the triangles. We shade triangles with a smaller function value darker. The color-coded Delaunay triangulations for the two example point sets are displayed in Figures 7.18 and 7.19. In Shannon, conjugate Shannon, and Fisher geometry, we get randomly mixed dark and light triangles. In Euclidean and weighted Euclidean geometry, there are clear but opposing gradients: the function values gradually increase or decrease in the direction of the diagonal for the point set in $(0, 2]^2$ and from the boundary to the center of the triangle for the points in Δ^2.

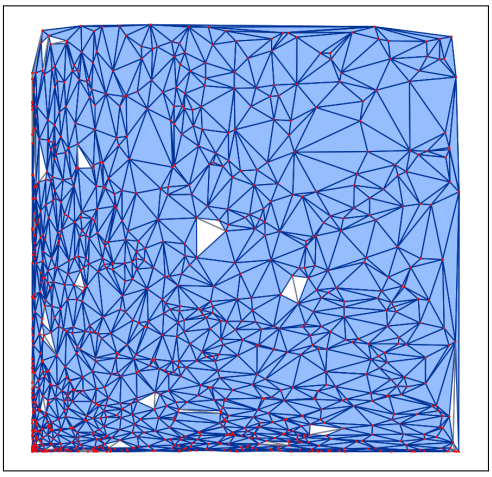
Different filtrations of the same Delaunay triangulation can be compared by counting the inversions, which are the pairs of simplices whose orderings are different in the two filtrations. Recall that $D_S = D_W$ and $E_S = E_W$, let $d_S \colon D_S \to \mathbb{R}$ and $e_S \colon E_S \to \mathbb{R}$ be the rise functions in Shannon geometry, and let $d_W \colon D_W \to \mathbb{R}$ and $e_W \colon E_W \to \mathbb{R}$ be the squared Delaunay radius functions in weighted Euclidean geometry. The numbers of inversions, normalized with the total number of pairs, are

$$I(d_S, d_W) = 0.476,$$
$$I(e_S, e_W) = 0.467.$$

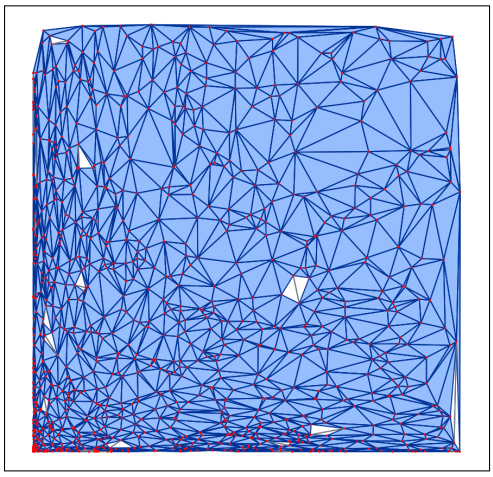
Slightly fewer than half the pairs are inversions in both examples. This is plausible because d_W orders the simplices along the diagonal while d_S preserves the random character of the point sample. Similarly, e_W orders the simplices radially, from the center of Δ^2 to its boundary, while e_S preserves again the random character of the sample.



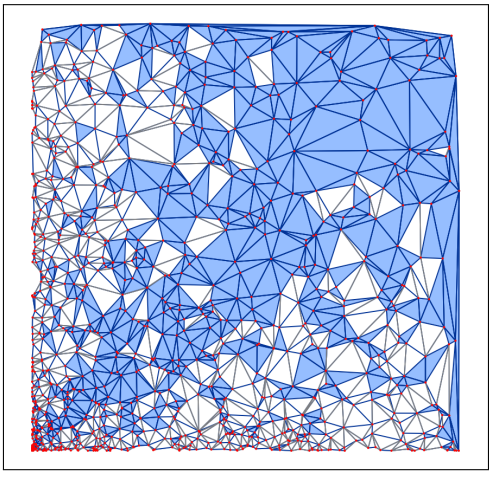
(a) Shannon.



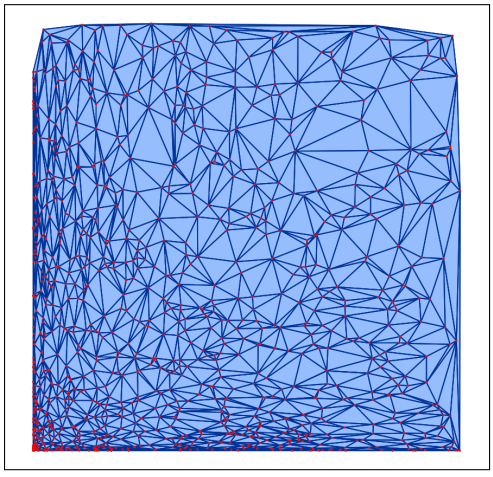
(b) Conjugate Shannon.



(c) Fisher.

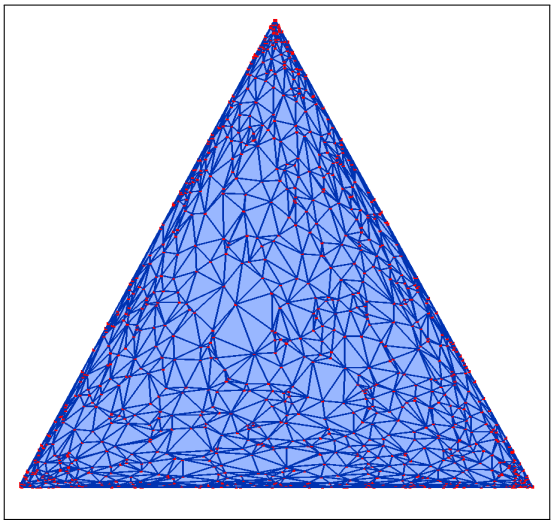


(d) Euclidean.

