Computational Homology Applied to Discrete Objects

Aldo Gonzalez-Lorenzo

AIX-MARSEILLE UNIVERSITÉ | UNIVERSIDAD DE SEVILLA

École Doctorale en Mathématiques et Informatique de Marseille

Instituto de Matemáticas Universidad de Sevilla

THÈSE DE DOCTORAT
en Informatique

TESIS DOCTORAL
en Matemáticas

Thèse présentée pour obtenir le grade universitaire de docteur
Memoria presentada para optar al título de Doctor

Aldo GONZALEZ LORENZO

Computational Homology Applied to Discrete Objects

Soutenue le 24/11/2016 devant le jury :

Massimo FERRI	Università di Bologna	Rapporteur
Jacques-Olivier LACHAUD	Université de Savoie	Rapporteur
Pascal LIENHARDT	Université de Poitiers	Examinateur
Aniceto MURILLO	Universidad de Málaga	Examinateur
Jean-Luc MARI	Aix-Marseille Université	Directeur de thèse
Alexandra BAC	Aix-Marseille Université	Directeur de thèse
Pedro REAL	Universidad de Sevilla	Directeur de thèse

"The scientists of today think deeply instead of clearly. One must be sane to think clearly, but one can think deeply and be quite insane."

Nikola Tesla

Abstract

Computational Homology Applied to Discrete Objects

Homology theory formalizes the concept of hole in a space. For a given subset of the Euclidean space, we define a sequence of homology groups, whose ranks are considered as the number of holes of each dimension. Hence, β_0, the rank of the 0-dimensional homology group, is the number of connected components, β_1 is the number of tunnels or handles and β_2 is the number of cavities. These groups are computable when the space is described in a combinatorial way, as simplicial or cubical complexes are. Given a discrete object (a set of pixels, voxels or their analog in higher dimension) we can build a cubical complex and thus compute its homology groups.

This thesis studies three approaches regarding the homology computation of discrete objects. First, we introduce the *homological discrete vector field*, a combinatorial structure which generalizes the discrete gradient vector field and allows us to compute the homology groups. This notion allows us to see the relation between different existing methods for computing homology. Next, we present a linear algorithm for computing the Betti numbers of a 3D cubical complex, which can be used for binary volumes. Finally, we introduce two measures (the *thickness* and the *breadth*) associated to the holes in a discrete object, which provide a topological and geometric signature more interesting than only the Betti numbers. This approach provides also some heuristics for localizing holes, obtaining minimal homology or cohomology generators, opening and closing holes.

∗

Homologie algorithmique pour les objets discrets

La théorie de l'homologie formalise la notion de trou dans un espace. Pour un sous-ensemble de l'espace Euclidien, on définit une séquence de groupes d'homologie, dont leurs rangs sont interprétés comme le nombre de trous de chaque dimension. Ainsi, β_0, le rang du groupe d'homologie de dimension zéro, est le nombre de composantes connexes, β_1 est le nombre de tunnels ou anses et β_2 est le nombre de cavités. Ces groupes sont calculables quand l'espace est décrit d'une façon combinatoire, comme c'est le cas pour les complexes simpliciaux ou cubiques. À partir d'un objet discret (un ensemble de pixels, voxels ou leur analogue en dimension supérieure) nous pouvons construire un complexe cubique et donc calculer ses groupes d'homologie.

Cette thèse étudie trois approches relatives au calcul de l'homologie sur des objets discrets. En premier lieu, nous introduisons le *champ de vecteurs discret homologique*, une structure combinatoire généralisant les champs de vecteurs gradients discrets, qui permet de calculer les groupes d'homologie. Cette notion permet de voir la relation entre plusieurs méthodes existantes pour le calcul de l'homologie et révèle également des notions subtiles associées. Nous présentons ensuite un algorithme linéaire pour calculer les

nombres de Betti dans un complexe cubique 3D, ce qui peut être utilisé pour les volumes binaires. Enfin, nous présentons deux mesures (l'*épaisseur* et la *largeur*) associées aux trous d'un objet discret, ce qui permet d'obtenir une signature topologique et géométrique plus intéressante que les simples nombres de Betti. Cette approche fournit aussi quelques heuristiques permettant de localiser les trous, d'obtenir des générateurs d'homologie ou de cohomologie minimaux, d'ouvrir et de fermer les trous.

$$*$$

Homología computacional para los objetos discretos

La teoría de la homología formaliza la noción de agujero en un espacio. Dado un subconjunto del espacio Euclídeo, se define una secuencia de grupos de homología, cuyos rangos se consideran el número de agujeros de cada dimensión. Así, β_0, el rango del grupo de homología de dimensión 0, es el número de componentes conexas, β_1 es el número de túneles o asas y β_2 es el número de cavidades. Estos grupos son calculables cuando el espacio es descrito de manera combinatoria, como ocurre con los complejos simpliciales o cúbicos. También, dado un objeto discreto (un conjunto de píxeles, vóxeles o elementos de dimensión superior), podemos construir un complejo cúbico y así calcular sus grupos de homología.

Esta tesis estudia tres enfoques relativos al cálculo de la homología en los objetos discretos. En primer lugar, introducimos el *campo de vectores discreto homológico*, una estructura combinatoria que generaliza el campo de vectores gradiente discreto y que permite calcular los grupos de homología. Este concepto permite ver la relación entre varios métodos existentes para el cálculo de la homología. Posteriormente presentamos un algoritmo lineal para calcular los números de Betti de un complejo cúbico 3D, y por tanto, de un volumen binario. Por último introducimos dos medidas (el *espesor* y la *amplitud*) asociadas a los agujeros de un objeto discreto, las cuales proporcionan una firma topológica y geométrica más interesante que simplemente los números de Betti. El cálculo de estas medidas además también aporta unas heurísticas para localizar los agujeros, obtener generadores de homología o cohomología mínimos, abrir o cerrar agujeros.

French Extended Abstract – Résumé étendu

La théorie de l'homologie formalise la notion de trou dans un espace. Pour un sous-ensemble de l'espace Euclidien, on définit une séquence de groupes d'homologie, dont leurs rangs sont interprétés comme le nombre de trous de chaque dimension. Ainsi, β_0, le rang du groupe d'homologie de dimension zéro, est le nombre de composantes connexes, β_1 est le nombre de tunnels ou anses et β_2 est le nombre de cavités. Ces notions peuvent aussi être définies pour des dimensions supérieures, mais il n'y a plus d'intuition géométrique pour elles. Les groupes d'homologie sont calculables quand l'espace est décrit d'une façon combinatoire, comme c'est le cas pour les complexes simpliciaux ou cubiques. Puisqu'un objet discret (un ensemble de pixels, voxels ou leur analogue en dimension supérieure) peut être transformé en complexe cubique, nous pouvons aussi calculer ses groupes d'homologie.

Cette thèse étudie trois approches relatives au calcul de l'homologie sur des objets discrets.

Le champ de vecteurs discret homologique

Le *champ de vecteurs discret homologique* (abrégé HDVF en anglais) est introduit au Chapitre 3. Le HDVF est une structure combinatoire définie sur un CW-complexe fini (notion généralisant les complexes simpliciaux et cubiques entre autres) qui induit une *réduction* (cf. Section 2.3.4), donc nous pouvons déduire ses nombres de Betti, un ensemble de générateurs de homologie ou cohomologie, etc.

Étant donné un CW-complexe K, un HDVF est un pair d'ensembles disjoints de ses cellules $X = (P, S)$ tels que la restriction de la matrice du bord sous ces deux ensembles (c'est-à-dire, la sous-matrice avec les colonnes correspondantes aux cellules de S et les lignes correspondantes aux cellules de P) est inversible. Nous démontrons (cf. Theorem 3.9) qu'un HDVF induit une réduction.

Nous étudions ensuite comment calculer efficacement un HDVF avec sa réduction. En nous appuyant sur des formules connues du calcul matriciel, nous trouvons que la meilleure option consiste à ajouter les cellules dans le HDVF par couples (dont une cellule est ajoutée à P et l'autre à S) et mettre à jour la réduction à chaque étape. Nous évitons ainsi toute inversion de matrice ou calcul du déterminant. Nous déduisons alors que le calcul d'un HDVF et sa réduction a une complexité $\mathcal{O}(n^3)$.

Nous introduisons après cinq opérations basiques pour transformer un HDVF. Bien que la notion de HDVF soit inspirée de la théorie discrète de Morse et que certains concepts sont purement des généralisations de cette théorie, nous remarquons que ces opérations sont nouvelles puisqu'elles sont basées sur le formalisme du HDVF.

La section suivante est dédiée à l'étude de la relation entre le HDVF et d'autres méthodes d'homologie algorithmique. Nous déduisons que le HDVF généralise le champ de vecteur gradient discret (discrete gradient vector field) et le champ de vecteur gradient discret itéré (iterated discrete gradient vector field). Nous démontrons aussi comment le calcul de la forme normale de Smith (la méthode classique pour calculer les nombres de Betti) est équivalent au calcul d'un HDVF. Par conséquence, on peut calculer l'homologie persistante avec un HDVF en utilisant les opérations basiques.

Nous finissons cette partie avec une étude expérimentale (et non théorique) de la complexité du calcul du HDVF. Bien que nous estimons sa complexité comme $\mathcal{O}(n^3)$, nous apprécions qu'elle est en moyenne $\mathcal{O}(n^2)$ ou même $\mathcal{O}(n^{1.4})$ si nous calculons seulement le HDVF sans sa réduction.

Calcul rapide des nombres de Betti sur un complexe cubique 3D

Cette partie est le fruit d'une étroite collaboration avec Mateusz Juda. Nous introduisons un algorithme efficient pour calculer les nombres de Betti sur un complexe cubique 3D. Cet algorithme est essentiellement basé sur le calcul du nombre de composantes connexes dans deux graphes, donc sa complexité est linéaire.

Calculer les groupes d'homologie d'un CW-complexe de dimension quelconque requiert des techniques générales, telles que la méthode basée sur le calcul de la forme normale de Smith ou le HDVF. Par contre, si le CW-complexe est un complexe cubique 3D, il existe une astuce pour calculer ses nombres de Betti s'il est un complexe cubique 3D. Soit K un tel complexe, nous démontrons (cf. Proposition 4.3) que son nombre de Betti de dimension 0, $\beta_0(K)$, est le nombre de composantes connexes d'un certain graphe défini sur les cellules de dimension 0 et 1 de K. Nous prouvons ensuite qu'il existe une relation entre les nombres de Betti de K et les nombres de Betti de son complémentaire dans un sur-complexe acyclique (cf. Proposition 4.4). Ceci permet de démontrer que $\beta_2(K)$ est le nombre de composantes connexes d'un certain graphe défini sur les cellules de dimension 2 et 3 de $L - K$, où L est un sur-complexe de K acyclique. Ces résultats formalisent l'idée intuitive que $\beta_0(K)$ est le nombre composantes connexes de K et $\beta_2(K)$, le nombre de cavités ou composantes connexes bornées de son complémentaire. Ces propositions sont toutes démontrées en s'appuyant sur le formalisme des HDVFs.

Étant donné que $\beta_0(K)$ et $\beta_2(K)$ peuvent être obtenus en comptant des composantes connexes, on peut déduire $\beta_1(K)$ grâce à la formule d'Euler-Poincaré. Nous proposons une approche simple et itérative pour calculer ses quantités en utilisant l'algorithme classique pour compter des composantes connexes avec un parcours en profondeur. Ensuite nous introduisons un algorithme récursif avec une technique *diviser pour régner* qui permet de paralléliser partiellement le calcul du nombre de composantes connexes. Cette approche est spécialement conçue pour les complexes cubiques et n'est donc pas valable pour des complexes simpliciaux.

Nous comparons notre implémentation avec la bibliothèque CAPD::RedHom et nous montrons que nous obtenons de meilleurs temps d'exécution ainsi qu'elle permet de traiter des complexes plus grands grâce à ses moindres besoins de mémoire.

Mesurer les trous

Nous introduisons dans ce chapitre deux mesures « pour les trous » d'un objet discret : l'*épaisseur* et la *largeur*. Soit O un objet discret (de dimension quelconque), nous définissons une *filtration* (cf. Section 2.3.5) basée sur la transformée de distances signée de l'objet sur son complexe cubique associé. En prenant un sous-ensemble des points du diagramme de persistance de cette filtration, nous obtenons un pair de valeurs pour chaque trou de l'objet. Autrement dit, nous obtenons $\beta_q(O)$ pairs, pour tout $q \geq 0$. Nous obtenons ainsi une information géométrique supplémentaire sur les nombres de Betti d'un objet, ce qui permet de comprendre mieux sa topologie.

La définition de l'épaisseur et de la largeur est donnée dans Definition 5.1. Notons que cette définition dépend de la distance considérée. De plus, on peut considérer deux types différents de complexe cubique associé a un objet, un lié à la $2n$-connectivité, l'autre à la $(3^n - 1)$-connectivité. Nous démontrons ensuite que ces mesures sont robustes (cf. Theorem 5.2).

La suite de ce chapitre étudie différentes applications de ces mesures.

Nous montrons comment les mesures peuvent être représentées par des boules à l'intérieur ou l'extérieur de l'objet qui indiquent d'une certain façon la position des trous. Nous montrons l'utilité de cette approche avec plusieurs exemples.

Nous introduisons aussi deux algorithmes basés sur les mesures qui produisent des générateurs d'homologie et de cohomologie. Ces générateurs semblent être minimales en pratique, bien que nous pouvons trouver des exceptions.

Nous définissons par la suite l'ouverture et la fermeture d'un trou (cf. Definition 5.2). Ces notions formalisent l'idée de qu'on peut éliminer un trou en ajoutant ou enlevant de la matière. Nous présentons alors un algorithme pour ouvrir les trous (cf. Algorithm 8) et une heuristique pour les fermer (cf. Algorithm 9).

Spanish Extended Abstract – Resumen extendido

La teoría de la homología formaliza la noción de agujero en un espacio. Dado un subconjunto del espacio Euclídeo, se define una secuencia de grupos de homología, cuyos rangos se consideran el número de agujeros de cada dimensión. Así, β_0, el rango del grupo de homología de dimensión 0, es el número de componentes conexas, β_1 es el número de túneles o asas y β_2 es el número de cavidades. Estos números pueden definirse en dimensiones superiores, pero carecen de interpretación geométrica clara. Los grupos de homología son calculables cuando el espacio es descrito de manera combinatoria, como ocurre con los complejos simpliciales o cúbicos. También, dado un objeto discreto (un conjunto de píxeles, vóxeles o elementos de dimensión superior), podemos construir un complejo cúbico y así calcular sus grupos de homología.

Esta tesis estudia tres enfoques relacionados con el cálculo de la homología en objetos discretos

El campo de vectores homológico discreto

El *campo de vectores homológico discreto* (abreviado HDVF en inglés) es introducido en el Capítulo 3. El HDVF es una estructura combinatoria definida sobre un CW-complejo finito (generalización de complejo simplicial y cúbico entre otros) que induce una *reducción* (cf. Sección 2.3.4), por lo que podemos deducir sus números de Betti, un conjunto de generadores de homología o cohomología, etc.

Dado un CW-complejo K, un HDVF es un par de conjuntos disjuntos de células $X = (P, S)$ tales que la restricción de la matriz del operador frontera a estos dos conjuntos (es decir, la submatriz con las columnas correspondientes a las células de S y las filas correspondientes a las células de P) es invertible. El Teorema 3.9 muestra que un HDVF induce una reducción.

A continuación estudiamos cómo calcular eficazmente un HDVF con su reducción. Basándonos en varias fórmulas conocidas del cálculo matricial, encontramos que la mejor opción consiste en añadir las células al HDVF por parejas (una célula en P y la otra en S) y actualizar la reducción en cada paso. Así deducimos que podemos calcular un HDVF y su reducción asociada en $\mathcal{O}(n^3)$.

Después presentamos cinco operaciones básicas para transformar un HDVF. Aunque el concepto de HDVF esté inspirado en la teoría discreta de Morse y que ciertas ideas sean simples generalizaciones de esta teoría, queremos destacar que estas operaciones son nuevas dado que están basadas en el formalismo del HDVF.

La siguiente sección está dedicada al estudio de la relación entre el HDVF y otros métodos de la homología computacional. Deducimos que el HDVF generaliza el campo vectorial gradiente discreto (discrete gradient vector

field) y el campo vectorial gradiente discreto iterado (iterated gradient vector field). También demostramos cómo el calculo de la forma normal de Smith (el método clásico para calcular los números de Betti) es equivalente a calcular un HDVF. Por tanto, podemos calcular la homología persistente con un HDVF utilizando las operaciones básicas introducidas anteriormente.

Terminamos esta parte con un estudio experimental (que no teórico) de la complejidad del calcular un HDVF. Aunque hayamos visto que la complejidad es $\mathcal{O}(n^3)$, podemos apreciar que en promedio es $\mathcal{O}(n^2)$ o incluso $\mathcal{O}(n^{1,4})$ si solo calculamos el HDVF sin su reducción.

Cálculo rápido de los números de Betti de un complejo cúbico 3D

Este capítulo es fruto de una estrecha colaboración con Mateusz Juda. Introducimos un algoritmo eficiente para calcular los números de Betti de un complejo cúbico 3D. Este algoritmo se basa principalmente en el conteo de componentes conexas de dos grafos, por lo que su complejidad es lineal.

Calcular los grupos de homología de un CW-complejo de dimensión cualquiera requiere técnicas generales, tales como el método basado en el cálculo de la forma normal de Smith ou el HDVF. Sin embargo, si el CW-complejo es un complejo cúbico 3D, hay otra manera de calcular sus números de Betti. Sea K el dicho complejo, probamos (cf. Proposición 4.3) que su número de Betti de dimensión 0, $\beta_0(K)$, es el número de componentes conexas de un grafo definido sobre las células de dimensión 0 y 1 de K. A continuación demostramos que existe una relación entre los números de Betti de K y de su complementario en un super-complejo acíclico (cf. Proposición 4.4). Esto permite probar que $\beta_2(K)$ es el número de componentes conexas de un grafo definido sobre las células de dimensión 2 y 3 de $L - K$, para todo L super-complejo acíclico de K. Estos resultados formalizan la idea intuitiva de que $\beta_0(K)$ es el número de componentes conexas de K y $\beta_2(K)$, el número de cavidades o componentes conexas acotadas de su complementario. Todas estas proposiciones son demostradas usando el formalismo de los HDVFs.

Dado que $\beta_0(K)$ y $\beta_2(K)$ pueden ser calculados contando componentes conexas, podemos deducir $\beta_1(K)$ gracias a la fórmula de Euler-Poincaré. Proponemos un método simple e iterativo para calcular estas cantidades utilizando el algoritmo clásico para contar componentes conexas mediante una búsqueda en anchura. A continuación presentamos un algoritmo recursivo con una técnica *divide y vencerás* que permite paralelizar parcialmente el cálculo del número de componentes conexas. Este método está concebido especialmente para los complejos cúbicos y no es válido para los complejos simpliciales.

Comparamos nuestra implementación con la biblioteca CAPD::RedHom y mostramos que no solo obtenemos mejores tiempos de ejecución sino que además podemos procesar complejos mas grandes gracias a un menor uso de memoria

Medir los agujeros

En este capítulo presentamos dos medidas "para los agujeros"de un objeto discreto: el *espesor* y la *amplitud*. Sea O un objeto discreto (de dimensión

cualquiera), definimos una *filtración* (cf. Sección 2.3.5) basada en la transformada de la distancia con signo del objeto sobre su complejo cúbico asociado. Tomando un subconjunto de los puntos del diagrama de persistencia de esta filtración, obtenemos un par de valores por cada agujero del objeto. Dicho de otra manera, obtenemos $\beta_q(O)$ pares para cada $q \geq 0$. De este modo obtenemos una información geométrica suplementaria sobre los números de Betti de un objeto, lo que permite entender mejor su topología.

La definición del espesor y la amplitud se encuentra en la Definición 5.1. Nótese que esta definición depende de la distancia considerada. Además, podemos considerar dos tipos de complejo cúbico asociado al objeto, uno relacionado a la $2n$-conectividad y el otro a la $(3^n - 1)$-conectividad. Seguidamente demostramos que estas medidas son robustas ante la presencia de ruido (cf. Teorema 5.2).

El resto de este capítulo estudia varias aplicaciones de estas medidas.

Mostramos cómo las medidas puedes ser representadas con bolas en el interior o el exterior del objeto que indican de cierto modo la posición de los agujeros. La utilidad de esta aplicación es demostrada con varios ejemplos.

También presentamos dos algoritmos basados en las medidas que producen generadores de homología y cohomología. Estos generadores parecen mínimos en la práctica, aunque podemos encontrar excepciones.

Posteriormente definimos la apertura y el cierre de un agujero (cf. Definición 5.2). Estos conceptos formalizan la idea de que un agujero puede ser eliminado añadiendo o quitado materia. Para ello, introducimos un algoritmo para abrir los agujeros (cf. Algoritmo 8) y una heurística para cerrarlos (cf. Algoritmo 9).

Acknowledgements

I would like to thank my advisors Jean-Luc Mari, Alexandra Bac and Pedro Real for their guidance, encouragement, and for providing me the freedom to work on the topics that interested me most. I am also indebted to many colleagues in the G-Mod team for their help in so many technical problems: Arnaud Polette, Joris Ravaglia, Jules Morel, Eric Remy and Romain Raffin to cite a few.

I also want to thank all those colleagues that I have met—personally or virtually—during these three years with whom I have had so many interesting discussions: Erin Chambers, Frédéric Chazal, Guillaume Damiand, Paweł Dłotko, Herbert Edelsbrunner, Laurent Fuchs, Tao Ju, Lu Liu, Clément Maria, Patrick Min, Vidit Nanda, Steve Oudot, Sophie Viseur and the DGtal team. I am particularly grateful to Mateusz Juda, whose collaboration led to successful results that I could otherwise never have imagined.
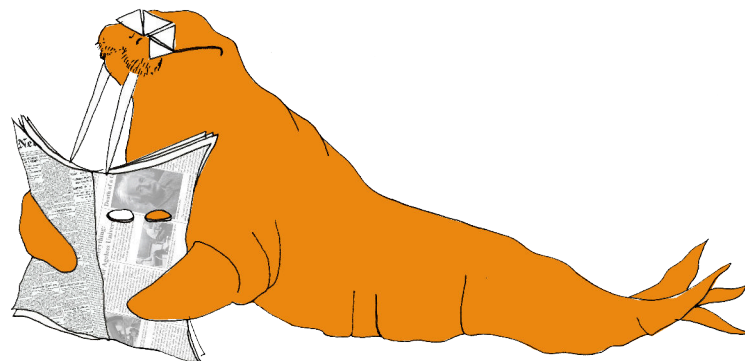
I thank my friends, flatmates, family and my love, Apolline, who always supported me and, in exchange, now know a little more about holes.

Contents

For David Robert Jones.

Chapter 1

Introduction

1.1 Overview

Understanding an object can be addressed by determining its volume, its convexity, its curvature, its medial axis or any other geometric descriptor. A higher level analysis can be made through topology, which tolerates continuous deformations. This could be seen as a less interesting approach, since we could not distinguish a coffee mug from a donut, but it actually provides a more essential information of the object. Homology is a powerful tool as its formalizes the concept of *hole* in algebraic terms.

One of the clearest concepts in topology is that of *connectivity*. It tells us how many disjoint parts there are in an object. Let us introduce this notion for a simple family of spaces: graphs. Consider a graph $G = (V, E)$. We recall that a *path* is a sequence of incident edges $[e_1, e_2, \ldots, e_r]$, that is, each pair of consecutive edges share a vertex. Then, two vertices $u, v \in V$ are said to be connected if there is a *path* connecting them, namely $[\{u, w_1\}, \{w_1, w_2\}, \cdots, \{w_r, v\}]$. Being connected is an equivalence relation, so we can define classes and the quotient set. Given a vertex $u \in V$, the class $[u]$ is the set of all vertices which are connected to u, that is, the *connected component* containing u. Thus, the quotient set under this relation is the collection of the connected components of the graph, and its cardinal is the number of connected components.

Homology theory extrapolates this concept to higher dimensions. Instead of graphs, we consider *simplicial complexes* (or any similar structure such as cubical complexes or CW complexes). Next, we define a homology group H_q for each dimension $q \geq 0$. Let us assume that our simplicial complex is embedded in $\mathbb{R}^3$. Then its homology groups are isomorphic to free groups of the form $\mathbb{Z}^\beta$, so their rank is β. For each dimension $q \geq 0$, the rank of H_q—called β_q, the q-th Betti number—is the number of q-holes.

We can interpret the 0-holes as connected components, so β_0 (as the cardinal of the previously defined quotient set) is the number of connected components. The 1-holes correspond to tunnels or handles and the 2-holes, to cavities or voids. It is straightforward to generalize these notions to higher dimensions, though we lose geometric intuition. For instance, a hollow square has one 1-hole (a handle) and a hollow cube has one 2-hole (a void), so a hollow four-dimensional cube contains a 3-hole, even if we cannot conceive what it is. Also, if the simplicial complex is embedded in a higher-dimensional space, like the Klein bottle, its homology groups can contain a torsion subgroup, which reveals the presence of "strange" holes.

*

Homology groups are computable, so given a space with a finite and combinatorial description (such as a simplicial complex), we can figure out how many holes of each dimension it has. Moreover, we can even "draw" the holes, although this presents many disadvantages. Consequently, homology allows us to compare and understand objects with regard to their topology, that is, their holes.

This theory can be considered to have begun with the Euler-Poincaré characteristic during the 18th century. However, its practical applications have not been exploited until the last twenty years due to its computational complexity. There are applications in dynamical systems [89, 93], material science [27, 112], electromagnetism [62, 40], geometric modeling [43], image understanding [2, 52, 96, 30] and sensor networks [38]. The general idea is to use homology to analyze and understand high dimensional structures in a rigorous way. *Persistent homology* has revolutionized these applications, and more than likely a second revolution will come with the *zigzag persistent homology*.

<div align="center">✳</div>

The content of this thesis is organized into three main chapters:

The homological discrete vector field   This research was motivated by the works of Helena Molina-Abril and Pedro Real about *homological spanning forests* (see [90] for a general picture), which were the starting point of this PhD thesis. From this, we define a combinatorial structure, namely the *homological discrete vector field* (HDVF), which encodes a reduction on a CW complex. This concept passed through different formalisms until we found a clear definition that allows a deep understanding of its nature. Roughly speaking, we defined it firstly as a discrete vector field (possibly with cycles) iteratively built and we ended up defining it as a collection of cells satisfying an algebraic condition regarding the boundary of the complex. This concept, which we prove to be equivalent to different methods in computational homology, however shows an interesting combinatorial relation between different computed homology groups for a same complex.

The HDVF thus improves the homological spanning forest in different ways: it works for any dimension, it always computes (if the ground ring is a field) the homology groups without needing a later diagonalization and its clear definition allows us to prove theorems using its formalism. Nevertheless, there are still many interesting open questions.

Fast computation of Betti numbers on three-dimensional cubical complexes   This research, which was conducted in collaboration with Mateusz Juda, develops an algorithm for efficiently computing only the Betti numbers of a 3D cubical complex (that is, without homology generators nor a reduction). Its description is simple and follows from a constructive proof involving the HDVF framework. Also, the regular structure of the cubical complexes allows us to efficiently parallelize the algorithm. We show that this algorithm outperforms the existing software specialized in cubical homology.

Measuring holes Let us introduce the problem that originated this research by an example. Consider a cubic portion of cheese with edge length of 5 cm. If we find out that there are 10 holes—that is, $\beta_2 = 10$—we cannot really tell if the cheese is *full of holes* or if there are just some small bubbles inside. A direct approach is to weigh the cheese to figure out the proportion of void in the portion, but this does not tell us if all holes have similar size or not. Moreover, this will not tell us anything about the 1-holes—there may be some tunnels going through the portion or torus-shaped holes inside the cheese. The idea we had is to gradually *expand* the cheese and see when the holes disappear. Moreover, because we are topologists and we always think about duality, we can also *shrink* or *erode* the cheese and see when the holes disappear, which gives us an idea of the *fragility* of the holes.

This example gives a good intuition about the two measures that we introduce in Chapter 5. By using persistent homology and the signed distance transform of a discrete object we obtain a pair of values—the *thickness* and the *breadth*— for each hole, even though holes cannot be canonically located. These measures have many good properties: their definition is general for objects of any dimension and for any type of holes and they are stable under small perturbations on the boundary of the objects. More surprisingly, they seem useful to visualize holes, find small generators of homology or cohomology and close or open holes. Let us point out that we can do this for all the holes or just for some of them—which we can choose regarding their measures. These last results are quite visual, so we illustrate them by presenting some examples.

1.2   How to read this dissertation

The common background for the three main chapters is presented in Chapter 2, while the specific preliminaries for each chapter are included in it.

The three chapters are ordered chronologically, but this order also reveals an increasing interest in discrete objects. Roughly speaking, Chapter 3 presents a framework for computing the homology of CW-complexes (including cubical complexes); Chapter 4 shows how to count the number of holes in a 3D discrete object and Chapter 5 studies further the geometry of its holes, even in higher dimensions. Consequently, the chapters are not completely independent:

- Chapter 3 can be read without the other two.

- Chapter 4 uses concepts from Chapter 3 in the proofs, but they can be omitted if one is only interested in the algorithm.

- Chapter 5 can almost be read without referring to Chapter 3, except for Section 5.4 and 5.5.

Note that each chapter contains its own conclusion. The general conclusion in Chapter 6 recalls the main results of each topic and explains some future works not strictly related to the research developed herein.

Chapter 2

Common Background

S INCE this document contains different works, we describe in this chapter the common background for all of them. Then, each other chapter contains its own specific background. If the reader is only interested in one of the works of this thesis, he/she could skip this part and refer to it later if necessary.

2.1   General Topology

We begin by defining a topology for introducing all the relations between topological spaces. This allows us to better understand the difference between homotopy and homology theory. Some kinds of topological spaces are also introduced.

Definition 2.1. *A* topological space *is a pair* (X, T) *where* X *is a set and* $T \subset 2^X$ *(called* topology*) is a collection of subsets of* X *(called* opens*) such that*

1. $\{\emptyset, X\} \subset T$: *the empty set and the full set are opens.*

2. $\bigcup_{i \in I} A_i \in T$: *any arbitrary (possibly infinite) union of opens is an open.*

3. $\bigcap_{i=1}^n A_i \in T$: *any finite intersection of opens is an open.*

For instance, the pair (X, T) where $X = \{1, 2, 3\}$ and $T = \{\emptyset, \{1\}, \{1, 2\}, \{1, 3\}, \{3\}, X\}$ is a topological space. A much more common example is the Euclidean space $X = \mathbb{R}^n$ with the so called *usual topology*

$$T_u = \left\{ A \subset \mathbb{R}^n \mid A = \bigcup_{i \in I} B(x, r) \right\}$$

where $B(x, r) = \left\{ y \in \mathbb{R}^n \mid \sqrt{\sum_{i=1}^n (y_i - x_i)^2} < r \right\}$ denotes the n-dimensional ball centered at x with radius r.

There is also a canonical way of defining a topology for a subspace of a topological space.

Definition 2.2. *Let* (X, T) *be a topological space and* $Y \subset X$. *Thus* (Y, T_Y) *is a topological space with the topology*

$$T_Y = \{A \cap Y \mid A \in T\}.$$

T_Y *is called the* subspace topology *(or* relative topology*).*

Thus, any subset of $\mathbb{R}^n$ has an usual topology. We can now define a continuous map.

Definition 2.3. *A map between two topological spaces (X, T) and (X', T') is continuous if for all $A' \in T'$, $f^{-1}(A') \in T$.*

This coincides with the well-known definition of continuous function in calculus: a function $f : \mathbb{R} \to \mathbb{R}$ is continuous if we can draw its graph $\{(x, f(x))\}$ without lifting the pencil from the paper. More generally, a function between two topological spaces $f : A \to B$ is continuous if every two "close" points in A are sent to "close" points in B. The definition of topological spaces provides the notion of "closeness".

In the context of this work we always consider subspaces of the Euclidean space. Thus, we refer to topological spaces just as spaces, and the usual topology T_u or the subspace topology T_Y are always considered and not mentioned explicitly.

We can now define different relations between spaces.

Definition 2.4. *Two spaces $X, Y \subset \mathbb{R}^n$ are homeomorphic if there exists a map $h : X \to Y$ which is continuous, bijective and h^{-1} is also continuous. Such map is called a homeomorphism.*

Definition 2.5. *Two spaces $X, Y \subset \mathbb{R}^n$ are ambient isotopic if there is a continuous map*

$$F : \mathbb{R}^n \times [0, 1] \to \mathbb{R}^n$$

such that $F(x, 0) = x$, $F(X, 1) = Y$ and $F(x, t)$ is a homeomorphism for each $t \in [0, 1]$

Definition 2.6. *Two continuous maps $f : X \to Y$ and $g : X \to Y$ are homotopic if there exists a continuous map*

$$H : X \times [0, 1] \to Y$$

such that $H(x, 0) = f(x)$ and $H(x, 1) = g(x)$. The map H is called a homotopy. Thus, two spaces $X, Y \subset \mathbb{R}^n$ are homotopy equivalent (or have the same homotopy type) if there exist two continuous maps

$$f : X \to Y, \quad g : Y \to X$$

such that gf is homotopic to 1_X and fg is homotopic to 1_Y.

Let us recapitulate what we have done. The notion of topological space allows us to define a continuous map, though we are only interested in a very restricted class of topological spaces: the Euclidean space $\mathbb{R}^n$ and its subspaces. We then define three relations between topological spaces: ambient isotopic, homeomorphic and homotopy equivalent.

Intuitively, two spaces are homeomorphic if they are the same up to a continuous deformation. In other words, X is homeomorphic to Y if it can be deformed to Y by stretching and bending it, without tearing or gluing. This is a key concept in topology, and this is why topology is sometimes called "rubber-sheet geometry", since spaces are regarded as deformable objects. Thus, many different spaces are considered as the same, but there are others that are not "equivalent" (homeomorphic), and this is the *big topological question*: are two spaces homeomorphic?

The other two relations are introduced to complement the notion of homeomorphic spaces. The ambient isotopy relation regards also the embedding of the space in a bigger space. If two spaces are ambient isotopic

then they are homeomorphic. A typical example of two spaces that are homeomorphic but not ambient isotopic is illustrated in Figure 2.1. This notion is not used in the rest of this dissertation, but it is interesting since it shows that homeomorphic spaces can be more different than expected.
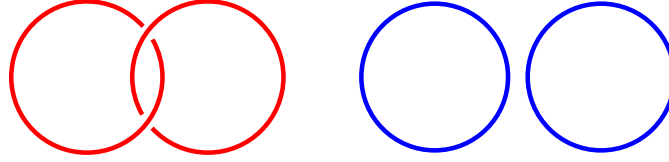
FIGURE 2.1: Two spaces that are homeomorphic but not ambient isotopic.

The homotopy relation is also considered since it helps to answer to the *big topological question*. If two spaces are homeomorphic then they are homotopy equivalent, and the converse is false. Figure 2.2 illustrates an example. Consequently, if we can prove that two spaces are not homotopy equivalent then we deduce that they are not homeomorphic. We will briefly see in Section 2.2.1 how this can be done.
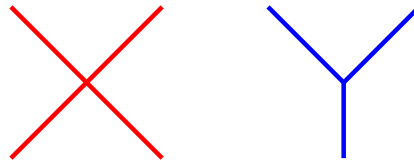
FIGURE 2.2: Two spaces that are homotopy equivalent but not homeomorphic.

2.2 Algebraic Topology

We give in this section some ideas about how to answer to the *big topological question*. This could seem too simplistic for a pure topologist, but this essay is about computational topology so more advanced constructions are out of the scope.

If two spaces are homeomorphic, one can prove it by providing a homeomorphism h between the spaces. If they are not, rather than trying all the possible maps between both spaces, one can use an *invariant*, that is, a property that is preserved under homeomorphisms. For example, if one topological space is connected, compact, Hausdorff, separable, . . ., then any homeomorphic space shares this property. We do not define these properties since they are not useful in the context of computational topology. Instead, we explain two fundamental topics in algebraic topology which provide algebraic objects (such as groups, vector spaces or just numbers) as invariants.

We were generous in the previous section by giving very basic definitions, but we assume here that the reader is familiar with basic algebraic notions such as group, homomorphism, vector space, basis, dimension, etc.

2.2.1   Homotopy

We introduce homotopy theory for completeness, but we do not use it in this monograph.

Let X be a topological space, a *loop* in X is a continuous map $\gamma : [0,1] \to X$ such that $\gamma(0) = \gamma(1)$. We say that $\gamma(0)$ is its *basepoint*. We can define the binary relation

$$\gamma \sim \gamma' \quad \Longleftrightarrow \quad \gamma \text{ and } \gamma' \text{ are homotopic.}$$

This is actually an equivalence relation, and we denote the equivalence classes by $[\gamma]$. We can also "compose" two loops with the same basepoint:

$$\gamma \circ \gamma' : [0,1] \longrightarrow X$$
$$t \longmapsto \begin{cases} \gamma(2t) & \text{if } t \leq 1/2 \\ \gamma'(2t-1) & \text{if } t > 1/2 \end{cases}$$

We thus define $[\gamma] \circ [\gamma'] := [\gamma \circ \gamma']$. Given a basepoint $x_0 \in X$, the set of all the equivalence classes of loops in X with basepoint x_0 is denoted by $\Pi_1(X, x_0)$. This set with the operation $\circ$ is a group and we call it the *fundamental group* of X at the basepoint x_0. We can drop the basepoint if X is path-connected.

Why defining this? Because the fundamental group is a homotopy invariant: if two topological spaces are homotopy equivalent then their fundamental groups are isomorphic. Consequently, if two spaces have different fundamental groups (the groups are not isomorphic) then they are not homotopy equivalent and thus they are not homeomorphic. Hence, we transform a topological problem into an algebraic one, since we have to tell whether two groups are isomorphic. That is how we can answer to the *big topological question* with homotopy.

Note that this construction can be generalized to higher dimensions, giving the so called *homotopy groups*.

2.2.2   Homology

Homology theory is the main topic of this essay. We first introduce it in algebraic terms to later see it in terms of simplicial or cubical complexes.

Chain Complexes and Homology

We first give a definition of homology in pure algebraic terms. Let $\mathfrak{R}$ be a ring (we can assume that $\mathfrak{R} = \mathbb{Z}$ for simplicity).

Definition 2.7. *An $\mathfrak{R}$-module is an abelian group $(M, +)$ with an operation $\cdot : \mathfrak{R} \times M \to M$ such that:*

- $1 \cdot m = m$ *for all $m \in M$. 1 denotes the multiplicative identity of $\mathfrak{R}$;*

- $(r + s) \cdot m = (r \cdot m) + (s \cdot m)$ *for all $r, s \in \mathfrak{R}, m \in M$;*

- $r \cdot (m + n) = (r \cdot m) + (r \cdot n)$ *for all $r \in \mathfrak{R}, m, n \in M$;*

- $(r \cdot s) \cdot m = r \cdot (s \cdot m)$ *for all $r, s \in \mathfrak{R}, m \in M$.*

If $\mathfrak{R} = \mathbb{Z}$ then an $\mathfrak{R}$-module is an abelian group. If $\mathfrak{R}$ is a field then an $\mathfrak{R}$-module is a vector space. We use this concept because it provides a uniform formalism for both algebraic structures. Thus, the concepts of homomorphism (of groups) or linear map (of vector spaces) can be generalized to a morphism of $\mathfrak{R}$-modules.

Definition 2.8. *A* chain complex (C, d) *is a sequence of $\mathfrak{R}$-modules* $\{\mathsf{C}_q\}_{q \geq 0}$ *(called* chain groups*) and morphisms* $\{d_q\}_{q \geq 0}$ *(called* boundary operators*)*

$$\cdots \xrightarrow{d_3} \mathsf{C}_2 \xrightarrow{d_2} \mathsf{C}_1 \xrightarrow{d_1} \mathsf{C}_0 \xrightarrow{d_0} 0$$

such that $d_{q-1}d_q = 0$ *for every* $q > 0$.

The elements of a chain group C_q are called *q-chains*. A *q*-chain x is a *cycle* if $x \in \ker(d_q)$ and a *boundary* if $x \in \operatorname{im}(d_{q+1})$. Since $d_{q-1}d_q = 0$, every boundary is a cycle and thus the quotient $\ker(d_q)/\operatorname{im}(d_{q+1})$ is well defined.

Definition 2.9. *The* homology groups *of a chain complex* (C, d) *are the quotients*

$$H_q(\mathsf{C}) = \ker(d_q)/\operatorname{im}(d_{q+1})$$

Singular Homology

We see now how we apply this algebraic structure to the context of topology. Note that the rest of this section is more complicated than the similar concepts defined in Section 2.3.2 and 2.3.3 and it is apparently out of the scope of this thesis, but it provides a general picture of how we use homology for answering the *big topological question*.

Let n be a non-negative integer, the *standard n-simplex* is the set

$$\Delta^n = \left\{ (\lambda_0, \dots, \lambda_n) \in \mathbb{R}^{n+1} \mid \sum_{i=0}^{n} \lambda_i = 1, \lambda_i \geq 0 \right\}$$

In other words, it is the convex hull of the vectors in the canonical basis $\{e_1, \dots, e_{n+1}\}$ of $\mathbb{R}^{n+1}$.