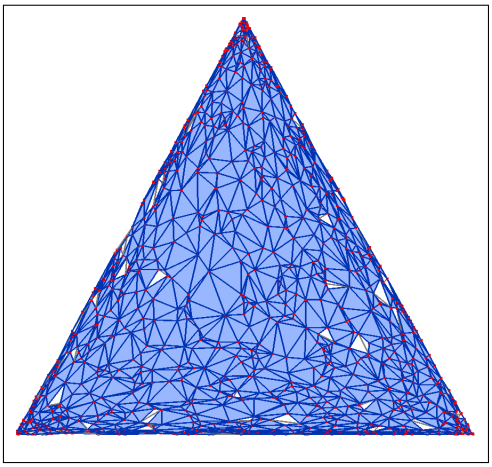


(e) Weighted Euclidean.

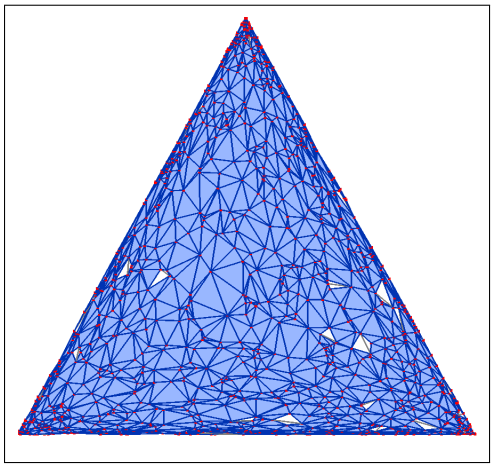
Figure 7.16: Comparison of the Delaunay triangulations of the same point set as used in Figure 7.12 for (a) Shannon, (b) conjugate Shannon, (c) Fisher, (d) Euclidean, and (e) weighted Euclidean geometry. The triangles and edges are colored depending on whether or not they belong to the Delaunay triangulation in Shannon geometry.



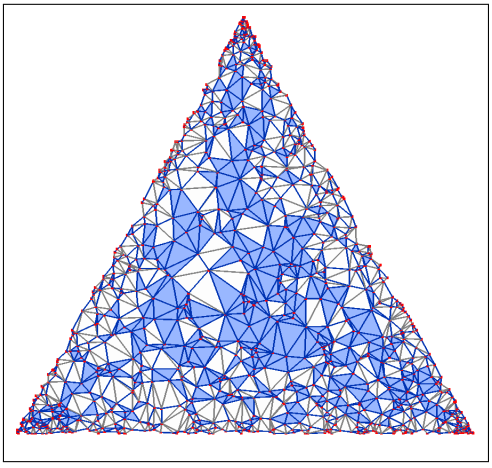
(a) Shannon.



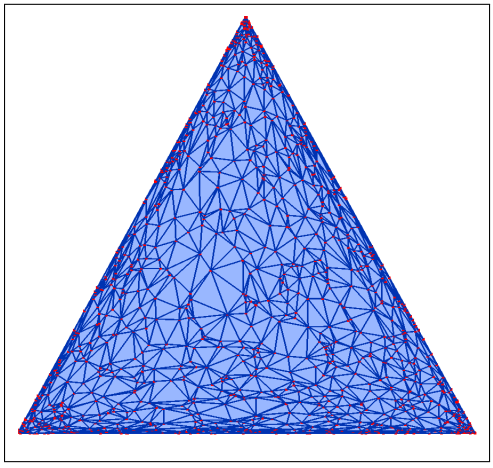
(b) Conjugate Shannon.



(c) Fisher.

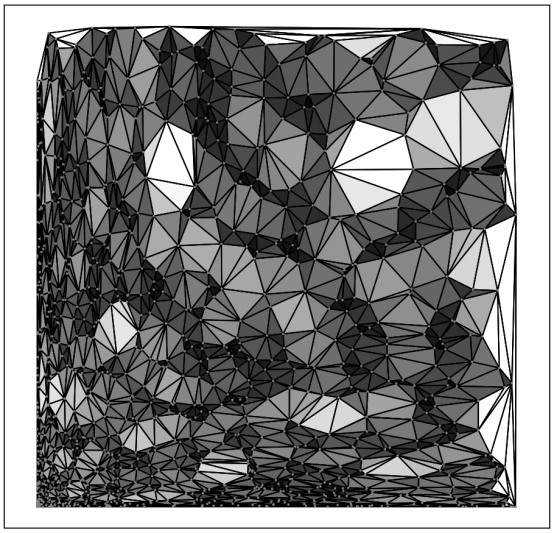


(d) Euclidean.

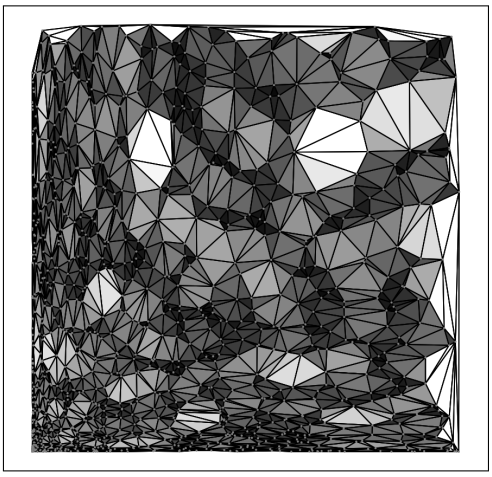


(e) Weighted Euclidean.

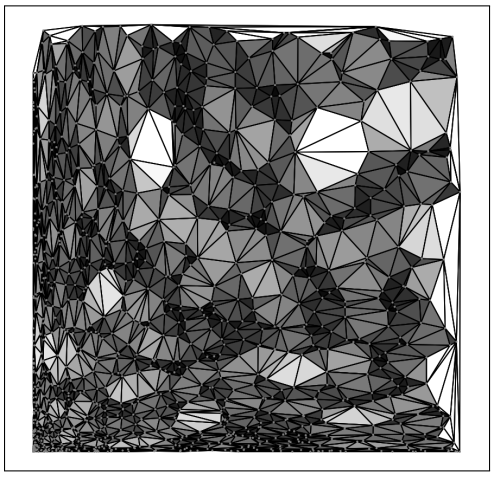
Figure 7.17: Comparison of the Delaunay triangulations of the same point set as used in Figure 7.14 for (a) Shannon, (b) conjugate Shannon, (c) Fisher, (d) Euclidean, and (e) weighted Euclidean geometry. The triangles and edges are colored depending on whether or not they belong to the Delaunay triangulation in Shannon geometry.



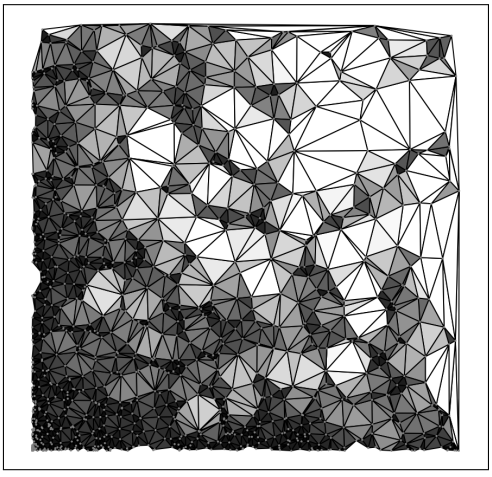
(a) Shannon.



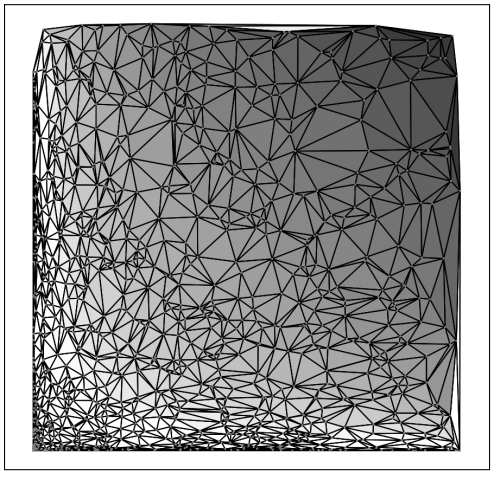
(b) Conjugate Shannon.



(c) Fisher.

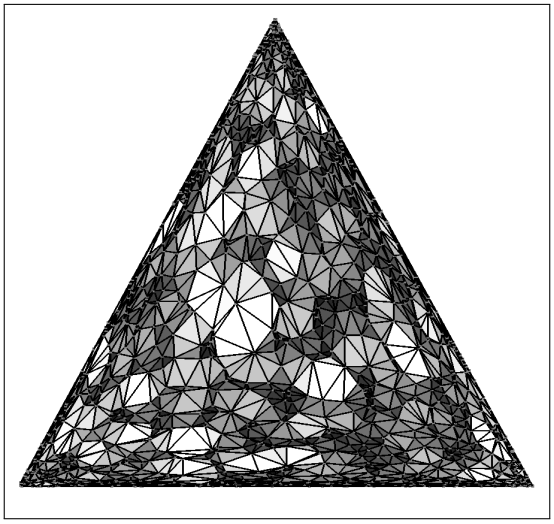


(d) Euclidean.

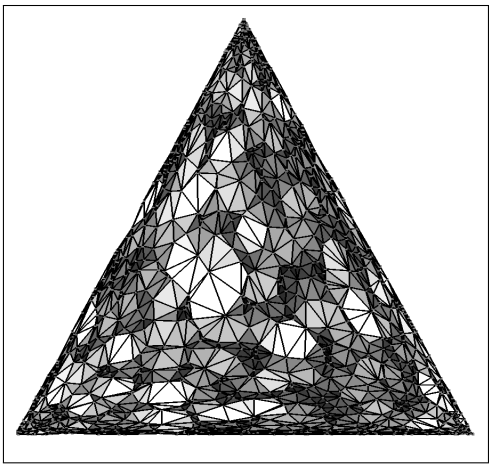


(e) Weighted Euclidean.

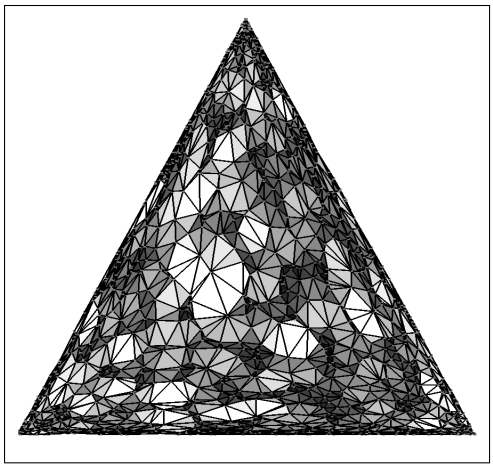
Figure 7.18: Color-coded Delaunay triangulations of the same point set as used in Figure 7.12 for (a) Shannon, (b) conjugate Shannon, (c) Fisher, (d) Euclidean, and (e) weighted Euclidean geometry. Triangles with a smaller function value have a darker shade.



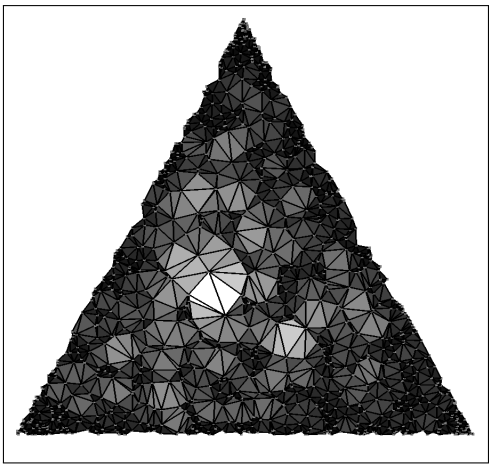
(a) Shannon.