Definition 2.10. *Let* X *be a topological space, a* singular *q*-simplex *is a continuous map*

$$\sigma : \Delta^q \longrightarrow X.$$

Moreover, its i-th $(q-1)$*-face is the singular* $(q-1)$*-simplex*

$$\sigma \epsilon_q^i : \Delta^{q-1} \longrightarrow X$$

where $\epsilon_q^i : \Delta^{q-1} \to \Delta^q$, $\epsilon_q^i(x_0, \dots, x_{q-1}) = (x_0, \dots, 0, \dots, x_{q-1})$, *with the number 0 at the i-th coordinate.*

It is helpful to think of σ as a subset of X rather than a proper map. Now we define the (singular) chain complex (C, d) associated to the space X:

- For each $q \geq 0$, C_q is the free $\mathfrak{R}$-module generated by the singular *q*-simplices

$$\mathsf{C}_q = \left\{ \sum_{i \in I} \lambda_i \cdot \sigma_i \mid \lambda_i \in \mathfrak{R}, \sigma_i \text{ is a singular } q\text{-simplex}, I \text{ is finite} \right\}$$

- d_q is defined over the singular q-simplices as the alternating sum of its $(q-1)$-faces

$$d_q(\sigma) = \sum_{i=1}^{q+1} (-1)^{i+1} \cdot \sigma \epsilon_q^i$$

It is easy to check that $d_{q-1} d_q = 0$. Thus for each topological space X we obtain a sequence of homology groups $H_q(X)$. The interest of this is that if two topological spaces are homotopy equivalent then their homology groups are isomorphic. Consequently, if two spaces have different (not isomorphic) homology groups for some dimension $q \geq 0$ then they are not homotopy equivalent and thus they are not homeomorphic.

Cohomology

There is a dual theory to homology which, given its similarity, we briefly describe in this section.

Definition 2.11. *A* cochain complex (C, d) *is a sequence of $\mathfrak{R}$-modules $\{\mathsf{C}^q\}_{q \geq 0}$ (called* cochain groups*) and morphisms $\{d^q\}_{q \geq 0}$ (called* coboundary operators*)*

$$\cdots \xleftarrow{d^3} \mathsf{C}_2 \xleftarrow{d^2} \mathsf{C}^1 \xleftarrow{d^1} \mathsf{C}^0 \xleftarrow{d^0} 0$$

satisfying that $d^{q+1} d^q = 0$ for every $q \geq 0$.

It is essentially the same as a chain complex, except that the morphisms go in the other direction. The cohomology groups are defined similarly.

Definition 2.12. *The* cohomology groups *of a cochain complex (C, d) are the quotients*

$$H^q(\mathsf{C}) = \ker(d^{q+1}) / \operatorname{im}(d^q)$$

The singular cohomology groups of a topological space X are naturally defined through the cochain complex (C^*, d^*) where $\mathsf{C}^* = Hom(C, \mathfrak{R})$ is the set of homomorphism from C to $\mathfrak{R}$ and d^* denotes the dual of the boundary operator, that is, if $x \in \mathsf{C}^*$ then $d^*(x) = x \circ d : C \to \mathfrak{R}$. Intuitively, we see the coboundary of (the dual of) a singular q-simplex as a sum of the (duals of the) $(q+1)$-simplices having it in their boundary.

2.3   Computational topology

Computational topology needs topological spaces that can be described with a finite representation. One could think that it should be possible to compute the homology of a differential manifold from its equation, as computer algebra systems can derive an equation, but this is not yet the case. Algebraic topologists prefer to work with much simpler spaces, such as simplicial complexes, cubical complexes or, more generally, CW complexes.

In this section we briefly speak about computational homotopy in Section 2.3.1 and then we describe different kinds of combinatorial spaces for which we can compute the homology groups with an algorithm in Section 2.3.2 and 2.3.3. Section 2.3.4 introduces an algebraic object that we will use in several parts of this digression. Last, we succinctly present the theory of persistent homology in Section 2.3.5.

2.3.1 Homotopy

The fundamental group is classically computed with the Seifert–van Kampen theorem. Unfortunately, it provides a *group representation* (a set of letters with relations between them) of the fundamental group, which is computationally useless. The problem of telling if the fundamental group of a space is the trivial group is undecidable, as it reduces to the *halt problem*. Thus, there is no algorithm that extracts any information from the group representation given by the Seifert–van Kampen theorem and hence the fundamental group cannot be used (computationally) as a topological invariant. Nevertheless, let us point out that [11] succeeds in comparing different fundamental groups by extracting some computable invariants.

2.3.2 Simplicial homology

The easiest way of understanding homology in computational terms is through simplicial complexes.

Simplicial complexes

Given a collection of points $\{p_0, \ldots, p_m\} \subset \mathbb{R}^n$, their *convex hull* is

$$\langle p_0, \ldots, p_m \rangle = \left\{ \sum_{i=0}^{m} \lambda_i p_i \mid \lambda_i \geq 0, \sum_{i=0}^{m} \lambda_i = 1 \right\}$$

We recall also that a collection of points $\{p_0, \ldots, p_m\} \subset \mathbb{R}^n$ is in *general position* if the set of vectors $\{\overrightarrow{p_0 p_i} \mid i \geq 1\}$ is linearly independent. In other words, no $(m-1)$-dimensional flat contains all the points.

Definition 2.13. *A q-simplex is the convex hull of $q+1$ points in general position. Let $\sigma = \langle p_0, \ldots, p_q \rangle$ be a simplex, its* faces *are the simplices $\tau = \langle I \rangle$ for every subset $I \subset \{p_0, \ldots, p_q\}$.*

A q-simplex σ is said to be of dimension q and we denote it by $\sigma^{(q)}$ when this is not clear. Note that this notation is also used for other types of complexes.

Definition 2.14. *A (finite) simplicial complex K is a collection of simplices such that (1) for every $\sigma \in K$, its faces are also contained in K and (2) for every $\sigma, \tau \in K$, its intersection is empty or a common face. Its* dimension *is the maximal dimension of its simplices. For each $q \geq 0$, we denote by K_q the set of the q-simplices of K.*

We now define the (simplicial) chain complex (C, d) associated to the simplicial complex K:

- For each $q \geq 0$, C_q is the free $\mathfrak{R}$-module generated by the q-simplices of K

$$\mathsf{C}_q = \left\{ \sum_i \lambda_i \cdot \sigma_i \mid \lambda_i \in \mathfrak{R}, \sigma_i \in K_q \right\}$$

- d_q is defined over the q-simplices as the alternating sum of its $(q-1)$-faces

$$d_q \left(\langle x_0, \ldots, x_q \rangle \right) = \sum_{i=0}^{q} (-1)^{i+1} \cdot \langle x_0, \ldots, \hat{x}_i, \ldots x_q \rangle$$

where $\hat{x}_i$ means that the point x_i has been removed. For the 0-simplices we define $d_0 = 0$.

It is easy to check that for every $q > 0$, $d_{q-1}d_q = 0$. Thus (C, d) is a chain complex and the (simplicial) homology groups of K are well defined. Note that K is a topological space and thus we can define its singular homology groups. Fortunately, they are isomorphic to the simplicial homology groups.

More about the homology groups

Let us now look more closely at the homology groups. They are quotient spaces, where each element is a class under the equivalence relation

$$\forall x, y \in \mathsf{C}_q, \quad x \sim y \Leftrightarrow x - y \in \mathrm{im}(d_{q+1}).$$

They are finitely generated $\mathfrak{R}$-modules, so there exists a generating set (a basis if it is a vector space). By the fundamental theorem of finitely generated abelian groups [37, §5.2], there are two different "normalizations" of this generating set:

1. The $\mathfrak{R}$-module is isomorphic to $\mathfrak{R}^{\beta_q} \times \mathfrak{R}/\lambda_1\mathfrak{R} \times \mathfrak{R}/\lambda_2\mathfrak{R} \times \ldots$, where each λ_i divides λ_{i+1}. This is called the *invariant factor decomposition*.

2. The $\mathfrak{R}$-module is isomorphic to $\mathfrak{R}^{\beta_q} \times \mathfrak{R}/\lambda_1\mathfrak{R} \times \mathfrak{R}/\lambda_2\mathfrak{R} \times \ldots$, where each λ_i is a power of some prime number. This is called the *primary decomposition*.

As most of the literature about computational homology, we use the first decomposition. The number β_q is called the q-th *Betti number* and $\lambda_1, \ldots, \lambda_t$ are the *torsion coefficients* of dimension q. Let us recall that if the ambient space is $\mathbb{R}^3$ there are no torsion coefficients.

The homology groups depend on the ground ring $\mathfrak{R}$. Most of the works in computational homology choose $\mathfrak{R} = \mathbb{Z}_2$ since the operations between chains are simpler, the homology groups are vector spaces and thus there are no torsion coefficients. However, the homology groups with any ground ring $\mathfrak{R}$ can be deduced from the homology groups with coefficients in $\mathbb{Z}$ by the universal coefficient theorem [67, §3.A].

Can we see the homology groups? If our simplicial complex is in $\mathbb{R}^3$ then there are no torsion coefficients and we can consider the ground ring as $\mathbb{Z}_2$. Chains are just sets of simplices and the elements of the homology groups (which are equivalence classes since they are quotient groups) are collections of sets of simplices. Figure 2.3 shows three representatives for the same element of the homology group. This means that the difference between any two of them (actually their symmetric difference) is a boundary, that is, it belongs to $\mathrm{im}(d_2)$.

Instead of seeing all the elements in the homology groups it may be more interesting to visualize only a basis of these groups and choose a representative for each generator. Therefore, there is a double choice. Figure 2.4 shows two generators for the one-dimensional homology group. The first set of cycles gives the image that we expect from a set of generators: they correspond to the holes of the simplicial complex. We have also
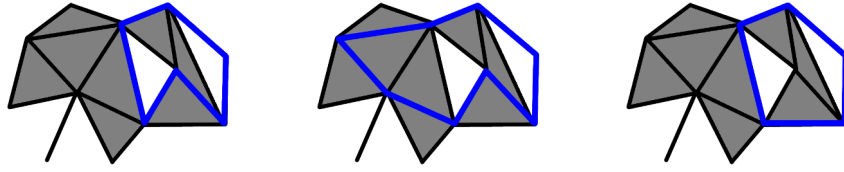
FIGURE 2.3: Three cycles of C_1 belonging to the same class
in $H_1(C)$.

added the second set of cycles which is a basis for $H_1(C)$ but where the
holes are not so well located.
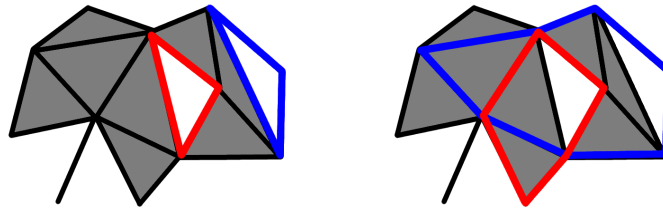
FIGURE 2.4: Two possible representations for the basis of
$H_1(C)$.

This provides a good picture about homology. On the one hand, we
consider the Betti numbers of a simplicial complex as the number of holes
of each dimension, which usually corresponds with the intuition. Note that
a wire-frame cube and tetrahedron (see Figure 2.5) have five and three 1-
holes respectively, instead of six and four. One usually sees an extra hole,
which is the sum of the other holes, but this depends on the point of view.
Thus, the Betti numbers let us formally define the number of holes in a
space regardless of its embedding. On the other hand, we cannot say where
these holes are, as there is a large number of possible sets of generators and
there is no canonical choice.
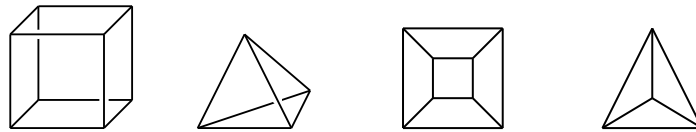
FIGURE 2.5: A wire-frame cube and tetrahedron seen from
two different points of view.

How to compute the homology groups

The simplicial homology groups are computable. Since the boundary op-
erators are linear, they can be encoded as a sequence $\{D_q\}_{q=1}^n$ of matrices,
called *boundary matrices*. The classical method for computing the homology
groups was introduced in [99] and consists in computing the Smith normal
form of the boundary matrices.

Definition 2.15. *The Smith normal form of a (not necessarily square) matrix $A \in M_{n \times m}(\mathfrak{R})$ with entries in a ring $\mathfrak{R}$ is the matrix*

$$N = \left[\begin{array}{cc} \Delta & 0 \\ 0 & 0 \end{array} \right] \in M_{n \times m}(\mathfrak{R})$$

such that Δ is a diagonal square matrix

$$\Delta = \left[\begin{array}{cccc} \alpha_1 & 0 & 0 & 0 \\ 0 & \alpha_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \alpha_r \end{array} \right]$$

with α_i dividing α_{i+1} for $1 \leq i < r$ and there are two invertible matrices P and Q such that $PAQ = N$.

The method present in [99] obtains the invariant factor decomposition of the homology groups. A variant of this method introduced in [102] obtains also a basis of the homology groups. A very clear description of this method can be found in [8].

Let us point out that computing the Smith normal form of a matrix is similar to perform a Gaussian elimination, except that every pivot must divide all the remaining entries. If $\mathfrak{R}$ is a field then there is no difficulty. If not, one has to perform elementary operations on the rows and columns of the matrix until an entry dividing all the others appears. Then this entry is chosen as a pivot and we make all the other entries in its row and column into zeros.

Storjohann introduced in [111] an algorithm with super-cubical complexity for computing the Smith normal form of a matrix over the integers or over the integers modulo d. Let us also point out that the computation of the Smith normal form can produce huge integers [65].

2.3.3   Cubical homology

In this section we introduce the cubical complexes and their homology groups. These complexes are very similar to the simplicial complexes, except that they are built with q-dimensional squares instead of q-dimensional triangles.

Let us fix our ambient space as $\mathbb{R}^n$. An *elementary interval* is an interval of the form $[k, k+1]$ or a degenerate interval $[k, k]$, where $k \in \mathbb{Z}$.

Definition 2.16. *An* elementary cube *is the Cartesian product of n elementary intervals*

$$\begin{aligned} \sigma &= [x_1, x_1 + \delta_1] \times \cdots \times [x_n, x_n + \delta_n] & x_i \in \mathbb{Z}, \delta_i \in \{0, 1\} \\ &=: [x, \delta] & x \in \mathbb{Z}^n, \delta \in \{0, 1\}^n \end{aligned}$$

Its Khalimsky coordinates *is the vector $\sigma_K = 2x + \delta \in \mathbb{Z}^n$, the sum of the intervals endpoints. The number of non-degenerate intervals in this product σ (or the number of odd entries in its Khalimsky coordinates) is the* dimension *of σ. An elementary cube of dimension q will be called a q-*cube*.*
Given two elementary cubes σ and τ, we say that σ is a face *of τ if $\sigma \subset \tau$.*

For instance, the Khalimsky coordinates of the elementary cube $\sigma = [1,1] \times [2,3] \times [1,2]$ are $(2,5,3)$ and hence it is a 2-cube.

Definition 2.17. *A (finite) nD cubical complex K is a collection of elementary cubes such that for every $\sigma \in K$, its faces are also contained in K. Its dimension is the maximal dimension of its elementary cubes. For each $q \geq 0$, we denote by K_q the set of the q-cubes of K.*

Let us point out that we do not demand any condition about the intersection of different elementary cubes due to the regular structure of the cubical complex.

We now define the (cubical) chain complex (C, d) associated to the cubical complex K:

- For each $q \geq 0$, C_q is the free $\mathfrak{R}$-module generated by the q-cubes of K

$$\mathsf{C}_q = \left\{ \sum_i \lambda_i \cdot \sigma_i \mid \lambda_i \in \mathfrak{R}, \sigma_i \in K_q \right\}$$

- d_q is defined over the q-cubes as the alternating sum of its $(q-1)$-faces along each axis

$$d_q\left([x, \delta]\right) = \sum_{i=1}^n (-1)^{o(i)} \cdot \left([x + \delta_i \cdot e_i, \delta - \delta_i \cdot e_i] - [x, \delta - \delta_i \cdot e_i]\right)$$

where $o(i)$ denotes the number of ones in $(\delta_1, \ldots, \delta_i)$ (or equivalently, the number of non-degenerate intervals among the i first elementary intervals of $[x, \delta]$), $x + \delta_i \cdot e_i = (x_1, \ldots, x_i + \delta_i, \ldots, x_n)$ and $\delta - \delta_i \cdot e_i = (\delta_1, \ldots, 0, \ldots, \delta_n)$. For the 0-cubes we define $d_0 = 0$.

Again, it is easy to check that for every $q > 0$, $d_{q-1}d_q = 0$. Thus (C, d) is a chain complex and the (cubical) homology groups of K are well defined. Note again that K is a topological space whose singular homology groups are isomorphic to its cubical homology groups.

2.3.4 Effective Homology

Computing the homology groups using the Smith normal form is practically impossible for large complexes due to its high complexity. A solution to reduce the amount of information to compute is the notion of *reduction*. It is a strong relation between two chain complexes that guarantees that they have isomorphic homology groups. This is the main tool in *effective homology theory* [106]. We typically reduce the initial chain complex to another one much smaller (called *reduced complex*). In the following we omit the subscripts whenever it is clear from the context.

Definition 2.18. *A reduction between two chain complexes (C, d) and (C', d') is a triplet of graded homomorphisms $\rho = (h, f, g)$ such that:*

- $h_q : \mathsf{C}_q \to \mathsf{C}_{q+1}$ *for every $q \geq 0$*

- $f_q : \mathsf{C}_q \to \mathsf{C}'_q$ *is a chain map: $f_{q-1}d_q = d'_q f_q$*

- $g_q : \mathsf{C}'_q \to \mathsf{C}_q$ *is also a chain map: $g_{q-1}d'_q = d_q g_q$*

- $gf = 1_C - dh - hd$

- $fg = 1_{C'}$

- $hh, \ fh, \ hg = 0$

A reduction is usually represented with the following diagram:

For instance, consider the following chain complexes (C, d) and (C', d'), whose chain groups are freely generated with $\mathfrak{R} = \mathbb{Z}$:

$$C_0 = \langle \sigma_1, \sigma_2, \sigma_3 \rangle, \quad C_1 = \langle \sigma_4, \sigma_5, \sigma_6 \rangle, \quad C_2 = \langle \sigma_7 \rangle, \quad C_3 = 0, \quad \cdots$$

$$d_1 = \begin{bmatrix} -1 & 0 & -1 \\ 0 & -1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad d_2 = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}$$

$$C'_0 = \langle \tau_1, \tau_2 \rangle, \quad C_1 = \langle \tau_3 \rangle, \quad C_2 = 0, \quad \cdots$$

$$d'_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad d'_2 = 0$$

Hence (h, f, g) is a reduction, where

$$h_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad h_1 = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}$$

$$f_0 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad f_1 = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

$$g_0 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad g_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

We have followed Sergeraert's terminology. There are equivalent or similar definitions in the literature: *contraction* [45, §12], *strong deformation retraction* [82, §2], *Eilenberg-Zilber data* [63, §4] or *trivialized extension* [98, §2].

If we remove the last line of conditions, the resulting relation is called a *chain homotopy equivalence*. It ensures that both chain complexes have isomorphic homology groups, where the isomorphisms are the induced maps f and g in the homology groups. However, these last conditions provide more information: they decompose the chain complex into two subcomplexes: $\ker(f)$ and $\operatorname{im}(g)$, where the former is acyclic (its homology groups are all trivial) and the latter is isomorphic to the reduced chain complex (C', d'). This can be thought as a homotopical thinning of the complex,

where we remove parts from the original chain complex without modifying its homology.

A reduction is *perfect* if $d' = 0$. In such case, $H(\mathsf{C}) \cong H(\mathsf{C}') = \mathsf{C}'$ and thus the homology groups are directly obtained. Moreover, $g(\mathsf{C}')$ is a basis for $H(\mathsf{C})$. Also, let $x \in \mathsf{C}_q$ be a cycle. If it is a boundary then

$$f(x) = fd(y) = d'f(y) = 0$$

since $d' = 0$. Hence,

$$g \underbrace{f(x)}_{0} = x - dh(x) - hd(x) = x - dh(x) \Rightarrow x = dh(x)$$

That is, x is a boundary if and only if $f(x) = 0$ and in that case $x = d(y)$ for the chain $y = h(x)$. These facts should justify the interest of having a perfect reduction.

Let us point out that if the homology groups of a chain complex have a torsion subgroup then there is no perfect reduction, since a perfect reduction involves homology groups freely generated, and hence of the form $\mathbb{Z}^{\beta}$. Also, a reduction can always be obtained via the Smith normal form computation as described in [8, p. 48]. This reduction is perfect if the homology groups are torsion-free. Otherwise, its reduced boundary matrices are in the Smith normal form.

If $\rho = (h, f, g)$ is a reduction from (C, d) to (C', d'), it is easy to prove that $\rho^* = (h^*, g^*, f^*)$ is a reduction between the cochain complexes (C, d^*) and $(\mathsf{C}', (d')^*)$. Consequently, a perfect reduction also provides a basis for the cohomology groups, namely $f^*(C)$.

2.3.5 Persistent homology

Persistent homology studies the global behavior of the homology groups of a complex that changes along time. Formally, we consider a nested sequence of simplicial complexes.

Definition 2.19. *A* filtration *of a simplicial complex K is a sequence of subcomplexes*

$$\emptyset = K^0 \subset K^1 \subset \cdots \subset K^m = K$$

It can also be described by a function $f : K \to [1, m]$ mapping each cell to the index of the first subcomplex containing it.

Note that $f(\sigma) < f(\tau)$ for all $\sigma < \tau$ since a cell must appear after its faces. We can thus build a filtration from any function on a complex by taking the maximum of the images of all the faces of a cell (even itself).

Persistent homology formalizes the idea that, in a filtration, some holes last longer (are more persistent) than others. The inclusion between the subcomplexes of the filtration, which induces a chain map between their chain groups and a homomorphism between their homology groups, plays a central role. Let (C^i, d^i) be the chain complex associated to the subcomplex K^i and $\iota^{i,p} : \mathsf{C}^i \to \mathsf{C}^{i+p}$ the inclusion from K^i to K^{i+p}.

Definition 2.20. *The p-persistent q-th homology* group of K^i is

$$H_q^{i,p} = \ker(d_q^i)/(\mathrm{im}(d_{q+1}^{i+p}) \cap \ker(d_q^i)) \cong \mathrm{im}\left(\left[\iota_q^{i,p}\right]\right)$$

Intuitively, it is the set of cycles in K^i that remain non-bounding for the p following steps. The rank of its free subgroup is called the *p-persistent q-th Betti number* of K^i. This definition has three parameters: i, p and q. If we were able to define the birth and death time of each homology class in the filtration, then we could easily deduce the p-persistent Betti numbers (see the k-triangle Lemma in [44]). Zomorodian and Carlsson introduced a correspondence in [119] that associates each filtration F to its *persistent diagram*, a set of intervals $PD(F) = \cup_{q \geq 0} PD_q(F) = \{(i,j) \mid 0 \leq i \leq j \leq \infty\}$. Unfortunately, this is only possible if the ground ring is a field. A simple algorithm is given, which amounts to compute the Smith normal form by choosing pivots in the order determined by the filtration.

The set $PD_q(F)$ can be interpreted as points in the plane or as intervals in the line. The points of the form (i,j) with $j < \infty$ correspond to q-holes that *are born* in K^i and *die* in K^j, while the points of the form (i, ∞) conform the q-holes that are present in K and appear in K^i. If the filtration is described as a function f, we denote its persistent diagram by $PD(f)$. Cohen-Steiner et al. proved in [19] that $PD(f)$ is stable under small perturbations of f.

2.4 Digital Geometry

A *discrete object* is a finite subset of $\mathbb{Z}^n$. It is also called *binary image* (if $n = 2$) or *binary volume* (if $n = 3$) in order to make the difference against a grayscale image or a color image. Its elements are called *pixels* when $n = 2$, *voxels* when $n = 3$ or *points* in general.

We endow a discrete object with a connectivity relation. Let us recall some usual connectivity relations. Let be $x = (x_1, \ldots, x_n) \in \mathbb{Z}^n$,

$$||x||_1 = |x_1| + \cdots + |x_n| \quad \text{and} \quad ||x||_\infty = \max\{|x_1|, \cdots, |x_n|\}$$

Thus, if $x, y \in \mathbb{Z}^2$,

- x and y are 4-connected if $||x - y||_\infty \leq 1$ and $||x - y||_1 \leq 1$

- x and y are 8-connected if $||x - y||_\infty \leq 1$ and $||x - y||_1 \leq 2$

Also, if $x, y \in \mathbb{Z}^3$,

- x and y are 6-connected if $||x - y||_\infty \leq 1$ and $||x - y||_1 \leq 1$

- x and y are 18-connected if $||x - y||_\infty \leq 1$ and $||x - y||_1 \leq 2$

- x and y are 26-connected if $||x - y||_\infty \leq 1$ and $||x - y||_1 \leq 3$

With this notation, the number accompanying the "connected" word tells the number of points connected to a point in $\mathbb{Z}^n$. Note that these definitions can be extended to any dimension. The reflexive and transitive closure of this connectivity relation allows us to define connected components of a discrete object.

We can actually build cubical complexes from discrete objects in order to obtain higher-dimensional topological information through the homology groups. We introduce two kinds of cubical complexes associated to a discrete object regarding the $2n$-connectivity and the $(3^n - 1)$-connectivity.

Primal associated cubical complex Let X be a discrete object, we denote by $K_p[X]$ its primal associated cubical complex. Let us give a constructive definition: for each point $x = (x_1, \cdots, x_n)$ of X we add to the cubical complex the n-cube $[x_1, x_1 + 1] \times \cdots \times [x_n, x_n + 1]$ together with its faces. This construction can be found in [16].

Dual associated cubical complex We denote the dual associated cubical complex by $K_d[X]$. Let us first adapt the notion of *clique* to our context: a *d-clique* is a maximal (in the sense of inclusion) set of points of $\mathbb{Z}^n$ such that the intersection of their corresponding n-cubes is a d-cube. First, for every point (in fact n-clique) $x = (x_1, \cdots, x_n)$ of the discrete object, we add the 0-cube $\sigma = [x_1, x_1] \times \cdots \times [x_n, x_n]$. Then, for every d-clique ($d < n$) in the discrete object, we add to the cubical complex a $(n - d)$-cube such that its vertices are the points of the d-clique. This approach was used in [84].

We can also define the dual associated cubical complex in a different fashion. Consider K the full nD cubical complex and, for each point $x = (x_1, \cdots, x_n)$ not in X, remove from K the 0-cube $\sigma = [x_1, x_1] \times \cdots \times [x_n, x_n]$ and its cofaces. The resulting cubical complex coincides with $K_d[X]$.

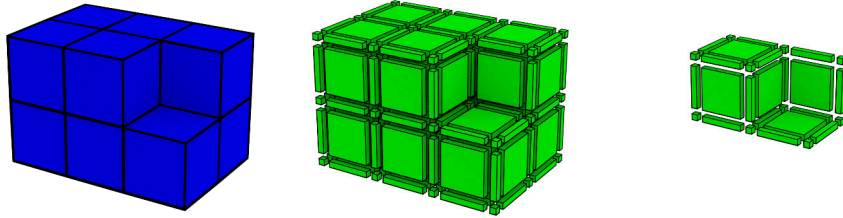Figure 2.6 illustrates a binary volume and its two associated cubical complexes.

FIGURE 2.6: Left: a binary volume. Center: its primal associated cubical complex. Right: its dual associated cubical complex

Chapter 3

The Homological Discrete Vector Field

THIS chapter is based on the journal paper [60] and a paper submitted for publication [56]. We introduce a combinatorial structure called *homological discrete vector field* (HDVF) that captures the homology (and cohomology) of a complex.

3.1 Introduction

Morse theory [87] is a tool in differential topology that deduces some information of the topology of a manifold by studying a differentiable function on it. In the late 90s, Robin Forman introduced a discrete version, the discrete Morse theory [49, 50], which was defined for CW complexes and discrete functions. Several theorems of Morse theory were translated into the discrete context but, in our opinion, the most notable result was the simplification of a CW complex, which can be used to compute its homology groups.

The classical method for computing the homology groups is based on the Smith normal form (SNF) [99], which has super-cubical complexity [111]. Some advances in the computation of the SNF have been achieved, but the best results in computing the homology groups of a complex have been obtained by reducing the number of cells in the complex (see [78, 94, 95]).

Among other approaches, let us mention the following two, which are closely related to our work: effective homology theory [106] and discrete Morse theory [50]. Both of them are explained in Section 2.3.4 and 3.3.4. The former has the advantage that it "controls" the homology because it contains all the homological information [103]; the latter is very concise and easy to implement. Effective homology theory deals with linear maps which should be typically encoded as enormous matrices; discrete Morse theory handles only graphs, but does not always compute the Betti numbers with accuracy. The use of reductions (the main concept of effective homology theory) has proved to be successful in the context of image analysis [55, 54, 52, 5] or in a more general setting providing more advanced topological information [53, 103].

We aim at finding an intermediate solution, avoiding the respective drawbacks of both of these methods whilst maintaining their advantages. Roughly speaking, discrete Morse theory gives an approximation of the homology of an object by establishing arrows on it. In this chapter we allow cycles in this "collection of arrows", which is normally forbidden, so that

we can go beyond the limits of the classical discrete Morse theory. Moreover, we can control when our approach produces the exact homological information. These allowed cycles must not be confused with the ideas found in [48]. The process of adding these arrows must be simultaneously accompanied by the computation of the linear maps of the effective homology theory, which is unnecessary when there are no cycles. The clearest advantage of our approach is that we only use linear space for saving these maps, instead of quadratic, and due to the geometric nature of the HDVF, we can also define heuristics for the assignment of the arrows. Also, our framework works for any dimension, any kind of CW complex and any ring of coefficients.

3.2   Previous Works

This chapter somehow creates a new problem instead of solving an existing one. This justifies the shortness of this section.

Discrete Morse theory was introduced in [49, 50]. It was then reformulated in terms of matchings in [15, 80]. Discrete Morse theory is often used for simplifying a complex in order to accelerate the computation of its homology. Thus, it can be seen as an optimization problem, in which one wants to find a discrete gradient vector field (a matching in the Hasse diagram of a CW complex) with as many edges as possible. It was proved that this is an NP-hard problem (see [83, 73]). Nevertheless, there has been an extensive research on this optimization problem without aiming at finding a perfect solution in the general case, such as in [83, 73, 46, 91]. There has been a parallel and successful research about simplifying a CW complex in [78, 94, 95]. These works were recently related to discrete Morse theory in [66], which states that reductions and coreductions are particular strategies for establishing a discrete gradient vector field.

3.3   Preliminaries

3.3.1   CW Complex

Computational topology needs topological spaces that can be described through a finite representation. A rigorous presentation of CW complexes would be too long for this chapter, so we give an intuitive introduction and we let the reader satisfy its curiosity by consulting [86].

A *CW complex*, or *cell complex*, is a collection of closed unit balls (up to homeomorphism) of different dimensions, called *cells*, that are "glued" together by their boundary: every cell of dimension $q \geq 1$ (q-cell) has a map from its boundary to the lower dimensional cells. A q-cell σ is denoted $\sigma^{(q)}$ whenever its dimension is not clear from the context. We are certainly interested in the case where the number of cells is finite.

To be honest, we only use the notion of CW complex for comprehending simplicial complexes, cubical complexes or even polyhedra [110, §1.1]. We could have chosen to work with S-complexes [94] but we have preferred the CW complexes, as in [39].

We say that a cell σ is a *face* of another one τ if it is contained in its boundary. A special case is when they have consecutive dimensions, in which we say that it is a *primary face* and we write $\sigma < \tau$.

Given a CW complex, we can define its *Hasse diagram*. It is a directed graph whose vertices represent the cells and whose arrows go from each cell to its primary faces. In this chapter we usually do not make the distinction between the vertices and the cells they represent, so we mix these terms.

3.3.2   Homology of a CW complex

The definition of the homology groups of a CW complex is similar to the case of simplicial or cubical complexes.

Let $\mathfrak{R}$ be a ring. We usually consider $\mathfrak{R}$ to be $\mathbb{Z}_2 = \mathbb{Z}/2\mathbb{Z}$ or $\mathbb{Z}$. We say that an element of $\mathfrak{R}$ is a *unit* if it is invertible for the multiplication. We denote by $\mathfrak{R}^*$ the set of the units of $\mathfrak{R}$. For instance, $\mathbb{Z}_2^* = \{1\}$ and $\mathbb{Z}^* = \{-1, 1\}$.

Given a CW complex K, we define its associated chain complex $\mathsf{C}(K)$ as follows:

- C_q is the free $\mathfrak{R}$-module generated by the q-dimensional cells of K;

- d_q gives the "algebraic" boundary, which is the linear operator that maps every cell to the sum of its primary faces with specific coefficients. These coefficients, which are not unique, can be computed with the algorithm present in [39, §3.1].

We will usually use the term *complex* for the CW complex or its associated chain complex. This chain complex can be seen as a sequence of matrices that express the relation of inclusion between the cells. However, note that not every chain complex is the chain complex associated to a CW complex.

The elements of the chain group C_q, which are formal linear combinations of cells, are called *q-chains*. If $x = \sum_{i \in I} \lambda_i \sigma_i$ then $\langle x, \sigma_i \rangle := \lambda_i$ denotes the coefficient of σ_i in the chain x.

3.3.3   Homological Information

Given the definition of the homology groups, we could ask ourselves how much information we want to obtain. If we want to use homology as a topological invariant, it should be enough to know its Euler-Poincaré characteristic or, more generally, its Betti numbers and torsion coefficients. Moreover, if we want to use homology to better understand the shape of the complex, we could be interested in knowing a representative of each class of homology that is a generator. These representatives, which we directly call *homology generators*, are not unique at all, and it is an interesting and ill-defined problem to find a set of well-shaped generators. Furthermore, we can decompose a given cycle onto the computed homology generators.

Since not all works in computational homology try to obtain the same information, we propose the following classification of homological information:

Level 0 : The Euler-Poincaré characteristic [67, p. 146].

Level 1 : The Betti numbers. They are the rank of the free part of the homology groups.

Level 2 : Invariant factor decomposition of the homology groups.

Level 3 : Homology group with generators: $\left(\mathbb{Z}\left[z_1\right] \times \cdots \times \mathbb{Z}\left[z_{\beta_q}\right]\right) \times \mathbb{Z}\left[t_1\right]/\lambda_1\mathbb{Z}\left[c_1\right] \times \mathbb{Z}/\left[t_2\right]\lambda_2\mathbb{Z}\left[c_2\right] \times \cdots$

Level 4 : Homology group with generators and decomposition of cycles.

Each level of homological information can be trivially deduced from the upper ones. We have decided to start from level 0 since the Euler-Poincaré characteristic is the easiest computable homological information. Persistent homology usually works at level 1 (in each complex of the filtration), which is equivalent to level 2 because the ground ring considered is usually a field; Munkres' original theorem/method arrives to level 2 and the *modified-SNF* [102] reaches the third level. Effective homology theory arrives to the fourth level whenever we have a perfect reduction (see Section 2.3.4), since for a given cycle $x \in \ker(d_q)$, $f(x)$ decomposes it onto a linear combination of generators.

This classification could be extended with (co)homology operations, the cohomology ring or even the homotopy groups.

3.3.4   Discrete Morse Theory

Discrete Morse theory was introduced by Robin Forman as a discretization of the Morse theory [50]. One of the main ideas is to obtain some homological information by means of a function defined on a CW complex. This function is equivalent to a discrete gradient vector field and we rather use this notion.

A *discrete vector field* (DVF) on a CW complex is a matching on its Hasse diagram, that is a collection of edges such that no two of them have a common vertex. From a Hasse diagram and a discrete vector field we can define a *Morse graph*: it is a graph similar to the Hasse diagram except for the arrows contained in the matching, which are reversed. These arrows are called *integral arrows*, and the others, *differential arrows*.

Given a DVF over a CW complex K, its cells can be partitioned into the following classes:

Primary (P) The cells having an out-going integral arrow.

Secondary (S) The cells having an in-going integral arrow.

Critical (C) The cells not incident to any integral arrow.

Since the DVF is a matching, it is immediate that $K = P \sqcup S \sqcup C$. This notation is inspired by [91, Def. 1], but this classification was previously introduced in [66, Def. 3.1] and [12, §5] with a different notation.

A $\mathcal{V}$-*path* is a path on the Morse graph that alternates between integral and differential arrows. Its *length* is the number of integral arrows contained. A *discrete gradient vector field* (DGVF) is a discrete vector field that does not contain any closed $\mathcal{V}$-path. As mentioned above, a *critical vertex* (or critical cell) is a vertex that is not paired by the matching. Figure 3.1 shows the usual representation of a DGVF over a cubical complex.

One of the main results of discrete Morse theory is that the number of critical q-cells is greater than or equal to the q-th Betti number. When they are equal, we say that the DGVF is *perfect*. An *optimal* DGVF contains the least possible number of critical cells. Every perfect DGVF is obviously optimal, but the converse is false. Therefore, a DGVF gives an estimation of the Betti numbers without using any algebraic method. We could say that it is a "combinatorial" tool.
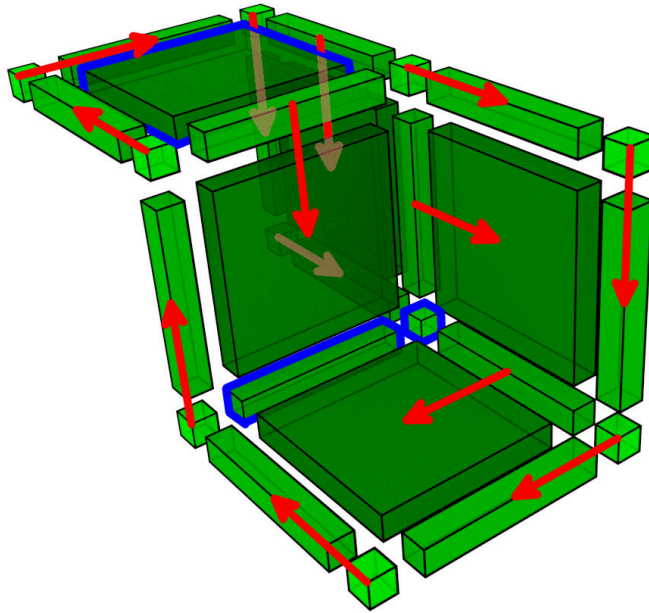

FIGURE 3.1: A DGVF over a cubical complex. The critical cells are highlighted in blue. The integral arrows are shown in red. The differential arrows are omitted.

Given a DGVF $\mathcal{V}$ on a CW complex, we define its *Morse complex* $(\mathfrak{R}[C], d_M)$, where $\mathfrak{R}[C]$ denotes the graded free $\mathfrak{R}$-module generated by the critical cells of $\mathcal{V}$ and d_M is the linear map that sends each critical $(q + 1)$-cell σ to the sum of critical q-cells which are connected to σ by a $\mathcal{V}$-path. An accurate definition of this map (called reduced boundary) will be given in Section 3.5.

Let us point out that starting from a DGVF $\mathcal{V}$, an associated reduction $(h, f, g) : (\mathsf{C}, d) \Rightarrow (f(\mathsf{C}), d)$ can be defined [49, 91]. Firstly, let us define a linear operator V which maps vertices containing an outward integral arrow to the head of this arrow with its sign. Formally,

$$V(\sigma) = \begin{cases} \langle d(\tau), \sigma \rangle \cdot \tau, & (\sigma, \tau) \in \mathcal{V} \\ 0, & \text{otherwise} \end{cases}$$

where $\langle \cdot, \cdot \rangle$ is the inner product associated to the basis of cells. In other words, V maps each primary cell to its secondary cell. Then,

$$h(\sigma) = \sum_{k \geq 0} V(1 - dV)^k(\sigma)$$

$$f(\sigma) = (1 - dh - hd)(\sigma)$$

$$g(\sigma) = \sigma$$

Notice that the sum in the equation for h is actually finite since the DGVF has no cycles. The image of h can be easily interpreted as the sum of the secondary cells in all the $\mathcal{V}$-paths leaving a primary cell. Furthermore, the map f coincides with the *stabilization map* Φ^∞ introduced in [49].

Let us point out that this reduction can be encoded as a matching in the Hasse diagram instead of using a sequence of matrices. Thus, the DGVF actually "compresses" the reduction, which obviously involves a computational cost for its "decompression". This is not a general property of reductions, as the following example shows.

Consider the simplicial complex with one 1-simplex (and its two 0-faces) whose boundary matrix is

$$d_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Then there is a reduction $(h, f, g) : (\mathsf{C}, d) \Rightarrow (f(\mathsf{C}), d_{|f(\mathsf{C})})$, where

$$h_0 = \begin{bmatrix} 2 & 3 \end{bmatrix}, \quad f_0 = \begin{bmatrix} 3 & 3 \\ -2 & -2 \end{bmatrix}, \quad f_1 = \begin{bmatrix} 0 \end{bmatrix}, \quad g = inc$$

In this case we cannot find how to "compress" the reduction as a matching, so we can only explicitly encode it as a sequence of matrices.

3.3.5 Some Matrix Properties

The proofs in this work use several matrix properties that may not be trivial for the reader. We prefer to recall some of them in order to ease the reading of the proofs.

Lemma 3.1. *Let be $A = BCD$ the product of three matrices. Then,*

$$a_{i,j} = B_{i,\cdot} C D_{\cdot,j}$$

Proof. $a_{i,j} = L_i A R_j$, where L_i is a row vector with zeros everywhere except for the i-th position, and R_j is a column vector with zeros everywhere except for the j-th position. Therefore, $a_{i,j} = L_i A R_j = L_i B C D R_j = B_{i,\cdot} C D_{\cdot,j}$. $\square$

Lemma 3.2. *Let be $A \in M_{m \times n}(\mathbb{Z})$, $B \in M_{n \times n}(\mathbb{Z})$, B invertible and $C = A \cdot B^{-1}$. Then,*

$$c_{(i,j)} = \frac{1}{\det(B)} \det(\tilde{B}),$$

where $\tilde{B}$ is the matrix identical to B except for the j-th row, which has been replaced by the i-th row of A.

Lemma 3.3. *(Schur determinant formula) Let be*

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

a block matrix ($A \in M_{n\times n}(\mathbb{Z}), B \in M_{n\times k}(\mathbb{Z}), C \in M_{k\times n}(\mathbb{Z})$ and $D \in M_{k\times k}(\mathbb{Z})$). If A is invertible then

$$\det(M) = \det(A) \cdot \det(D - CA^{-1}B).$$

Lemma 3.4. *(The Banachiewicz identity) Let be*

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

a block matrix ($A \in M_{n\times n}(\mathbb{Z}), B \in M_{n\times k}(\mathbb{Z}), C \in M_{k\times n}(\mathbb{Z})$ and $D \in M_{k\times k}(\mathbb{Z})$). If A and $D - CA^{-1}B$ are invertible, then M is invertible and

$$M^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}$$

We recall that the transpose of a matrix A is denoted $A^{\top}$.