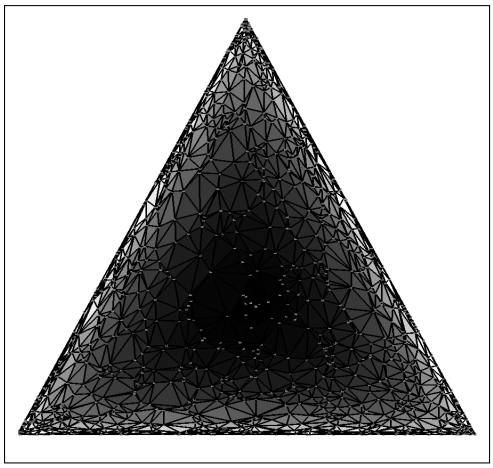
(b) Conjugate Shannon.



(c) Fisher.



(d) Euclidean.



(e) Weighted Euclidean.

Figure 7.19: Color-coded Delaunay triangulations of the same point set as used in Figure 7.14 for (a) Shannon, (b) conjugate Shannon, (c) Fisher, (d) Euclidean, and (e) weighted Euclidean geometry. Triangles with a smaller function value have a darker shade.

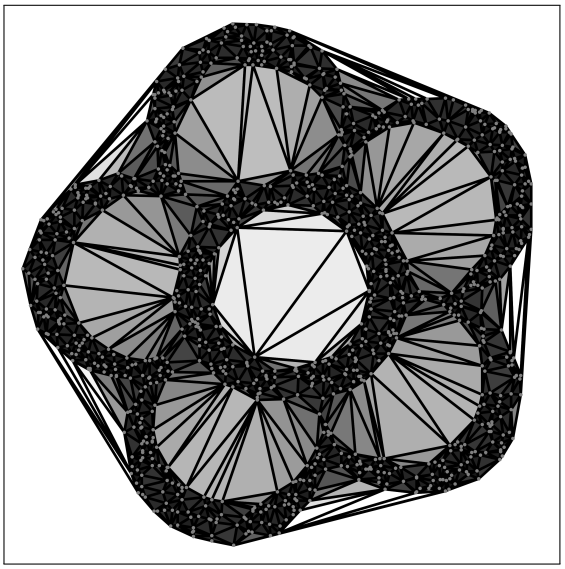
Chapter 8

Conclusion

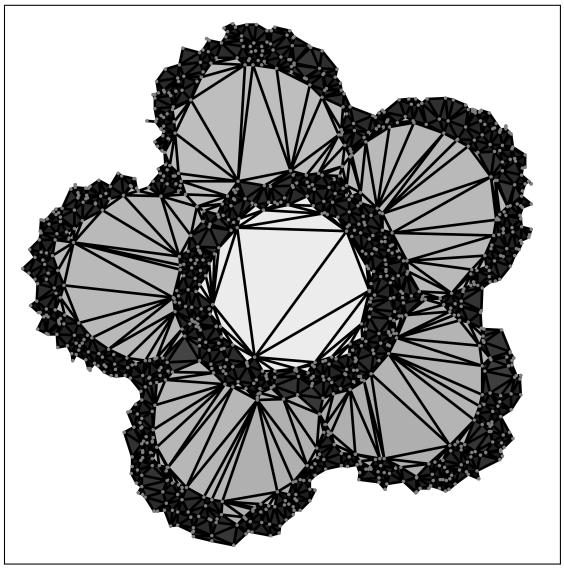
We recapitulate the results that we have obtained and the tools that we have developed in this work by illustrating them with the point set of Figure 1.1, which also shows its Delaunay triangulation and the Alpha and Wrap complex for $r = 50$.

In Chapter 3, we first stated algorithms for computing the Delaunay and Wrap radius values for the simplices of a given Delaunay triangulation and then proposed algorithms for locally updating the Wrap complex and for computing a relaxed version of it, see Figure 8.1. In Chapter 4, we presented experimentally estimated combinatorial and topological statistics of Alpha and Wrap complexes, including amongst others the numbers of simplices and Betti numbers, see Figure 8.2. In Chapter 5, we introduced the tri-partition of the p-simplices of an ordered simplicial complex and constructed canonical bases for homology and cohomology, see Figure 8.3. In Chapter 6, we presented two approaches to opening and closing holes in an ordered simplicial complex, see Figure 8.4. In Chapter 7, we extended the concepts of the Alpha and the Wrap complex to Bregman spaces and compared Delaunay triangulations in different geometries, see Figure 8.5.

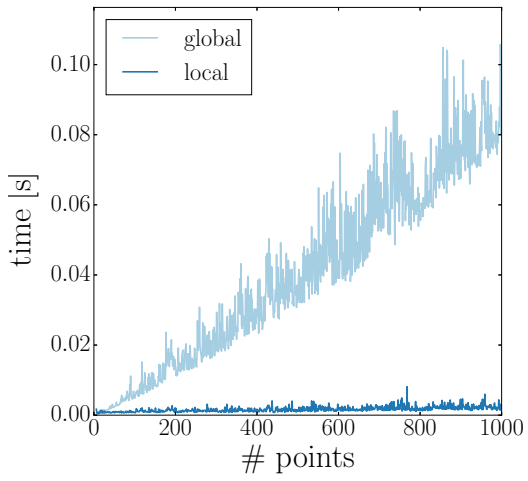
All in all, this thesis has studied the structure of ordered complexes in general and of subcomplexes of the Delaunay triangulation in particular in various ways and with different applications ranging from shape reconstruction to information theory in mind. The results presented here are only the starting point for a deeper analysis and motivate, for example, the following questions:



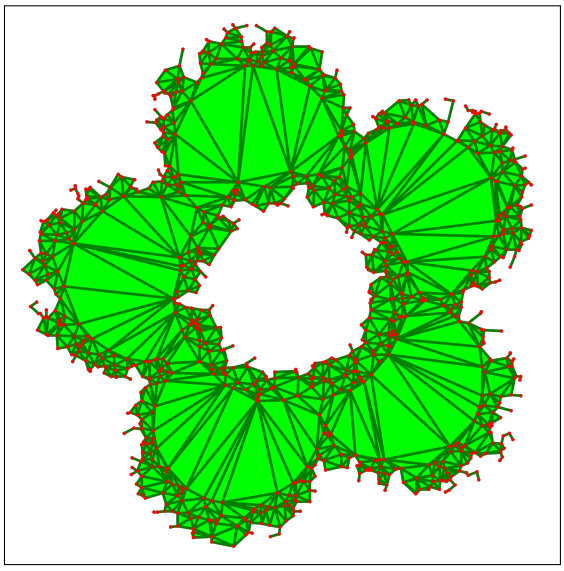
(a) Delaunay radii.



(b) Wrap radii.



(c) Dynamic updates.



(d) Relaxed Wrap complex.

Figure 8.1: *First row:* Delaunay and Wrap simplices with the shade of color depending on their radius value. Darker simplices have a smaller radius, while lighter simplices have a larger radius. *Second row:* On the *left*, we compare the times for computing the Alpha and Wrap complexes incrementally by local updates with the times for computing them from scratch in every step. On the *right*, we see the relaxed Wrap complex for radius $r = 80$ and threshold $\varepsilon = \infty$.

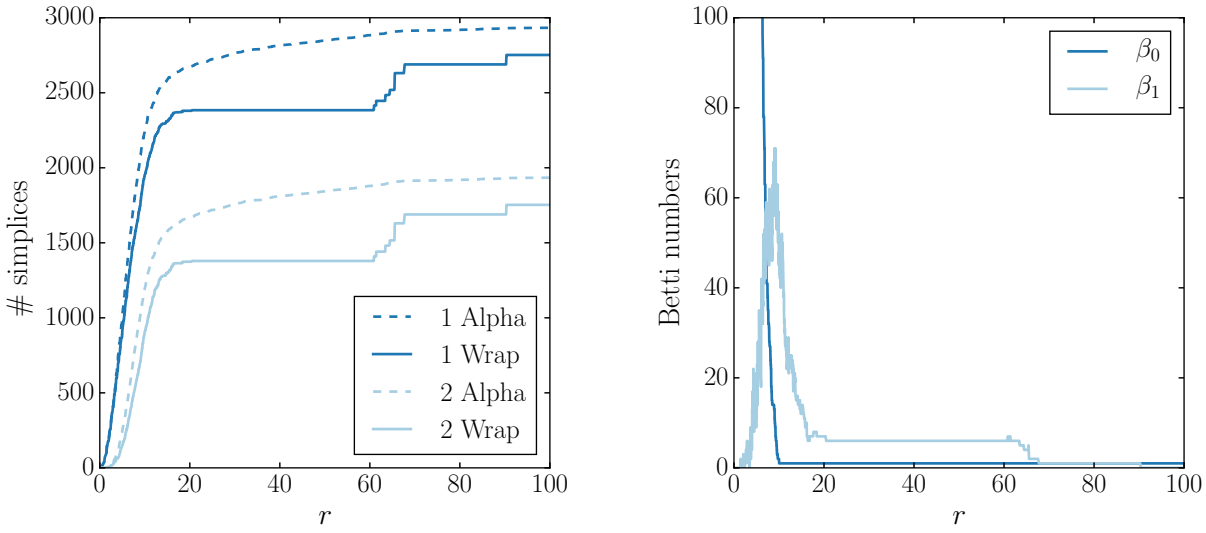


Figure 8.2: *Left:* Numbers of 1- and 2-dimensional simplices in the Alpha and Wrap complexes. *Right:* Betti numbers for the complexes.

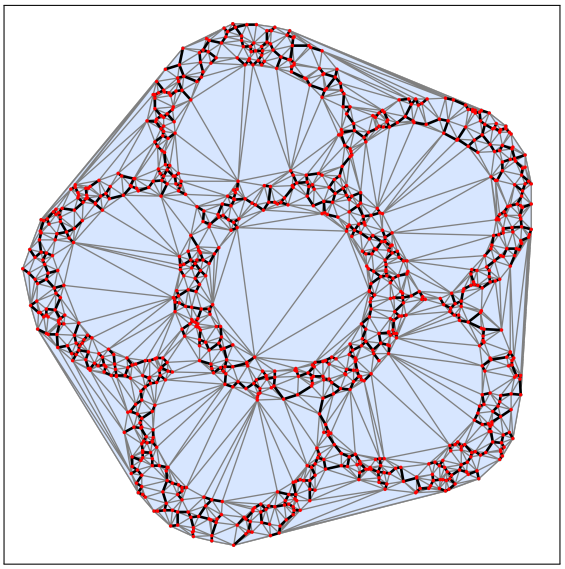
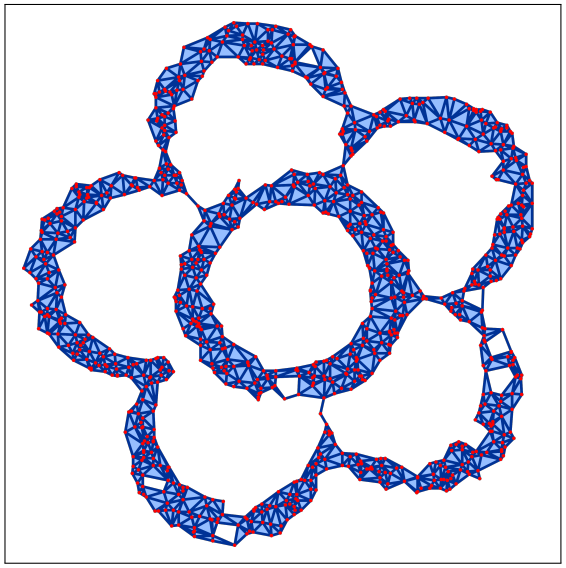
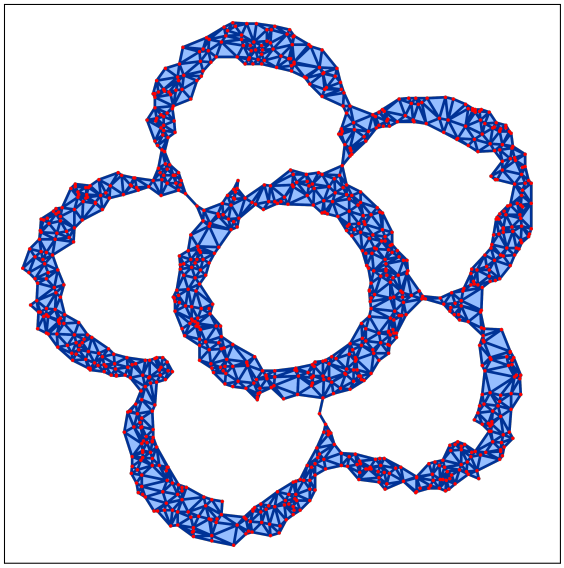