Lemma 3.5. *(Sherman-Morrison formula) Let $A \in M_{n\times n}(\mathbb{Z})$ and $u, v \in \mathbb{Z}^n$. If A is invertible and $1 + v^{\top}Au \neq 0$ then*

$$\left(A + uv^{\top}\right)^{-1} = A^{-1} - \frac{A^{-1}uv^{\top}A^{-1}}{1 + v^{\top}Au}.$$

Lemma 3.6. *Let $A \in M_{m\times n}(\mathbb{Z})$, we say that it is an $[r, c]$-matrix if each row contains at most r non-zero entries and each column contains at most c non-zero entries. Thus,*

- *If $A \in M_{m\times n}(\mathbb{Z})$ is an $[r, c]$-matrix and $B \in M_{m\times n}(\mathbb{Z})$ is an $[r', c']$-matrix, then $A + B$ is an $[r + r', c + c']$-matrix and it can be computed within $\mathcal{O}\left(\min(m \cdot (r + r'), (c + c') \cdot n)\right)$ operations.*

- *If $A \in M_{m\times n}(\mathbb{Z})$ is an $[r, c]$-matrix and $B \in M_{n\times p}(\mathbb{Z})$ is an $[r', c']$-matrix, then $A \cdot B$ is an $[rr', cc']$-matrix and it can be computed within $\mathcal{O}\left(m \cdot \min(r, c') \cdot p\right)$ operations.*

Lemma 3.7. *(Matrix inversion lemma) Let $A \in M_{n\times n}(\mathbb{Z})$ and $u, v \in \mathbb{Z}^n$. If A is invertible then*

$$\det(A + uv^{\top}) = \det(A) \cdot \left(1 + v^{\top}A^{-1}u\right).$$

Proof. We write

$$M = \left[\begin{array}{c|c} 1 & -v \\ \hline u & A \end{array}\right]$$

Since $\det(M) = \det(M^t)$, by the Schur determinant formula (see Lemma 3.3),

$$\det(M) = \det(M^t)$$
$$\det(1) \cdot \det(A + uv) = \det(A) \cdot \det(1 + vA^{-1}u)$$
$$\det(A + uv) = (1 + vA^{-1}u) \cdot \det(A)$$

FIGURE 3.2: Left: an iterated Morse decomposition, where the red arrow belongs to the first DGVF and the purple one, to the second DGVF. Right: a (standard) DGVF inducing the same reduction.

□

3.4 Motivation

The discrete Morse theory approach has a strong interest as it addresses the computation of homology as a purely combinatorial problem rather than an algebraic one. The associated reduction can be encoded just as a list of pairs of cells. It also provides an approximation of the Betti numbers that can sometimes be accurate (depending on the choice of the integral arrows) but that is always wrong for some well known spaces as, for instance, the Bing's house [6] (also called house with two rooms) or the dunce hat [117].

We can increase a DGVF (and thus improve the approximation) by canceling pairs of critical cells: find two critical cells $\tau^{(q+1)}$ and $\sigma^{(q)}$ connected by only one $\mathcal{V}$-path and exchange the integral and differential arrows in this path. This can be seen as reversing the direction of the $\mathcal{V}$-path. Note that, even though this transformation is expressed in combinatorial terms, computing the number of $\mathcal{V}$-paths is equivalent to compute the associated reduction.

Another approach for reducing the number of critical cells is to compute the Morse complex and to establish a new DGVF $\mathcal{V}'$ on it, which is useful when there is no unique $\mathcal{V}$-path between the critical cells. This is known as *iterated Morse decomposition* [42]. Regarding the associated reduction, reversing the only $\mathcal{V}$-path between two critical cells is equivalent to adding an integral arrow between them in the Morse complex. Figure 3.2 illustrates this.

Thus, reversing a $\mathcal{V}$-path can be seen as pushing an integral arrow from the Morse complex back to the original one. However, not all the integral arrows on the Morse complex are equivalent to reverse a $\mathcal{V}$-path: this is the case when there are several $\mathcal{V}$-paths between two critical cells. Figure 3.3 shows an example where there are three $\mathcal{V}$-paths between two critical cells. However, the 1-cell is a face of the 2-cell in the associated Morse complex, so we can add an integral arrow which does not correspond to a unique $\mathcal{V}$-path. The motivation for our work was to push all the integral arrows in the Morse complex back to the original one.

There is a different (but equivalent) point of view which is more surprising. Finding an optimal DGVF, with the minimal number of critical cells, is an NP problem. Canceling pairs of critical cells by reversing $\mathcal{V}$-paths could
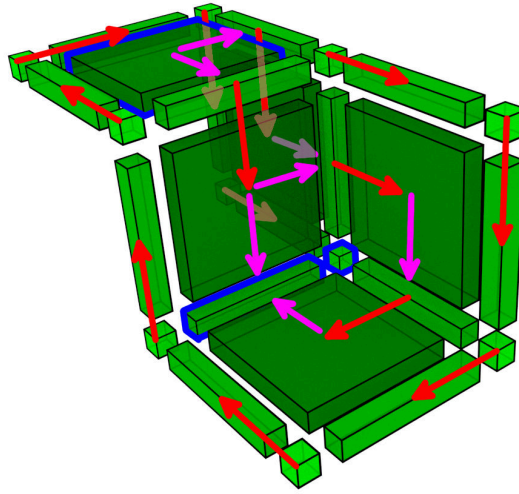
FIGURE 3.3: The same DGVF depicted in Figure 3.1. Some
differential arrows are shown in purple.

seem to be a solution to this problem, but we cannot do it in general be-
cause of the conditions in the definition of a DGVF. Thus, one could think
of removing one of them:

1. It must be a matching: if there are several $\mathcal{V}$-paths between two crit-
   ical cells, we could think of reversing all of them. Both critical cells
   would disappear and no cycles would thus appear. Sadly enough,
   this idea does not seem to give any homological information. We
   cannot affirm that this approach is impossible, since extra conditions
   could be added, but we can show a very discouraging example at Fig-
   ure 3.4. On the left, there is a DGVF with 3 $\mathcal{V}$-paths between the two
   leftmost critical cells of dimension 1 and 2. If we reverse all of them,
   there is just one $\mathcal{V}$-path between the two rightmost critical cells. If we
   cancel them, we would finish with just one critical cell of dimension
   0, while the complex has $\beta_1 = \beta_2 = 1$.

   A more detailed description of this example would take too much
   space, and we only wish to show that this does not seem a good idea.

2. There cannot be closed $\mathcal{V}$-paths: miraculously, this has been a success-
   ful idea. Only by adding one condition that we introduce in Sec-
   tion 3.6, we obtain a generalization of the DGVF. The methods for
   constructing such an object and the equations for computing its asso-
   ciated reduction are valid for a standard DGVF. We call this kind of
   DVF a *homological discrete vector field* (HDVF).

   Integrating all the integral arrows in the Hasse diagram of the original
complex is not a simple challenge. Nonetheless, it has already been noted
that a reduction from a CW complex can benefit from its geometric realiza-
tion and, in our opinion, this is the real advantage of discrete Morse theory.
Let us point out a few examples supporting this rather informal affirmation:
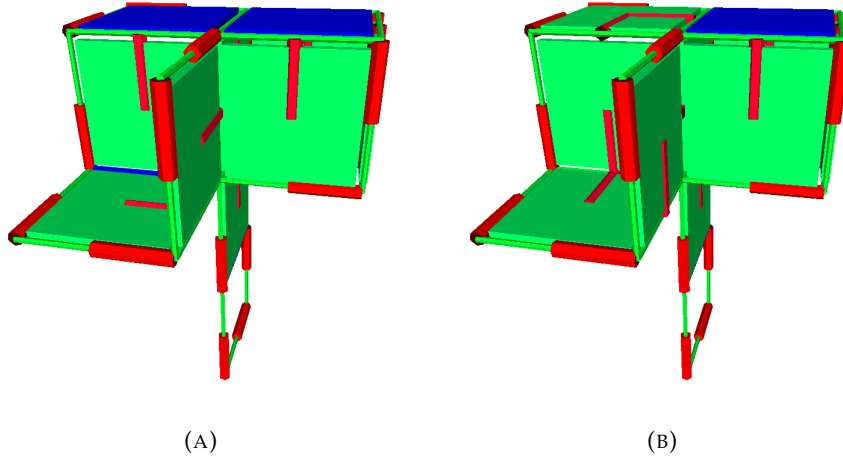
(A)   (B)

FIGURE 3.4: (a) A DGVF. (b) The result after reversing all
three $\mathcal{V}$-paths between the two leftmost critical cells.

- We have some a priori information about the boundary matrices of a simplicial complex: the columns of the matrix d_q have exactly q non-zero entries. Moreover, the boundary matrix d_q of a cubical complex has $2q$ non-zero entries in its columns and less than or equal to $2(n-q)$ in its rows, where n is the dimension in which the cubical complex is embedded.

- In [91, §6] a parallel method for establishing a DGVF was introduced for cubical complexes. This method seems impossible to extend to other kinds of CW complexes, so it is really based on the geometry of the complex. In terms of the reduction, it can be seen as doing a partial parallel diagonalization of the boundary matrices. Extending this approach to general chain complexes is not at all clear.

- Given an n-dimensional cubical complex (that is, a cubical complex embedded in $\mathbb{R}^n$), it is not difficult to set a DGVF such that the homology generators of dimension $n-1$ lie on the boundary of the complex. We can identify the $(n-1)$-holes of the complex by considering its complement. Choose a $(n-1)$-cell on the boundary of the complex next to one of those holes, and add integral arrows starting from its boundary, covering all that part of the boundary. Repeat this step for every hole and then cancel the remaining critical cells without modifying these integral arrows.

  This is not easily generalizable to other classes of CW complexes. Such an idea, that we could name as "modeling" or "shaping" the homology generators makes no sense when we establish a reduction from a general chain complex.

3.5 Introducing the HDVF

In the context of discrete Morse theory, we always try to set a DGVF with the maximum number of integral arrows (or equivalently, with the minimum number of critical cells) in order to obtain the best possible approximation of the Betti numbers. In the language of effective homology theory, the induced reduction greatly "reduces" the original chain complex.

Given a DGVF, we can improve it by incrementing the number of integral arrows. If we find two critical cells $\sigma < \tau$, such that inserting an integral arrow between them does not create a cycle, adding this integral arrow reduces by two the number of critical cells.

More generally, if there is only one $\mathcal{V}$-path between one cell σ' belonging to the boundary of a critical cell τ and another critical cell σ, we can reverse it and add the arrow (σ', τ). This means that the integral and differential arrows in the $\mathcal{V}$-path are exchanged. This can be considered as the general method for improving a DGVF (actually, in the previous case, the $\mathcal{V}$-path has length zero so there is no reversing). However, depending on the order in which we cancel the critical cells and on the CW complex itself, we can create several $\mathcal{V}$-paths between the other pairs of critical cells, so that we cannot cancel them anymore. This gives an intuition on why this optimization problem is NP [83].

In order to avoid this situation, we propose to allow cycles in the DVF, provided that we create them "smartly", so a reduction can still be defined. We cancel pairs of critical cells independently of the number of $\mathcal{V}$-paths, but considering the information given by the associated reduction. This means that the reduction must be known at every step, but do not panic: finding a $\mathcal{V}$-path amounts also to compute a reduction.

We recall that a DVF induces a partition $K = P \sqcup S \sqcup C$ of a CW complex.

Definition 3.1. *A homological discrete vector field (HDVF) $X = (P, S)$ on a CW complex K is a partition $K = P \sqcup S \sqcup C$ such that $d(S_{q+1})_{|P_q}$ is an invertible matrix (in $\mathfrak{R}$) for every $q \geq 0$, where P_q and S_q denote the restrictions of P and S to the q-cells and $d(S_{q+1})_{|P_q}$ is the submatrix of the boundary matrix d_{q+1} consisting in the columns associated to the secondary $(q + 1)$-cells and the rows associated to the primary q-cells.*

Note that the DVF is not explicit in the definition of the HDVF. When X is a DGVF, there is a unique DVF inducing its partition, but this is not the case for a HDVF. For instance, Figure 3.5 depicts three different DVFs inducing the same HDVF, since the primary and secondary cells in each complex are the same.

Deducing a DVF requires to find a perfect matching in a bipartite graph. The existence of this perfect matching, when the partition is a HDVF, follows from Proposition 3.8.

Proposition 3.8. *Let K be a CW complex endowed with a HDVF $X = (P, S)$. Then there exists a discrete vector field $\mathcal{V}$ that induces the partition $K = P \sqcup S \sqcup C$.*

Proof. In this proof we do not use the fact that $d(S_{q+1})_{|P_q}$ is invertible, but that $\det\left(d(S_{q+1})_{|P_q}\right) \neq 0$.

Let us fix a dimension q. By the Laplace expansion formula, there is a pair of cells (σ, τ) such that $\langle d(\tau), \sigma \rangle \neq 0$ and $\det\left(d(S_{q+1} \setminus \tau)_{|P_q \setminus \sigma}\right) \neq 0$. Thus, the discrete vector field $\mathcal{V}$ can be found recursively. $\qquad\square$
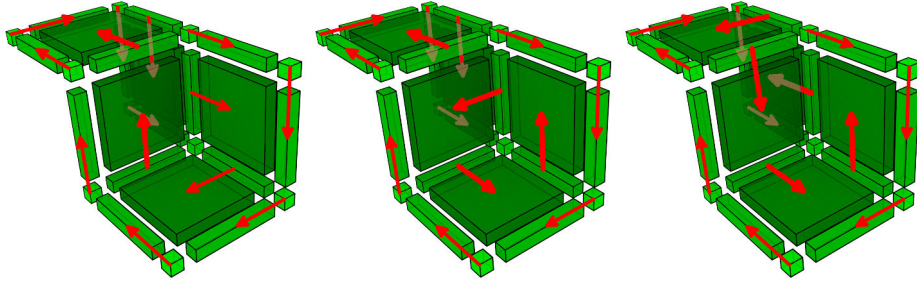
FIGURE 3.5: Three different matchings inducing the same HDVF.

The DVF can be computed using the Hopcroft-Karp algorithm [69] in $\mathcal{O}(m\sqrt{n})$ time, where n and m denote the number of vertices and edges in the Hasse diagram. It is interesting as it allows us to *visualize* the HDVF and its computation.

Let us now present the reduction induced by a HDVF. We showed in Section 3.3.4 a reduction induced by a DGVF. Since a DGVF has no cycles, the chain $(1 - dV)$ is nilpotent and hence the sum $\sum_{k \geq 0} V(1 - dV)^k$ is well defined. This does not hold for the HDVF, and therefore we must consider an appropriate reduction.

Note that all the operators of a reduction are linear, so they can be represented by matrices. An appropriate choice of bases can provide nice matrices and we have found a very good one: the basis $\mathcal{B} = \langle P_q, S_q, C_q \rangle$ for every chain group C_q. In the following we omit the subscripts to facilitate readability.

Theorem 3.9. *Let K be a CW complex endowed with a HDVF X. Then X induces the reduction* $(h, f, g) : (\mathsf{C}, d) \Rightarrow (\mathfrak{R}[C], d')$, *where the operators h, f, g and the reduced boundary d' are given by*

$$
h = \begin{array}{c|c|c|c}
 & P & S & C \\
\hline
P & 0 & 0 & 0 \\
\hline
S & H & 0 & 0 \\
\hline
C & 0 & 0 & 0 \\
\end{array}
\qquad
f = \begin{array}{c|c|c|c}
 & P & S & C \\
\hline
C & F & 0 & I \\
\end{array}
\qquad
g = \begin{array}{c|c}
 & C \\
\hline
P & 0 \\
\hline
S & G \\
\hline
C & I \\
\end{array}
\qquad
d' = \begin{array}{c|c}
 & C \\
\hline
C & D \\
\end{array}
$$

where

$$
\begin{aligned}
H &= (d(S)_{|P})^{-1} \\
F &= -d(S)_{|C} \cdot (d(S)_{|P})^{-1} \\
G &= -(d(S)_{|P})^{-1} \cdot d(C)_{|P} \\
D &= d(C)_{|C} + F \cdot d(C)_{|P} = d(C)_{|C} + d(S)_{|C} \cdot G
\end{aligned}
$$

Proof. Let us see that these linear operators satisfy the conditions of a reduction. By developing the matrix products by blocks, we can easily check that $hh = 0$, $fh = 0$, $hg = 0$ and $fg = 1_C$. The rest of the conditions precise more detail.

♦ $gf = 1_{\mathsf{C}} - dh - hd$: By developing the matrix product, we obtain

$$
\left[\begin{array}{c|c|c}
0 & 0 & 0 \\ \hline
GF & 0 & G \\ \hline
F & 0 & I
\end{array}\right]
=
\left[\begin{array}{c|c|c}
I - d(S)_{|P}H & 0 & 0 \\ \hline
-d(S)_{|S}H - Hd(P)_{|P} & I - Hd(S)_{|P} & -Hd(C)_{|P} \\ \hline
-d(S)_{|C}H & 0 & I
\end{array}\right]
$$

All the equalities can be deduced directly from the definition of H, F and G. The equality $GF = -d(S)_{|S}H - Hd(P)_{|P}$ is more difficult to see. Let us call

$$
\begin{aligned}
X &= GF + d(S)_{|S}H + Hd(P)_{|P} \\
&= Hd(C)_{|P}d(S)_{|C}H + d(S)_{|S}H + Hd(P)_{|P}
\end{aligned}
$$

Then,

$$
\begin{aligned}
d(S)_{|P}Xd(S)_{|P} &= \big[d(S)_{|P}H\big]\,d(C)_{|P}d(S)_{|C}\,\big[Hd(S)_{|P}\big] \\
&\quad + d(S)_{|P}d(S)_{|S}\,\big[Hd(S)_{|P}\big] + \big[d(S)_{|P}H\big]\,d(P)_{|P}d(S)_{|P} \\
&= d(C)_{|P}d(S)_{|C} + d(S)_{|P}d(S)_{|S} + d(P)_{|P}d(S)_{|P}
\end{aligned}
$$

Then, the reader can check that $d(S)_{|P}Xd(S)_{|P} = (dd)(S)_{|P} = 0$, so $X = 0$.

We need now some properties whose proof is direct by developing the matrix product:

$$
d' = fdg = fd\left[\begin{array}{c} 0 \\ \hline 0 \\ \hline I \end{array}\right] = \big[\; 0 \mid 0 \mid I \;\big]\,dg \tag{3.1}
$$

$$
f = \big[\; 0 \mid 0 \mid I \;\big] \cdot (1_{\mathsf{C}} - dh) \tag{3.2}
$$

$$
g = (1_{\mathsf{C}} - hd) \cdot \left[\begin{array}{c} 0 \\ \hline 0 \\ \hline I \end{array}\right] \tag{3.3}
$$

♦ $d'f = fd$. Using (3.1) and (3.2),

$$
\begin{aligned}
d'f &= \big[\; 0 \mid 0 \mid I \;\big]\,dgf \\
&= \big[\; 0 \mid 0 \mid I \;\big]\,d(1_{\mathsf{C}} - dh - hd) \\
&= \big[\; 0 \mid 0 \mid I \;\big]\,(1_{\mathsf{C}} - dh)d = fd
\end{aligned}
$$

♦ $gd' = dg$. Symmetrically,

$$gd' = gfd \left[\begin{array}{c} 0 \\ \hline 0 \\ \hline I \end{array}\right]$$

$$= (1_{\mathsf{C}} - dh - hd)d \left[\begin{array}{c} 0 \\ \hline 0 \\ \hline I \end{array}\right]$$

$$= d(1_{\mathsf{C}} - hd) \left[\begin{array}{c} 0 \\ \hline 0 \\ \hline I \end{array}\right] = dg$$

♦ $d'd' = 0$. Using (3.2) and (3.3),

$$d'd' = (fdg)(fdg) = fdg(d'f)g = f(dd)gfg = 0$$

□

We say that a HDVF is *perfect* if its associated reduction is perfect.

The previous theorem allows us to prove the desired property that the number of critical cells approximates the Betti numbers also in the HDVF.

Theorem 3.10. *Let K be a CW complex endowed with a HDVF X. Then, for every $q \geq 0$, the number of q-critical cells is greater than its q-th Betti number.*

Proof. A HDVF induces a reduction to a chain complex C' with isomorphic homology groups, whose rank in each dimension q is the number of critical q-cells. This proves the theorem. □

Let us point out that this reduction is not directly a generalization of the reduction introduced in [91]. Though, it has a similar form if we consider the reduction $\rho' = (h', f', g') = (h, 1 - dh - hd, \iota)$ between (C, d) and $(f'(\mathsf{C}), d)$.

Using the same language as discrete Morse theory, this class extending the DGVF allows us to find the correct number of critical cells in complexes which do not admit a perfect DGVF, such as the Bing's house or the dunce hat [3]. Instead of providing the explicit (and enormous) description of each complex and its HDVF, we prefer to show illustrations and to comment the construction of those HDVFs.

The cubical complex version of the Bing's house has been created by the authors. It contains 60 0-cubes, 129 1-cubes and 70 2-cubes. The first DGVF defined on it contains 13 critical cubes (see Figure 3.6-(a)): 1 0-cube, 6 1-cubes and 6 2-cubes. Let us comment that it is not the best DGVF possible. Starting from this DGVF, and after canceling pairs of critical cells by reversing $\mathcal{V}$-paths, it remains only 1 critical 0-cube, which corresponds to the Betti numbers of the complex. Obviously, these $\mathcal{V}$-paths were chosen to preserve the HDVF structure. Consequently, the Morse graph contains two cycles.

For the dunce hat we used a simplicial complex from [64] consisting of 8 0-simplices, 24 1-simplices and 17 2-simplices. We can set a DGVF
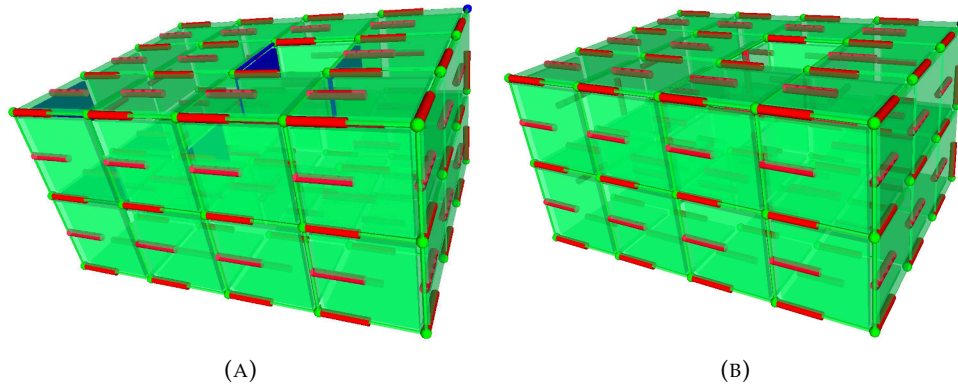
FIGURE 3.6: (a) A DGVF over the Bing's house. (b) A perfect HDVF obtained on the Bing's house. There is only one critical 0-cell (in blue).

containing 3 critical cells (see Figure 3.7-(a)): one of each dimension. After reversing one $\mathcal{V}$-path between the critical cells of dimension 1 and 2, we obtain a HDVF with only 1 critical 0-cell, which is in accordance with the Betti numbers of the complex. The two cycles created in the homological DVF are shown in green.
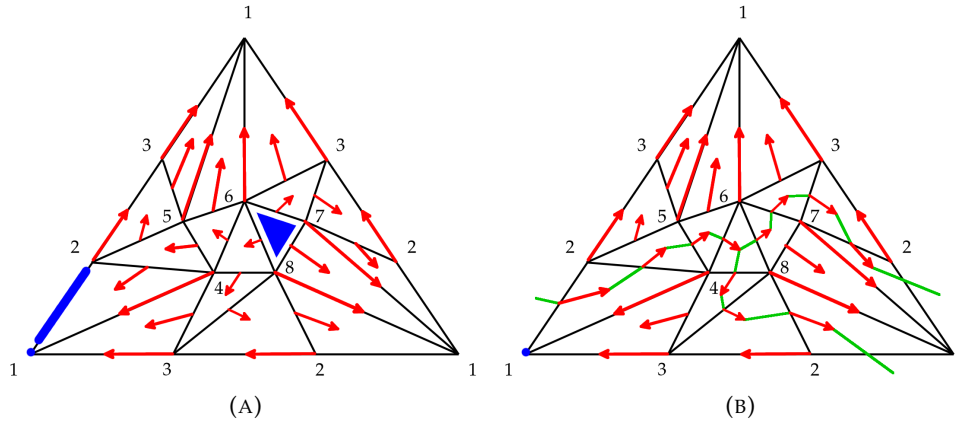

FIGURE 3.7: (a) A DGVF over the dunce hat with three critical cells in blue. (b) The HDVF obtained after improving the DGVF. The only critical cell is the 0-cell denoted by 1. The two cycles in the Morse graph are displayed in green.

3.6 Computing a HDVF

We explain in this section how we can compute a HDVF and its reduction efficiently. We do it in terms of the partition $K = P \sqcup S \sqcup C$ instead of the DVF $\mathcal{V}$, but we briefly describe how to obtain the DVF.

3.6.1 Computing the Reduced Complex

Our first proposition states when we can add a pair of cells to a HDVF so that the matrix $d(S)_{|P}$ is still invertible.

Proposition 3.11. *Let K be a CW complex endowed with a HDVF $X = (P, S)$. Let $\sigma^{(q)}$ and $\tau^{(q+1)}$ be two critical cells. If $\langle d'(\tau), \sigma \rangle$ is a unit then $X' = (P \cup \{\sigma\}, S \cup \{\tau\})$ is a HDVF.*

Proof. We only need to prove that the matrix $d(S')_{|P'}$ is invertible, where $S' = S \cup \{\tau\}$ and $P' = P \cup \{\sigma\}$. This matrix has the form

$$
d(S')_{|P'} = \quad
\begin{array}{c|c|c}
 & S & \tau \\
\hline
P & d(S)_{|P} & v \\
\hline
\sigma & u & w
\end{array}
$$

where $u = d(S)_{|\sigma}$, $v = d(\tau)_{|P}$ and $w = d(\tau)_{|\sigma}$.

We know that $d(S)_{|P}$ is invertible. Let us prove that $w - u(d(S)_{|P})^{-1}v$ is also invertible. By hypothesis, $\langle d'(\tau), \sigma \rangle = \pm 1$. Since

$$
D = d(C)_{|C} - d(S)_{|C} \cdot (d(S)_{|P})^{-1} \cdot d(C)_{|P}
$$

then, by Lemma 3.1 (without specifying the indices),

$$
\begin{aligned}
\langle d'(\tau), \sigma \rangle &= d(\tau)_{|\sigma} - d(S)_{|\sigma} \cdot (d(S)_{|P})^{-1} \cdot d(\tau)_{|P} \\
&= w - u \cdot (d(S)_{|P})^{-1} \cdot v
\end{aligned}
$$

Consequently, by the *Schur determinant formula* (c.f. Lemma 3.3),

$$
\det(d(S')_{|P'}) = \det(d(S')_{|P'}) \cdot \det(w - u(d(S)_{|P})^{-1}v)
$$

is a unit, so $d(S)_{|P}$ is invertible. $\qquad\square$

Once we have added two critical cells to a HDVF, we do not need to comute a new DVF inducing the expanded HDVF. Instead of this, we can deduce the corresponding DVF by inverting one of the $\mathcal{V}$-paths connecting both critical cells. The following proposition proves that such $\mathcal{V}$-path exists.

Proposition 3.12. *Let K be a CW complex endowed with a HDVF X. Let $\sigma^{(q)}$ and $\tau^{(q+1)}$ be two critical cells. If $\langle d'(\tau), \sigma \rangle$ is a unit then there is a $\mathcal{V}$-path between them.*

Proof. Let V denote the matrix associated with the DVF introduced in Section 3.3.4. Thus,

$$
\begin{aligned}
d'(C) &= d(C)_{|C} - d(S)_{|C} \cdot (d(S)_{|P})^{-1} \cdot d(C)_{|P} \\
&= d(C)_{|C} - d(S)_{|C} \cdot V(P)_{|S} \cdot (V(P)_{|S})^{-1} \cdot (d(S)_{|P})^{-1} \cdot d(C)_{|P} \\
&= d(C)_{|C} - dV(P)_{|C} \cdot (dV(P)_{|P})^{-1} \cdot d(C)_{|P}
\end{aligned}
$$

Hence $\langle d'(\tau), \sigma \rangle = d'(\tau)_{|\sigma} = -dV(P)_{|\sigma} \cdot (dV(P)_{|P})^{-1} \cdot d(\tau)_{|P} + d(\tau)_{|\sigma}$. If $\sigma < \tau$ then it is obvious. Otherwise, $\langle d'(\tau), \sigma \rangle = -dV(P)_{|\sigma} \cdot (dV(P)_{|P})^{-1} \cdot d(\tau)_{|P}$.

As this term is non zero, there must be some σ_i, σ_j in P such that

$$dV(\sigma_i)_{|\sigma} \neq 0$$
$$d(\tau)_{|\sigma_j} \neq 0$$
$$(dV(P)_{|P})^{-1}_{(i,j)} \neq 0$$

The first two inequalities imply that there exist two $\mathcal{V}$-paths $\sigma_i \nearrow b \searrow \sigma$ and $\tau \searrow \sigma_j$.

The third one implies that there is a path from σ_j to σ_i. Let us see a short proof of this in a more general context.

Let A be the adjacency matrix of a weighted digraph such that $\det(A) \neq 0$. We know that $A^k_{(i,j)} \neq 0, i \neq j$ implies that there is a path from the vertex j to i. Following the Cayley-Hamilton theorem [51, §4.4, Thm. 2],

$$A^n + c_{n-1}A^{n-1} + \cdots c_1 A + (-1)^n \det(A) I_n = 0$$
$$\Rightarrow A^{-1} = \frac{(-1)^{n+1}}{\det(A)} \left(A^{n-1} + c_{n-1}A^{n-2} + \cdots c_1 I_n \right)$$

Thus, $A^{-1}_{(i,j)} \neq 0 \Rightarrow \exists k \geq 0, A^k_{(i,j)} \neq 0$.

Then, there is a $\mathcal{V}$-path $\sigma_j \nearrow b_1 \searrow a_2 \cdots \searrow \sigma_i$.
By concatenating these paths, we obtain

$$\tau \searrow \sigma_j \nearrow b_1 \searrow a_2 \cdots \searrow \sigma_i \nearrow b \searrow \sigma$$

$\square$

Algorithm 1 gives a general pipeline for computing a HDVF.

Algorithm 1: Compute a HDVF

Input: A CW complex K
Output: A HDVF X
1 $X \leftarrow (\emptyset, \emptyset)$;
2 **repeat**
3 | Find two critical cells σ, τ such that $\langle d'(\tau), \sigma \rangle$ is a unit;
4 | Add (σ, τ) to X;
5 | Update the boundary matrices D of the reduced chain complex;
6 **until** *idempotency*;

If we also want to obtain the DVF, for each pair of cells (σ, τ) that we add to the HDVF we have to find a $\mathcal{V}$-path between them and reverse it.

The core of Algorithm 1 lies at line 5. We now present three methods for updating the matrix D_{q+1} after adding a pair of critical cells $(\sigma^{(q)}, \tau^{(q+1)})$ and we study their complexity.

Let us now point out an important aspect about the complexity. We denote by n the number of cells in the CW complex K. If K is a simplicial complex and D_q is its initial (not reduced) boundary matrix, then the number of non-zero entries in each column of D_q is q. Also, if K is a cubical complex embedded in $\mathbb{R}^d$ then each column of D_q has $2q$ non-zero entries

and each row contains at most $2(d-q)$ non-zero entries. Therefore, it is interesting not only to consider dense boundary matrices, but also sparse ones along their columns or along their rows and columns. We denote these three types of matrices *dense*, *d-bounded* and *dd*-bounded* respectively.

A common point between the following three methods is that we must remove the row of τ in D_{q+2} and the column of σ in D_q.

Method I: inverting H

Given the equations for the reduction associated to a HDVF X (cf. Theorem 3.9), the most trivial way to update the boundary matrices is to invert the new matrix $d(S'_{q+1})_{|P'_q}$ and to compute

$$D'_{q+1} = d(C'_{q+1})_{|C'_q} - d(S'_{q+1})_{|C'_q} \cdot (d(S'_{q+1})_{|P'_q})^{-1} \cdot d(C'_{q+1})_{|P'_q}$$

We estimate the complexity of this operation. Remark that all these matrices have at most n columns and n rows. Inverting the matrix $d(S'_{q+1})_{|P'_q}$ can be done in matrix multiplication time, so it requires $\mathcal{O}(n^\omega)$ operations, where $\omega \leq 2.374$ [26].

In order to understand the complexity of this method in the context of the three types of boundary matrices, we recall the following notation. A matrix $A \in M_{n \times n}(\mathbb{Z})$ is called $[r,c]$-matrix if each row contains at most r non-zero entries and each column contains at most c non-zero entries. Thus, dense matrices are $[n,n]$-matrices, *d-bounded* matrices are $[n,c]$-matrices and *dd*-bounded* are $[r,c]$-matrices for some constants r and c. Lemma 3.6 states the complexity of the sum and the multiplication of $[r,c]$-matrices.

Let us suppose that the three matrices $d(C'_{q+1})_{|C'_q}$, $d(S'_{q+1})_{|C'_q}$ and $d(C'_{q+1})_{|P'_q}$ are $[r,c]$-matrices, where $K = P' \sqcup S' \sqcup C'$ denotes the new partition and D' is the new reduced boundary after adding (σ, τ) to X. Thus we can obtain D'_{q+1} by performing $\mathcal{O}(n^2 \cdot (c+r))$ operations:

$$D'_{q+1} = \underbrace{d(C'_{q+1})_{|C'_q}}_{[r,c]} - \underbrace{d(S'_{q+1})_{|C'_q}}_{[r,c]} \cdot \underbrace{(d(S'_{q+1})_{|P'_q})^{-1}}_{[n,n]} \cdot \underbrace{d(C'_{q+1})_{|P'_q}}_{[r,c]}$$

$$[r,c] + ([r,c] \cdot [n,n]) \cdot [r,c]$$

$$[r,c] + [n,n] \cdot [r,c] \quad (n^2 c \text{ operations})$$

$$[r,c] + [n,n] \quad (n^2 r \text{ operations})$$

$$[n,n] \quad (n^2 \text{ operations})$$

Consequently, if the boundary matrices are dense, *d-bounded* or *dd*-bounded*, the complexity of this method is $\mathcal{O}(n^3)$, $\mathcal{O}(n^3)$ and $\mathcal{O}(n^{2.374})$ respectively.

Method II: using the Banachiewicz formula for H

Let us see how we can obtain $(d(S)_{|P})^{-1}$ without inverting the matrix. In the following we omit the subscripts whenever they are clear from the context.

Proposition 3.13. *After adding the pair of critical cells (σ, τ), the matrix $(d(S')_{|P'})^{-1}$ can be obtained within $\mathcal{O}(n^2)$ operations.*

Proof. We write

$$H_q = \begin{bmatrix} A \end{bmatrix}^{-1} \qquad\qquad =: H$$

$$F_q = -\begin{bmatrix} u \\ \hline B \end{bmatrix} \cdot H_q = -\begin{bmatrix} uH \\ \hline BH \end{bmatrix} \qquad =: \begin{bmatrix} F_{11} \\ \hline F_{21} \end{bmatrix}$$

$$G_{q+1} = -H_q \cdot \begin{bmatrix} v \mid C \end{bmatrix} = -\begin{bmatrix} Hv \mid HC \end{bmatrix} \quad =: \begin{bmatrix} G_{11} \mid G_{12} \end{bmatrix}$$

$$D_{q+1} = -\begin{bmatrix} uH \\ \hline BH \end{bmatrix} \cdot \begin{bmatrix} v \mid C \end{bmatrix} + \begin{bmatrix} w & s \\ \hline r & E \end{bmatrix} =$$

$$= \begin{bmatrix} -uHv + w & -uHC + s \\ \hline -BHv + r & -BHC + E \end{bmatrix} \qquad =: \begin{bmatrix} D_{11} \mid D_{12} \\ \hline D_{21} \mid D_{22} \end{bmatrix}$$

where

$$A = d(S_{q+1})_{|P_q}$$
$$u = d(S_{q+1})_{|\sigma} \qquad B = d(S_{q+1})_{|C'_q} \qquad r = d(\tau)_{|C'_q}$$
$$v = d(\tau)_{|P_q} \qquad C = d(C'_{q+1})_{|P_q} \qquad s = d(C'_{q+1})_{|\sigma}$$
$$w = d(\tau)_{|\sigma} \qquad\qquad E = d(C'_{q+1})_{|C'_q}$$

Remark that $D_{11} = \langle d'(\tau), \sigma \rangle$ is a unit, so D_{11}^{-1} exists. By the Banachiewicz identity (cf. Lemma 3.4),

$$H'_q = \begin{bmatrix} A & v \\ \hline u & w \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} H + HvD_{11}^{-1}uH & -HvD_{11}^{-1} \\ \hline -D_{11}^{-1}uH & D_{11}^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} H + G_{11}D_{11}^{-1}F_{11} & G_{11}D_{11}^{-1} \\ \hline D_{11}^{-1}F_{11} & D_{11}^{-1} \end{bmatrix}$$

The complexity of this method is dominated by the computation of the upper-left block, which requires $\mathcal{O}(n^2 + n(c + r)) = \mathcal{O}(n^2)$ operations:

$$H + HvD_{11}^{-1}uH$$
$$[n, n] + ([n, n] \cdot [1, c]) \cdot ([r, 1] \cdot [n, n])$$
$$[n, n] + [1, n] \cdot [n, 1] \quad (nc + nr \text{ operations})$$
$$[n, n] + [n, n] \quad (n^2 \text{ operations})$$
$$[n, n] \quad (n^2 \text{ operations})$$

$\square$

Thus, the boundary matrices can be obtained within $\mathcal{O}\left(n^2 + n^2(c + r)\right) = \mathcal{O}\left(n^2(c + r)\right)$ operations. Consequently, if the boundary matrices are dense, d-bounded or dd^*-bounded, the complexity of this method is $\mathcal{O}(n^3)$, $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2)$ respectively. This means that this method is theoretically better for a cubical complex.

Method III: continuing that idea

Following the notation in Proposition 3.13, we can directly obtain D_{q+1} without computing H_q.

Proposition 3.14. *The matrix D_{q+1} can be obtained within $\mathcal{O}(n^2)$ operations.*

Proof. Following the notation of the proof of Proposition 3.13,

$$
\begin{aligned}
D'_{q+1} &= d(C'_{q+1})_{|C'_q} - d(S'_{q+1})_{|C'_q} \cdot (d(S'_{q+1})_{|P'_q})^{-1} \cdot d(C'_{q+1})_{|P'_q} \\
&= E - \begin{bmatrix} B \mid r \end{bmatrix} \left[\begin{array}{c|c} H + HvD_{11}^{-1}uH & -HvD_{11}^{-1} \\ \hline -D_{11}^{-1}uH & D_{11}^{-1} \end{array} \right] \begin{bmatrix} C \\ s \end{bmatrix} \\
&= E - BHC - BHvD_{11}^{-1}uHC + rD_{11}^{-1}uHC + BHvD_{11}^{-1}s - rD_{11}^{-1}S \\
&= (E - BHC) - (r - BHv)D_{11}^{-1}(s - uHC) \\
&= D_{22} - D_{21}D_{11}^{-1}D_{12}
\end{aligned}
$$

Consequently, D'_{q+1} can be obtained within $\mathcal{O}(n^2)$ operations. $\qquad\square$

Table 3.1 compares the three methods against the three possible types of boundary matrices.

	Method I	Method II	Method III
Dense	n^3	n^3	n^2
d-bounded	n^3	n^3	n^2
dd^*-bounded	$n^{2.373}$	n^2	n^2

TABLE 3.1: Comparison of the three methods.

Thus the third method outperforms the two others for each type of boundary matrix. We can now formulate the complexity of Algorithm 1.

Theorem 3.15. *Algorithm 1 can be computed within $\mathcal{O}(n^3)$ operations.*

Proof. Let us consider the worst case in which we found a perfect HDVF and no critical cell remains (which is impossible since there is at least one connected component). Thus we add $n/2$ pairs of cells. Finding a pair of cells consists in choosing a unit in the boundary matrices, so it can be done within $\mathcal{O}(n^2)$ operations. Then, by using the third method, the complexity of the algorithm is $\mathcal{O}\left(\frac{n}{2}(n^2 + n^2)\right) = \mathcal{O}(n^3)$. $\qquad\square$

Note that this result does not depend (and does not take advantage) of the boundary matrices type. Computing also the DVF does not affect the complexity of the algorithm theoretically. We can find and reverse a $\mathcal{V}$-path in $\mathcal{O}(n^2)$ time, so obtaining the DVF associated to the HDVF requires also at most $\mathcal{O}(n^3)$ operations.

In Algorithm 1 we do not propose any rule for choosing the pair of critical cells. It could be convenient to choose a pair (σ, τ) such that $D_{12} = 0$ or $D_{21} = 0$, so updating the reduced boundary D is simpler. This corresponds to an *elementary reduction* (τ is the only coface of σ) or an *elementary coreduction* (σ is the only face of τ) [94]. It has been noted that it is preferable to look for elementary coreductions than for elementary reductions [95, 39, 66, 76].

3.6.2 Computing also the reduction

At some point, it may be interesting to obtain the reduction induced by the HDVF. We have seen with the second method that it is better not to invert $d(S)_{|P}$, so H may be computed at each step using the formula in Proposition 3.13. Then, F and G may be computed at the end of the algorithm using the formula in Theorem 3.9, which needs $\mathcal{O}(n^3)$ operations if the boundary matrices are not dd^*-bounded. However, if we want to know the operators f and g throughout the computation of the HDVF it is better to use the following result.

Proposition 3.16. *After adding the pair of critical cells $(\sigma^{(q)}, \tau^{(q+1)})$, the matrices F'_q and G'_{q+1} can be obtained within $\mathcal{O}(n^2)$ operations.*

Proof. Using the notation introduced in Proposition 3.13, it is easy to prove that

$$
\begin{aligned}
F'_q &= - \begin{bmatrix} B & | & r \end{bmatrix} \cdot H'_q \\
&= \begin{bmatrix} F_{21} - D_{21}D_{11}^{-1}F_{11} & | & -D_{21}D_{11}^{-1} \end{bmatrix} \\
G'_{q+1} &= -H'_q \cdot \begin{bmatrix} C \\ \hline s \end{bmatrix} \\
&= \begin{bmatrix} G_{12} - G_{11}D_{11}^{-1}D_{12} \\ \hline -D_{11}^{-1}D_{12} \end{bmatrix}
\end{aligned}
$$

Thus, we can update f and g within $\mathcal{O}(n^2)$ operations. $\qquad\square$

Consequently, a HDVF and its reduction can be computed within $\mathcal{O}(n^3)$ operations.

3.6.3 Some questions about the algorithm

We partially answer some questions concerning Algorithm 1 in this section.

Question 1: does Algorithm 1 return a perfect HDVF?

First of all, we recall that a CW complex with a torsion subgroup in one of its homology groups does not admit a perfect HDVF, since a perfect reduction involves homology groups of the form $\mathbb{Z}^\beta$.

Moreover, let us show that Algorithm 1 can return a non-perfect HDVF even when the homology groups are torsion-free. We consider the ring of coefficients $\mathfrak{R} = \mathbb{Z}$ and the chain complex

$$
0 \xrightarrow{\ 0\ } A \xrightarrow{\ d\ } B \xrightarrow{\ 0\ } 0
$$

where $A = \langle a_1, \ldots, a_5 \rangle \cong \mathbb{Z}^5$, $B = \langle b_1, \ldots, b_5 \rangle \cong \mathbb{Z}^5$ and the linear operator d is defined by the matrix

$$
\begin{bmatrix}
1 & 1 & -1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 \\
0 & 1 & -1 & 0 & -1 \\
-1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & -1 & 1
\end{bmatrix}
$$

If Algorithm 1 adds the pairs (b_4, a_1), (b_3, a_5) and (b_5, a_4) then it stops since the reduced boundary matrix is

$$D = \begin{bmatrix} 4 & -3 \\ 3 & 0 \end{bmatrix}$$

and it does not contain any unit. However, if it adds the pairs (b_1, a_1), (b_2, a_4), (b_5, a_3), (b_3, a_5) and (b_4, a_2) then it does find a perfect HDVF. Nevertheless, this counterexample considers a chain complex, so the question remains open for simplicial or cubical complexes.

If $\mathfrak{R}$ is a field, the answer is yes.

Proposition 3.17. *Let K be a CW complex. Then Algorithm 1 returns a perfect HDVF whenever the ring of coefficients is a field.*

Proof. Note that, if $\mathfrak{R}$ is a field, every non-zero element is a unit. Thus, Algorithm 1 only stops when the reduced boundary matrices are zero, in which case the returned HDVF is perfect. $\qquad\square$

Question 2: can Algorithm 1 compute every HDVF?

We provide again a counterexample. Let $\mathfrak{R} = \mathbb{Z}$ and consider the chain complex $0 \xrightarrow{0} \mathbb{Z}^6 \xrightarrow{d} \mathbb{Z}^6 \xrightarrow{0} 0$ where the linear operator d is defined by the matrix

$$\begin{bmatrix} -1 & 1 & 0 & 1 & -1 & 1 \\ 1 & 0 & 1 & -1 & 0 & -1 \\ 1 & 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & -1 \\ -1 & -1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

One can prove by exhaustion over all the possible sequences of pairs (b_i, a_j) that Algorithm 1 never finds the unique perfect HDVF, which contains all the elements of the bases. Hence, this is also a counterexample for the first question.

However, if $\mathfrak{R}$ is a field, the answer is yes.

Proposition 3.18. *Any HDVF can be obtained with Algorithm 1 whenever the ring of coefficients is a field.*

Proof. Let X be a HDVF. Using the Laplace expansion for the determinant along the first row of $A = d(S)_{|P}$, we get $\det(d(S)_{|P}) = \sum_j \lambda_{1,j} \det(A_{1,j}) \neq 0$. Thus there exists some j such that $\det(A_{1,j}) \neq 0$, so X can be obtained by adding a pair (σ, τ) to a smaller HDVF. The result follows from induction over the size of the matrix $d(S)_{|P}$. $\qquad\square$

Question 3: is there a perfect HDVF?

The two first questions have been partially answered, since we do not have any counterexample involving a simplicial or cubical complex with $\mathfrak{R} = \mathbb{Z}$. Indeed, they are strongly related to the existence of a perfect HDVF. If a CW complex does not admit a perfect HDVF, the first question has a negative answer. On the other hand, if a CW complex admits a perfect HDVF and the

second question has a positive answer, then Algorithm 1 can find a perfect HDVF, though it may not find it always.

We have already seen that a CW complex whose homology groups have torsion coefficients does not admit a perfect HDVF. In addition, as a consequence of Proposition 3.17, every CW complex admits a perfect HDVF whenever $\mathfrak{R}$ is a field. Nevertheless, we ignore what happens when $\mathfrak{R} = \mathbb{Z}$ and the homology groups are torsion-free. In order to find a counterexample, we executed Algorithm 1 for all the torsion-free simplicial complexes in Benedetti and Lutz's library of triangulations [4] and we always found a perfect HDVF. Moreover, the HDVFs returned for the simplicial complexes with just one torsion coefficient per dimension (i.e., `Hom_C5_K4`, `RP4`, `RP4#K3_17`, `RP4#11S2xS2` and `RP5_24`) had their reduced boundary matrix already in SNF. Hence, even if they are not perfect HDVFs, the homology groups can be directly read from them.

We point out that the simplicial complex `hyperbolic_dodecahe-dral_space` presented an interesting behavior. Its 1-dimensional homology group is $H_1 = (\mathbb{Z}_5)^3$. Due to its small size (718 simplices), we executed Algorithm 1 500 000 times with random choices of pairs of cells and we only found 72 HDVFs whose reduced boundary matrices were in SNF. The other simplicial complex with more than one torsion coefficient, `PG128_PG128P7`, is much larger (13 462 simplices) and we still have not found any HDVF whose reduced boundary matrix is in SNF.

3.6.4 Another algorithm for computing a HDVF

Algorithm 1 consists in iteratively adding a pair of critical cells to the HDVF. Nevertheless, we can also add several pairs of cells to a HDVF at the same time.

Let X be a HDVF and $\Sigma = \{\sigma_1, \ldots, \sigma_r\}$ and $T = \{\tau_1, \ldots, \tau_r\}$ be two sets of critical cells of codimension 1 (that is, $\dim(\sigma_i) = \dim(\tau_i) - 1$). If the matrix $d'(T)_{|\Sigma}$ is invertible in $\mathfrak{R}$ then $X' = (P \cup \Sigma, S \cup T)$ is a HDVF. The proof is similar to that of Proposition 3.11.

As a consequence, Algorithm 1 is not the unique way of computing a HDVF. However, we prefer it for its simplicity and we do not study in this work the above alternative approach.

Let us point out that, if we can add several pairs of cells at the same time, then the second question of the previous section is true since we can add all the pairs of cells in a HDVF at once.

3.7 Deforming a HDVF

In Section 3.6 we described how the reduction changes after adding a pair of critical cells to the HDVF. This can be seen as a basic operation on a HDVF, in which two critical cells γ and γ' are transformed into a primary and a secondary cell respectively. In this section we extend this idea to define five basic operations that allow us to modify a HDVF.

3.7.1 Basic Operations

Let K be a CW complex endowed with a HDVF $X = (P, S)$. Let $\sigma \in P$, $\tau \in S$ and $\gamma, \gamma' \in C$. Thus,

- $X' = \mathtt{A}(X, \gamma, \gamma') = (P \cup \{\gamma\}, S \cup \{\gamma'\})$ is a HDVF identical to X except for γ, which is a primary cell, and γ', which is a secondary cell

- $X' = \mathtt{R}(X, \sigma, \tau) = (P \setminus \{\sigma\}, S \setminus \{\tau\})$ is a HDVF identical to X except for σ and τ, which are critical cells

- $X' = \mathtt{M}(X, \sigma, \gamma) = ((P \setminus \{\sigma\}) \cup \{\gamma\}, S)$ is a HDVF identical to X except for σ, which is a critical cell, and γ, which is a primary cell

- $X' = \mathtt{W}(X, \tau, \gamma) = (P, (S \setminus \{\tau\}) \cup \{\gamma\})$ is a HDVF identical to X except for τ, which is a critical cell, and γ, which is a secondary cell

- $X' = \mathtt{MW}(X, \sigma, \tau) = ((P \setminus \{\sigma\}) \cup \{\tau\}, (S \setminus \{\tau\}) \cup \{\sigma\})$ is a HDVF identical to X except for τ, which is a primary cell, and σ, which is a secondary cell

The operation $\mathtt{A}$ has been largely explained in Section 3.6 and $\mathtt{R}$ consists in removing a pair of cells from the HDVF. $\mathtt{M}$ exchanges a primary cell with a critical one, while $\mathtt{W}$ exchanges a secondary cell with a critical one. $\mathtt{MW}$ is like combining $\mathtt{M}$ and $\mathtt{W}$ except that no critical cell is needed.

Let us see the conditions under which we can perform each operation.

Proposition 3.19. *Let K be a CW complex endowed with a HDVF X. Let $\sigma \in P$, $\tau \in S$ and $\gamma, \gamma' \in C$. Thus,*

1. *$\mathtt{A}(X, \gamma, \gamma')$ is a HDVF if $\langle d'(\gamma'), \gamma \rangle$ is a unit*

2. *$\mathtt{R}(X, \sigma, \tau)$ is a HDVF if $\langle h(\sigma), \tau \rangle$ is a unit*

3. *$\mathtt{M}(X, \sigma, \gamma)$ is a HDVF if $\langle f(\sigma), \gamma \rangle$ is a unit*

4. *$\mathtt{W}(X, \tau, \gamma)$ is a HDVF if $\langle g(\gamma), \tau \rangle$ is a unit*

5. *$\mathtt{MW}(X, \sigma, \tau)$ is a HDVF if $\langle dh(\sigma), \tau \rangle$ and $\langle hd(\tau), \sigma \rangle$ are units*

Proof. The first statement only rephrases Proposition 3.11.

For the second statement we need to prove that $d_q(S'_{q+1})_{|P'_q}$ is invertible after removing the two cells. In the following we omit the subscripts. We write

$$d(S)_{|P} = \left[\begin{array}{c|c} A & B \\ \hline C & d(S')_{|P'} \end{array} \right], \quad M = \left[\begin{array}{c|c} 1 & 0 \\ \hline C & d(S')_{|P'} \end{array} \right]$$

where $A = d(\tau)_{|\sigma}$, $B = d(S')_{|\sigma}$ and $C = d(\tau)_{|P'}$. Note that $\det(M) = \det\left(d(S')_{|P'}\right)$. Then

$$\det\left(d(S')_{|P'}\right) = \det(M)$$
$$= \det\left(d(S)_{|P} + \left[\begin{array}{c|c} 1-A & -B \\ \hline 0 & 0 \end{array} \right]\right)$$
$$= \det\left(d(S)_{|P} + \left[\begin{array}{c} 1 \\ \hline 0 \end{array} \right] \cdot \left[\begin{array}{c|c} 1-A & -B \end{array} \right]\right)$$
$$= \det\left(d(S)_{|P}\right) \cdot \left(1 + \left[\begin{array}{c|c} 1-A & -B \end{array} \right] \cdot H \cdot \left[\begin{array}{c} 1 \\ \hline 0 \end{array} \right]\right) \quad \text{(cf. Lemma 3.7)}$$
$$= \det\left(d(S)_{|P}\right) \cdot \left(1 + \left[\begin{array}{c|c} 1 & 0 \end{array} \right] \cdot H \cdot \left[\begin{array}{c} 1 \\ \hline 0 \end{array} \right] - \left[\begin{array}{c|c} A & B \end{array} \right] \cdot H \cdot \left[\begin{array}{c} 1 \\ \hline 0 \end{array} \right]\right)$$
$$= \det\left(d(S)_{|P}\right) \cdot (1 + H_{11} - 1) = \det\left(d(S)_{|P}\right) \cdot H_{11}$$

where H_{11} denotes $h(\sigma)_{|\tau} = \langle h(\sigma), \tau \rangle$.

The third statement is also proved using Lemma 3.7. We write

$$d_q(S)_{|P} = \left[\frac{a}{M} \right], \quad d_q(S)_{|P'} = \left[\frac{b}{M} \right]$$

where $a = d(S)_{|\sigma}$ and $b = d(S)_{|\gamma}$. We note that

$$F = - \left[\frac{b}{N} \right] \cdot H$$

where $N = d(S)_{|C \setminus \gamma}$ and thus

$$\langle f(\sigma), \gamma \rangle = -b \cdot h$$

where $h = h(\sigma)_{|S}$. Then

$$
\begin{aligned}
\det \left(d_q(S)_{|P'} \right) &= \det \left(d_q(S)_{|P} + \left[\frac{1}{0} \right] \cdot (b - a) \right) \\
&= \det \left(d_q(S)_{|P'} \right) \cdot \left(1 + (b - a) \cdot H \cdot \left[\frac{1}{0} \right] \right) \\
&= \det \left(d_q(S)_{|P'} \right) \cdot (1 + (b - a) \cdot h) \\
&= \det \left(d_q(S)_{|P'} \right) \cdot (1 + b \cdot h - a \cdot h) \\
&= \det \left(d_q(S)_{|P'} \right) \cdot (1 - \langle f(\sigma), \gamma \rangle - 1) \\
&= - \det \left(d_q(S)_{|P'} \right) \cdot \langle f(\sigma), \gamma \rangle
\end{aligned}
$$

$d_q(S)_{|P'}$ is thus invertible.

We omit the proof of the last two statements since they are similar to the third one. $\qquad \square$

All these operations can be applied in terms of the DVF by reversing a $\mathcal{V}$-path between the two cells considered. This $\mathcal{V}$-path is not unique, but its existence can be proved using the same argument present in Proposition 3.12. In the case of MW, there are two $\mathcal{V}$-paths to reverse. Figure 3.8 illustrates the operations M, W and MW on a cubical complex.

Some of these operations were introduced in [91]. Namely, the *arrow reversing* is the operation M between 0-cells, the *edge rotation* is MW between 1-cells and the *face rotation* is MW between 2-cells. These three operations were announced as local deformations of a DGVF but, since no condition was given, this is in general false: applying an *edge rotation* or a *face rotation* to a DGVF can produce a non-gradient discrete vector field. Only the *arrow reversing* preserved the structure of DGVF since, in dimension 0, the existence of a $\mathcal{V}$-path between a primary cell σ and a critical cell γ implies that $\langle f(\sigma), \gamma \rangle$ is a unit.
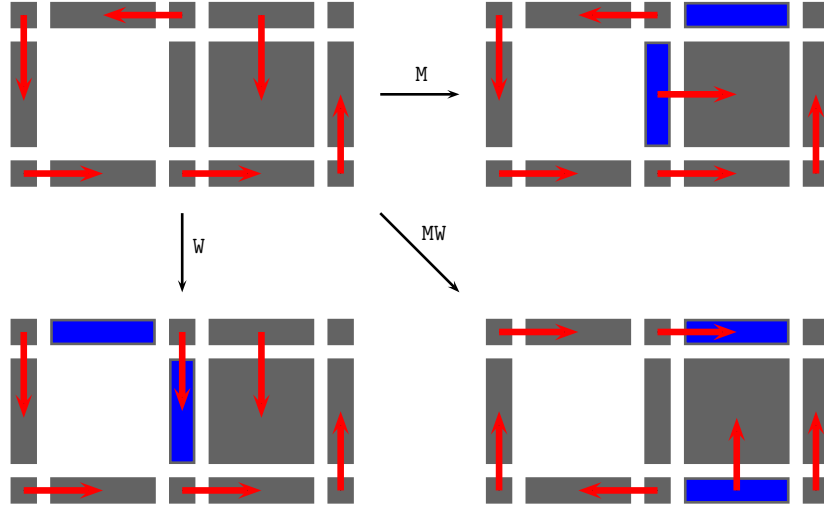
FIGURE 3.8: A HDVF on a cubical complex and the result after applying M, W and MW. Blue cells are those which are exchanged.

3.7.2   Delineating (co)homology generators

Given a perfect HDVF, the operations M, W and MW are interesting since they change the reduction and thus the generators of the homology and the cohomology groups. The next proposition specifies how a generator changes when the operations M or W affect its associated critical cell.

Proposition 3.20. *Let K be a CW complex endowed with a perfect HDVF $X = (P, S)$ with $\mathfrak{R} = \mathbb{Z}_2$. Let $\sigma \in P$, $\tau \in S$ and $\gamma \in C$. Then,*

1. *If $\langle f(\sigma), \gamma \rangle$ is a unit, the cohomology generators associated to γ in X and σ in $\mathtt{M}(X, \sigma, \gamma)$ are the same.*

2. *If $\langle g(\gamma), \tau \rangle$ is a unit, the homology generators associated to γ in X and τ in $\mathtt{W}(X, \sigma, \gamma)$ are the same.*

Proof. The proof of these statements is quite lengthy, but it provides a partial description of the reduction after perturbing the HDVF.

For the first statement we write

$$H_q = \left[\frac{u}{A} \right]^{-1} \qquad\qquad =: \left[\ H_1 \mid H_2\ \right]$$

$$F_q = -\left[\frac{v}{B} \right] \cdot H_q = -\left[\frac{vH_1 \mid vH_2}{BH_1 \mid BH_2} \right] \qquad =: \left[\frac{F_{11} \mid F_{12}}{F_{21} \mid F_{22}} \right]$$

where $u = d(S)_{|\sigma}$, $A = d(S)_{|P\setminus\sigma}$, $v = d(S)_{|\tau}$ and $B = d(S)_{|C\setminus\gamma}$

Then,

$$
\begin{aligned}
H' &= \left(d(S)_{|P'}\right)^{-1} \\
&= \left(d(S)_P + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot (v - u)\right)^{-1} \quad \text{(cf. Lemma 3.5)} \\
&= H + F_{11}^{-1}\left(H \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot (v - u) \cdot H\right) \\
&= H \cdot \left(I - F_{11}^{-1} \left[\begin{array}{c|c} F_{11} & F_{12} \\ \hline 0 & 0 \end{array}\right] - F_{11}^{-1} \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & 0 \end{array}\right]\right) \\
&= H \cdot \left[\begin{array}{c|c} -F_{11}^{-1} & -F_{11}^{-1} F_{12} \\ \hline 0 & I \end{array}\right] \\
&= \left[\begin{array}{c|c} -H_1 F_{11}^{-1} & H_2 - H_1 F_{11}^{-1} F_{12} \end{array}\right]
\end{aligned}
$$

Consequently,

$$
\begin{aligned}
F' &= -\begin{bmatrix} u \\ B \end{bmatrix} \cdot \left[\begin{array}{c|c} -H_1 F_{11}^{-1} & H_2 - H_1 F_{11}^{-1} F_{12} \end{array}\right] \\
&= \left[\begin{array}{c|c} F_{11}^{-1} & F_{11}^{-1} F_{12} \\ \hline F_{21} F_{11}^{-1} & F_{22} - F_{21} F_{11}^{-1} F_{12} \end{array}\right]
\end{aligned}
$$

The proof of the second statement is similar. We write

$$
H_q = \left[\begin{array}{c|c} u & A \end{array}\right]^{-1} \qquad\qquad =: \begin{bmatrix} H_1 \\ H_2 \end{bmatrix}
$$

$$
G_{q+1} = -H_q \cdot \left[\begin{array}{c|c} v & B \end{array}\right] = -\left[\begin{array}{c|c} H_1 v & H_1 B \\ \hline H_2 v & H_2 B \end{array}\right] \qquad =: \left[\begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array}\right]
$$

where $u = d(\tau)_{|P}$, $A = d(S \setminus \tau)_{|P}$, $v = d(\gamma)_{|P}$ and $B = d(C \setminus \gamma)_{|P}$. Then it is easy to prove that

$$
H' = \left[\begin{array}{c} -G_{11}^{-1} H_1 \\ \hline H_2 - G_{11}^{-1} G_{21} H_1 \end{array}\right]
$$

$$
G' = \left[\begin{array}{c|c} G_{11}^{-1} & G_{11}^{-1} G_{12} \\ \hline G_{11}^{-1} G_{21} & G_{22} - G_{11}^{-1} G_{21} G_{12} \end{array}\right]
$$

$\square$

We can thus use these operations to change the shape of the generators. Figure 3.9 shows a cubical complex endowed with a HDVF. We want to have a one-dimensional homology generator around the hole. For doing this, it suffices that all the 1-cells are secondary except for one which is critical. Thus, we use `M` on the top 1-cell to put there the critical cell. Then, for the other three 1-cells, we use `MW` to make them secondary. At the end, the homology generator induced by the HDVF stands at the desired location.

It is unclear whether this application is computationally feasible. The problem is: given a perfect HDVF X and a set of cycles S, can we find a perfect HDVF X' whose homology generators contain this set? We may
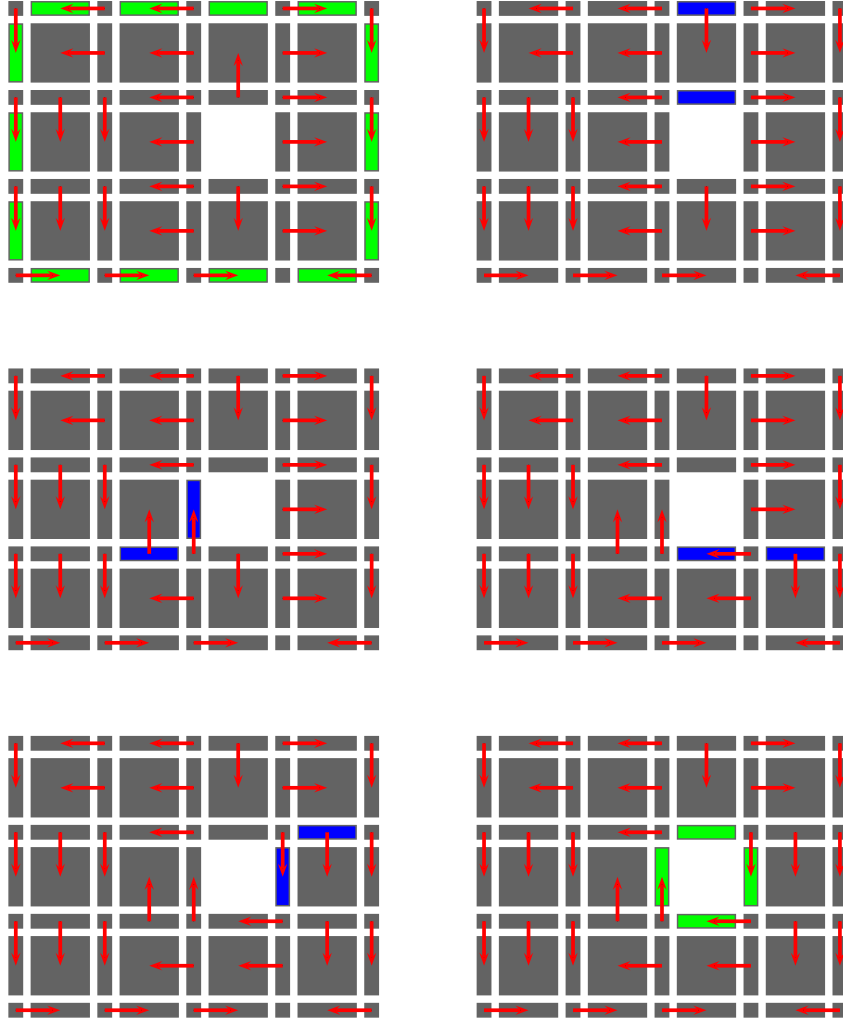
FIGURE 3.9: Example of multiple applications of the operations on a cubical complex. Blue cells are those that have changed. The one-dimensional homology generator is depicted in green at the beginning and at the end.

first check that the cycles are linearly independent. This can be done by computing the rank of the matrix $f(S)$. If the rank is maximal, then the cycles can be part of a homology basis. But even if the cycles are linearly independent, the HDVF X' does not exist in general, and Figure 3.10 provides a counterexample. Thus, this problem must be studied further in order to find conditions under which such HDVF exists. A possible hint to follow is that every cycle must have a cell not included in any other cycle, which is the intuition that led to our counterexample.

Assuming that the HDVF X' exists, it is possible to find a sequence of operations that transform one HDVF into the other: it suffices to successively apply R to X for removing all the pairing in the DVF, and then build the other HDVF using A (this is guaranteed if $\Re$ is a field by Proposition 3.18). Thus, the interesting question is to find a minimal sequence of operations that transform X into X'.
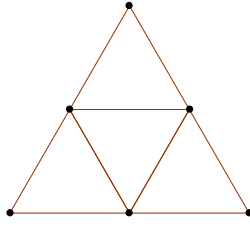
FIGURE 3.10: There exists no HDVF on this simplicial complex whose homology generators are the four triangles consisting of three 1-simplices.

3.7.3 Connectivity between HDVFs

The new definitions let us state that Algorithm 1 can compute any HDVF which is the result of applying only the operation A to an empty HDVF. We explained in Section 3.6.3 that we ignore if every HDVF on a simplicial or cubical complex can be found through Algorithm 1. Thus it is natural to wonder if any HDVF on a simplicial or cubical complex can be obtained by a sequence of operations (not only A) on an empty HDVF. This can also be formulated as follows: are all the HDVF on a simplicial or cubical complex connected via a sequence of operations? This is still an open question.

3.8 Relation with other Methods in Computational Homology

There are several methods for computing homology in the literature which seem to be equivalent. The simple formulation of the HDVFs allows us to clearly see these equivalences.

3.8.1 Iterated Morse decomposition

First, let us prove that the HDVF generalizes the notion of DGVF.

Proposition 3.21. *Every DGVF is a HDVF.*

Proof. We need to prove that the matrices $d(S_{q+1})_{|P_q}$ are invertible for each $q \geq 0$. In the following we omit the subscripts since the proof is the same for every dimension q.

Let $\mathcal{V} = \{(\sigma_i, \tau_i)\}_{i=1}^m$ be a DGVF. Consider the weighted digraph whose vertices are the primary cells and where arrows connect two vertices whenever there is a $\mathcal{V}$-path of length 1 between them. Formally, (σ_i, σ_j) is an arrow if $\langle d(\tau_i), \sigma_j \rangle \neq 0$ and $\sigma_i \neq \sigma_j$. Its weight is the value $-\langle d(\tau_i), \sigma_i \rangle \cdot \langle d(\tau_i), \sigma_j \rangle$. It is immediate to see that the matrix associated to this graph is $I - d(S)_{|P}V$, where $S = \{\tau_i\}_{i=1}^m$, $P = \{\sigma_i\}_{i=1}^m$ and the diagonal matrix $V = (v_{i,j})$ is such that for every i, $v_{i,i} = \langle d(\tau_i), \sigma_i \rangle$ and zero elsewhere. Note that V is invertible. Since there are no closed $\mathcal{V}$-paths in the DGVF, the matrix $I - d(S)_{|P}V$ is nilpotent, and thus $d(S)_{|P}V$ is invertible. As V is invertible, we deduce that $d(S)_{|P}$ is invertible. $\square$

A DGVF is a limited tool for computing homology. A more elaborate tool is the iterated Morse decomposition [42], which consists in iteratively:

(1) computing a DGVF and (2) considering the resulting Morse complex for a next DGVF. We prove now that every iterated Morse decomposition is also a HDVF.

Proposition 3.22. *Every iterated Morse decomposition is a HDVF.*

Proof. For clarity, we assume that the iterated Morse decomposition consists of only two DGVFs $\mathcal{V}^1$ and $\mathcal{V}^2$. We recall that the second DGVF is defined on the chain complex consisting of the critical cells of $\mathcal{V}^1$ and the boundary operator

$$d' = d(C^1)_{|C^1} - d(S^1)_{|C^1} \cdot H \cdot d(C^1)_{|P^1}.$$

If we write

$$d(S^1 \cup S^2)_{|P^1 \cup P^2} = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right]$$

then

$$d(S^1)_{|P^1} = A \quad \text{and} \quad d'(S^2)_{|P^2} = D - CA^{-1}B$$

Given the previous proposition, these two matrices are invertible. Thus, by Lemma 3.3,

$$\det \left(d(S^1 \cup S^2)_{|P^1 \cup P^2} \right) = \det \left(d(S^1)_{|P^1} \right) \cdot \det \left(d'(S^2)_{|P^2} \right)$$

which is a unit. $\qquad\qquad\square$

It is easy to see that if a HDVF has been created using Algorithm 1, then the list of pairs of cells $[(\sigma_i, \tau_i)]_{i=1}^m$ is an iterated Morse decomposition. As we showed in Section 3.6, it is not true in general that every HDVF can be computed with Algorithm 1, so we cannot deduce that every HDVF is an iterated Morse decomposition. However, this does not mean that it is false. This question remains open.

3.8.2 The Smith normal form

The classic algorithm for computing homology groups computes the Smith normal form (SNF) [99]. We prove in this section that the reduced boundary matrices obtained in Algorithm 1 are similar to the diagonalization performed for the computation of the SNF.

Let K be a CW complex and X a trivial HDVF ($P = S = \emptyset$). Let us choose some pivot in some boundary matrix D_q. For simplicity, we assume that the pivot is the element $D_q(1,1) = \langle d_q(\tau), \sigma \rangle$ and we omit the subscript. In order to make all the other entries in its row and column into zeros we perform

$$\forall j \neq 1, \quad D(\cdot, j) \leftarrow D(\cdot, j) - D(1, j)D(1, 1)^{-1}D(\cdot, 1)$$
$$\forall i \neq 1, \quad D(i, \cdot) \leftarrow D(i, \cdot) - D(i, 1)D(1, 1)^{-1}D(i, \cdot)$$

Using the notation of Proposition 3.13, this is equivalent to

$$D' = D - \left[\frac{D_{11}}{D_{21}}\right] D_{11}^{-1} \left[\ 0 \mid D_{12}\ \right]$$

$$D'' = D' - \left[\frac{0}{D'_{21}}\right] D_{11}^{-1} \left[\ D'_{11} \mid D'_{12}\ \right]$$

By developing both equations we obtain that the pseudo-diagonalized boundary matrix is

$$\left[\begin{array}{c|c} D_{11} & 0 \\ \hline 0 & D_{22} - D_{21}D_{11}^{-1}D_{12} \end{array}\right],$$

where the bottom-right block is the reduced boundary computed in Algorithm 1 after inserting the pair of cells (σ, τ).

Proposition 3.23. *Let K be a CW complex. Then, Algorithm 1 performs a partial diagonalization of the boundary matrices of K.*

Proof. The proof is direct from the previous argument. □

We have just seen that computing a HDVF is equivalent to compute the SNF of the boundary matrices using only the pivot operation, that is, given an invertible entry in the matrix, we make all the entries in its row and column into zeros. Computing the SNF needs also another type of operation: if there is no entry dividing all the others, we make elementary operations on the rows and columns so such an entry appears. For this reason, Algorithm 1 cannot always return a perfect HDVF if $\mathfrak{R} = \mathbb{Z}$, since in the computation of the SNF we can arrive to a matrix without units even if the SNF contains only units in its diagonal (see Section 3.6.3 for an example).

3.8.3 Persistent homology

Proposition 3.23 also implies that persistent homology can be computed with a variation of Algorithm 1. The classical algorithm for persistent homology [119] is based on the Smith normal form. The main difference with a standard homology computation is that cells are considered in the order given by the filtration. Therefore, Algorithm 2 computes the persistence intervals of a filtration using the same calculations as Algorithm 1.

Algorithm 2 is a mere translation of the algorithm described in [119] into the HDVF framework. The purpose of doing so is to show that we can obtain a reduction for every step of the filtration and that we can apply the conclusions of Section 3.9 to the persistent homology theory.

3.9 Experimental Complexity

We fix in this section $\mathfrak{R} = \mathbb{Z}_2$, so we are sure that we obtain a perfect HDVF and thus we compute the homology of the CW complex.

Computing the homology groups of a CW complex is considered in general a problem with $\mathcal{O}(n^3)$ time complexity. Only [88] proves that it can be computed in matrix multiplication time, but there is no implementation of

---

Algorithm 2: Compute a HDVF associated to a filtration

Input: A CW complex K and a filtration $F = \left\{ \sigma^j \right\}_{j=1}^n$

Output: The persistence intervals of F

for $k = 0$ *to* $\dim(K)$ **do**

    $L_k \leftarrow \emptyset$;

$X \leftarrow (\emptyset, \emptyset)$;

for $j = 1$ *to* n **do**

    if $d'(\sigma^j) \neq 0$ **then**

        $i \leftarrow \max \left\{ j' : \langle d'(\sigma^j), \sigma^{j'} \rangle = 1 \right\}$;

        $X \leftarrow \mathtt{A}(X, \sigma^i, \sigma^j)$ (and update the boundary matrices D);

        $L_{\dim(\sigma^j)} \leftarrow L_{\dim(\sigma^j)} \cup \left\{ (\deg \sigma^i, \deg \sigma^j) \right\}$;

for $j = 0$ *to* n **do**

    if σ^j *is critical* **then**

        $L_{\dim(\sigma^j)} \leftarrow L_{\dim(\sigma^j)} \cup \left\{ (\deg \sigma^j, \infty) \right\}$;

---

this algorithm. Nevertheless, it has been noticed that in practice the execution time is linear for homology [39, §4] and persistent homology [119, §4]. We estimated in Theorem 3.15 the complexity of our algorithm by bounding the number on non-zero entries in rows and columns by n, obtaining that Algorithm 1 can find a HDVF within $(n/2) \cdot n^2 = n^3/2$ operations or $(n/2) \cdot \left(n^2 + n^2 + n^2 + n^2 \right) = 2n^3$ if we also want to obtain the associated reduction. Since these bounds are not tight, it should not be surprising that the complexity in practice is lower than cubic.

One advantage of the HDVF framework is that we can easily count the number of operations that we perform along its computation. At each step of Algorithm 1, updating the matrices H, F, G and D requires $|F_{11}||G_{11}|$, $|D_{21}||F_{11}|$, $|G_{11}||D_{12}|$ and $|D_{21}||D_{12}|$ operations respectively, where $|v|$ denotes the number of non-zero entries in the vector v. Thus, updating the reduced boundary requires

$$|D_{21}||D_{12}|$$

operations (plus some operations to remove rows and columns). Moreover, updating all the reduction requires

$$|F_{11}||G_{11}| + |D_{21}||F_{11}| + |G_{11}||D_{12}| + |D_{21}||D_{12}| = (|F_{11}| + |D_{12}|)(|G_{11}| + |D_{21}|)$$

operations.

Let us study now the average complexity for two random models.

Random cubical complexes We introduce a random model for constructing cubical complexes. We denote it by $K(p, m)$ and it is similar to the *closed faces model* introduced in [116]. Let $m \in \mathbb{Z}^+$ and $p \in \mathbb{R}$, $0 \leq p \leq 1$. A cubical complex in $K(p, m)$ is built by adding each cubical cell $\sigma \in [0, m]^3$ (with its faces) to the complex with probability p. Note that each cell σ belongs to the cubical complex with probability $1 - (1 - p)^c$, where c denotes the number of cofaces (in the full cubical complex $[0, m]^3$) of σ, including itself. Thus, lower-dimensional cells are more frequent.

For each cubical complex K we denote by $X(K)$ its size. Also, $Y^d(K)$ denotes the number of operations performed for computing a HDVF (which is the sum of $|D_{21}||D_{12}|$ along its computation) and $Y^r(K)$ denotes the number of operations needed for computing a HDVF and its reduction. We thus know that for each K,

$$Y^d(K) \leq \frac{1}{2}X(K)^3 \quad \text{and} \quad Y^r(K) \leq 2X(K)^3$$

Let us point out two concerns:

1. If K has large Betti numbers then there are several critical cells in the perfect HDVF, so we perform less than $n/2$ steps. The relation between the parameter p and the number of critical cells in a perfect HDVF is unknown, and this could help to understand our experiments. Figure 3.11 shows the Betti numbers for a large number of cubical complexes in $K(p, 100)$ and their sum. We appreciate that the number of critical cells is at most 5% of the size of the complex, so it does not seem to be significant.

2. We do not make any smart choice for the pair (σ, τ) in each step of the algorithm. This means that the quantities $Y_d(K)$ and $Y_r(K)$ are not optimized. We could have implemented a better choice that tries to minimize these values, but we have chosen not to do it since we want to obtain a really general result that can be also applied to compute persistent homology.

In our experiment we fix $m = 25$ and we build 2217 cubical complexes with probability p uniformly distributed in $[0, 1]$. We want to show that the average complexity of Algorithm 1 is $\mathcal{O}(n^\alpha)$ for some $\alpha \in \mathbb{R}$. Note, however, that our sample does not seem to fit to a polynomial function. For achieving this we fit our sample $\{(\log(X_i), \log(Y_i^d))\}$ to a linear function $y = b \cdot x$. Using R [104] we obtain the 99.99% confidence interval $[1.372086, 1.384346]$ for b. Thus, we can (statistically) affirm that $b < 1.4$. Consequently, $Y^d < X^{1.4}$ and the average-case complexity is $\mathcal{O}(n^{1.4})$. Figure 3.12-(top) shows the plot of $\{(\log(X_i), \log(Y_i^d))\}$ together with the fitted linear function passing by the origin.

We repeat this study for Y^r (thus computing also the reduction). The 99.99% confidence interval for b is $[1.940787, 1.959928]$ so we can affirm that in average, computing a perfect HDVF with its reduction requires $\mathcal{O}(n^2)$. Figure 3.12-(bottom) shows the plot of $\{(\log(X_i), \log(Y_i^r))\}$ together with the fitted linear function passing by the origin.

Random volumes We may be interested in studying cubical complexes which come from binary volumes. We thus introduce the following random model. Let be $m \in \mathbb{Z}^+$. We consider an empty binary volume of size $m \times m \times m$ and we add $\lfloor m/10 \rfloor^3$ blocks of voxels of size $\lfloor m/10 \rfloor \times \lfloor m/10 \rfloor \times \lfloor m/10 \rfloor$ at random (uniform) position. We can see this process as cutting a volume in small pieces and shuffling them. This binary volume can be transformed into a cubical complex by substituting each voxel for a 3-cube (see the primal associated cubical complex in Section 2.4). We denote this model by $V(m)$.
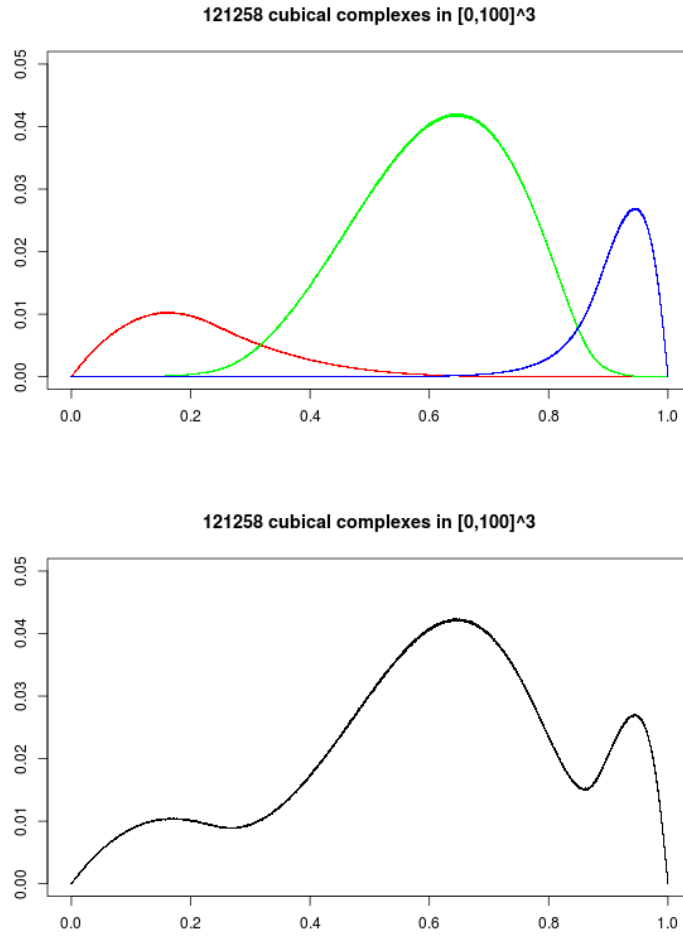
FIGURE 3.11: We have computed the Betti numbers for 121258 cubical complexes in $K(p, 100)$ with $p \sim U(0, 1)$. Top: for each cubical complex K we plot the points $(X(K), \beta_0)$ (red), $(X(K), \beta_1)$ (green) and $(X(K), \beta_2)$ (blue) divided by the total number of cubes $(2 \cdot 100 + 1)^3$. Bottom: for each cubical complex K we plot the point $(X(K), \beta_0 + \beta_1 + \beta_2)$ divided by the total number of cubes.

When building 1675 cubical complexes with the random model $V(25)$ we obtain very similar results. By fitting $\left\{ (\log(X_i), \log(Y_i^d)) \right\}$ to a linear function $y = b \cdot x$ we obtain the 99.99% confidence interval $[1.377406, 1.377593]$ for b. For Y^r, the interval is $[1.941323, 1.941854]$. Both fitted linear functions are depicted in Figure 3.13.