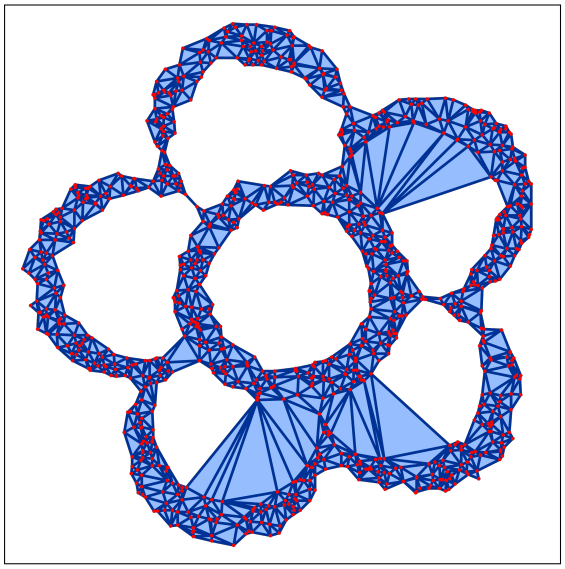
Figure 8.3: Partition of the edge set into a maximal 1-tree (black) and a maximal 1-cotree (thin gray edges), the third set is empty.



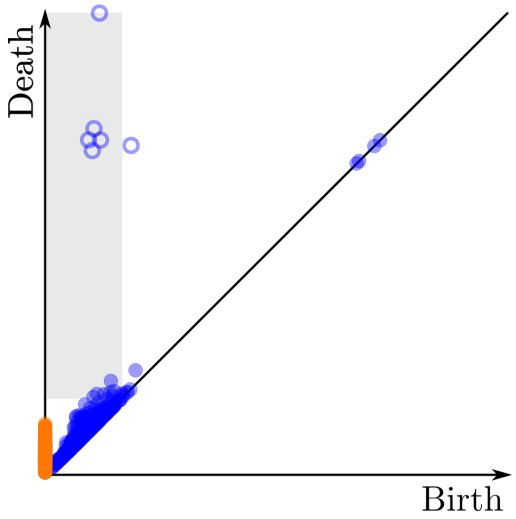
(a) Alpha complex for $r = 15$.



(b) Filling by adding canonical chains.

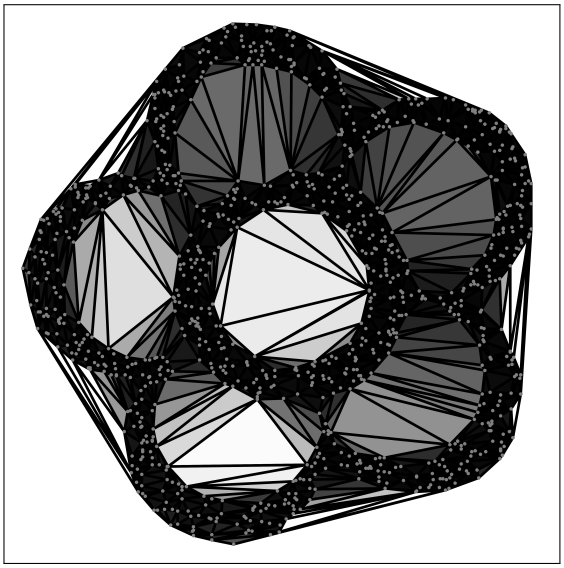


(c) Recursive filling.

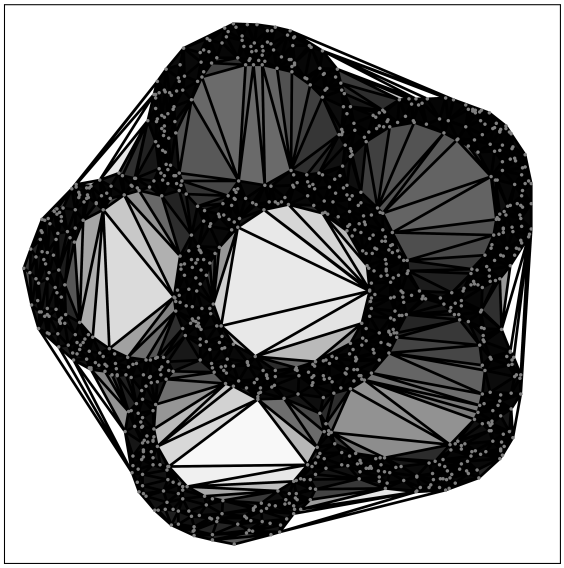


(d) Persistence diagram after recursive filling.

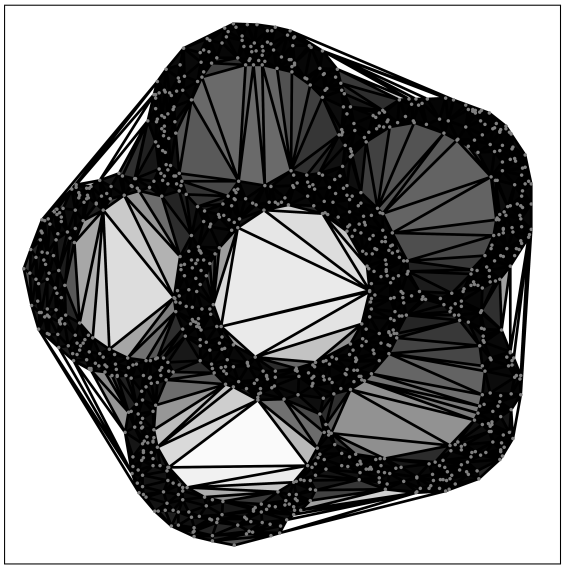
Figure 8.4: We fill the holes with persistence smaller than 10 in the Alpha complex for $r = 15$ (b) by adding canonical chains and (c) with the recursive operations respecting the dependences between holes. In the second case, we know how the status of any hole is affected and we can visualize the changes in the persistence diagram by drawing filled, empty, and dashed circles for holes in the past, presence, and future, respectively. The gray quadrant highlights the points corresponding to holes that are in the presence for the original subcomplex.