3.10   Conclusion and future work

We have introduced a combinatorial structure that can be interpreted as a new class of discrete vector field, namely the homological discrete vector field, together with the theorems providing homological results for such extended DGVF. We have shown that this extended class of DGVF successfully reaches the correct number of critical cells in complexes for which standard DGVFs cannot, such as the Bing's house or the dunce hat.
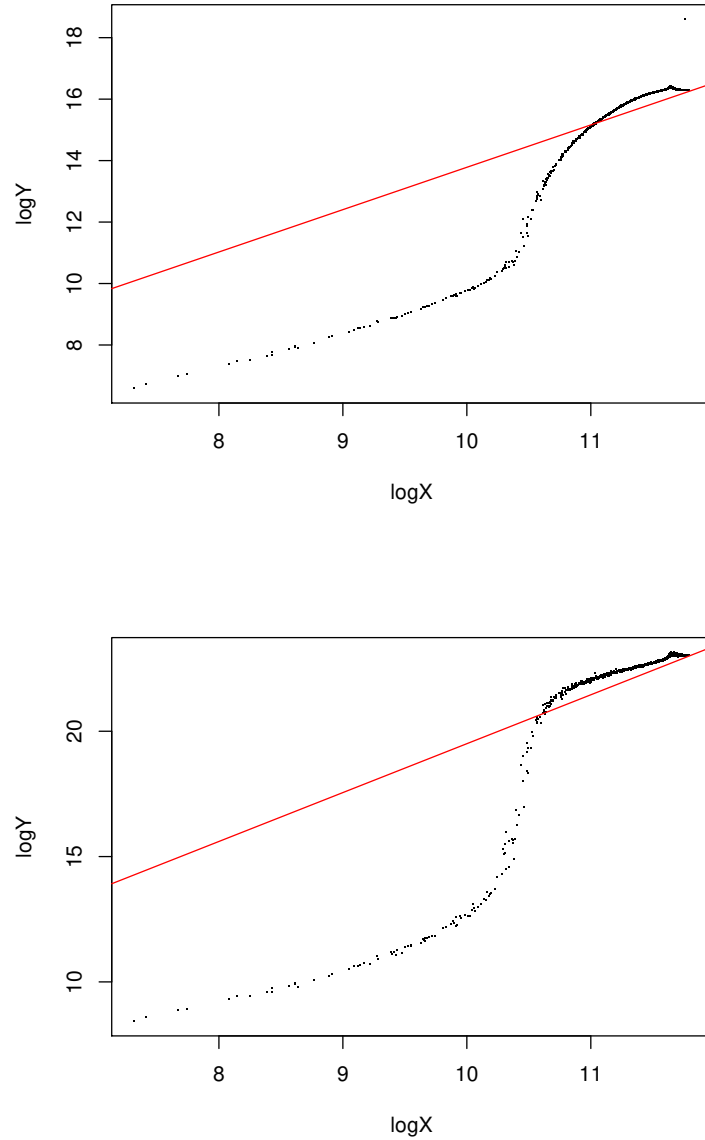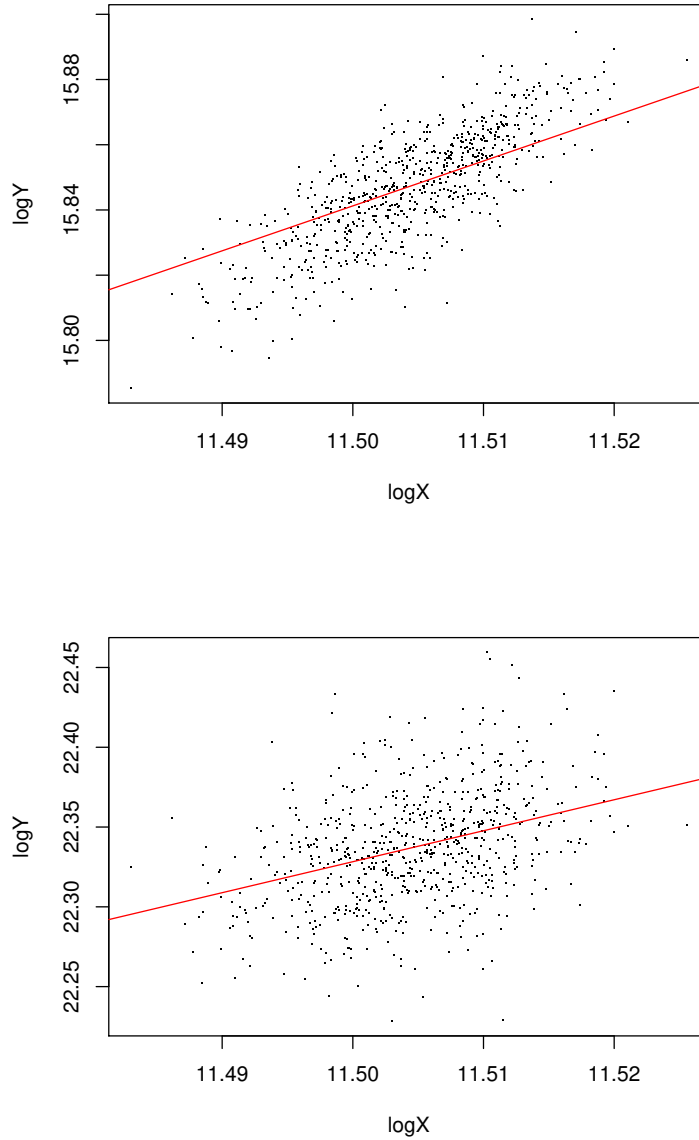
FIGURE 3.12: Top: plot of $\left\{(\log(X_i), \log(Y_i^d))\right\}$ for cubical complexes in $K(25, p)$ together with the linear regression model passing by the origin. Bottom: the same for $\left\{(\log(X_i), \log(Y_i^r))\right\}$

We provide a simple sequential algorithm for computing a HDVF in Section 3.6. The reduction is updated by using formulas that depend on the previous step reduction, without inverting the matrix $d(S)_{|P}$. The worst-case complexity of this algorithm is $\mathcal{O}(n^3)$. Finally, we partially answer some questions about the algorithm and the existence of perfect HDVFs.

Section 3.7 introduces five basic operations that allow us to switch pairs of cells belonging to the sets P, S or C of a HDVF. This extends and corrects a similar idea present in [91]. These operations allow us to change the shape of the homology (or cohomology) generators of a perfect HDVF. However, we do not explore how to do this in practice since there are some theoretical

FIGURE 3.13: Top: plot of $\left\{(\log(X_i), \log(Y_i^d))\right\}$ for cubical complexes in $V(25)$ together with the linear regression model passing by the origin. Bottom: the same for $\left\{(\log(X_i), \log(Y_i^r))\right\}$

problems that must be solved before designing an algorithm, namely the existence of a sequence of operations that transform one HDVF into any other.

Section 3.8 is devoted to the relation between the HDVF framework and other homology algorithms. We prove that computing a HDVF is equivalent to compute a DGVF, an iterated Morse decomposition and the classical homology algorithm using the Smith normal form. It remains as an open question if every iterated Morse decomposition can be computed through a HDVF (the converse is proved). We also show how to compute persistence intervals using the HDVF framework. A very interesting task is to do the

same for zigzag persistent homology [13].

In Section 3.9 we study the average-case complexity of our algorithm through an experimental approach. The validity of this study can be questioned, since we consider only two random models for cubical complexes with fixed parameters which do not sample the whole space of CW complexes. However we show that we can use the HDVF framework for giving a more concrete sense to the well accepted idea that homology and persistent homology can be computed in practice in almost linear time. We show, using simple linear regression, that a perfect HDVF (which provides the Betti numbers of the complex) can be obtained within $\mathcal{O}(n^{1.4})$ operations in average. If we also want the reduction (and thus the homology groups), it requires $\mathcal{O}(n^2)$ operations in average. However, we have not used any optimization technique such as reduction and coreductions [94], which should give even better estimations.

Chapter 4

Fast Computation of Betti Numbers on Three-Dimensional Cubical Complexes

T HIS chapter is based on the conference paper [61], which was co-written with Mateusz Juda. We explain how to efficiently compute (only) the Betti numbers of a 3D cubical complex without manipulating any boundary matrix.

4.1 Introduction

Computing homology usually needs algebraic methods. It seems that they all are based on the Smith normal form as shown in Section 3.8. However, there are Betti numbers that are easier than others.

Consider a simple shape in the real plane $\mathbb{R}^2$. It is well known that β_0 is the number of connected components and β_1 is the number of bounded connected components of the complement.

Thus, if we can count the connected components and detect which ones are bounded, we can obtain both Betti numbers without resorting any algebraic method. Let us mention two simple scenarios where this is possible:

1. A subcomplex of a simplicial complex triangulating a rectangle. We can compute the number of connected components with the usual algorithm on the connectivity graph of the subcomplex and its complement. The unbounded connected components of the complement correspond to those connected components which contain a simplex from the boundary of the rectangle.

   Figure 4.1 illustrates this. Two simplicial complex $K \subset L$ are shown in Figure 4.1a, while the connectivity graphs of K and $L-K$ are depicted in Figure 4.1b. Note that $L - K$ is not a simplicial complex since some simplices in $L - K$ do not have their faces in $L - K$. There are two connected components in the connectivity graph of K (in red), so $\beta_0 = 2$. On the other hand, there are four connected components in the connectivity graph of $L - K$ (in green), one of them containing all the simplices in the boundary of the square, so $\beta_1 = 4 - 1$.
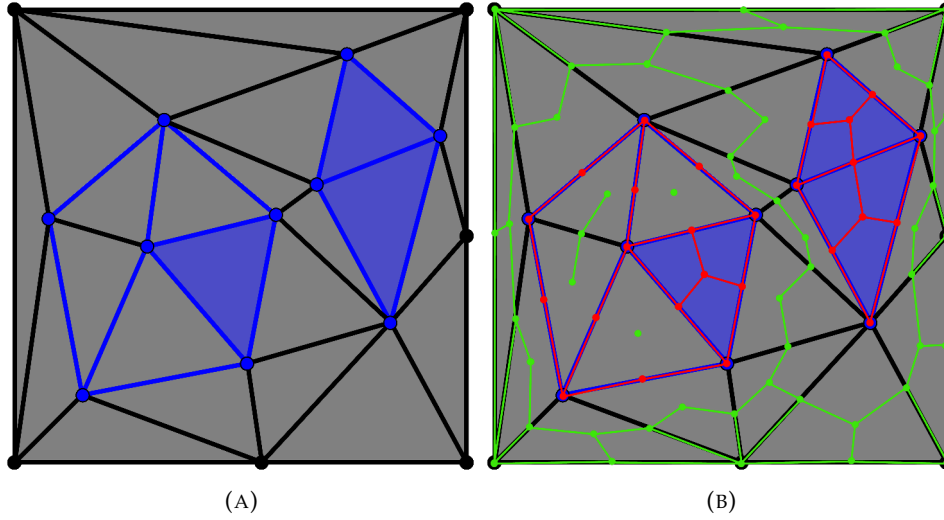
(A)   (B)

FIGURE 4.1: Left: a simplicial complex L (in gray) and a subcomplex K (in blue). Right: the connectivity graphs of K and $L - K$.

2. A 2D cubical complex. This case is particularly simple because a cubical complex is always a subcomplex of a bigger cubical complex. The same ideas apply to this setting.

We now focus only on the second scenario, since the first one is very restrictive.

Can we adapt this idea to the three-dimensional space? Let K be a 3D cubical complex. Then β_0 is still the number of connected components, but now β_2 is the number of bounded connected components of the complement. Alexander duality generalizes this fact to any dimension and any Betti number:

Theorem 4.1 (Alexander duality). *Let K be an nD cubical complex. Then $H_q(K)$ and $H^{n-1-q}(S^n - K)$ are isomorphic for reduced homology and cohomology.*

This is why homology starts being interesting starting from three dimensions. The first homology group H_1 describes the handles and tunnels of an object, which cannot be easily deduced as connected components or voids. Nevertheless, since β_0 and β_2 are easy to compute, we can compute β_1 via the simplest topological invariant: the Euler-Poincaré characteristic. The *Euler-Poincaré characteristic* of a 3D cubical complex K is the alternating sum of its cubes. Formally,

$$\chi(K) = k_0 - k_1 + k_2 - k_3,$$

where k_q denotes the number of cubes of dimension q in K.

Theorem 4.2 (Euler-Poincaré formula). *Let K be a 3D cubical complex. Then $\chi(K) = \beta_0(K) - \beta_1(K) + \beta_2(K)$.*

Therefore, $\beta_1(K) = \beta_0(K) + \beta_2(K) - \chi(K)$.

Consequently, we can obtain the Betti numbers of a 3D cubical complex only by counting connected components and computing the Euler-Poincaré characteristic.

Delfinado and Edelsbrunner introduced in [29] an algorithm with almost linear time complexity that computes the Betti numbers of a filtered simplicial complex which is a subcomplex of a triangulation of S^3. They also sketched the cases where no filtration is given or where the simplicial complex is embedded just in $\mathbb{R}^3$. This last algorithm was developed further by Dey and Guha in [32]. Its distinctive feature is that β_2 is found by recognizing closed surfaces on the boundary of the complex using a graph approach, so the simplicial complex does not need to be a subcomplex of a triangulation of S^3. The algorithm is first defined for three-manifolds, and if the simplicial complex is not a three-manifold, a technique for converting it is described. Sadly, there is no available implementation of this brilliant algorithm. Our work shares several ideas with these articles, though we focus our research on cubical complexes and exploit their structure, which provides a simpler algorithm. Juda and Mrozek presented in [75] an optimal algorithm which computes $\mathbb{Z}_2$ Betti numbers and homology generators of a special class of pseudomanifolds. This is an extension of Delfinado and Edelsbrunner's work for cubical and simplicial complexes. Our work shares several ideas with these articles, though we focus our research on cubical complexes and exploit their structure, which provides a simpler algorithm.

4.2 The Iterative Algorithm

In the following, K denotes a 3D cubical complex.

We first prove that $\beta_0(K)$ can be computed by counting connected components. This is a well known fact, but we include this proof to introduce this kind of reasoning which we also use in Proposition 4.5.

Proposition 4.3. *Let K be a 3D cubical complex and $G_0(K) = (V, E)$ denote the graph such that*

- *$V = K_0$, the 0-cells of K*

- *$E = \{\{u, v\} \mid u \text{ and } v \text{ have a common coface}\}$.*

Thus, $\beta_0(K)$ is the number of connected components in the graph $G_0(K)$.

Proof. Observe that E corresponds to the 1-cells of K, which connect its two faces.

Let $\mathcal{F}$ be a spanning forest of $G_0(K)$. Assume that $\mathcal{F} = \{T_1, \ldots, T_r\}$ where each T_i is a connected component. For each connected component T_i of $\mathcal{F}$, choose a 0-cell r_i as root and put arrows on the other 0-cells pointing to the coface following the tree towards its root. This provides a discrete vector field $\mathcal{V}$ on K where the only critical 0-cells are $r_1, \ldots, r_t$. We now prove that it is a DGVF and that the number of critical 0-cells is minimal, so the statement of the proposition follows.

As the arrows of $\mathcal{V}$ are induced by trees, it is clear that there are no closed $\mathcal{V}$-paths. Hence, $\mathcal{V}$ is a DGVF and thus a HDVF (see Proposition 3.21).

The number of critical 0-cells is minimal if we cannot cancel a critical 0-cell with a critical 1-cell. This is true if $d'(\gamma) = 0$ for each $\gamma \in C_1$, which we prove now. Following Theorem 3.9, we recall that $d' = fd$ and

$$F = -d(S)_{|C} H$$

$$= -d(S)_{|C} \left(H + I - \underbrace{d(S)_{|P}^{-1} d(S)_{|P}}_{H} \right)$$

$$= -d(S)_{|C} \left(I + H(I - d(S)_{|P}) \right) = -d(S)_{|C} + F(I - d(S)_{|P})$$

Thus, if $(\sigma, \tau) \in \mathcal{V}$, $f(\sigma) = -d_{|C}(\tau) + f(\sigma - d_{|P}(\tau))$. By induction, f of a primary 0-cell gives the root r_i of its connected component. Consider any $\gamma \in C_1$. Since $\mathcal{F}$ is a spanning forest, the two faces of γ must be contained in the same connected component of G_0 (since adding this edge to $\mathcal{F}$ must create a cycle). Thus,

$$d'(\gamma) = fd(\gamma) = r_i - r_i = 0$$

$\square$

Figure 4.2 illustrates the construction done in this proof. On the left there is a 2D cubical complex together with its graph $G_0(K)$. In the middle there is a spanning forest of $G_0(K)$ (in red). On the right we can appreciate the induced DGVF after choosing as root the top leftmost 0-cell of the complex.

(A)   (B)   (C)
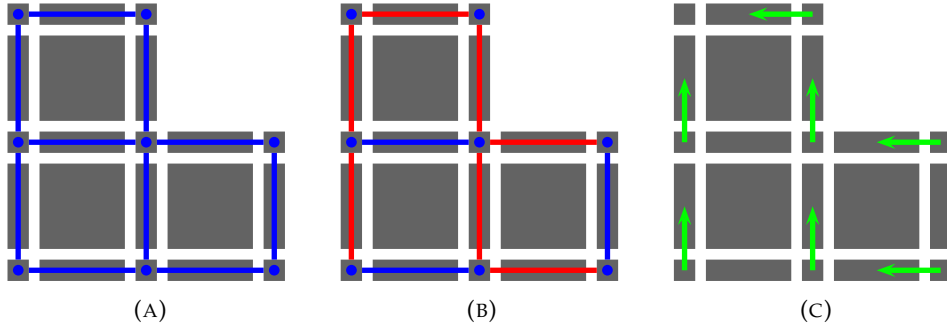
FIGURE 4.2: Illustration of the construction done in the
proof of Proposition 4.3

We now present a proposition similar to Alexander duality which tells how the homology of a 3D cubical complex and its complement in an acyclic supercomplex are linked.

Let L be a 3D cubical complex such that $K \subset L$ and $\beta(L) = (1, 0, 0, 0)$. We typically consider $L = [0, m]^3$ for some $m > 0$, assuming that the coordinates of the cells of K are all positive.

Proposition 4.4. *Let K and L be two 3D cubical complexes such that $K \subset L$ and $\beta(L) = (1, 0, 0, 0)$. Then,*

$$\beta_q(K) = \begin{cases} \beta_1(L - K) + 1 & \text{if } q = 0 \\ \beta_{q+1}(L - K) & \text{else} \end{cases}$$

Proof. Note that, since $K \subset L$, the boundary matrix (for all dimensions together) of L is of the form

$$D = \left[\begin{array}{c|c} D_1 & \cdot \\ \hline 0 & D_2 \end{array} \right]$$

where $D_1 = d(K)_{|K}$ and $D_2 = d(L-K)_{|L-K}$. As $D \cdot D = 0$ and $D_1 \cdot D_1 = 0$, $D_2 \cdot D_2 = 0$ too. Let $X_1 = (P_1, S_1)$ and $X_2 = (P_2, S_2)$ be two perfect HDVFs for K and $L - K$ respectively. Let $X_{12} = (P_{12}, S_{12}) := (P_1 \cup P_2, S_1 \cup S_2)$ be their union. It is a HDVF for L:

$$\det(d(S_{12})_{|P_{12}}) = \det\left(\left[\begin{array}{c|c} A & \cdot \\ \hline 0 & B \end{array} \right]\right)$$

$$= \det(A) \cdot \det(B) \in \mathfrak{R}^*$$

Moreover, as A and B are invertible,

$$d(S_{12})_{|P_{12}}^{-1} = \left[\begin{array}{c|c} A^{-1} & \cdot \\ \hline 0 & B^{-1} \end{array} \right]$$

Let $Y = (P, S)$ be a perfect HDVF for L that extends X_{12}. Thus

$$d(S)_{|P} = \left[\begin{array}{c|c|c|c} A & \cdot & Y_1 & \cdot \\ \hline 0 & B & 0 & Y2 \\ \hline X_1 & \cdot & \cdot & \cdot \\ \hline 0 & X_2 & 0 & \cdot \end{array} \right]$$

where the last two columns (resp. rows) are T_1 and T_2 (resp. Σ_1 and Σ_2), the new secondary (resp. primary) cells belonging to K and $L - K$ respectively. By the Schur determinant formula,

$$\det(d(S)_{|P}) = \det\left(\left[\begin{array}{c|c} A & \cdot \\ \hline 0 & B \end{array} \right]\right) \cdot$$

$$\det\left(\left[\begin{array}{c|c} \cdot & \cdot \\ \hline 0 & \cdot \end{array} \right] - \left[\begin{array}{c|c} X_1 & \cdot \\ \hline 0 & X_2 \end{array} \right] \left[\begin{array}{c|c} A^{-1} & \cdot \\ \hline 0 & B^{-1} \end{array} \right] \left[\begin{array}{c|c} Y_1 & \cdot \\ \hline 0 & Y2 \end{array} \right]\right)$$

so the rightmost determinant must be a unit. However, developing the equation we obtain

$$\left[\begin{array}{c|c} \cdot & \cdot \\ \hline 0 & \cdot \end{array} \right] - \left[\begin{array}{c|c} X_1 H_1 Y_1 & \cdot \\ \hline 0 & X_2 H_2 Y_2 \end{array} \right] = \left[\begin{array}{c|c} 0 & \cdot \\ \hline 0 & 0 \end{array} \right]$$

since X_1 and X_2 are perfect HDVFs. Thus, Σ_2 and T_1 must be empty. This means that all the new couples of cells in Y are from K to $L - K$.

Let $q > 0$. Since $\beta_q(L) = 0$ and $\beta_{q+1}(L) = 0$, all the critical q-cells of K must cancel with the critical $(q + 1)$-cells of $L - K$ and vice versa. Hence, $\beta_q(K) = \beta_{q+1}(L - K)$.

For $q = 0$, since $\beta_0(L) = 1$ and $\beta_1(L) = 0$, all the critical 1-cells of $L - K$ must cancel with all the critical 0-cells of K but one and vice versa. Therefore, $\beta_0(K) = \beta_1(L - K) + 1$. $\qquad \square$

Thus, in order to compute $\beta_2(K)$ we can compute $\beta_3(L-K)$. The following proposition tells that this can be achieved also via counting connected

components

Proposition 4.5. *Let $K \subset L$ be two 3D cubical complexes. Consider the graph $G_3(L - K) = (V, E)$ such that*

- *$V = (L - K)_3 \cup \{\epsilon\}$, the 3-cells of $L - K$ plus an extra (abstract) vertex*

- *$E = \{\{u, v\} \mid u \text{ and } v \text{ have a common face}\} \cup \{\{u, \epsilon\} \mid u \text{ contains a free face}\}$.*

Thus, $\beta_3(L - K)$ is the number of connected components in the graph $G_3(L - K)$ minus one.

Proof. Observe that E corresponds to the 2-cells of $L - K$, which can have two cofaces (type $\{u, v\}$) or just one (type $\{u, \epsilon\}$).

Let $\mathcal{F}$ be a spanning forest of $G_3(L - K)$. Assume that $\mathcal{F} = \{T_1, \ldots, T_r\}$ where each T_i is a connected component and $\epsilon \in T_1$. Fix ϵ as the root of T_1 and, for the other connected components T_i of $\mathcal{F}$, choose a 3-cell r_i as root. Put arrows on the 2-cells pointing to the coface in the opposite direction towards its root. This provides a discrete vector field $\mathcal{V}$ on K where the only critical 3-cells are $r_2, \ldots, r_t$. We now prove that it is a DGVF and that the number of critical 3-cells is minimal.

As the arrows of $\mathcal{V}$ are induced by trees, it is clear that there are no closed $\mathcal{V}$-paths. Hence, $\mathcal{V}$ is a DGVF and thus a HDVF.

The number of critical 3-cells is minimal if we cannot cancel a critical 3-cell with a critical 2-cell. This is true if $d'(\gamma) = 0$ for each $\gamma \in C_3$, or equivalently, if $(d')^*(\gamma) = 0$ for each $\gamma \in C_2$, where $(d')^*$ denotes the dual of d'. We recall that $d' = dg$, so $(d')^* = g^* d^*$ and

$$
G = -Hd(C)_{|P}
$$

$$
= -\left(H + I - d(S)_{|P} \underbrace{d(S)_{|P}^{-1}}_{H} \right) d(C)_{|P}
$$

$$
= -\left(I + (I - d(S)_{|P})H \right) d(C)_{|P} = -d(C)_{|P} + (I - d(S)_{|P})G
$$

Thus, if $(\sigma, \tau) \in V$, $g^*(\tau) = d_{|C}^*(\sigma) + g^*(\tau - d_{|P}^*(\sigma))$. By induction, g^* of a secondary 2-cell gives the root r_i of its connected component (if $i \neq 1$) or the empty chain (if $i = 1$). Consider any $\gamma \in C_2$.

1. If γ has two cofaces, they must be contained in the same connected component of G_3. Thus, $(d')^*(\gamma) = g^* d^*(\gamma) = r_i - r_i = 0$

2. If γ has only one coface then it belongs to T_1. Thus $(d')^*(\gamma) = g^* d^*(\gamma) = 0$

Therefore, $\mathcal{V}$ contains $r - 1$ critical 3-cells and this number is minimal, which completes the proof. $\qquad\square$

Once again, Figure 4.3 illustrates the construction done in this proof in the two-dimensional space. On the left there is a 2D cubical complex with the cells of $L - K$ colored in light blue, with the graph $G_2(L - K)$ superimposed. In the middle there is a spanning forest of $G_2(L - K)$ (in red). On the right we can appreciate the induced DGVF after choosing as root the top leftmost 2-cell of the complex.
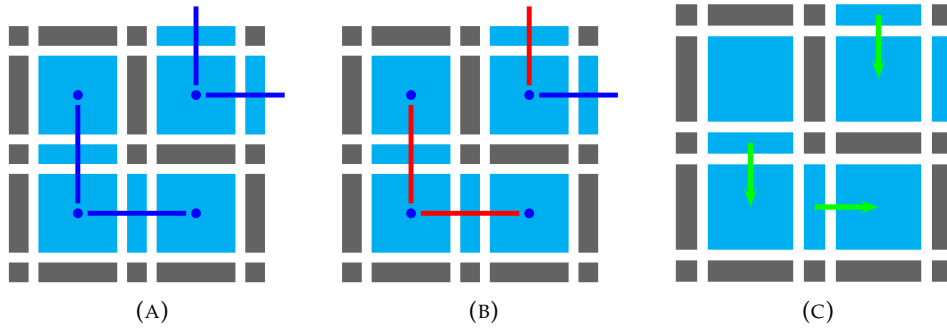
FIGURE 4.3: Illustration of the construction done in the proof of Proposition 4.5

The previous propositions can directly be extended to higher dimensions. Also, L does not need to be a 3D cubical complex of the form $[0, m]^3$. We have considered the bounding box of K, but any acyclic supercomplex can be used. We recall that these statements are true for simplicial complexes, but obtaining an acyclic supercomplex of a simplicial complex of the same dimension does not seem to be a trivial task.

We can count connected components using breadth first search. Algorithm 3 explicitly illustrates this.

Algorithm 3: Count connected components in a graph with BFS

> **Input:** A graph $G = (V, E)$
> **Output:** Number of connected components of G
> $n \leftarrow 0$;
> **foreach** $u \in V$ **do**
> > **if** u *not marked* **then**
> > > Q.push(u); mark u;
> > > **while** Q *not empty* **do**
> > > > $u \leftarrow Q$.pop();
> > > > **foreach** v *such that* $\{u, v\} \in E$ **do**
> > > > > Q.push(v); mark v;
> > >
> > > $n \leftarrow n + 1$;
>
> **return** n

We actually do not need to build the graphs $G_0(K)$ nor $G_3(L - K)$ to count its connected components, since they are included in K and $L - K$.

Algorithm 3 has complexity $\mathcal{O}(|V| + |E|)$. In $G_0(K)$, $|V|$ (resp. $|E|$) is the number of 0-cubes (resp. 1-cube) of K. Also, in $G_3(L - K)$ $|V|$ (resp. $|E|$) is the number of 3-cubes (resp. 2-cube) of $L - K$. In addition, computing $\chi(K)$ only needs counting the cubes of K with the appropriate sign. Hence, Algorithm 4—called *ViteBetti* from the French word "vite" (fast)—computes the Betti numbers of a 3D cubical complex in $\mathcal{O}(n)$ time, where n denotes the number of cubes in L.

Algorithm 4: ViteBetti

Input: A 3D cubical complex K
Output: Its Betti numbers: β_0, β_1, β_2
$\chi \leftarrow \chi(K)$;
$\beta_0 \leftarrow$ number of connected components of $G_0(K)$;
$\beta_2 \leftarrow$ number of connected components of $G_3(L - K) - 1$;
$\beta_1 \leftarrow \beta_0 + \beta_2 - \chi$;
return β_0, β_1, β_2

4.3   The Recursive Algorithm

We showed in the previous section that the computation of the Betti numbers of a 3D cubical complex reduces to (1) compute the Euler-Poincaré characteristic and (2) to find the number of connected components of two graphs. We present in this section a way to parallelize Algorithm 4 with a divide-and-conquer approach.

Computing the Euler-Poincaré characteristic can be achieved in $\mathcal{O}(\log(n))$ time on a parallel machine. Assuming that the 3D cubical complex K is encoded as a binary 3D array A_K (called *CubeMap* in [115]), it suffices to split the array into two parts, recursively sum both of them and then sum the two values. The recursion stops whenever only two elements remain, in which case we sum them with their corresponding signs. The recursion depth of this method is $\lceil \log_2(n) \rceil$, so the problem can be solved in $\mathcal{O}(\log(n))$ with $n/2$ processors. A simpler method consists in dividing the binary array into p parts, summing each of them in parallel and then summing all the partial results. This approach needs $\mathcal{O}(n/p + p)$ steps, so it can be done in $\mathcal{O}(\sqrt{n})$ time with $p = \sqrt{n}$ processors.

There has been an extensive research about computing connected components in parallel, see [68, 107, 114] for some examples. We present here a simple recursive method that works well for commodity computers with few processors.

Algorithm 3 described how to count connected components by traversing the graph. This approach is not suited for parallelization since it uses a queue data structure. Another well known approach to compute connected components is to use the *disjoint-set* data structure (see [20, Chap. 21]). This data structure maintains a collection $S = \{S_1, \ldots, S_k\}$ of disjoint sets. Each set in S is identified by a representative, which is a member of the set. The following operations can be performed on a disjoint-set data structure:

- `MakeSet(`u`)` - creates a new set whose only member (and thus representative) is u.

- `Find(`u`)` - returns a pointer to the representative of the (unique) set containing u.

- `Union(`u, v`)` - merges the sets containing u and v into a new set which is the union of these two sets.

To compute connected components of a graph it is enough to call `Union`(u, v) for each pair (u, v) of adjacent vertices. A parallel version of such algorithm requires synchronization, so in practice it cannot be implemented efficiently. However, the regular structure of the cubical complex allows us to propose a different approach where synchronization is not needed. The idea is to recursively cut the graph in two halves, find the connected components in each half and then merge them.

We recall that both $G_0(K)$ and $G_3(L-K)$ (without the special vertex ϵ) are grid graphs and that their vertices are identified to points of $\mathbb{Z}^3$ via the Khalimsky coordinates. We define the *left slice*, *right slice* and *middle slice* of a subset of vertices W in dimension d by x respectively as

$$S(W, x_-, d) := \{u \in W \mid u = (u_1, u_2, u_3), u_d < x\}$$
$$S(W, x_+, d) := \{u \in W \mid u = (u_1, u_2, u_3), x \leq u_d\}$$
$$S(W, x_0, d) := \{u \in W \mid u = (u_1, u_2, u_3), x - 1 \leq u_d \leq x\}.$$

These three operations allow us to divide a set of vertices into two parts (the left and the right slice) plus a small subset that intersects both.

Algorithm 5 recursively computes the connected components of a grid graph. Observe that, at each step of the recursion, the set W is divided into two parts. Each of them can be treated independently since there are no edges between both sets. We then consider the middle slice to combine both parts, which is not subdivided (since $\epsilon = \infty$).

Algorithm 5: RecursiveCC

Input: $G = (V, E)$ a grid graph, $\epsilon > 0$
$S \leftarrow$ disjoint-set data structure;
RecursiveCC$(V, 0, \epsilon)$;
return $|S|$;
Procedure `RecursiveCC`(W, d, ϵ)
    Input: $W \subset V$, $d \in \mathbb{Z}$, $\epsilon > 0$
1   **if** $|W| > \epsilon$ **then**
2        $d \leftarrow ((d+1) \mod 3) + 1$;
        $x \leftarrow$ middle point among the dth Khalimsky coordinates of W;
3   RecursiveCC$(S(W, x_-, d), d, \epsilon)$;
4   RecursiveCC$(S(W, x_+, d), d, \epsilon)$;
5   RecursiveCC$(S(W, x_0, d), d, \infty)$;
6   **else**
7   **foreach** $u \in W$ **do**
8   MakeSet(u)
9   **foreach** $u \in W$ **do**
10   **foreach** $v \in W$ *incident to* u **do**
11   Union(u, v);

At the end of Algorithm 5 all the edges have been treated, so the disjoint-set data structure S contains the connected components of the graph. In order to use this method for counting connected components in Algorithm 4 we need to clarify what happens with the vertex ϵ of $G_3(L - K)$. Instead of directly counting the connected components of $G_3(L - K)$, we do it for

the induced subgraph without ϵ, which is a grid graph. Then, by merging all the sets in S containing a vertex incident to ϵ in $G_3(L - K)$, we obtain the number of connected components of $G_3(L - K)$. However, it is more convenient to add a flag to the sets containing such vertices (at line 8) and then consider only one of those sets when counting $|S|$.

4.4   Results

We compare in this section our algorithm with the library CAPD::RedHom [77], specialized in Betti numbers computation on cubical complexes. We have used the default settings, which executes shaving, coreduction algorithm, discrete Morse theory reduction and finally algebraic reductions. Three versions of our algorithm are tested: *VB-i* for the iterative version introduced in Section 4.2, *VB-r* for the recursive version described in Section 4.3 and *VB-rp* for the same algorithm using parallel computing.

Our algorithms are implemented in C++ and compiled by the GNU compiler g++ (version 5.2.1) with option -O3. The parallel algorithm *VB-rp* uses the Threading Building Blocks (TBB) library [70] (version 4.4).

We have tested the algorithm with random 3D cubical complexes. A $(2 \cdot m + 1)^3$ cubical complex consists in a cube of m^3 3-cubes with their faces. Each cubical cell (and its faces) is added with a fixed probability p (see the random model $K(p, m)$ in Section 3.9). We have made 10 random cubical complexes of size 51^3, 101^3, 201^3, 301^3, 401^3 and 501^3 and probability 0.25, 0.5 and 0.75. We use $\epsilon = 10^5$ as threshold for *VB-r* and *VB-rp*, which seems to be the best one for this implementation. We computed these 180 cubical complexes on a Dell PC with 3.70GHz $\times$ 8 Intel Xeon E5-1630 v3 CPU and 31.3 GB RAM. The experimental results are obtained by averaging the execution time of the algorithms, while the reading time (which exceeds the execution time for the bigger complexes) is omitted. Table 4.1 shows the results obtained. The Betti numbers calculated by each of the algorithms are obviously the same.

Size	RedHom	VB-i	VB-r	VB-rp
51^3	0.1842	0.0026	0.0026	0.0023
101^3	1.268	0.0142	0.0148	0.0091
201^3	10.78	0.1309	0.1232	0.0552
301^3	40.89	0.4303	0.4176	0.1583
401^3	101.26	1.436	0.983	0.3092
501^3	—	3.609	1.977	0.5494

TABLE 4.1: Execution time (in seconds) versus the size of the cubical complex.

We can appreciate that our algorithm clearly improves the execution time of RedHom. Moreover, RedHom uses a memory consuming data structure for its computation which does not allow us to process complexes bigger than 401^3, which can be achieved by *ViteBetti*.

The recursive version of our algorithm, *VB-r*, is clearly faster than *VB-i*. The TBB library automatically chooses the number of parallel threads, but we can appreciate that *VB-rp* is more than three times faster than *VB-r* for cubical complexes of size bigger than 401^3.

4.5 Conclusion

We have introduced an algorithm for computing the Betti numbers of a 3D cubical complex in linear time without algebraic calculations. It is based on the counting of connected components and it exploits the Alexander duality and the Euler-Poincaré formula. Its correctness is proved using the HDVF framework, which allows us to connect the Betti numbers β_0 and β_2 of the complex with the number of connected components of two graphs associated to the complex and its complement. Moreover, Proposition 4.4 gives a version of the Alexander duality in the context of cubical complexes, and its proof is not only restricted to the three-dimensional space.

We have presented two versions of this algorithm: the iterative and the recursive one. The former is the most obvious, while the latter takes advantage of the regular structure of the cubical complex in a divide-and-conquer way and it is suitable for parallel computation. These two versions have been implemented and compared with the library RedHom [77], bringing better execution times and memory consumptions.

We count connected components using breadth first search in the iterative version or disjoint-set data structure in the recursive version. However, in the image processing community this type of approach is avoided because random access in large arrays produces several many page faults. Thus, many raster scan-based algorithms have been designed (see [105, 85, 36] for instance). We plan to test these approaches to examine if they can improve our algorithm.

Other perspective is to use a similar algorithm for computing the Betti numbers of a binary volume, without building its associated cubical complex. The resulting algorithm is similar to that introduced by Nakamura and Aizawa in [100], though their work lacks some formalism since the quantities they compute (which they call Betti numbers) are not clearly related to homology.

Chapter 5

Measuring Holes

T HIS chapter is partially based on the conference paper [59]. We relate some issues where we transfer some geometry into the homology.

5.1 Introduction

It is clear from the definition of the homology groups (see Section 2.2.2) that the geometry of the space is neglected. Homology groups are defined through the boundary operators, which just encode the incidence relation between the cells of the complex. The shape of the cells, their size or their position are not considered. This is completely coherent with the topological approach, which considers spaces up to deformations, so we should not complain about it. However, there is a *trick* for capturing some geometry: persistent homology.

We recall that persistent homology studies the homology not only of a complex, but of a sequence of complexes. Moreover, these complexes are nested (otherwise, we may use *zigzag persistent homology* [13]). By studying the inclusion maps between the complexes, persistent homology succeeds to connect the homology groups of the complexes and produces the *persistence intervals*, which tell the lifetime of the holes.

Let us mention now a point that may seem obvious, but which usually avoids many misunderstandings. Given a set of points in $\mathbb{R}^d$, we can obtain a sequence of complexes by considering the Čech, Vietoris-Rips or alpha complexes [43, Chap. III] with an incrementing parameter. Then, the persistence intervals give a multi-scale homological information of the cloud of points. Research works in persistent homology often use this pipeline in order to study *topological inference*. However, persistent homology is defined on a filtration and not on a set of points.

By taking a filtration based on a distance function, we can link geometry and homology. Fortunately, this works quite well. We briefly introduce how we do it. In the following we only consider discrete objects (see Section 2.4) with their associated cubical complex.

Size of a hole   The objects depicted in Figure 5.1 have the same Betti numbers ($\beta_0 = \beta_1 = 1$). Thus, they are equal from a homological point of view. However, they look very different, notably because the hole on the right is *bigger* than the hole on the left. That is why we want to define a measure on holes, so that we can discriminate between objects.

We can say that the hole on the left is *smaller* than the hole on the right because we need to add a smaller area to the object to fill the hole. Figure 5.2 shows how we can fill the 1-holes by adding a *patch* (in red).
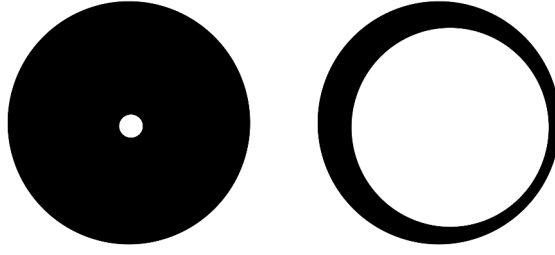
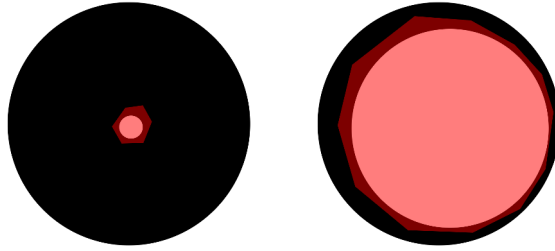FIGURE 5.1: Two objects with isomorphic homology groups.

FIGURE 5.2: By adding a patch (in red), the holes disappear.

This motivates another idea: we can erase holes also by removing a part of the object. The hole on the left is *thicker* than the hole on the right because we need to remove a bigger area to break the hole. Figure 5.3 shows how we can break the 1-holes by removing a *patch* (in blue). Consequently, there is another measure for the holes, related to their fragility.
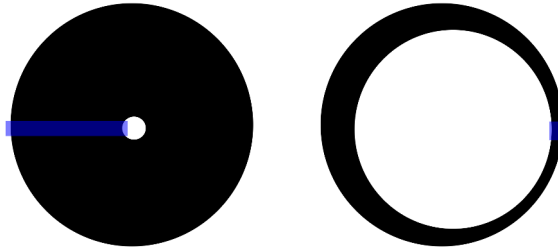

FIGURE 5.3: By removing a part (in blue), the holes disappear.

This duality seems to be justified by the definition of the homology groups. A non-trivial element $[x] \neq 0$ of a homology group is a cycle $(d(x) = 0)$ which is not a boundary $(x \notin d(\mathsf{C}))$. If we add some cells to the complex, it may become a boundary of a new chain and thus $[x] = 0$. On the other hand, if we remove some cells from the complex, some of them in x and in its homologous chains, x may stop being a cycle and thus $x \notin \ker(d)$. We will also see in Section 5.4 that the two measures are related to the duality between homology and cohomology, which is quite surprising.

We properly define these two measures in Section 5.2 and we prove that they are robust.

Visualizing a hole   We have previously mentioned that an advantage of homology is that the Betti numbers have an easy interpretation (in a three-dimensional space). Thus, they serve not only as an invariant, but as a tool for understanding a shape. Moreover, the elements of the homology groups are classes of chains, so we can display them by marking the cells contained in one of the chains in the class. We usually say that, by displaying the homology generators (a representative, actually), we can see the holes. Figure 5.4-(left) is a typical illustration of the generators of H_1 of a torus. However, this is not the general case, as Figure 5.4-(right) also shows two generators which are not *well-shaped*.
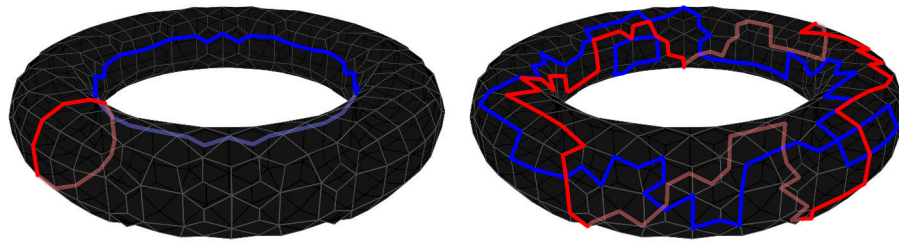
FIGURE 5.4: Different representatives for the two homology generators of dimension 1.

In Section 5.3 we show a different visualization of the holes which uses their homology groups in a less direct way. Its computation is completely related to the computation of the two measures.

Small generators   Finding small homology generators is not only a challenging work, but it also usually provides *well-shaped* generators. Let us assume that $\mathfrak{R} = \mathbb{Z}_2$, so chains are sets of cells. Let K be a CW complex and $q \geq 0$, we want to find a set of q-dimensional cycles $\{x_1, \ldots, x_r\}$ such that $\{[x_1], \ldots, [x_r]\}$ is a base for $H_q(K)$ and the sum of the volumes of the cells of the cycles is minimum. This problem has widely been studied, mainly on simplicial complexes triangulating a 2-manifold, but also with other geometric criteria. We introduce in Section 5.4 two algorithms for computing a set of generators for the homology and the cohomology groups. These generators are usually small, though this is not guaranteed. They make use of the two measures of the holes and find the generators by constructing a filtration on the object for each hole.

Opening and closing holes   Let us recover the previously introduced problem on opening and closing holes. Given an object O, which is the minimal set of voxels that we have to add (or remove) so that it becomes an acyclic space (i.e., $\beta_0 = 1$ and $\beta_q = 0$ for $q > 0$)? We can be even more demanding and ask to open or close just one hole, maybe depending on its measure. Given a cycle x, $[x] \neq 0$, we wonder which is the minimal set of voxels S such that:

- $x \notin \mathrm{im}(\iota)$ where ι denotes the inclusion map $\iota : H(O \setminus S) \longrightarrow H(O)$

- $\iota(x) = 0$ where ι denotes the inclusion map $\iota : H(O) \longrightarrow H(O \cup S)$

It seems natural to solve this problem using homology. We present in Section 5.5 two algorithms that provide a set of cubical cells for cracking or filling a hole. Note that these algorithms do not take any cycle as input, but only the generators obtained during the computation of the measures. Unfortunately, we can only provide a minimal set of voxels by solving an Integer Programming problem, which is a hard problem.

There are many works combining homology and geometry. Most of them study only the first homology group H_1 of simplicial complexes triangulating a 2-manifold. Simplicial complexes are usually endowed with a weight function (which can be constant in the simplest case) on its simplices, so we can ask for a base for H_1 minimizing this weight function [47, 17]. Dey et al. [34, 31] developed a similar work with the restriction of classifying the 1-holes into tunnels and handles. Other works search the shortest cycle homologous to a given one [28, 33]. Dey et al. introduced an algorithm in [35] that computes the shortest base for H_1 on any simplicial complex. On the other hand, Chen and Freedman proved in [18] that all these problems are NP-hard in general dimension, unless we minimize the *radius* of the cycles.

Localized homology [118] was intended to be the general tool for combining homology and geometry. It considers a topological space together with a cover, and provides a homology base where each element belongs to a set of the cover. This theory can be used by considering specific covers related to the problem we want to solve.

Most of these works use persistent homology, and many of them suffer from a high complexity because they consider many cycles to find the smallest ones.

5.2   The measures

In this section we study how to measure the holes of a discrete object.

Chen and Freedman [17] introduced a measure for homology classes which we briefly describe. Let K be a simplicial complex where each edge has a nonnegative weight (it can be its length or just 1). We define the *discrete geodesic distance* from a 0-simplex σ to a simplex τ as follows. If τ is a 0-simplex, it is the length of the minimum weight path connecting σ to τ along the 1-simplices of K. If the dimension of τ is higher, we consider the maximal discrete geodesic distance to its 0-dimensional faces. Thus, the *geodesic ball* $B(\sigma, r)$ is the set of simplices whose discrete geodesic distance to σ is less than or equal to r. Note that a geodesic ball is a subcomplex.

Hence, the measure of a homology class $[x]$ is the smallest radius r such that the class vanishes in the relative homology group $H(K, B(\sigma, r)) = H(K)/H(B(\sigma, r))$ for some $\sigma \in K$. This measure is usually called *radius*. This definition has the advantage that it measures any homology class in any CW complex. We could compute an optimal homology base, that is, a set of cycles which minimizes the sum of its radiuses, and use this measure to characterize the complex. However, computing the measure of a class is known to take $\mathcal{O}(n^4)$ time and their algorithm for finding an optimal q-homology base runs in $\mathcal{O}(\beta_q n^3 \log^2(n))$ time.

Our measures are only defined for discrete objects and cannot consider individual homology classes, but they can be computed faster (there is no optimization problem) and they have very good geometric properties as it will be shown in this chapter.

Let $O \subset \mathbb{Z}^d$ be a discrete object (see Section 2.4). We fix $d = 3$ for simplicity, but the generalization to any dimension is direct. We have two ways of building the cubical complex K associated to O depending on which connectivity relation we choose: the 6 or the 26-connectivity relation.

The *distance transform dt_O* of O is the map that sends every voxel $x \in O$ to

$$dt_O(x) = d(x, O) = \min \{d(x, y) \mid y \notin O\}$$

where

$$d(x, y) = \sqrt{\sum_{i=1}^{3} (x_i - y_i)^2}$$

is the Euclidean distance. However, we can also consider other distances such as the Manhattan distance (L_1), the chessboard distance (L_∞), distances based on chamfer masks [92, 9] or sequences of chamfer masks [97, 101].

The *signed distance transform sdt_O* of O is the map $sdt_O : \mathbb{Z}^3 \to \mathbb{R}$ defined as follows:

$$sdt_O(x) = \begin{cases} -dt_O(x) = -\min \{d(x, y) \mid y \notin O\} & \text{if } x \in O \\ dt_{\mathbb{Z}^3 \setminus O}(x) = \min \{d(x, y) \mid y \in O\} & \text{if } x \notin O \end{cases}$$

Let us point out some simple properties about the sublevel sets $L_t^-(sdt_O) := sdt_O^{-1}(]-\infty, t])$ of the signed distance transform.

1. $O = sdt_O^{-1}(]-\infty, 0])$

2. $\mathbb{Z}^3 = sdt_O^{-1}(]-\infty, \infty[)$

3. $sdt_O^{-1}(]-\infty, a]) \subset sdt_O^{-1}(]-\infty, b])$ whenever $a < b$

Figure 5.5 shows five sublevel sets at different values. Observe that the sequence of objects $\left(sdt_O^{-1}(]-\infty, t[)\right)_{t=-\infty}^{0}$ looks like an erosion of O, while $\left(sdt_O^{-1}(]-\infty, t[)\right)_{t=0}^{\infty}$ seems a dilation.

We now define the filtration associated to the signed distance transform. A simple formulation of this filtration is

$$F = \left(K[L_t^-(sdt_O)]\right)_{t \in \mathbb{R}}$$

where $K[Y]$ denotes the primal or the dual associated cubical complex of Y.

$PD_q(F) \subset \mathbb{R}^2$ denotes the persistence diagram in dimension q of this filtration. We denote $TB_q = \{(x, y) \in PD_q(F) \mid x < 0, y > 0\}$. It is obvious that TB_q contains $\beta_q(K)$ pairs, that is, there are as many pairs in TB_q as q-holes in O.

Definition 5.1. *Let $O \subset \mathbb{Z}^3$ be a discrete object. Let us fix a distance function $d : \mathbb{Z}^3 \times \mathbb{Z}^3 \to \mathbb{R}$ and a connectivity relation. Let $q \geq 0$,*
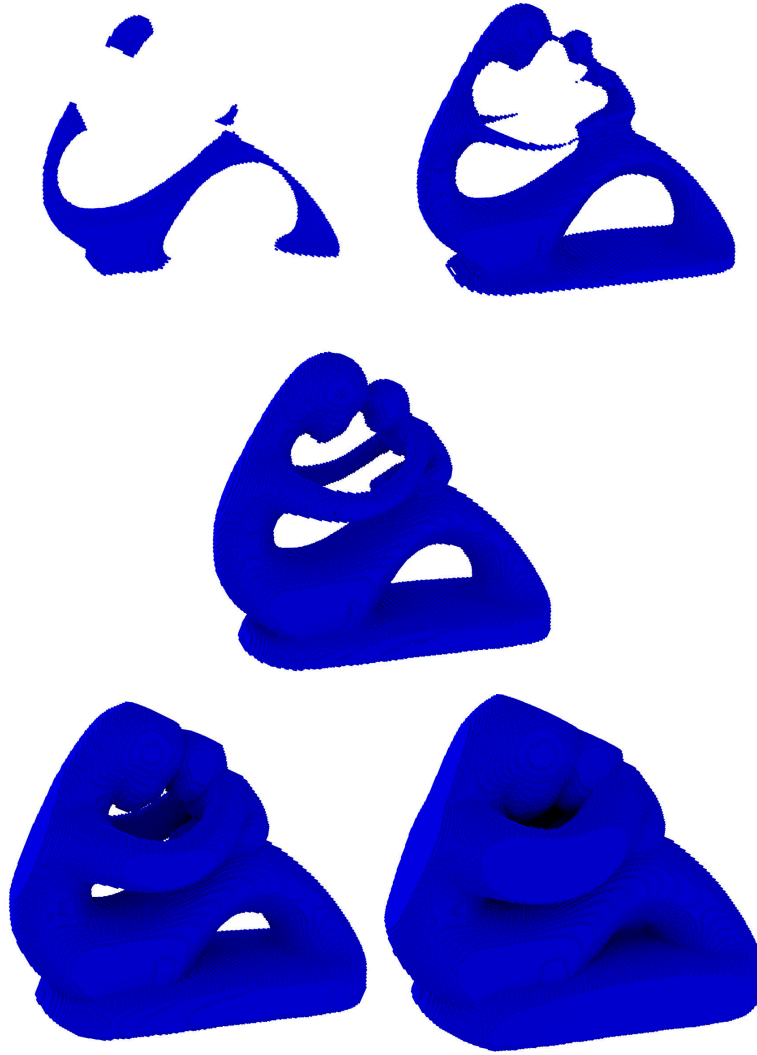
FIGURE 5.5: Sublevel sets of the signed distance transform
at values -10 (top-left), -5 (top-right), 0 (middle), 5 (bottom-
right) and 10 (bottom-right)

- *The* thickness *of the q-holes of O are the values* $\{-x \mid (x, y) \in TB_q\}$

- *The* breadth *of the q-holes of O are the values* $\{y \mid (x, y) \in TB_q\}$

Observe that the thickness and the breadth of the holes appear in pairs. We can thus represent them as points in $\mathbb{R}^2$ in the *thickness-breadth diagram*, just like the persistence diagrams. We call these points *thickness-breadth pairs*. The interpretation of the thickness-breadth diagram is similar to that of the persistence diagram: points close to the axes are holes with one small measure which may be originated by the presence of noise in the discrete object. Section 5.3 contains many examples of thickness-breadth diagrams.

We speak about measures of the holes since we obtain as many values as holes in the object. However, we cannot measure a given hole (that is, a cycle x such that $[x] \neq 0$). This is why we have preferred to talk about *measures for the homology groups* in [59], which seems a more correct formulation. Nevertheless, we think that speaking about measures for the holes sounds clearer.

5.2.1 On the computation of the measures

In this section we give a more detailed description of how the measures are computed.

Let us assume that the discrete object is contained in a bounding box $BB := [0, w_1] \times [0, w_2] \times [0, w_3] \subset \mathbb{Z}^3$. The signed distance transform can be obtained by computing the distance transform of O and $BB \setminus O$ in $\mathcal{O}(w_1 w_2 w_3)$ [74].

Next, the filtration induced by sdt_O depends on the associated cubical complex that we consider.

- Primal associated cubical complex: let $K := K_p[BB]$ be the primal cubical complex associated to the bounding box BB. We define the filtration induced by sdt_O in terms of a function f_O defined on K. Recall that each 3-cube is identified with a voxel of BB. Thus, for every 3-cube $\sigma \in K$, $f_O(\sigma)$ takes the value of sdt_O on its associated voxel. For the rest of the cubes, the value of f_O is assigned so its sublevel sets are complexes. Namely, $f_O(\sigma) = \min \left\{ f_O(\tau^{(3)}) \mid \tau > \sigma \right\}$.

  In other words, f_O maps each cube $\sigma \in K$ to the first value t such that $\sigma \in K_p[L_t^-(sdt_O)]$.

- Dual associated cubical complex: the description is similar. Let $K := K_d[BB]$ be the dual cubical complex associated to the bounding box BB. Since every 0-cube is identified with a voxel of BB, $f_O(\sigma)$ takes the value of sdt_O on its associated voxel for each 0-cube $\sigma \in K$. For the rest of the cubes, $f_O(\sigma) = \max \left\{ f_O(\tau^{(0)}) \mid \tau < \sigma \right\}$.

Let $a_0 < a_1 < \cdots < -1 < 1 < \cdots < a_n$ be the different values of f_O over K. Thus, we can consider the filtration F:

$$K^0 = f_O^{-1}(]-\infty, a_0]) \subset \cdots \subset K[O] \subset \cdots \subset K^n = f_O^{-1}(]-\infty, a_n]) = K$$

Persistent homology is a very active field of research. The computation of the persistence intervals of a filtration has cubical worst-case complexity. An algorithm in matrix multiplication time was introduced in [88]. However, the most recent algorithms [10, 7] are observed to have near linear complexity. An algorithm adapted for cubical complexes was developed in [115].

Algorithms for persistent homology consider a kind of *elementary* filtration where each step consists in adding only one cell to the previous complex. Thus, we need to decompose the filtration F. Some heuristics are given in [7] for this decomposition in order to accelerate the computation of the persistence intervals. We thus obtain some sets P_q of pairs of cells $(\sigma^{(q)}, \tau^{(q)})$ for $q \geq 0$. The q-dimensional persistence diagram is

$$PD_q(f) = \{(f(\sigma), f(\tau)) \mid (\sigma, \tau) \in P_q\} \cup \{(f(\gamma), \infty) \mid \gamma \text{ not paired}\}$$

Observe that this set does not depend on how the filtration F is refined. However, the pair of cells associated with each point does depend. This is relevant for the following sections. There are typically many points of the form (x, x) in the persistence diagram which are usually ignored.

Regarding the thickness-breadth diagram, there exists only one point (x, ∞) corresponding to the first connected component that appears in the filtration. This point can be plotted as $(x, -1)$.

5.2.2 The robustness of the measures

We prove in this section the robustness of the measures. The idea is that if we slightly deform an object, its measures suffer small changes.

The celebrated article [19] introduced a theorem for the stability of the persistence diagrams. We recall that $||x||_\infty = \max \{|x_1|, |x_2|\}$ for $x \in \mathbb{R}^2$ and $||f||_\infty = \max \{|f(a)| \mid a \in A\}$ for a function $f : A \to \mathbb{R}$. We can compare two persistence diagrams via the Hausdorff distance.

Let X, Y be two multisets (sets with repetitions) of points in $\mathbb{R}^2$. Their *Hausdorff distance* is

$$d_H(X, Y) = \max \left\{ \max_x \min_y ||x - y||_\infty, \max_y \min_x ||x - y||_\infty \right\}$$

where $x \in X$ and $y \in Y$. Thus, if $d_H(X, Y) = \epsilon$ then for every $x \in X$ there is a $y \in Y$ such that $||x - y||_\infty \leq \epsilon$ and vice versa.

Let us adapt one of the theorems of [19] to our context. Let f and g be two functions on a cubical complex defining a filtration, that is, their sublevel sets are cubical complexes (each cube contains all its faces in the sublevel set). Let $PD(f)$ and $PD(g)$ be their respective persistence diagrams (all dimensions taken together). Therefore,

$$d_H(PD(f), PD(g)) \leq ||f - g||_\infty$$

Consequently, if the two functions are similar, their associated persistence diagrams are also similar.

Let now X and Y be two discrete objects. We call f_X and f_Y the filtration functions induced by sdt_X and sdt_Y respectively.

Lemma 5.1. *Let $x, y \in \mathbb{Z}^3$ be two 26-neighbors and $A \subset \mathbb{Z}^3$. Then*

$$|sdt_A(x) - sdt_A(y)| \leq 2\sqrt{3}$$

Proof. We recall that, as x and y are 26-neighbors, $|x - y| := ||x - y||_2 \leq \sqrt{3}$. We have to consider four cases:

- $x, y \notin A$. Let us call $p_x, p_y \in A$ their closest points in A. Thus

$$sdt_A(x) = |x - p_x| \leq |x - p_y|$$
$$\leq |x - y| + |y - p_y| = |x - y| + sdt_A(y)$$

  Thus,

$$sdt_A(x) - sdt_A(y) \leq |x - y|$$

  By symmetry,

$$|sdt_A(x) - sdt_A(y)| \leq |x - y| \leq \sqrt{3} \leq 2\sqrt{3}$$

- $x \notin A, y \in A$. Then

$$sdt_A(x) \leq |x - y|$$
$$-sdt_A(y) \leq |y - x|$$

Thus,

$$sdt_A(x) - sdt_A(y) \leq 2 \cdot |x - y| \leq 2\sqrt{3}$$

By symmetry,

$$|sdt_A(x) - sdt_A(y)| \leq 2\sqrt{3}$$

- The other two cases follow the same arguments.

$\square$

Theorem 5.2. *Let X and Y be two discrete objects in $\mathbb{Z}^3$. Let us call*

$$\delta = d_H(X, Y) + d_H(\mathbb{Z}^3 \setminus X, \mathbb{Z}^3 \setminus Y) + 2\sqrt{3}.$$

Then, for every thickness-breadth pair $p_X = (x, y)$ of X such that $x, y > \delta$, there exists another thickness-breadth pair $p_Y = (x', y')$ of Y such that

$$||p_X - p_Y||_\infty \leq \delta$$

Proof. Let σ be an elementary cube. Let us consider the two possible cubical complexes associated to the discrete objects.

- Primal associated cubical complex: $f_X(\sigma) = f_X(\tau_X)$ for a 3-dimensional coface τ_X. Similarly, $f_Y(\sigma) = f_Y(\tau_Y)$. Let q_X and q_Y denote the voxels associated to the 3-cubes τ_X and τ_Y respectively.

- Dual associated cubical complex: $f_X(\sigma) = f_X(\tau_X)$ for a 0-dimensional face τ_X. Similarly, $f_Y(\sigma) = f_Y(\tau_Y)$. Let q_X and q_Y denote the voxels associated to the 0-cubes τ_X and τ_Y respectively.

In both cases q_X and q_Y are 26-neighbors, so $||q_X - q_Y|| \leq \sqrt{3}$. Therefore,

$$\begin{aligned}
|f_X(\sigma) - f_Y(\sigma)| &= |f_X(\tau_X) - f_Y(\tau_Y)| \\
&= |sdt_X(q_X) - sdt_Y(q_Y)| \\
&\leq |sdt_X(q_X) - sdt_Y(q_X)| + |sdt_Y(q_X) - sdt_Y(q_Y)|
\end{aligned}$$

By Theorem 2 of [81],

$$||sdt_X - sdt_Y||_\infty \leq d_H(X, Y) + d_H(\mathbb{Z}^3 \setminus X, \mathbb{Z}^3 \setminus Y)$$

Thus,

$$\begin{aligned}
|f_X(\sigma) - f_Y(\sigma)| &\leq |sdt_X(q_X) - sdt_Y(q_X)| + |sdt_Y(q_X) - sdt_Y(q_Y)| \\
&\leq d_H(X, Y) + d_H(\mathbb{Z}^3 \setminus X, \mathbb{Z}^3 \setminus Y) + 2\sqrt{3}
\end{aligned}$$

Consequently,

$$d_H(PD(f_X), PD(f_Y)) \leq ||f_X - f_Y||_\infty \leq \delta$$

As the thickness-breadth diagram is the intersection of the persistence diagram with the quadrant $\{(x, y) \in \mathbb{R}^2 : x, y \geq 0\}$, the theorem follows from this.　　　　　　　　　　　　　　　　　　　　　　　　　　　　　□

Observe that, if a thickness-breadth pair is close to the axes, a small perturbation in the dicrete object can make it disappear. In other words, a hole being not thick or broad enough can easily disappear after a small perturbation. Hence, in order to compare the thickness-breadth pairs of two discrete objects, they must be far enough from the axes. This is why we ask their values to be bigger than δ in Theorem 5.2.

Thus, we can bound the distance between two thickness-breadth diagrams via the Hausdorff distance of the two objects and their complements.

5.3 Thickness and breadth balls

In the previous section we represented the thickness and the breadth of the holes as points in the thickness-breadth diagram. There is an alternative way, in terms of balls.

Each thickness-breadth pair (t, b) has an associated pair of cubes (σ, τ). Therefore,

- its *thickness ball* is the ball centered at the barycenter of σ with radius t;

- its *breadth ball* is the ball centered at the barycenter of τ with radius b.

Observe that the thickness balls are contained in the object, while the breadth balls are outside. Note however that the cubes σ and τ are not unique since they depend on how we decompose the filtration induced by the signed distance transform.

The thickness and breadth balls allow us to represent both measures directly on the object. Moreover, the first impression we have when we encounter the breadth balls is that they are in the center of the holes.

It is well accepted to visualize holes as representatives for a set of homology generators. For each representative, which is a chain, we mark those cells with a non-negative coefficient. Nevertheless, these representatives can be visually unpleasant. In order to better formalize this aspect, some authors suggest that the best representatives are those which are minimal in terms of their length, area, volume, etc [47, 17, 31].

Breadth balls emerge as an interesting alternative to the representation of holes in terms of homology generators. Symmetrically, thickness balls look like cohomology generators.

In the following we show several examples of thickness-breadth diagrams and balls. We have considered several meshes from the AimAtShape repository[1] (except for *Buddha*[2]) and we have converted them to binary volumes using the software *binvox*[3]. The thickness-diagram and balls were computed with a specific software which does not take advantage on the

---

[1]http://visionair.ge.imati.cnr.it/
[2]Courtesy of the Stanford Computer Graphics Laboratory
[3]http://www.patrickmin.com/binvox/

latest results in persistent homology computation [10, 7], so we have omitted the time spent in these calculations.

Figure 5.6 illustrates a voxelized version of *Buddha*. The thickness balls are shown in red, while the breadth balls are in green. We only display the balls of the 1-holes, since the other ones are less visually interesting. The thickness-breadth diagram shows the 0-holes (red circle), 1-holes (green triangles) and 2-holes (blue square). Observe that the thickness of the only connected components, which is ∞, is represented as -1. Also, the small (in both thickness and breadth) 2-hole is due to an error in the voxelization process of the mesh. Figures 5.7–5.16 show other models.

5.4   Small generators

In this section we introduce a heuristic for obtaining *well-shaped* generators for the homology and cohomology groups based on the thickness and breadth balls.

In the previous section we claimed that the thickness and breadth balls seem to be a good alternative for localizing holes, instead of displaying the homology generators. Therefore, it is a natural question to wonder if we can use these balls to find well-shaped homology generators. Moreover, the duality thickness/breadth of the measures provides also results for the cohomology groups.

The *localization problem* [18] consists in finding the smallest representative cycle of a homology class with regard to a geometric measure. It seems natural that such cycles are good representatives for the holes. We explain some of these measures in a nutshell. Let K be a CW complex endowed with a weight function on its cells (its q-dimensional volume or just a constant). Some measures are:

Volume The volume of a chain is the sum of the weights of its cells.

Diameter The diameter of a chain is the maximal discrete geodesic distance (see Section 5.2) between the 0-dimensional faces of the cells in the chain. If K is embedded in a metric space we can also consider the maximal distance between the 0-dimensional faces.

Radius The radius of a chain is the radius of the smallest geodesic ball containing the chain. Again, if K is embedded in a metric space we can consider the radius of the smallest ball containing the chain.

Chen and Freedman [18] proved that finding a cycle minimizing the volume is an NP-hard problem, even if we look for an approximation. Considering the diameter is also an NP-hard problem, but we can compute a 2-approximation considering the radius, for which there is a polynomial time algorithm. Sadly, they also showed that considering the diameter or the radius does not always provide visually pleasant generators as they can wiggle.

Our heuristic provides a cycle and a cocycle associated to each thickness-breadth pair of a discrete object. The intuition is that a minimal homology generator must be around a breadth ball, while a minimal cohomology generator must traverse a thickness ball. However, given the complexity results exposed previously, we cannot expect to prove that these generators are optimal for the volume.
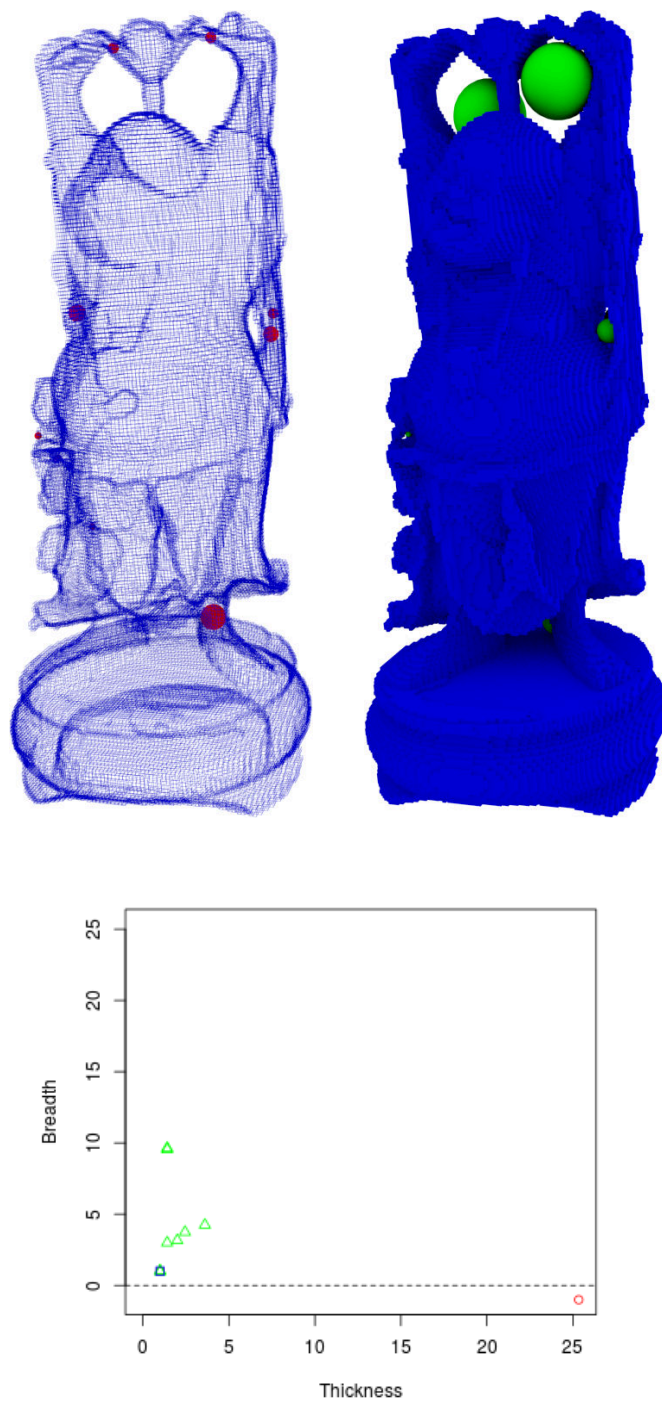
FIGURE 5.6: Buddha: thickness balls (in red), breadth balls
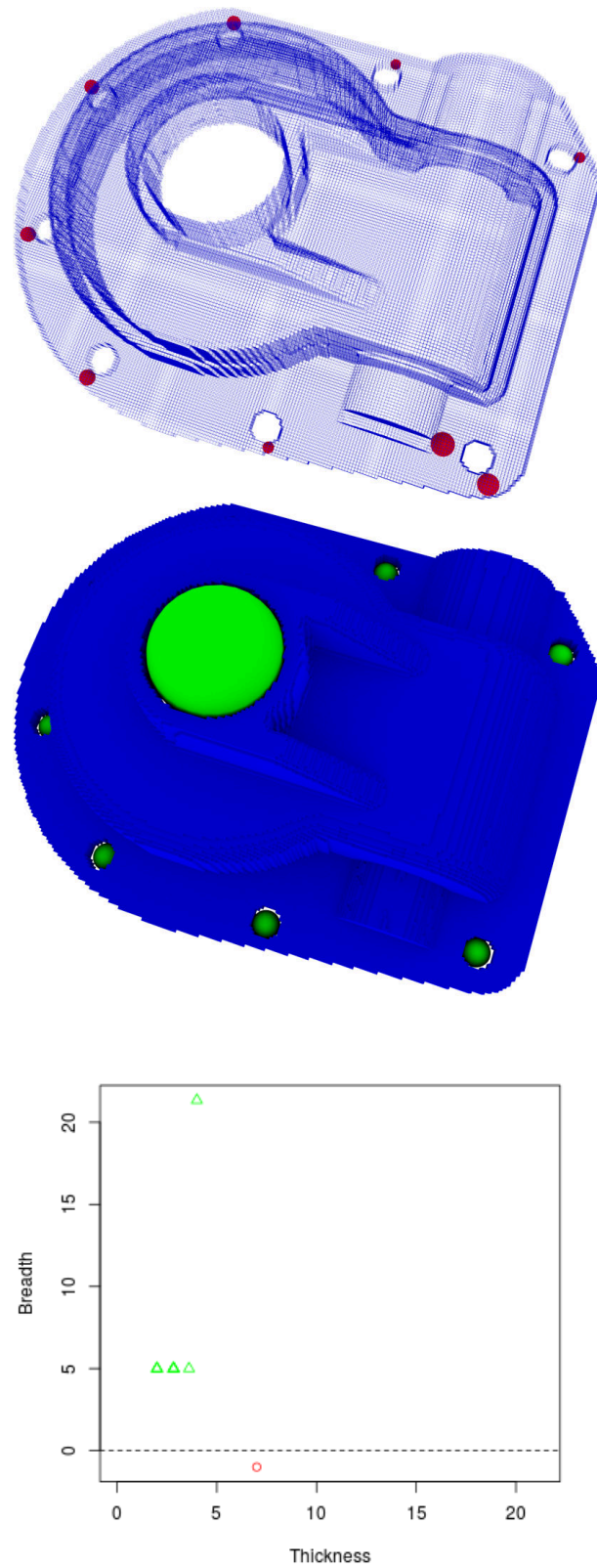(green) and thickness-breadth diagram.

FIGURE 5.7: Casting: there are two types of 1-holes according to the breadth. We also observe that the narrow (less broad) holes do not have the same thickness, as they are not equally close to the border.

FIGURE 5.8: Dancing: the breadth balls clearly localize the
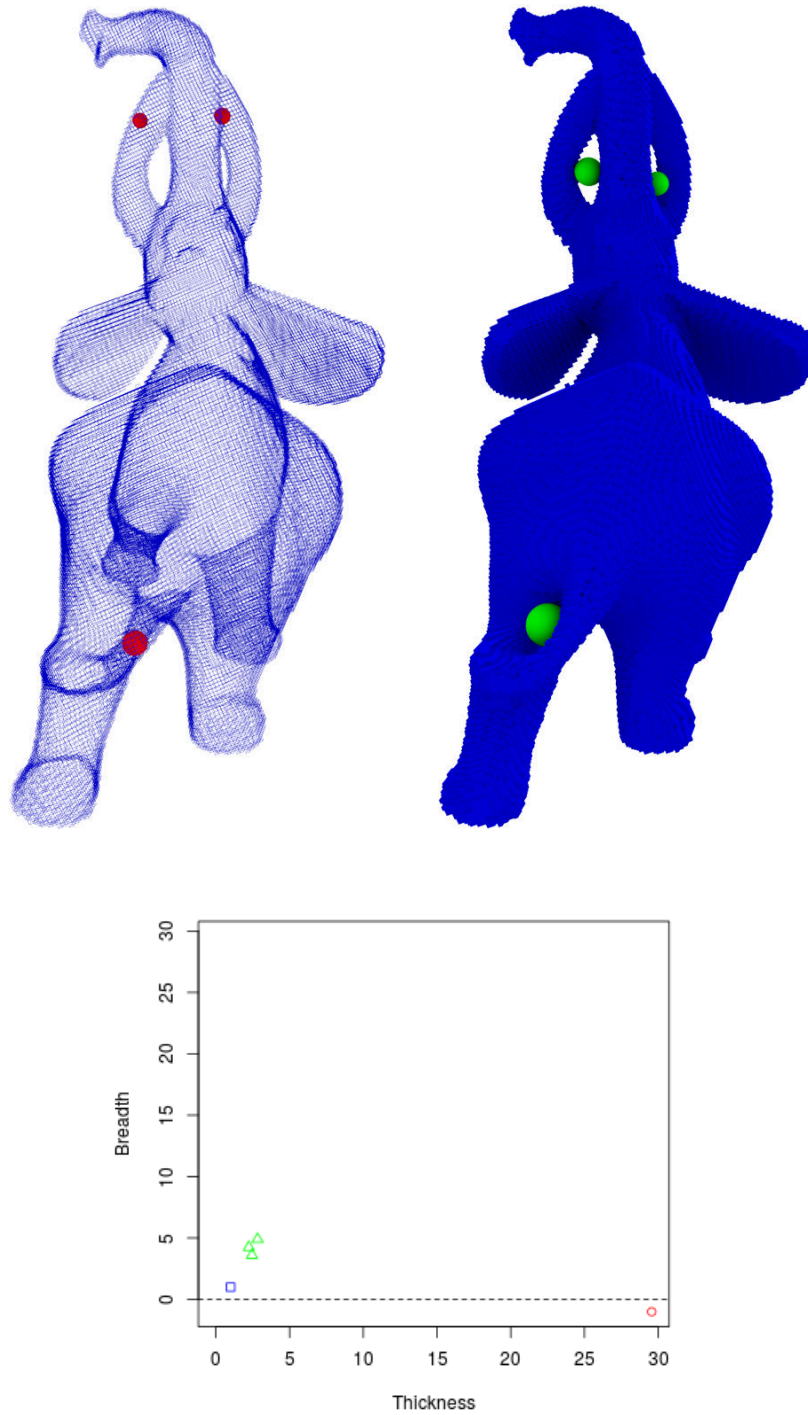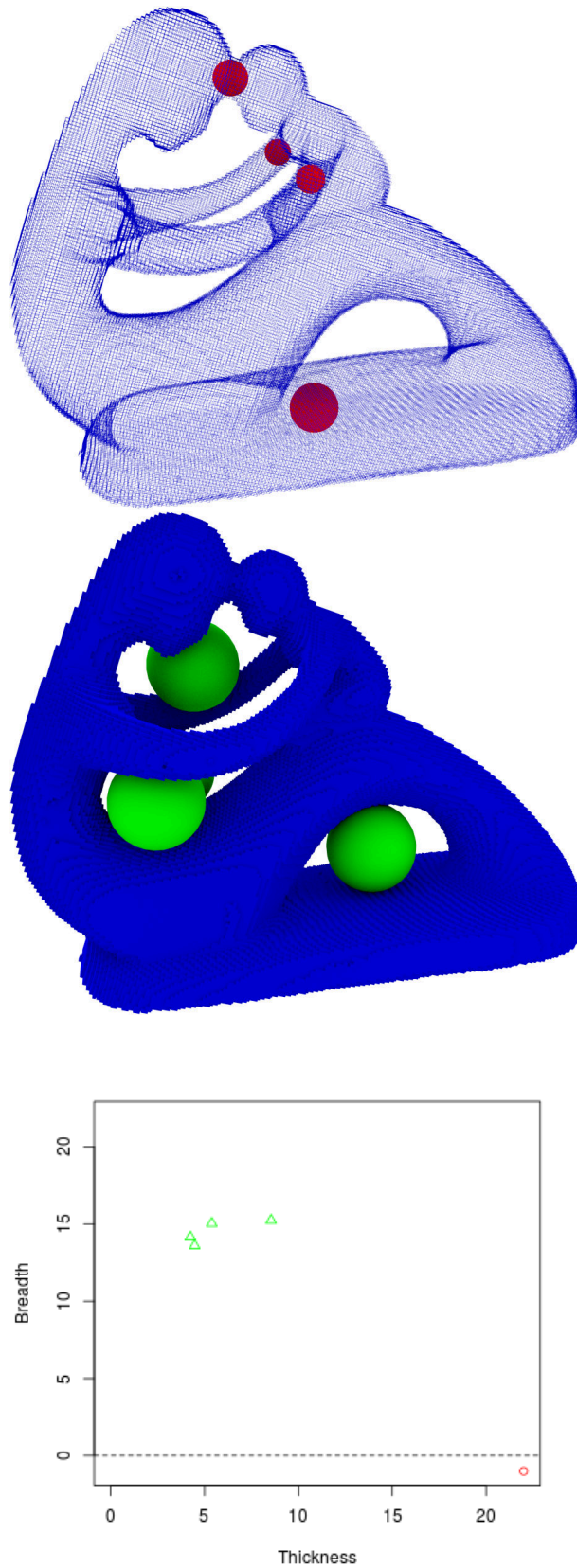1-holes of the object.