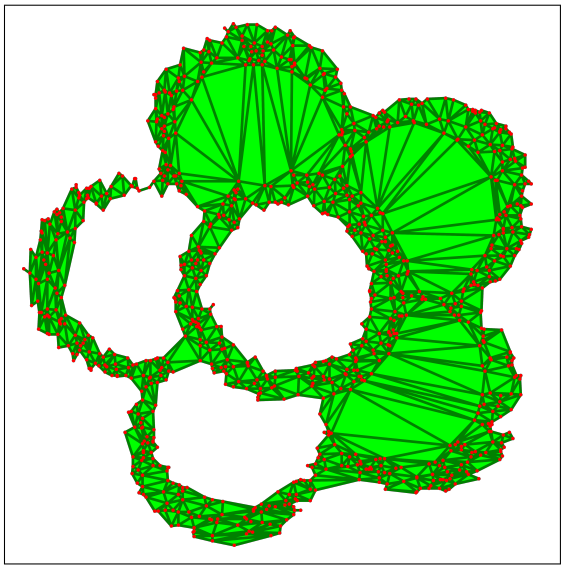
(a) Shannon geometry.



(b) Conjugate Shannon geometry.



(c) Fisher geometry.



(d) Wrap complex in Shannon geometry.

Figure 8.5: Delaunay triangulation in (a) Shannon, (b) conjugate Shannon, and (c) Fisher geometry. The colors of the simplices depend on their rise function values or squared Delaunay radii, respectively. Darker simplices have a smaller value, while lighter simplices have a larger value. (d) Wrap complex for $r = 10$ in Shannon geometry.

- We presented experimental results for the variances of the interval densities in the Alpha complexes of a Poisson point process. Can we derive analytic formulas for them? It is interesting that the expected numbers of $(1, 2)$-intervals and of $(2, 2)$-intervals are the same in $\mathbb{R}^2$, while their variances differ. Why is this the case?

- Fitting linear combinations of Gamma functions to the estimated interval densities in the Wrap complexes yielded surprisingly small errors. This suggests that the intervals might follow such a distribution. Can this be proven analytically?

- We have shown experimentally that the lower sets of critical simplices are typically small for Poisson generated points. Can we prove that their expected sizes are small?

- What can we say about the structure of the family of maximal p-trees for ordered complexes on random points? Is there a relation between the maximal p-tree and the maximal $(p + 1)$-tree?

- The dependence structure of holes of an ordered complex depends on the column-reduced matrix R. How does it change when we use different reduction algorithms or if we define it based on the row-reduced matrix Q instead? Does the dependence structure derived from Q improve the results for recursive unlocking and unfilling?

- We can use the dependence structure to decorate the persistence diagram with additional structural information. Can we use this approach to differentiate data sets with the same persistence diagram?

- Can we define a dependence structure of holes that only depends on the birth-death pairs?

- We compared the Delaunay triangulations of two random point sets in different geometries. Can we probabilistically quantify the sensitivity of the Delaunay triangulation to the dissimilarity measure?

- We computed the Delaunay triangulations in Fisher geometry and in conjugate Shannon geometry on a transformed set of points but then drew them by mapping the vertices to the original points. Does this necessarily give a geometric realization or can non-incident simplices overlap?

More generally, we ask if the combination of computational topology and generalized discrete Morse theory can help to study other questions in geometry, topology, or more distant fields. We would be especially interested in whether our methods can be successfully used for concrete applications in shape reconstruction and analysis, e.g. in the study of biomolecules or new materials.

The algorithms introduced in this paper were all integrated in a software for computing, analyzing, and manipulating 2- and 3-dimensional Alpha and Wrap complexes, called `wrap_2_3`. It can be downloaded from

- `https://git.ist.ac.at/katharina.oelsboeck/wrap_2_3-public.git`, or
- `https://bitbucket.org/koelsboe/wrap_2_3-public.git`.

Our implementations are in C++, for parts of the computation we use the CGAL library [10].

Bibliography

[1] Peter L. Antonelli et al. The geometry of random drift I-VI. *Advances in Applied Probability*, 9-12, 1977-80.

[2] Ulrich Bauer and Herbert Edelsbrunner. The Morse theory of Čech and Delaunay complexes. *Trans. Amer. Math. Soc.*, 369(5):3741–3762, 2017.

[3] Heinz Bauschke and Jonathan Borwein. Legendre functions and the method of random Bregman projections. *Journal of Convex Analysis*, 4, 1998.

[4] Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Joshua A. Levine, Andrei Sharf, and Claudio T. Silva. State of the art in surface reconstruction from point clouds. In *Eurographics 2014 - State of the Art Reports*. The Eurographics Association, 2014.

[5] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28:235–242, 2000. `www.rcsb.org`.

[6] Norman Biggs. Spanning trees of dual graphs. *Journal of Combinatorial Theory, Series B*, 11(2):127–131, 1971.

[7] Jean-Daniel Boissonnat, Olivier Devillers, and Samuel Hornus. Incremental construction of the Delaunay triangulation and the Delaunay graph in medium dimension. In *Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry*, SCG '09, pages 208–216, New York, NY, USA, 2009. ACM.

[8] Jean-Daniel Boissonnat, Frank Nielsen, and Richard Nock. Bregman Voronoi diagrams. *Discrete Comput. Geom.*, 44(2):281–307, 2010.