FIGURE 5.9: Elephant

FIGURE 5.10: Fertility: observe that there are five, and not four 1-holes. The thickness balls in the arms and between the heads clearly show the most fragile parts of these holes. The thickness ball in the base is less evident, as the hole formed by the legs does not have a tubular shape.
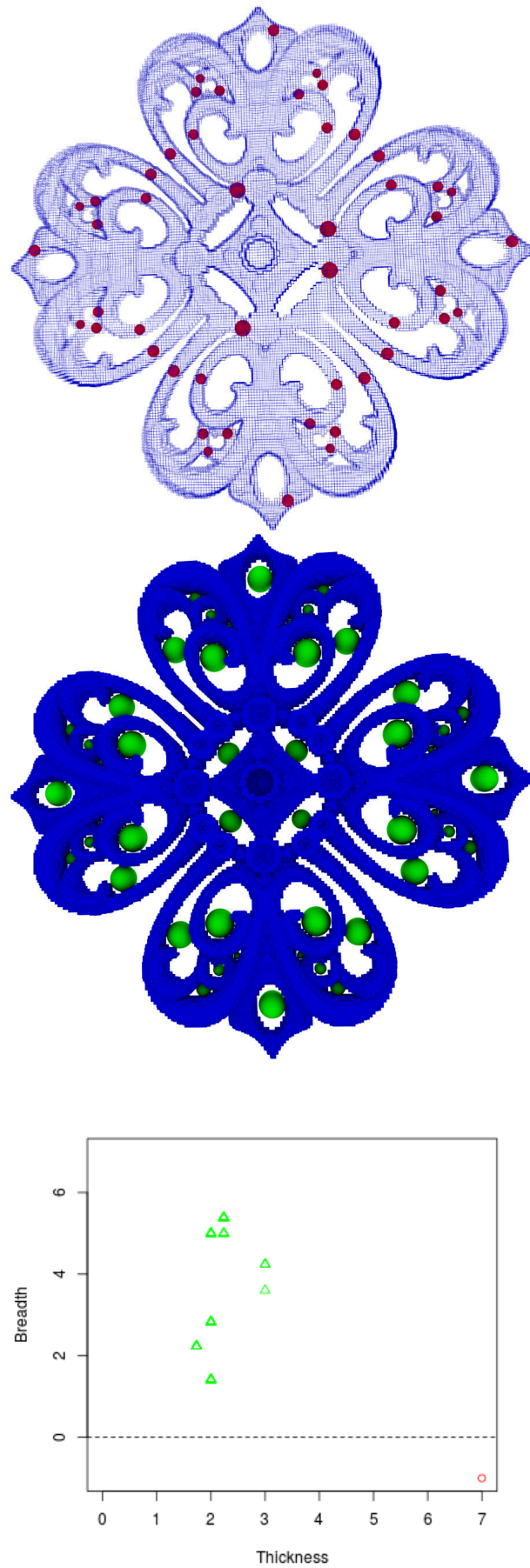
FIGURE 5.11: Filigree: the position of the thickness and the breadth balls respects quite well the symmetry of this object.
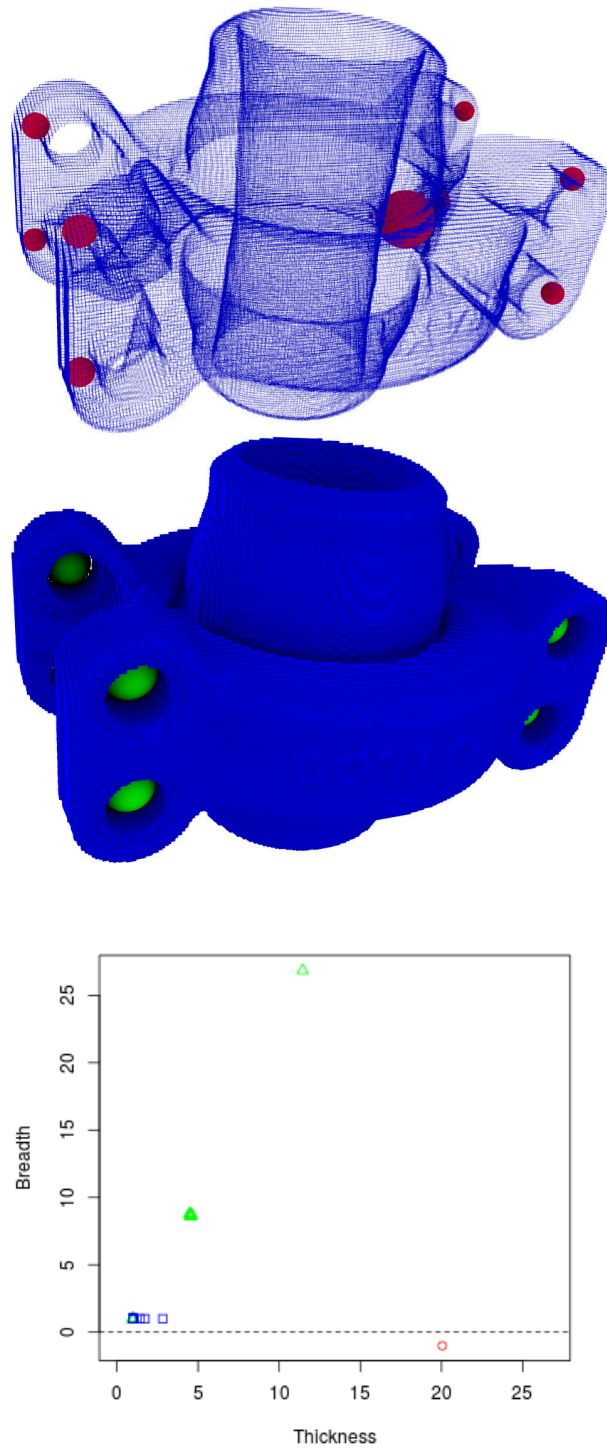
FIGURE 5.12: Grayloc: there are eight similar 1-holes and one bigger in both thickness and breadth in the middle. Several 2-holes were produced during the voxelization of the mesh.
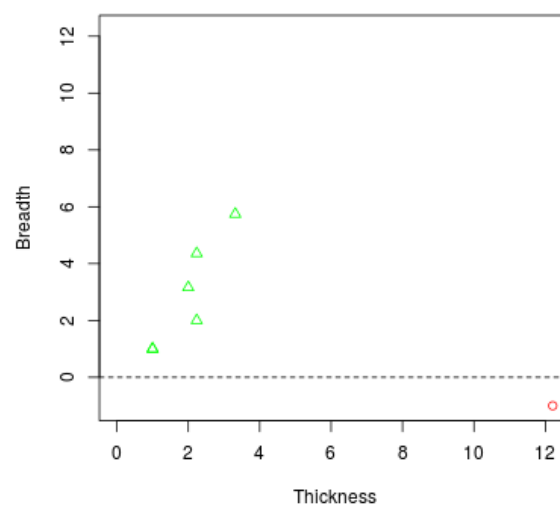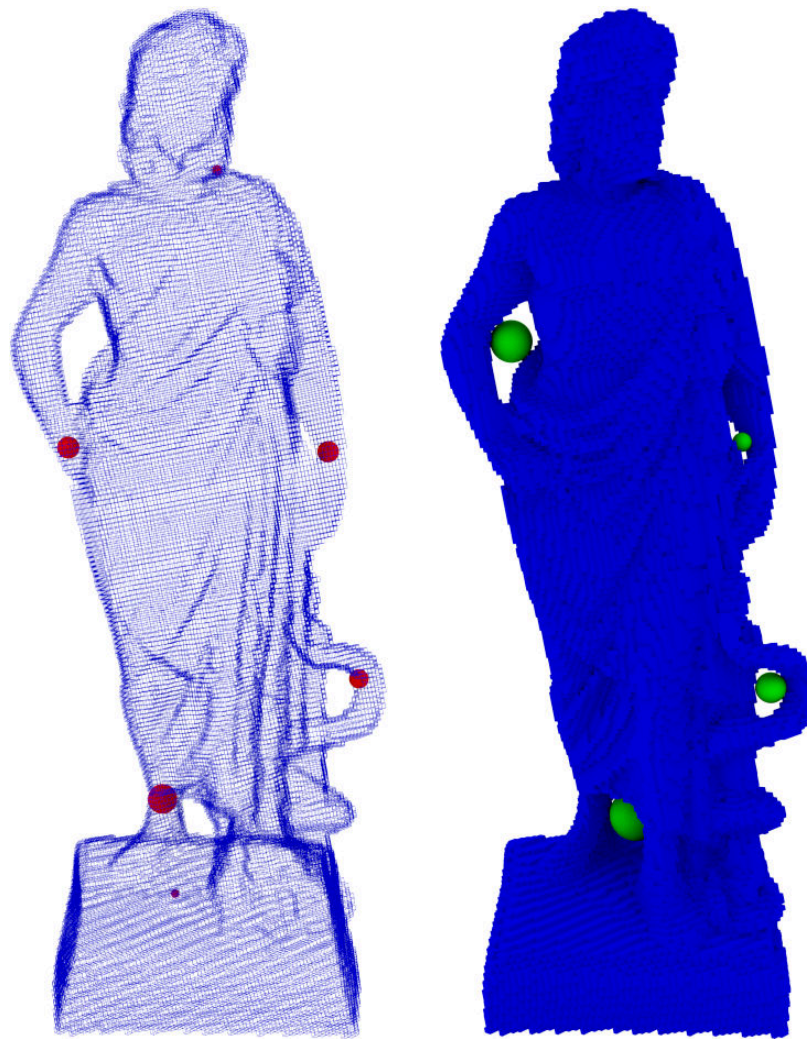
FIGURE 5.13: Greek: observe the two small holes in the base
and in the beard, which can be neglected as noise.
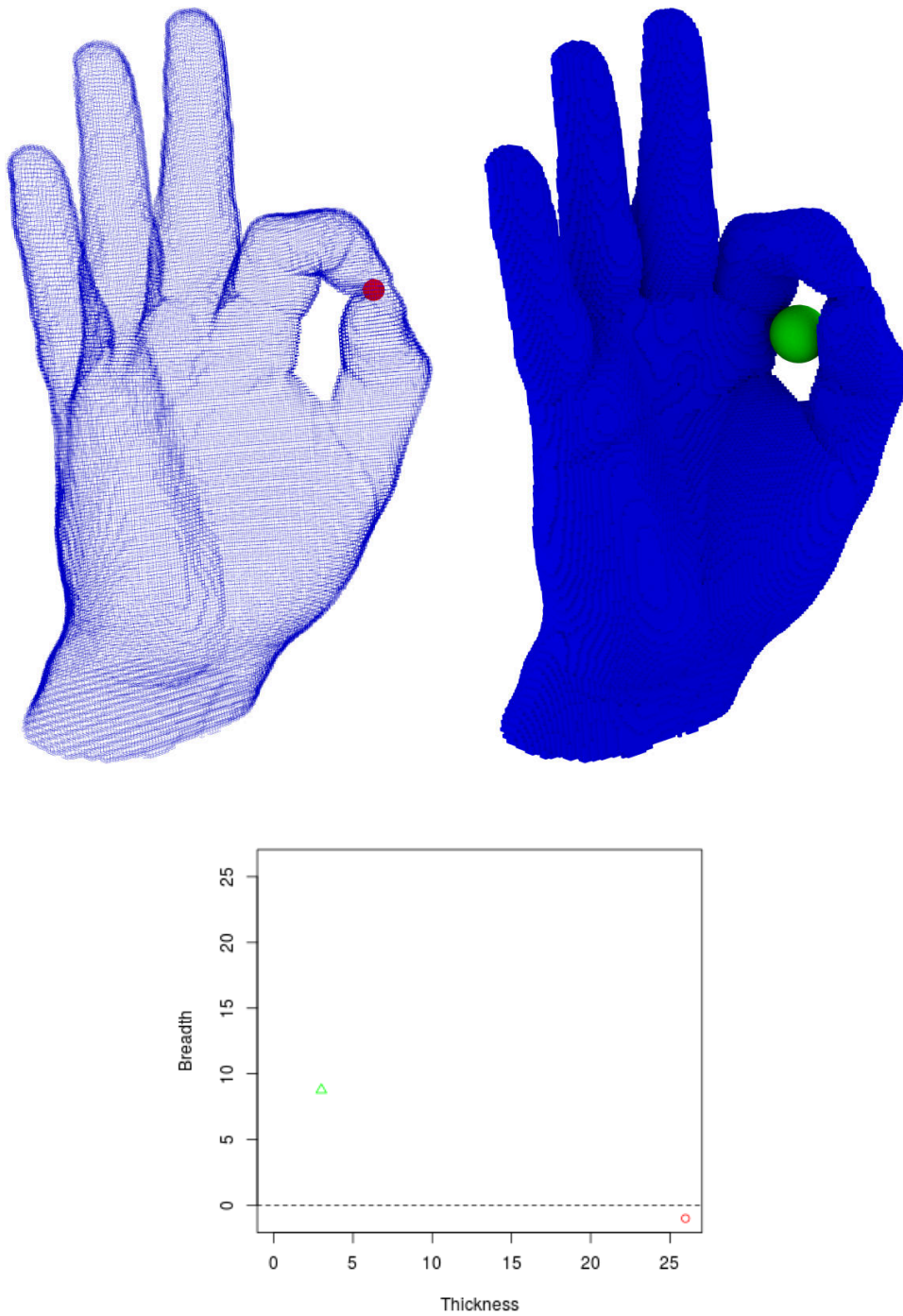
FIGURE 5.14: Hand: the thickness and the breadth balls are
placed in the most intuitive positions.
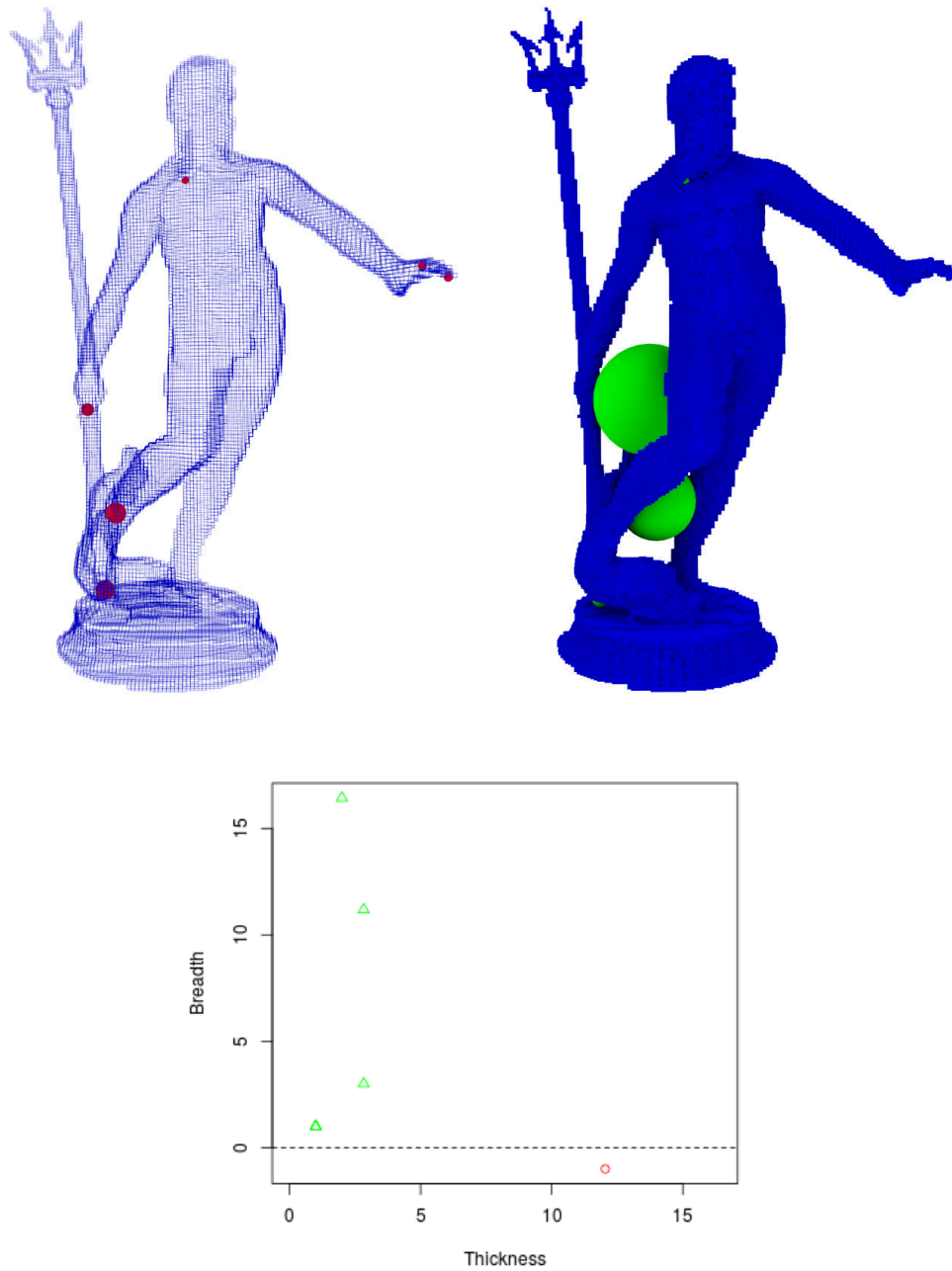
FIGURE 5.15: Neptune: there are three notable holes in this object. The rest, located in the beard and in the hand, can be considered as noise.
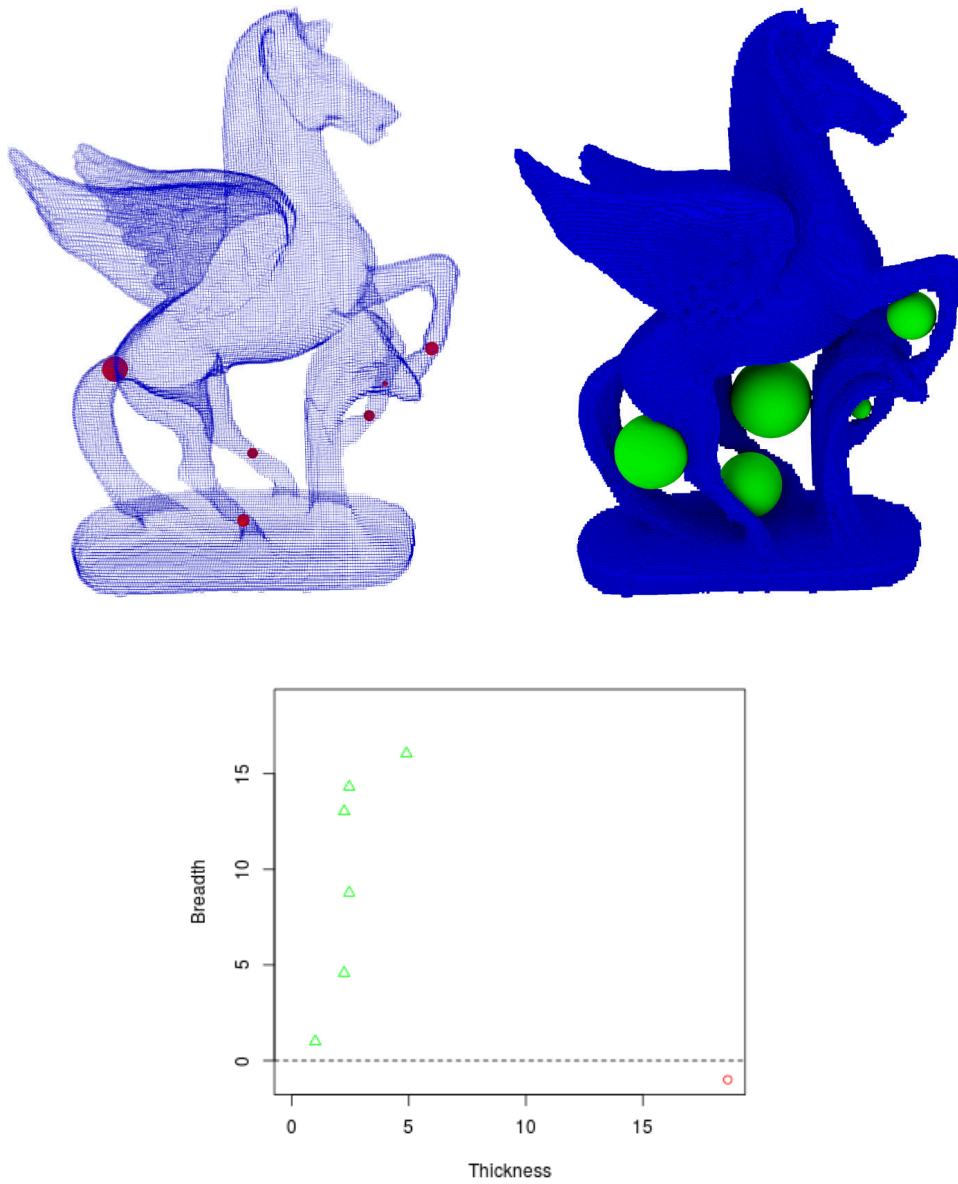
FIGURE 5.16: Pegasus: there are five significant holes and a small one near the right fron paw (see its thickness ball).

5.4.1 Homology generators

Let O be a discrete object for which we have computed the thickness and the breadth of its holes. Let (σ, τ) be a pair of cubes associated to a thickness-breadth pair and let $\rho_- = (h, f, g)$ be the reduction associated to the persistent homology computation performed for obtaining the measures of O before adding the cells with positive signed distance transform. Note that ρ_- is a reduction for $K[O]$. Actually, we only need the chain $f^*(\sigma)$. Algorithm 6 provides a cycle associated to the homology generator $g(\sigma)$.

Algorithm 6: Homology generator

Input: A discrete object O, its reduction ρ_-, a pair of cubes (σ, τ) associated to a thickness-breadth pair
Output: A cycle x such that $\langle f(x), \sigma \rangle \neq 0$
$\vec{p} \leftarrow$ some voxel associated to τ;
$F \leftarrow$ filtration induced by $d_{\vec{p}} : O \rightarrow \mathbb{R}$;
Compute the persistent homology on F. When a cycle x is found, check if $\langle f(x), \sigma \rangle \neq 0$. If true, return this cycle;

Let $\vec{p}$ be a voxel associated to τ: a 3-dimensional coface if we considered the primal associated cubical complex or a 0-dimensional face if we considered the dual associated cubical complex. It is not unique, so we choose one arbitrarily. Let $d_{\vec{p}} : O \rightarrow \mathbb{R}$ be the function that maps every voxel of O to its distance to $\vec{p}$. Consider the filtration F associated to this function (as we did for the signed distance transform in Section 5.2). When we compute the persistent homology of this filtration we find a boundary associated to a negative cell or a cycle x associated to a positive cell. Among these cycles, we take the first one for which $\langle f(x), \sigma \rangle \neq 0$.

The condition $\langle f(x), \sigma \rangle \neq 0$ means that if we write the class $[x]$ in terms of the homology base $g(C)$ associated to ρ_-, the coefficient of the generator $g(\sigma)$ is not zero. This is necessary to capture the hole we want. Figure 5.17 shows a binary image with two holes and its breadth balls (in blue). While searching for a small homology generator for the broader hole, Algorithm 6 finds the small cycle x in the middle before the bigger cycle y on the right. The condition $\langle f(x), \sigma \rangle \neq 0$ avoids to return x, which does not correspond to the hole we chose.
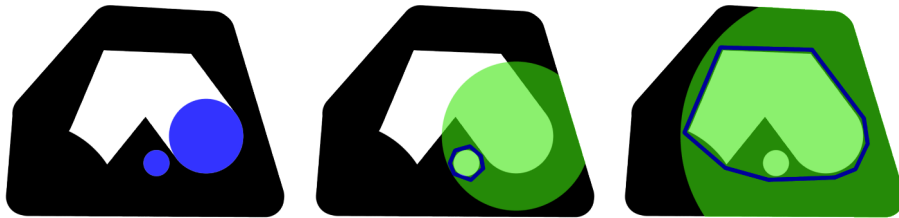

FIGURE 5.17: Left: a binary image with its breath balls in blue. Center and right: two cycles found during the computation of Algorithm 6.

It seems that we could obtain an approximation of a minimal homology base by computing the chains associated to each thickness-breadth pair, but this is unclear since we could obtain a set which is not linearly independent.

5.4.2   Cohomology generators

Representing holes by the generators of the cohomology groups seems an unused approach. This is possibly due to the fact that they are not manifold-like (as homology generators) since they are cocycles instead of cycles. Nevertheless, they can be visually interesting if we do not only display the cells in the cohomology generators but also all its cofaces. Note that when we display the homology generators we also display their faces, which is the dual statement of the previous sentence. We are interested in computing small cohomology generators since they display holes as thickness balls do.

The heuristic for obtaining a cochain associated to a thickness-breadth pair is very similar. Algorithm 7 provides a cocycle associated to the cohomology generator $f^*(\sigma)$.

Algorithm 7: Cohomology generator

Input: A discrete object O, its reduction ρ_-, a pair (σ, τ) of cubes
       associated to a thickness-breadth pair
Output: A cocycle x such that $\langle g^*(x), \sigma \rangle \neq 0$
$\vec{p} \leftarrow$ some voxel associated to σ;
$F \leftarrow$ cofiltration induced by $d_{\vec{p}} : O \rightarrow \mathbb{R}$;
Compute the persistent cohomology on F. When a cocycle x is
  found, check if $\langle g^*(x), \sigma \rangle \neq 0$. If true, return this cocycle;

The input is the same, unless we only need the cochain $g(\sigma)$. The main difference is that we do not consider a filtration but a *cofiltration*. In Algorithm 6 we compute the persistent homology of the filtration induced by $d_{\vec{p}}$ because we want to consider all the cycles that appear in this filtration. Using the algorithm for persistent homology avoids to consider cycles that are boundaries, but we actually do not need the persistence intervals of this filtration. In this context we want to find cocycles, so we take a dual approach.

Let K be a cubical complex. $L \subset K$ is a *sub-cocomplex* if for each cube of L, all its cofaces in K are also included in L. A *cofiltration* of K is a sequence of nested sub-cocomplexes $\emptyset = K_0 \subset \cdots \subset K_m = K$. Thus, a cube enters the cofiltration before its faces. We can adapt the persistent homology algorithm by considering coboundaries instead of boundaries, which gives a coboundary or a cocycle for each cube. As in Algorithm 6, we take the first cocycle such that $\langle g^*(x), \sigma \rangle \neq 0$.

Let us present a few results of these two algorithms. Figure 5.18 depicts a thickened wire-frame cube. We can appreciate its homology generators (in green) and cohomology generators (in red) of dimension 1. Observe that two of its cohomology generators are too close to be distinguished. A double torus is shown in Figure 5.19, whose generators are tight. The object in Figure 5.20 does not have four holes, but three. They are located by the generators. Let us point out that the generators obtained for these three objects are linearly independent and hence they conform a (co)homology base. However, as mentioned above, this is not guaranteed for our two algorithms.
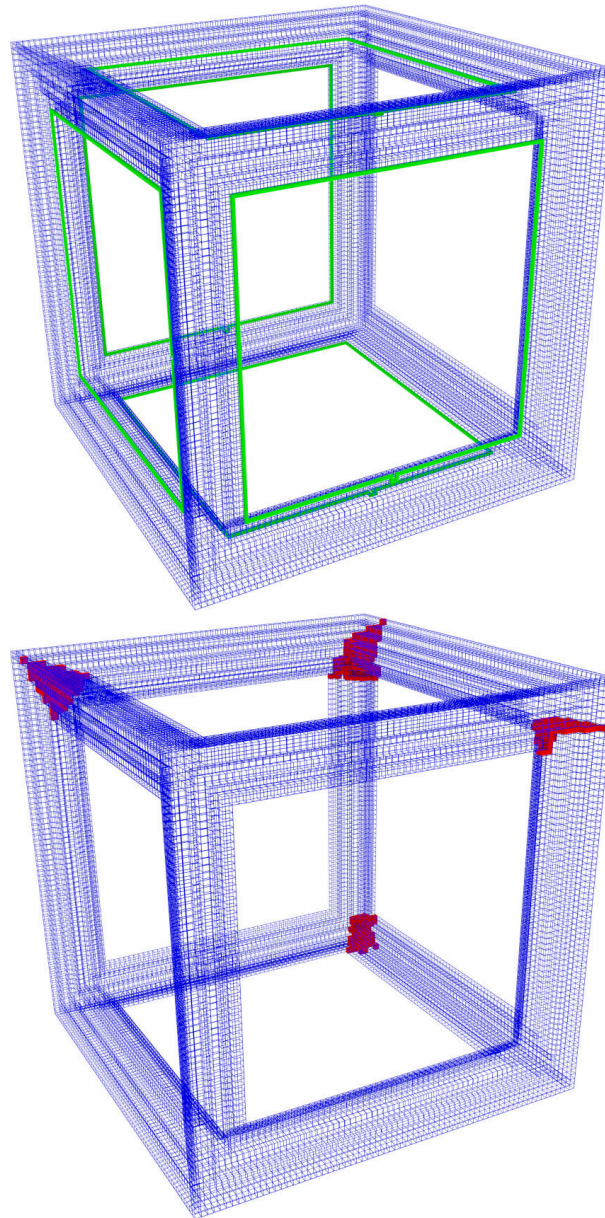
FIGURE 5.18: An object with its homology generators (top, in green) and homology generators (bottom, in red).

5.5 Opening or closing holes

If $[x]$ is a non-trivial homology class, it means that x is a cycle, but it is not a boundary. Thus, if we add a "coface" so x becomes a boundary, it will be a trivial class. This means that a homology generators is the boundary of a chain that is missing. Hence, in order to remove a hole, we can add such chain.

Let us see now why this reasoning does not work for cohomology. A non-trivial cohomology class $[x]$ is a cocycle that it is not a coboundary. Thus, if we want it to become a coboundary, we must add a "face" whose coboundary is x. However, this does not make sense since a complex already contains all its faces.

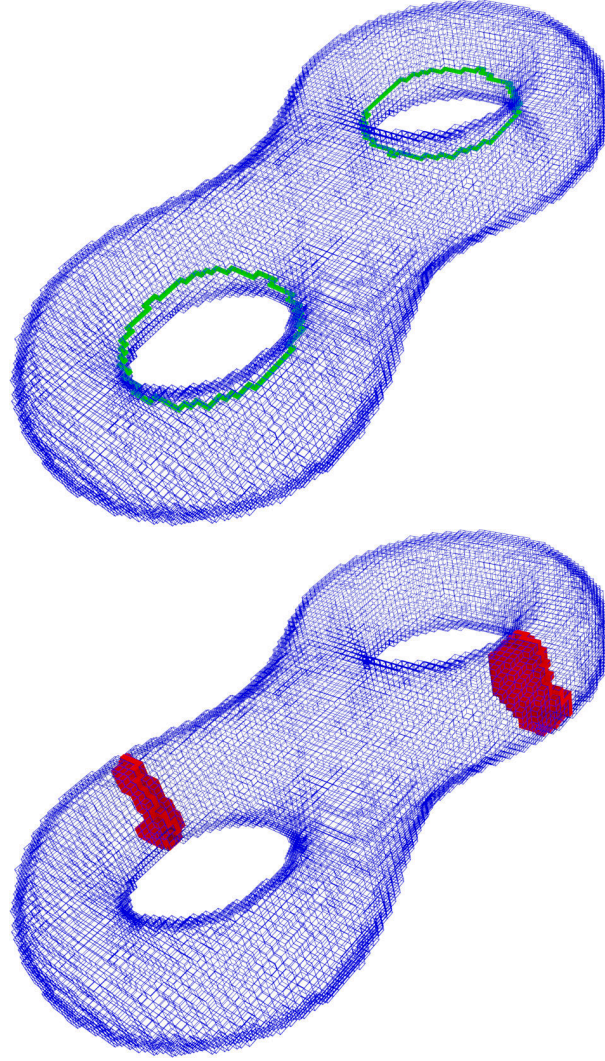We define now what is to open and to close a homology class. Given

FIGURE 5.19: A double torus. Model from the AimAtShape
repository.

two cubical complexes $K \subset L$, the inclusion map $\iota : K \to L$ induces a chain
map $\iota_{\#} : \mathsf{C}(K) \to \mathsf{C}(L)$ between their corresponding chain groups. Hence,
$\iota_{\#}$ induces a homomorphism $\iota_{*} : H(K) \to H(L)$ between their homology
groups, that is, $\iota_{*}([x])$ is the homology class of the chain x in $H(L)$.

Definition 5.2. *Let K be a cubical complex, x a cycle and S a set of cubes. Then,*

- *S* opens *the chain x if $K - S$ is a cubical complex, $\iota_{*} : H(K - S) \to H(K)$
  is injective and $[x] \notin \mathrm{im}(\iota_{*})$. We say that S is an* opening set *for x.*

- *S* closes *the chain x if $K \cup S$ is a cubical complex, $\iota_{*} : H(K) \to H(K \cup S)$
  is surjective and $[x] \in \ker(\iota_{*})$. We say that S is a* closing set *for x.*

Let $[x]$ be a non-trivial homology class of $H(K)$. The injectivity of ι_{*}
means that $K - S$ does not contain *new holes*, and $[x] \notin \mathrm{im}(\iota_{*})$ implies that
we have removed *the hole* $[x]$. A similar interpretation follows from the def-
inition of closing a chain. Note that, by closing a q-hole, several other holes
may disappear. For instance, closing a 1-hole can merge two connected
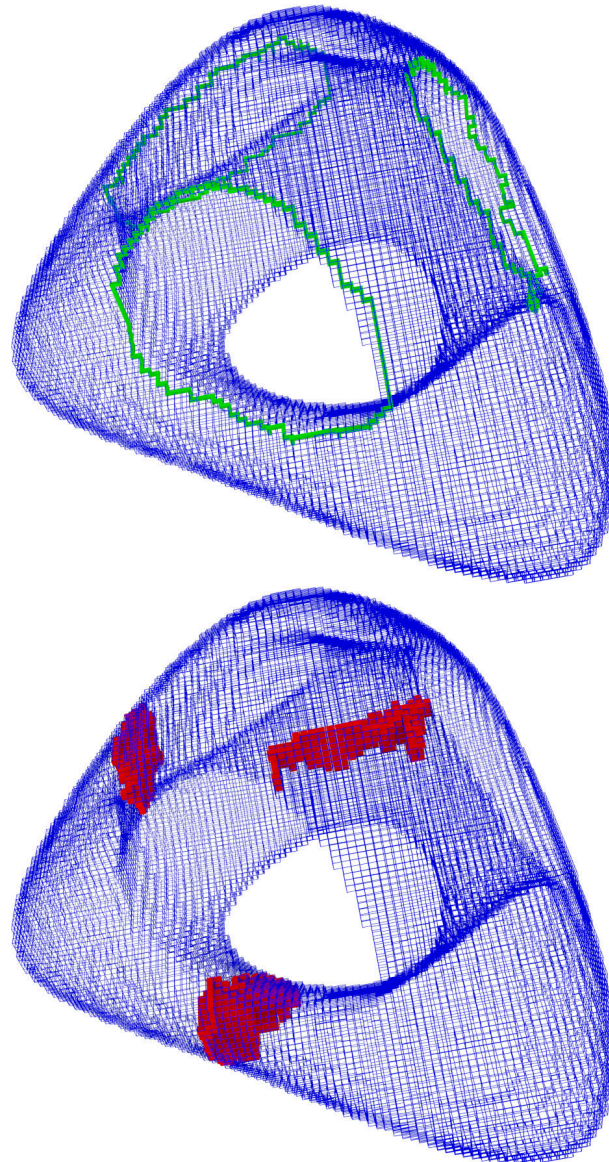components (think of two chained circles) and closing the 2-hole of a torus

FIGURE 5.20: An object with three 1-holes. Model from the
AimAtShape repository.

removes one of its 1-holes. Also, opening a 0-hole removes all the higher-
dimensional holes in the connected component and opening a 1-hole in a
torus removes its 2-hole.

The terms *open* and *close* are inspired by the work of Aktouf et al. [1].
They introduced the concept of *topological hull* for a three-dimensional dis-
crete object O: it is a minimal (for inclusion) superset $TH \supset O$ which has
no holes or cavities in the sense of digital topology [79]. Later, Janaszewski
et al. [71] made the distinction between closing a hole (adding a minimal
set of voxels to remove the hole) and filling a hole (the set is not minimal
since it tries to fit the local geometry of the object). Observe, however, that
these notions are defined for discrete homotopy and not for homology.

These definitions evoke the following problem: what is the minimal
opening/closing set (under any geometric criterion) for a given chain, or
more generally, what is the minimal set that opens/closes all the holes of

a cubical complex? This seems to be a hard problem. Intuitively, the intersection of a minimal closing (resp. opening) set with the object gives a small homology (resp. cohomology) generator. Thus, we can even expect this problem to be NP. Consequently, as in Section 5.4, we only provide algorithms that seem to work well, without any proof of optimality.

5.5.1 Opening a hole

The previous section, where several cohomology generators were shown, should have suggested an idea: removing a cohomology generator erases a hole. We cannot prove this fact because it is not true in general. Figure 5.21 illustrates a counter-example. It shows a small simplicial complex with one 1-hole. It has a cohomology generator involving three 1-simplices (marked in red). After removing them, and their cofaces, there is still a 1-hole in the complex. Note that we could have removed other cohomology generators which do open the hole.
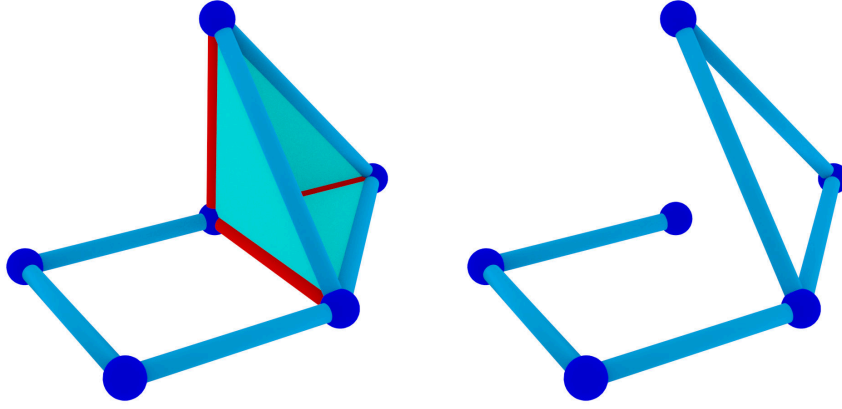

FIGURE 5.21: Left: a simplicial complex K with a cohomology generator x (in red). Right: after removing x (and its cofaces) from K, there is still a 1-hole.

Let K be a cubical complex endowed with a perfect HDVF $X = (P_X, S_X)$ whose critical cells are $C_X = \{\gamma_1, \ldots, \gamma_r\}$. If we have a subcomplex $L \subset K$ endowed with a perfect HDVF $Y = (P_Y, S_Y)$ such that $P_Y \subset P_X$, $S_Y \subset S_X$ and $C_Y \subset C_X - \{\gamma_1\}$, then we have opened the homology generator $g_X(\gamma_1)$ associated to the critical cell γ_1 in K:

- L is a cubical complex

- Since both HDVFs are perfect, $[g_X(\gamma_1)] = f_X g_X(\gamma_1) = \gamma_1$, which does not belong to C_Y, so $[g_X(\gamma_1)] \notin \text{im}(\iota_*)$

- $C_Y \subset C_X$ implies that ι_* is injective.

We can obtain such cubical complex by removing a critical cell and pairs of cells until we obtain a cubical complex. Algorithm 8 describes this procedure.

At the end we obtain a perfect HDVF for the subcomplex $K - 0$ which does not contain γ. Thus, $K - 0$ has at least one hole less, and no holes are created. Before proving it, we need a lemma that ensures that we can find the cells τ and σ at lines 6 and 9 respectively.

Algorithm 8: Hole opening

Input: A CW complex K endowed with a perfect HDVF $X = (P, S)$, a critical cell γ

Output: An opening set $\mathtt{O}$ for the homology generator $g(\gamma)$

1 Q.push($d^*(\gamma)$);
2 $\mathtt{O} \leftarrow \{\gamma\}$;
3 **while** Q *not empty* **do**
4      $a \leftarrow Q$.pop();
5   **if** $a \in P$ **then**
6   choose $\tau > a$ st. $\langle h(a), \tau \rangle \neq 0$; $X \leftarrow \mathtt{R}(X, a, \tau)$;
7          $\mathtt{O} \leftarrow \mathtt{O} \cup \{a\}$; Q.push($d^*(a)$);
8   **else if** $a \in S$ **then**
9   choose $\sigma < a$ st. $\langle h(\sigma), a \rangle \neq 0$; $X \leftarrow \mathtt{R}(X, \sigma, a)$;
10         $\mathtt{O} \leftarrow \mathtt{O} \cup \{\sigma\}$; Q.push($d^*(\sigma)$);
11   **else if** $a \in C$ **then**
12         $\mathtt{O} \leftarrow \mathtt{O} \cup \{a\}$; Q.push($d^*(a)$);

13 **return** $\mathtt{O}$

Lemma 5.3. *Let K be a CW complex endowed with a perfect HDVF $X = (P, S)$. Then,*

- *If $\sigma \in P$ then there exists $\tau \in S$, $\tau > \sigma$ such that $\langle h(\sigma), \tau \rangle \neq 0$.*

- *If $\tau \in S$ then there exists $\sigma \in P$, $\sigma < \tau$ such that $\langle h(\sigma), \tau \rangle \neq 0$.*

Proof. Let $\sigma \in K$ be a primary cell. Since $d(S)_{|P} \cdot H = I$, then $d(S)_{|\sigma} \cdot h(\sigma)_{|S} = 1$. Thus, there exist some $\tau \in S$ such that

$$\langle d(\tau), \sigma \rangle \neq 0 \quad \text{and} \quad \langle h(\sigma), \tau \rangle \neq 0$$

The second statement follows from $H \cdot d(S)_{|P} = I$. $\qquad\qquad\square$

Next, we need yet another lemma.

Lemma 5.4. *Let $K \subset L$ be two CW complexes endowed with two HDVFs $X \subset Y$ respectively. If Y is perfect then X is perfect too.*

Proof. The boundary matrix of L is of the form

$$D = \left[\begin{array}{c|c} D_1 & \cdot \\ \hline 0 & D_2 \end{array} \right]$$

where $D_1 = d(K)_{|K}$ and $D_2 = d(L - K)_{|L-K}$. Thus,

$$d(S)_P \cdot H = \left[\begin{array}{c|c} u & v \\ \hline 0 & w \end{array} \right] \cdot \left[\begin{array}{c|c} x & y \\ \hline z & t \end{array} \right] = \left[\begin{array}{c|c} I & 0 \\ \hline 0 & I \end{array} \right]$$

and

$$H \cdot d(S)_P = \left[\begin{array}{c|c} x & y \\ \hline z & t \end{array} \right] \cdot \left[\begin{array}{c|c} u & v \\ \hline 0 & w \end{array} \right] \cdot = \left[\begin{array}{c|c} I & 0 \\ \hline 0 & I \end{array} \right]$$

Note that u is invertible (since X is a HDVF). Hence,

$$xu + y0 = I \Rightarrow x = u^{-1}$$
$$zu + t0 = 0 \Rightarrow z = 0$$

Therefore,

$$H = \left[\begin{array}{c|c} u^{-1} & \cdot \\ \hline 0 & \cdot \end{array}\right]$$

Consequently, as Y is perfect,

$$d(C)_{|C} = 0 \Rightarrow \left[\begin{array}{c|c} A & \cdot \\ \hline 0 & \cdot \end{array}\right] = \left[\begin{array}{c|c} B & \cdot \\ \hline 0 & \cdot \end{array}\right] \cdot \left[\begin{array}{c|c} u^{-1} & \cdot \\ \hline 0 & \cdot \end{array}\right] \cdot \left[\begin{array}{c|c} D & \cdot \\ \hline 0 & \cdot \end{array}\right]$$

so $A = Bu^{-1}D$ and thus X is perfect. $\qquad\square$

We can now prove the correctness of Algorithm 8.

Proposition 5.5. *Algorithm 8 returns an opening set for the chain $g(\gamma)$.*

Proof. Let K be a CW complex endowed with a perfect HDVF X. Let γ be a critical cell. We have to prove that O opens the chain $g(\gamma)$.

First, let us prove that $K - $ O is a cubical complex. This is equivalent to prove that for any cube in O, its cofaces (in K) are also included in O. Indeed, whenever we add a cell to O, we add its cofaces to the queue Q. Let a be one of those cofaces. If a is primary or critical, it will be added to O when it is taken from Q. If it is secondary, it will become critical and will be added again to Q, so it will eventually be added to O.

We denote by X' the resulting HDVF at the end of the algorithm. Let us prove now that X' on $K - $ O is perfect. By construction, X' is a HDVF for $K - $ O and, since X is perfect for K, the result follows from Lemma 5.4.

As γ does not belong to $K - $ O and there are no new critical cells, it follows that O opens the chain $g(\gamma)$. $\qquad\square$

Thus, given a discrete object O and a thickness-breadth pair (σ, τ), we can remove the homology generator associated to σ using Algorithm 8 with the HDVF on $K[O]$ obtained while computing the measures. However, in practice, it seems that it suffices to remove the cohomology generator associated to σ in the HDVF and its cofaces. We have performed several experiments on objects with complex geometry and we have never found an example as Figure 5.21. Note that, in that example, there are other cohomology generators which do open the hole. We are not able to prove why this works, or in what cases it does.

5.5.2 Closing a hole

Closing a hole has also an intuitive answer which is false. Let x be a homology generator in a complex K. If we add a set of cells such that its boundary is x, then it becomes a trivial class in the homology group. However, this does not necessarily close the hole. Let K be the boundary of a Möbius strip, which is homotopy equivalent to S^1, and x the chain with all its 1-cells. By adding the interior of the strip, x becomes a boundary but there is still a 1-hole. This is illustrated in Figure 5.22.
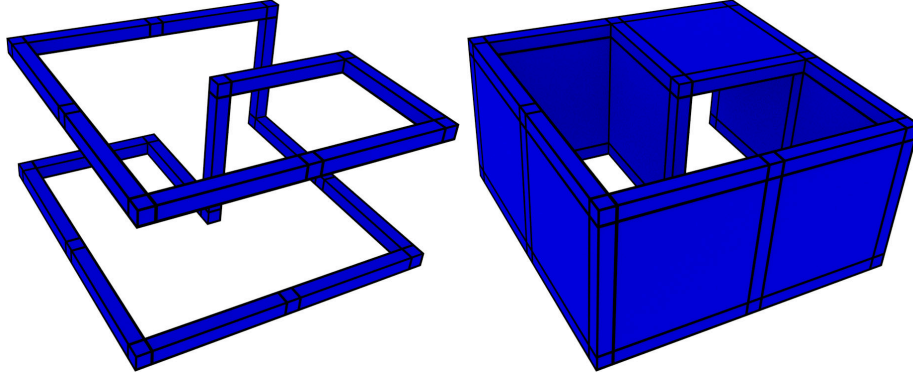
FIGURE 5.22: Left: boundary of a Möbius strip. Right: a
Möbius strip.