[9] Frédéric Cazals and Joachim Giesen. Delaunay triangulation based surface reconstruction. In *Effective Computational Geometry for Curves and Surfaces*, pages 231–276. Springer, Berlin, Heidelberg, 2006.

[10] The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 4.14 edition, 2019. `www.cgal.org`.

[11] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Proceedings of the Twenty-second Annual Symposium on Computational Geometry*, SCG '06, pages 119–126, New York, NY, USA, 2006. ACM.

[12] Vin de Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in persistent (co)homology. *Inverse Problems*, 27, 2011.

[13] Cecil Jose A. Delfinado and Herbert Edelsbrunner. An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design*, 12(7):771–784, 1995.

[14] Olivier Devillers. On deletion in Delaunay triangulations. *International Journal of Computational Geometry and Applications*, 12:193–205, 2002.

[15] Olivier Devillers, Stefan Meiser, and Monique Teillaud. Fully dynamic Delaunay triangulation in logarithmic expected time per operation. *Comput. Geom. Theory Appl.*, 2(2):55–80, 1992.

[16] Beno Eckmann. Harmonische Funktionen und Randwertaufgaben in einem Komplex. *Commentarii mathematici Helvetici*, 17:240–255, 1945.

[17] Herbert Edelsbrunner. The union of balls and its dual shape. *Discrete & Computational Geometry*, 13(3):415–440, 1995.

[18] Herbert Edelsbrunner. *Geometry and topology for mesh generation*, volume 7 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2001.

[19] Herbert Edelsbrunner. Surface reconstruction by wrapping finite sets in space. In *Discrete and Computational Geometry*, pages 379–404. Springer, 2003.

[20] Herbert Edelsbrunner and John L. Harer. *Computational topology.* American Mathematical Society, Providence, RI, 2010.

[21] Herbert Edelsbrunner, Anton Nikitenko, Katharina Ölsböck, and Peter Synak. Poisson-Delaunay mosaics and related complexes experimentally. In *Proceedings of the Abel Symposium 2018*, to appear.

[22] Herbert Edelsbrunner, Anton Nikitenko, and Matthias Reitzner. Expected sizes of Poisson-Delaunay mosaics and their discrete Morse functions. *Advances in Applied Probability*, 49(3):745–67, 2017.

[23] Herbert Edelsbrunner and Katharina Ölsböck. Holes and dependences in an ordered complex. *Computer Aided Geometric Design*, 73:1–15, 2019.

[24] Herbert Edelsbrunner and Katharina Ölsböck. Tri-partitions and bases of an ordered complex. *Discrete and Computational Geometry*, to appear.

[25] Herbert Edelsbrunner, Katharina Ölsböck, and Hubert Wagner. Shape reconstruction in information space. Manuscript, IST Austria, 2019.

[26] Herbert Edelsbrunner and Nimish R. Shah. Incremental topological flipping works for regular triangulations. *Algorithmica*, 15(3):223–241, 1996.

[27] Herbert Edelsbrunner, Žiga Virk, and Hubert Wagner. Topological data analysis in information space. In *35th International Symposium on Computational Geometry, SoCG 2019, June 18-21, 2019, Portland, Oregon, USA*, pages 31:1–31:14, 2019.

[28] Herbert Edelsbrunner and Hubert Wagner. Topological data analysis with Bregman divergences. In *33rd International Symposium on Computational Geometry, SoCG 2017, July 4-7, 2017, Brisbane, Australia*, pages 39:1–39:16, 2017.

[29] Herbert Edelsbrunner and Afra Zomorodian. Computing linking numbers of a filtration. *Homology Homotopy Appl.*, 5(2):19–37, 2003.

[30] Massimo Ferri. Persistent topology for natural data analysis - a survey. In *BIRS-IMLKE*, 2015.

[31] Robin Forman. Morse theory for cell complexes. *Advances in Mathematics*, 134(1):90–145, 1998.

[32] Ragnar Freij. Equivariant discrete Morse theory. *Discrete Mathematics*, 309(12):3821–3829, 2009.

[33] Leonidas J. Guibas, Donald E. Knuth, and Micha Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica*, 7(1):381–413, 1992.

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

[35] H. Helmholtz. Über Integrale der hydrodynamischen Gleichungen, welche den Wirbelbewegungen entsprechen. *Journal für die reine und angewandte Mathematik*, 55:25–55, 1858.

[36] W.V.D. Hodge. *The theory and applications of harmonic integrals*. Cambridge Univ. Press, Cambridge, 1941.

[37] Alireza Khatamian and Hamid Arabnia. Survey on 3d surface reconstruction. *Journal of Information Processing Systems*, 12:338–357, 2016.

[38] Yongjin Lee, Senja D. Barthel, Pawel Dlotko, S. Mohamad Moosavi, Kathryn Hess, and Berend Smit. Quantifying similarity of pore-geometry in nanoporous materials. *Nature Communications*, 8:1–8, 2017.

[39] R.E. Miles. On the homogeneous planar Poisson point process. *Mathematical Biosciences*, 6:85–127, 1970.

[40] Edgar A. Ramos and Bardia Sadri. Geometric and topological guarantees for the WRAP reconstruction algorithm. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1086–1095, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[41] Vanessa Robins. Betti number signatures of homogeneous Poisson point processes. *Phys. Rev. E*, 74:061107, 2006.

[42] P. Rosenstiehl and R.C. Read. On the principal edge tripartition of a graph. Advances in graph theory (Cambridge Combinatorial Conf., Trinity College, Cambridge, 1977). *Annals of Discrete Mathematics*, 3:195–226, 1978.

[43] Peer Stelldinger. Topologically correct surface reconstruction using Alpha shapes and relations to ball-pivoting. In *Proceedings - International Conference on Pattern Recognition*, pages 1–4, 2008.

[44] Kokichi Sugihara. Laguerre Voronoi diagram on the sphere. *Journal for Geometry and Graphics Volume*, 6:69–81, 2002.