Unfortunately, we have not found any algorithm like Algorithm 8 that closes a hole. Thus, we present a heuristic for closing holes, even though we know that it does not work in all the cases.

Suppose that (σ, τ) are the cells associated to a thickness-breadth pair of a discrete object O and that the persistent homology computation has been performed in the context of the HDVF (see Algorithm 2). This means that the homology generator x ($x = g_-(\sigma)$ for the reduction ρ_-) associated to σ became a boundary when τ entered the complex. The advantage of working with HDVFs is that we know that actually x is the boundary of $h(x) = h(\sigma)$ (for the reduction ρ associated to the last HDVF). Intuitively, if we add to $K[O]$ the cubes in $h(\sigma)$ together with its faces, the resulting cubical complex should contain (at least) one hole less since $[x]$ vanishes.

Algorithm 9: Hole closing

Input: A discrete object O, its reduction ρ, a pair (σ, τ) of cubes
associated to a thickness-breadth pair
Output: A set of cubes C such that $K[O] \cup \text{C}$ is a cubical complex and
$\iota([x]) = 0$, where $x = g_-(\sigma)$ is the homology generator
associated to σ in $K[O]$
C $\leftarrow h(\sigma)$ together with its faces;
return C

It is clear that $K[O] \cup \text{C}$ is a cubical complex and $\iota([x]) = 0$ since it is the boundary of $h(\sigma)$. On the other hand, ι is not surjective in general.

It is quite surprising that we have not found an algorithm for closing holes, given the strong relation between closing and opening holes. A more elaborate scheme for closing holes is to execute a version of Algorithm 8 on the complement of the object and append the removed part to the complex, but we cannot prove that this closes the given hole.

As in the previous section, we propose a few examples for validating these results. For each object, we display the opening (in red) and closing (in green) sets for all its critical 1-cells together. The opening sets have not been obtained through Algorithm 8, but by removing the cohomology generators. All the objects in figures 5.23, 5.24 and 5.25 present visually pleasant opening and closing sets. On the other hand, Figure 5.26 illustrates an

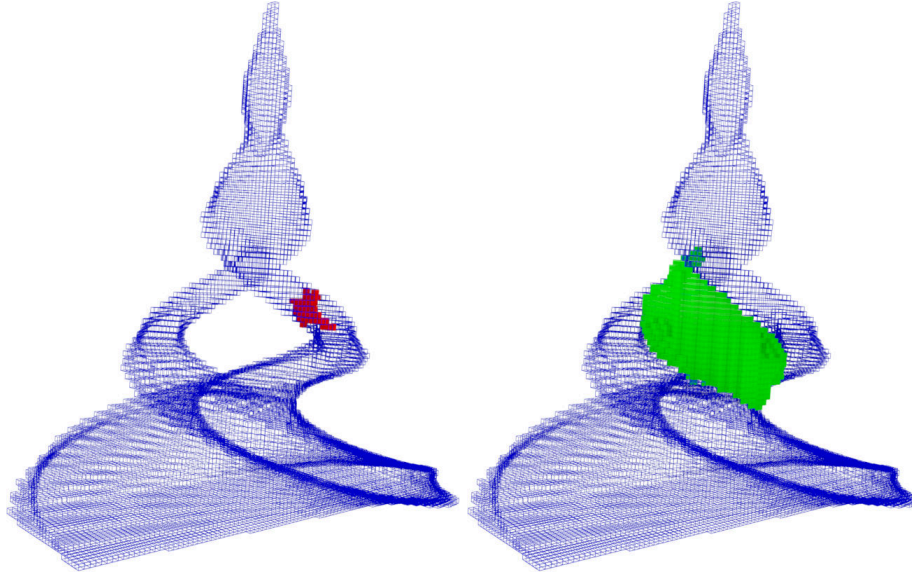object for which Algorithm 9 does not close the holes.

FIGURE 5.23: An object with its opening set (left, in red) and
its closing set (right, in green). Model from the AimAtShape
repository.

5.5.3 We want voxels, not cubes

We claimed in the introduction of this chapter that we wanted to close or
open holes in a discrete object by adding or removing voxels. However,
the algorithms proposed provide sets of cubes that close or open the holes
of the cubical complex associated to a discrete object. We can provide an
intuitive answer to this problem, but we cannot prove that they close or
open the holes without creating new holes.

Let σ be a cube, we denote by $\hat{\sigma}$ the set of voxels incident to σ: for the
primal associated cubical complex this is the set of the voxels associated to
the 3-dimensional cofaces of σ; for the dual associated cubical complex this
is the set of the voxels associated to the 0-dimensional faces of σ.

In the following we discuss the four possible cases.

Closing a hole given the primal associated cubical complex   Given a
closing set C for a homology generator x, we can consider a set of voxels S
such that its primal associated cubical complex $K_p[S]$ contains all the cubes
in C. This is actually a set-covering problem [20, § 35.3], since we look for a
minimal set of voxels *covering* all the cubes in C.

A trivial solution is to consider the set

$$\hat{\mathtt{C}} := \bigcup_{\sigma \in \mathtt{C}} \hat{\sigma}$$

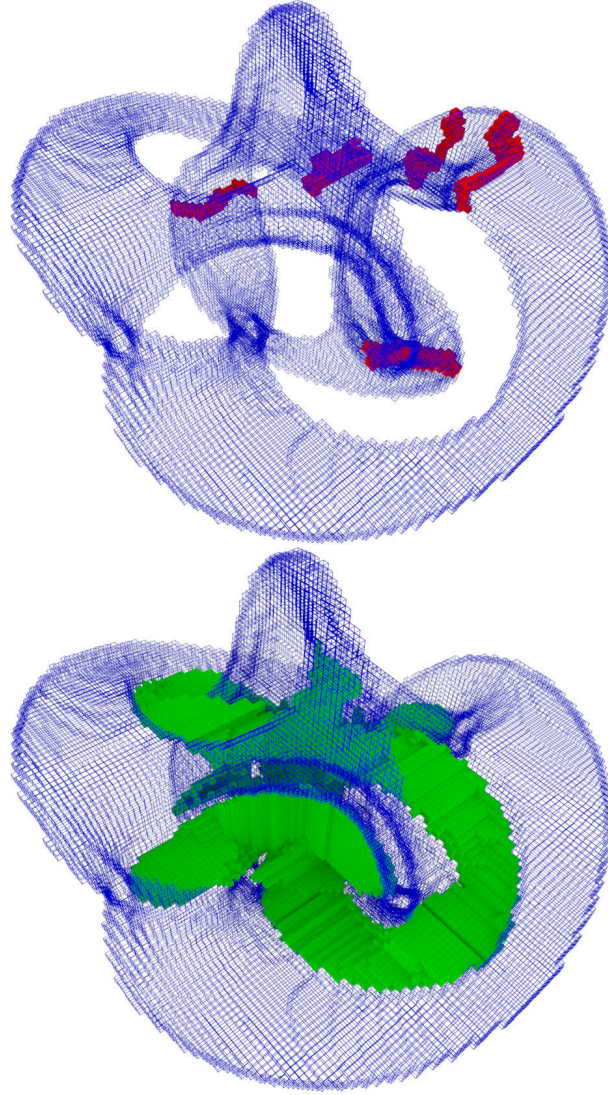of all the voxels incident to the cubes in C.

FIGURE 5.24: An object containing five 1-holes.

Obtaining an optimal (in the number of voxels in S) solution is an NP-hard problem, since it requires to solve the following integer linear programming problem:

$$\text{minimize} \sum_{p \in \hat{\mathtt{C}}} x_p$$
$$\text{subject to} \sum_{p \in \hat{\sigma}} x_p \geq 1, \forall \sigma \in \mathtt{C}$$
$$x_p \in \{0, 1\}, \forall p \in \hat{\mathtt{C}}$$

Fortunately, we can obtain a $(\ln(27)+1)$-approximation with a greedy algorithm, where the number 27 comes from the maximum number of cubes incident to a voxel. Roughly speaking, this algorithm takes the voxels having the greatest number of non-covered cubes until they are all covered.
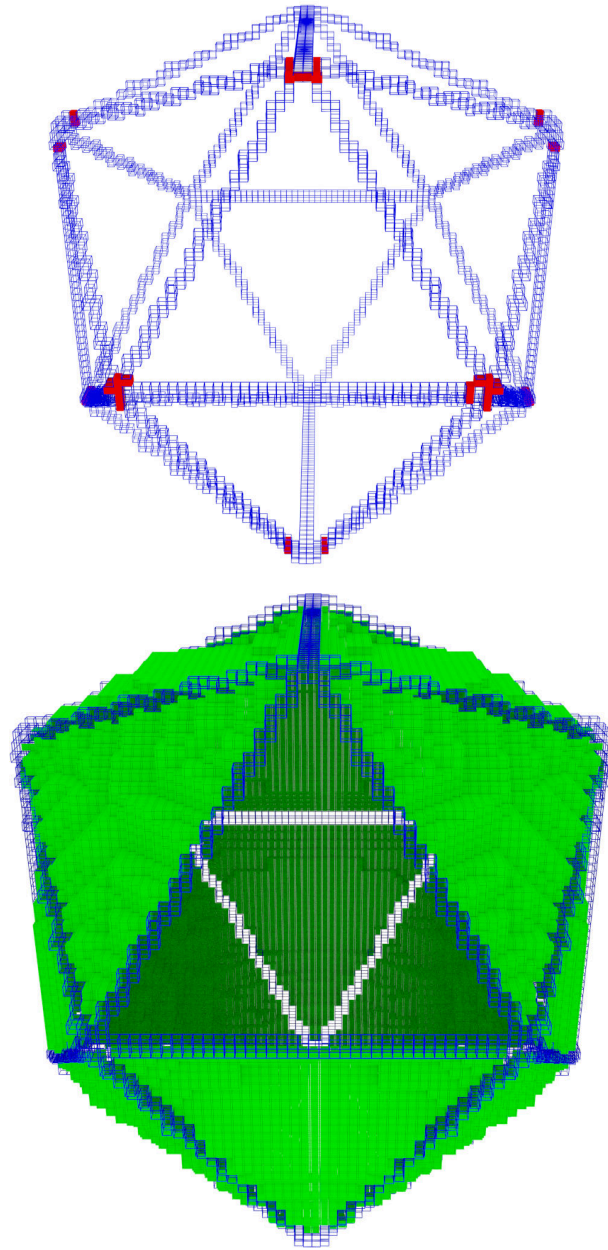
FIGURE 5.25: A wire-frame icosahedron.

Closing a hole given the dual associated cubical complex   In this case, finding a set of voxels covering a closing set C is trivial as it suffices to consider Ĉ, the voxels associated to the 0-dimensional faces of the cubes in C.

Opening a hole given the primal associated cubical complex   We now can consider a set of voxels S such that $K_p[X - S]$ does not contain any cube of O. The solution is easy: $S = \hat{O} = \bigcup_{\sigma \in O} \hat{\sigma}$.

Opening a hole given the dual associated cubical complex   Finding a minimal set of voxels S such that $K_d[X - S]$ does not contain any cube of O is again a set-covering problem, as we look for a set of voxels whose
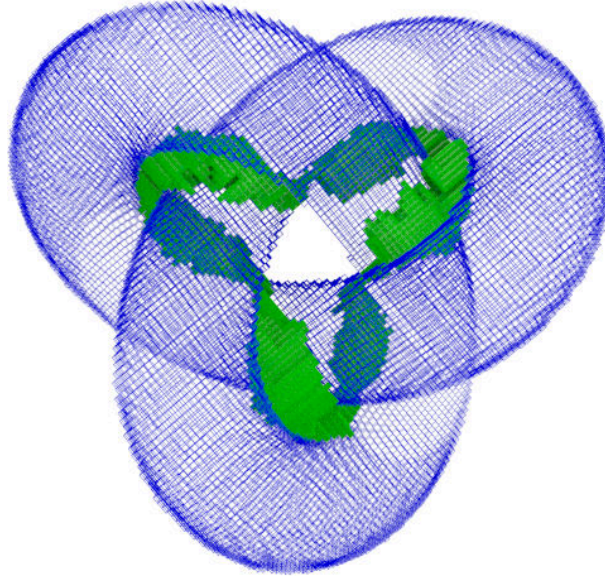
FIGURE 5.26: The output of Algorithm 9 does not close the
1-hole of this knot.

incident cubes (cofaces of the 0-cube) covers 0. The approximation ratio of
the greedy algorithm is again $\ln(27) + 1$.

5.6 Conclusion and future works

We have introduced two measures for the homology groups of discrete objects which provide a relevant geometric information of their holes. Their computation is direct from the definition, and its complexity is dominated by the persistent homology computation. These measures can be visualized in terms of balls, which display holes in an alternative way to homology or cohomology generators. Moreover, this framework seems to be useful for obtaining small generators of homology and cohomology without any guarantee of optimality. Furthermore, we have defined the hole closing and opening in homological terms and presented one algorithm and two heuristics for computing them on a discrete object.

There are several perspectives for this research:

1. To formalize the geometric intuition behind the breadth and the thickness.

   Apart from the definition of the measures, we would like to find an equivalent definition. The breadth of a 0-hole (connected component) seems to be the radius of the biggest ball contained in it. The breadth of a 1-hole seems to be the radius of the maximal ball that can *traverse* it. The breadth of a 2-hole seems to be the radius of the maximal ball that can be *contained* in the hole. The thickness of a 0-hole seems to be radius of the minimal ball that can connect it to another 0-hole. The thickness of a 1-hole seems to be half the diameter of the smallest set opening it. The thickness of a 2-hole seems to be half the length of the smallest set opening it.

   Note, however, that there are pathological cases. Consider two parallel rings close enough so, when they are dilated, they merge before

their 1-holes disappear, as depicted in Figure 5.27. One 1-hole vanishes because the two rings merge, and thus its breadth does not coincide with the radius of the ring. We must take this into consideration when trying to find a valid equivalent definition.
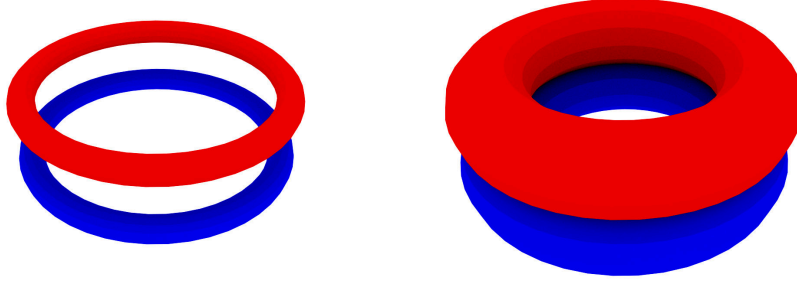

FIGURE 5.27: Example where the breadth of one ring does
not coincide with its radius.

2. To consider alternative measures.
   As we have seen, there are several geometric measures for chains. Let μ be one of such measures (e.g. volume, diameter, radius), we can define the breadth of the holes of an object as μ of the generators of a minimal homology base for μ. The thickness measure can be defined in terms of the cohomology groups. If they are unique, which is not trivial, it could be interesting to compare these measures with the ones we defined. However, note that they are more difficult to compute and that there is no notion of thickness-breadth pair since there is no canonical matching between the elements of the minimal bases for homology and cohomology.

3. The uniqueness of the thickness and breadth balls.
   Several algorithms depend on the pair of cells (σ, τ) associated to a thickness-breadth pair. Since they are not unique, as they depend on how we refine the filtration, it could be interesting to check what happens if we consider two sets of cells instead of just a pair. A hint can be to take σ and all the cubes in its homology generator with the same signed distance transform for the first set, and τ and all the cubes in $h(\sigma)$ with the same signed distance transform for the second one.

4. An algorithm for closing holes.
   We must keep on trying to find an algorithm that closes holes in the style of Algorithm 8. Also, Algorithm 9 presents the same disadvantages observed in [72], where a closing set for a 1-hole is far from being minimal. A promising hint is to consider the small homology generator provided by Algorithm 6, transform it into a discrete object and close its hole. However, it is not clear that this will close the chosen hole.

5. Simplicial complexes.
   The most challenging perspective is probably to adapt the definition of the breadth and the thickness to the context of simplicial complexes. The definition of our measures is theoretically valid for a subspace of the Euclidean space $\mathbb{R}^d$, so we could try to triangulate

the space (or at least the convex hull of the complex) and assign the signed distance transform values to the simplices. The main difficulty is thus how to define/compute a triangulation T of the space with a parameter, as we are going to compute an approximation of the measures in the continuous space.

Chapter 6

Conclusion

APART from the individual conclusions in each chapter, we summarize here the main results of this essay and, more importantly, we discuss the future perspectives of this work. Later we describe two works that have not been developed in this dissertation.

6.1 General conclusion

The main goal of this thesis is to study discrete objects—that is, binary images, volumes or their equivalent notions in higher dimensions—from a homological point of view. The initial objective was to extend the work on the homological spanning forest developed by H. Molina-Abril and P. Real [14, 91, 90], considering geometric features of the discrete objects such as the curvature or the medial axis. We will now consider the history of each chapter and the relation between them.

6.1.1 Homological Discrete Vector Field

The *homological spanning forest* naturally led to the concept of the homological discrete vector field (HDVF), which has a richer structure and better properties. However, finding the correct definition was far from being a trivial task.

When studying the homological spanning forest we were disappointed that it did not work well for classical examples in discrete Morse theory such as the Bing's house or the dunce hat. The homological discrete vector field was born while studying the former. One cannot find a perfect DGVF on it since at some point we find three $\mathcal{V}$-paths (instead of one) between two critical cells that should be paired and we cannot reverse any of them since this would create a closed $\mathcal{V}$-path in the DGVF. Despite this, we decided to reverse one. The operator h in a reduction is the most important since the others are defined through it. In the reduction associated to a DGVF, h is defined recursively and thus it is not well defined if there are closed $\mathcal{V}$-paths. We thought that we could compute h in terms of the chains $h(\sigma)$ where σ is a *confluence cell*, that is a cell where two or more closed $\mathcal{V}$-paths merge. Hence we obtain a system of linear equations, which we found that always had a solution. This approach was published in [57], but we did not understand why this works and we could not prove it. We later found the correct definition inspired by the formula

$$1 + x + x^2 + \cdots = \frac{1}{1-x}$$

The classical definition of the operator h is similar to the left side of the formula. As in our case the geometric sequence never vanishes, we must use the right side of the formula. Thus, we need something (actually a submatrix of the boundary matrices) to be invertible. We also changed the formulas for the reduction in order to have a clean definition. This is the formalism we used in [60].

Given a CW complex K, a homological discrete vector field is a pair of disjoint subsets (P, S) of the cells of K such that the boundary matrix of K restricted to these sets is invertible. Given this property, we can define a reduction on the chain complex associated to K, so we can compute its homology groups. This definition generalizes several concepts such as the discrete gradient vector field, the iterated discrete gradient vector field and the reduction induced by the Smith normal form.

This structure reveals an idea which is not evident in other methods for computing homology. There are many possible different HDVFs for the same CW complex. They are actually not completely independent, since we can define basic relations between them. Thus, a global structure containing all the possible HDVFs for a CW complex appears, which has not been completely understood.

There is a natural algorithm for building a HDVF. We have studied how to efficiently compute the associated reduction and we have estimated its complexity both in theory and in practice, by considering random cubical complexes. The results show that the worst-case complexity of building a HDVF is $\mathcal{O}(n^3)$ but significantly less in practice.

Open questions   We recall that a HDVF is perfect if its associated reduction is perfect, that is, if it reduces the chain complex associated to the CW complex to its homology groups.

We do not know if a perfect HDVF exists for every CW complex. We have found chain complexes (actually just matrices) for which this is false, so we think that there may be exceptions. Nevertheless, the boundary matrices of simplicial, cubical or regular CW complex have certain properties that may guarantee the existence of a perfect HDVF. We have tried to find a counter-example by brute-force and we have not succeeded. Observe, however, that this problem is solved if the ring of coefficients is a field.

The previous problem assumes that the considered CW complexes have torsion-free homology groups, because there exists no perfect reduction (and thus, HDVF) otherwise. However, we have found that there exist HDVFs whose reduced boundary matrix is already in the Smith normal form. Let us call such kind of HDVF a *pseudo-perfect* HDVF. Thus, like in the previous paragraph, we cannot conclude that any CW complex admits a pseudo-perfect HDVF.

Given the operations introduced in Section 3.7, we could define a kind of edit distance [20, § 15] between HDVFs. Given two HDVFs, their distance can be the minimal number of operations A, R, M, W, MW (or a subset of them) needed to transform one into the other. It is not clear that such a number exists, since one HDVF may be impossible to transform into another, so we should consider an extended metric (with infinite value for non-connected HDVFs). We find this problem fascinating, though we cannot see any practical application for this.

Possibly, the most useful perspective to this work is to compute zigzag persistence homology with the HDVF framework. We have explained in Section 3.8 how to compute (standard) persistent homology with HDVFs, which gives a very clear idea of what persistent homology represents. Zigzag persistent homology [13] is a more recent and complex theory which lacks a clear intuition. We are working on an algorithm for computing zigzag persistent homology using HDVFs and their operations. The advantage of doing this, rather than reducing its complexity, is to make this theory clearer and easier to understand.

6.1.2 Fast Computation of Betti Numbers on Three-Dimensional Cubical Complexes

In order to compute the homology groups of discrete objects, one needs to first transform them into cubical complexes. The algorithm introduced in this chapter computes (only) the Betti numbers of 3D cubical complexes. We thus can compute the Betti numbers of a discrete object in linear time with regard to the size of the bounding box.

This algorithm reduces the problem of computing the Betti numbers to count connected components in two graphs defined on the cubical complex. This is proved using the HDVF framework. The advantage of the cubical complex is not only that we know its complement (which is used), but that its regular structure allows us to easily subdivide the graphs into subgraphs with a reasonable number of edges between them.

Three versions of this algorithm have been implemented depending on how we count the number of connected components. Even the slowest one (the sequential version) outperforms the library RedHom, specialized in homological operations on cubical complexes.

Open questions Delfinado and Edelsbrunner sketched an algorithm in [29] for computing the Betti numbers of a simplicial complex embedded in $\mathbb{R}^3$ which is not necessarily a subcomplex of a triangulation of S^3, though it is not clearly proved. Later, Dey and Guha proved in [32] this algorithm, though complexes not being a three-manifold require a pre-processing step. The idea is similar to our algorithm, except that β_2 is obtained by recognizing closed surfaces on the boundary of the complex. We are trying to find a simpler algorithm and proof using the HDVF framework for both simplicial and cubical complexes.

We also aim at computing the Betti numbers directly on a 3D discrete object for the 6 and the 26-connectivity relation. We can do this by building the associated primal (or dual) cubical complex and using our algorithm, but it seems easy to do this directly on the discrete object. For instance, given a discrete object X with the 6-connectivity relation, β_0 is the number of connected components of X, β_2 is the number of connected components in its complement with the 26-connectivity relation (minus the unbounded component) and the Euler characteristic can be computed locally in each voxel of X. This idea is already present in [100], but the theoretical foundations of this work are not clear. In order to assign a topological space to a discrete object, we can use the primal associated cubical complex

for the 6-connectivity and the dual associated cubical complex for the 26-connectivity. As these complexes are locally built, it is easy to transfer the calculations of our algorithm in the cubical complex to the original object.

Finally, we want to explore other methods for counting connected components. In the image processing literature, connected components labeling is generally performed by traversing the volume in raster scan order (first incrementing the first coordinate, then the second and later the third one) and making an equivalence table between the voxels that are adjacent. This strategy avoids accessing the voxels in a random order (such as while performing a breadth first search), which provokes page faults. We intend to try these other methods to make our algorithm run faster, and also to treat huge complexes (bigger than 1001^3) which can be read by slices in order to fit in the virtual memory.

6.1.3 Measuring Holes

It took us a long time to combine digital geometry and homology. Motivated by a problem in geostatistics [23], where the Betti numbers do not give any information about the shape or size of the holes, we developed a simple definition of the size of the holes using the signed distance transform and persistent homology. It was a great surprise to find that holes do not have one natural measure but two, since we can erase them by cracking or filling them. Fortunately, these measures are stable under small perturbations, which makes them suitable for applications where objects come from acquisition devices or computer simulations.

Surprisingly, the computation of these measures provides a novel representation of holes in terms of balls (instead of homology generators) which we show to be useful through examples. We also include some recent research about obtaining homology and cohomology generators and about opening and closing holes using these measures.

Open questions   We want to emphasize three perspectives about this research.

The definition of the breadth and the thickness is not only valid for cubical complexes, but for any subspace of $\mathbb{R}^n$. We want to compute the measures for simplicial complexes embedded in $\mathbb{R}^3$. The challenge is how to triangulate—tetrahedrize, actually—the space according to the signed distance transform. We think that we can only compute an approximation of the measures, since we can only compute the signed distance transform in a finite number of points. We are convinced that computing small generators for (co)homology and opening or closing holes in this context will give better results since the output is not restricted to fit in the grid.

While the problem of finding small homology generators has been largely studied from a computational point of view, that of closing and opening holes seems to be overlooked. We do not aim at proving that finding a minimal closing or opening set is a NP problem, but we believe it is.

Even though the measures were conceived to solve a practical problem (studying the holes of different types of soils), we really do not have any application for them. Specific needs may give new ideas, such as using different distances. In particular, some recent works [108, 109, 113] study

the shape of the universe through persistent homology, and we think that the breadth and the thickness may be useful for this.

6.2 Other works

We present here two works that are not described in this thesis for different reasons.

6.2.1 Cellular skeletons

One of the main theorems in discrete Morse theory states that a simplicial complex endowed with a DGVF is homotopy equivalent to a CW complex with exactly one q-cell for each critical q-simplex. From a topological point of view, the *skeleton* of a discrete object is a subset which is homotopy equivalent. Thus, it is clear that a DGVF—moreover, a reduction—provides a kind of skeleton for a discrete object.

In [58] we introduced the concept of *cellular skeleton*. Given a reduction from the primal (or dual) associated cubical complex of a discrete object, it is the set of chains $g(C)$. Taking $\mathbb{Z}_2$ as ring of coefficients, each chain is a set of cells comprehending a manifold-like part of the skeleton. Thus, this is not just a subset of voxels but a chain complex.
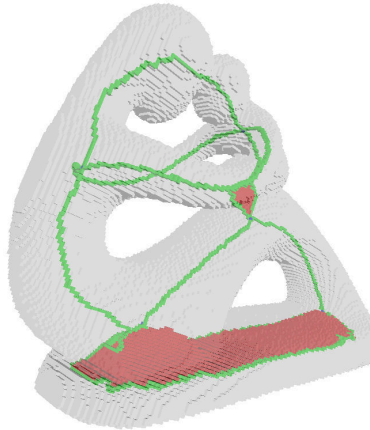

FIGURE 6.1: Cellular skeleton of *Fertility*.

These skeletons were computed using known algorithms for homotopy thinning for cubical complexes such as [84, 41, 16, 21, 22]. Then we used a *cell clustering* algorithm to reduce the number of cells without changing the shape of the skeleton. Figure 6.1 shows an example of the visualization of a cellular skeleton.

The cellular skeleton has several advantages. On one hand, the resulting skeleton preserves the topology of the original object since it is defined through a reduction. On the other hand, since the previously mentioned algorithms give (hopefully) centered skeletons, thus a subsequent homology computation on the cellular skeleton should give centered homology generators which, though not minimal, are visually pleasant.

The cell clustering algorithm was not fully understood in [58]. It takes a 3D cubical complex as input and it returns a reduction. We later conceived a general version for any dimension and we proved that every maximal

(without cofaces) cell of the complex belongs to one and only one chain $g(\gamma)$. This implies that the shape of the skeleton does not change. On the other hand, the resulting skeleton is not unique (since there are multiple choices in the algorithm) and it does not always return a minimal (in the number of cells) skeleton. These results have not been published, but they seem to be equivalent to the works of Damiand et al. [25, 24] in the context of *generalized maps*.

Sadly, we did not advance more in this direction due to the lack of practical applications for this theory.

6.2.2   Opening holes in discrete objects

There is a simple way of closing the holes of a discrete object. Let X be a 3D discrete object with the 26-connectivity. Consider an acyclic volume $Y \supset X$ containing it (its bounding box, for instance) and remove *simple* points—that is, voxels that can be removed without changing the homotopy type of the object—in $Y - X$ until idempotency. The final volume contains X and has no holes since it is homotopy equivalent to Y, so it *closes* the holes of X. Giving priority to voxels which are further from the object usually gives minimal fillings. This has been studied in [1, 71, 72].

Inspired by the duality in the measures (see Chapter 5), we would like to *open* holes in discrete objects. A simple algorithm consists in considering a point x inside the object and then adding simple points in the object until idempotency. The object obtained is a subset of the original object without holes. Again, considering the distance transform for the order in which the points are added should give a minimal result.

Simple examples in 2D such as those in Figure 6.2 show that this natural approach does not give optimal fractures. Instead of obtaining the minimal cuts opening the holes, the fractures seem to follow some angles.
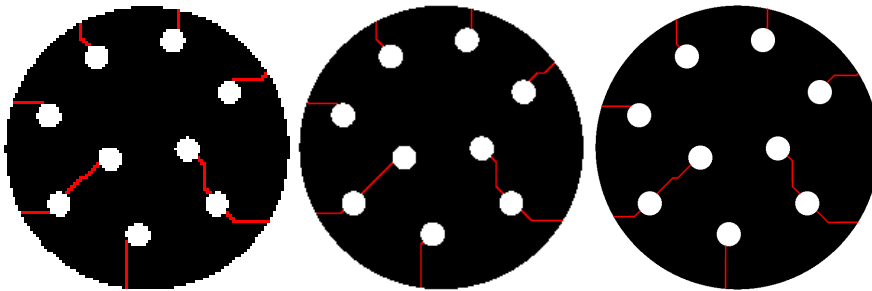

FIGURE 6.2: Result for the same image with different resolution. The cuts in the figure on the right have been thickened for visibility.

This idea is very recent and we have not had time enough to develop it. We decided to include it here for its simplicity and possible future perspectives. We plan to explore more advanced techniques for opening holes that converge to the minimal cuts.

Bibliography

[1] Zouina Aktouf, Gilles Bertrand, and Laurent Perroton. A three-dimensional holes closing algorithm. *Pattern Recogn. Lett.*, 23(5):523–531, March 2002.

[2] Madjid Allili and David Corriveau. Topological analysis of shapes using Morse theory. *Computer Vision and Image Understanding*, 105(3):188–199, 2007.

[3] Rafael Ayala, Desamparados Fernández-Ternero, and José Antonio Vilches. Perfect discrete Morse functions on 2-complexes. *Pattern Recogn. Lett.*, 33(11):1495–1500, August 2012.

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

[5] Ainhoa Berciano, Helena Molina-Abril, and Pedro Real. Searching high order invariants in computer imagery. *Appl. Algebra Eng. Commun. Comput.*, 23(1-2):17–28, 2012.

[6] R. H. Bing. Some aspects of the topology of 3-manifolds related to the Poincaré conjecture. *Lectures on Modern Mathematics*, 2, 1964.

[7] Jean-Daniel Boissonnat, Tamal K. Dey, and Clément Maria. The compressed annotation matrix: An efficient data structure for computing persistent cohomology. *Algorithmica*, 73(3):607–619, 2015.

[8] Dobrina Boltcheva, Sara Merino Aceitunos, Jean-Claude Léon, and Franck Hétroy. Constructive Mayer-Vietoris algorithm: Computing the homology of unions of simplicial complexes. Research Report RR-7471, INRIA, December 2010.

[9] Gunilla Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27(3):321–345, 1984.

[10] Peer-Timo Bremer, Ingrid Hotz, Valerio Pascucci, and Ronald Peikert, editors. *Topological Methods in Data Analysis and Visualization III, Theory, Algorithms, and Applications.* Springer, 2014.

[11] Piotr Brendel, Paweł Dłotko, Graham Ellis, Mateusz Juda, and Marian Mrozek. Computing fundamental groups from point clouds. *Applicable Algebra in Engineering, Communication and Computing*, 26(1):27–48, 2015.

[12] Kenneth S. Brown and Ross Geoghegan. An infinite-dimensional torsion-free FP_∞ group. *Inventiones mathematicae*, 77(2):367–381, 1984.

[13] Gunnar Carlsson and Vin Silva. Zigzag persistence. *Foundations of Computational Mathematics*, 10(4):367–405, 2010.

[14] Javier Carnero, Helena Molina-Abril, and Pedro Real. Triangle mesh compression and homological spanning forests. In *Computational Topology in Image Context - 4th International Workshop, CTIC 2012, Bertinoro, Italy, May 28-30, 2012. Proceedings*, pages 108–116, 2012.

[15] Manoj K. Chari. On discrete Morse functions and combinatorial decompositions. *Discrete Mathematics*, 217(1–3):101–113, 2000.

[16] John Chaussard and Michel Couprie. Surface thinning in 3D cubical complexes. In Petra Wiederhold and Reneta P. Barneva, editors, *Combinatorial Image Analysis*, volume 5852 of *Lecture Notes in Computer Science*, pages 135–148. Springer Berlin Heidelberg, 2009.

[17] Chao Chen and Daniel Freedman. Measuring and computing natural generators for homology groups. *Comput. Geom.*, 43(2):169–181, 2010.

[18] Chao Chen and Daniel Freedman. Hardness results for homology localization. *Discrete & Computational Geometry*, 45(3):425–448, 2011.

[19] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2006.

[20] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

[21] Michel Couprie. Hierarchic Euclidean skeletons in cubical complexes. In *Discrete Geometry for Computer Imagery - 16th IAPR International Conference, DGCI 2011, Nancy, France, April 6-8, 2011. Proceedings*, pages 141–152. 2011.

[22] Michel Couprie. Topological maps and robust hierarchical Euclidean skeletons in cubical complexes. *Computer Vision and Image Understanding*, 117(4):355–369, 2013.

[23] Asmae Dahrabou, Sophie Viseur, Aldo Gonzalez-Lorenzo, Jeremy Rohmer, Alexandra Bac, Pedro Real, Jean-Luc Mari, and Pascal Audigane. Topological comparisons of fluvial reservoir rock volumes using Betti numbers: Application to CO_2 storage uncertainty analysis. In *6th International Workshop on Computational Topology in Image Context (CTIC 2016), Lecture Notes in Computer Science (LNCS 9667)*, pages 101–112. Springer International Publishing, 2016. DOI:10.1007/978-3-319-39441-1_10.

[24] Guillaume Damiand, Rocío González-Díaz, and Samuel Peltier. Removal operations in nD generalized maps for efficient homology computation. In *Computational Topology in Image Context - 4th International Workshop, CTIC 2012, Bertinoro, Italy, May 28-30, 2012. Proceedings*, pages 20–29, 2012.

[25] Guillaume Damiand, Samuel Peltier, and Laurent Fuchs. Computing homology generators for volumes using minimal generalized maps. In *Combinatorial Image Analysis, 12th International Workshop, IWCIA 2008, Buffalo, NY, USA, April 7-9, 2008. Proceedings*, pages 63–74, 2008.

[26] A. M. Davie and A. J. Stothers. Improved bound for complexity of matrix multiplication. *Proceedings of the Royal Society of Edinburgh, Section: A Mathematics*, 143:351–369, 4 2013.

[27] Sarah Day, William D. Kalies, and Thomas Wanner. Verified homology computations for nodal domains. *Multiscale Modeling & Simulation*, 7(4):1695–1726, 2009.

[28] Éric Colin de Verdière and Francis Lazarus. Optimal system of loops on an orientable surface. *Discrete & Computational Geometry*, 33(3):507–534, 2005.

[29] 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.

[30] Olaf Delgado-Friedrichs, Vanessa Robins, and Adrian P. Sheppard. Skeletonization and partitioning of digital images using discrete Morse theory. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):654–666, 2015.

[31] Tamal K. Dey, Fengtao Fan, and Yusu Wang. An efficient computation of handle and tunnel loops via Reeb graphs. *ACM Trans. Graph.*, 32(4):32:1–32:10, July 2013.

[32] Tamal K. Dey and Sumanta Guha. Computing homology groups of simplicial complexes in $\mathbb{R}^3$. *J. ACM*, 45(2):266–287, 1998.

[33] Tamal K. Dey, Anil N. Hirani, and Bala Krishnamoorthy. Optimal homologous cycles, total unimodularity, and linear programming. *SIAM Journal on Computing*, 40(4):1026–1044, 2011.

[34] Tamal K. Dey, Kuiyu Li, Jian Sun, and David Cohen-Steiner. Computing geometry-aware handle and tunnel loops in 3D models. *ACM Trans. Graph.*, 27(3):45:1–45:9, August 2008.

[35] Tamal K. Dey, Jian Sun, and Yusu Wang. Approximating loops in a shortest homology basis from point data. In *Proceedings of the Twenty-sixth Annual Symposium on Computational Geometry*, SoCG '10, pages 166–175, New York, NY, USA, 2010. ACM.

[36] Michael B. Dillencourt, Hanan Samet, and Markku Tamminen. A general approach to connected-component labeling for arbitrary image representations. *J. ACM*, 39(2):253–280, April 1992.

[37] David Steven Dummit and Richard M. Foote. *Abstract algebra*. John Wiley & sons, Hoboken, NJ, 2004.

[38] Paweł Dłotko, Robert Ghrist, Mateusz Juda, and Marian Mrozek. Distributed computation of coverage in sensor networks by homological methods. *Applicable Algebra in Engineering, Communication and Computing*, 23(1-2):29–58, 2012.

[39] Paweł Dłotko, Tomasz Kaczyński, Marian Mrozek, and Thomas Wanner. Coreduction homology algorithm for regular CW-complexes. *Discrete and Computational Geometry*, 46(2):361–388, 2011.

[40] Paweł Dłotko and Ruben Specogna. Efficient cohomology computation for electromagnetic modeling. *CMES: Computer Modeling in Engineering & Sciences*, 60(3):247–278, 2010.

[41] Paweł Dłotko and Ruben Specogna. Topology preserving thinning of cell complexes. *IEEE Trans. Image Processing*, 23(10):4486–4495, 2014.

[42] Paweł Dłotko and Hubert Wagner. Computing homology and persistent homology using iterated Morse decomposition. *arXiv preprint arXiv:1210.1429*, pages 1–31, 2012.

[43] Herbert Edelsbrunner and John Harer. *Computational Topology - an Introduction*. American Mathematical Society, 2010.

[44] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, 2002.

[45] Samuel Eilenberg and Saunders Mac Lane. On the groups H(π, n), I. *Annals of Mathematics*, 58(1):55–106, 1953.

[46] Alexander Engstrom. Discrete Morse functions from Fourier transforms. *Experimental Mathematics*, 18(1):45–54, 2009.

[47] Jeff Erickson and Kim Whittlesey. Greedy optimal homotopy and homology generators. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*, pages 1038–1046, 2005.

[48] Robin Forman. Combinatorial vector fields and dynamical systems. *Mathematische Zeitschrift*, 228(4):629–681, 1998.

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

[50] Robin Forman. A user's guide to discrete Morse theory. *Séminaire Lotharingien de Combinatoire*, 48:B48c, 2002.

[51] Felix R. Gantmacher. *The theory of matrices*, volume 1. Chelsea Publishing Company, New York, N.Y., 1959.

[52] Rocío González-Díaz, María José Jiménez, and Belén Medrano. Cubical cohomology ring of 3D photographs. *Int. J. Imaging Systems and Technology*, 21(1):76–85, 2011.

[53] Rocío González-Díaz, María José Jiménez, Belén Medrano, and Pedro Real. Chain homotopies for object topological representations. *Discrete Applied Mathematics*, 157(3):490–499, 2009.

[54] Rocío González-Díaz, María José Jiménez, Belén Medrano, and Pedro Real. A tool for integer homology computation: λ-AT-model. *Image Vision Comput.*, 27(7):837–845, 2009.

[55] Rocío González-Díaz and Pedro Real. On the cohomology of 3D digital images. *Discrete Applied Mathematics*, 147(2-3):245–263, 2005.

[56] Aldo Gonzalez-Lorenzo. Computing homological discrete vector fields. *Foundations of Computational Mathematics*, "submitted".

[57] Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc Mari, and Pedro Real. Computing homological information based on directed graphs within discrete objects. In *16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2014, Timisoara, Romania, September 22-25, 2014*, pages 571–578, 2014. DOI:10.1109/SYNASC.2014.82.

[58] Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc Mari, and Pedro Real. Cellular skeletons: A new approach to topological skeletons with geometric features. In *Computer Analysis of Images and Patterns - 16th International Conference (CAIP 2015), Lecture Notes in Computer Science (LNCS 9257)*, pages 616–627. Springer International Publishing, 2015. DOI:10.1007/978-3-319-23117-4_53.

[59] Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc Mari, and Pedro Real. Two measures for the homology groups of binary volumes. In *Discrete Geometry for Computer Imagery - 19th IAPR International Conference (DGCI 2016), Lecture Notes in Computer Science (LNCS 9647)*, pages 154–165. Springer International Publishing, 2016. DOI:10.1007/978-3-319-32360-2_12.

[60] Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc Mari, and Pedro Real. Allowing cycles in discrete Morse theory. *Foundations of Computational Mathematics*, "to appear".

[61] Aldo Gonzalez-Lorenzo, Mateusz Juda, Alexandra Bac, Jean-Luc Mari, and Pedro Real. Fast, simple and separable computation of Betti numbers on three-dimensional cubical complexes. In *6th International Workshop on Computational Topology in Image Context (CTIC 2016), Lecture Notes in Computer Science (LNCS 9667)*, pages 130–139. Springer International Publishing, 2016. DOI:10.1007/978-3-319-39441-1_12.

[62] Paul W. Gross and P. Robert Kotiuga. *Electromagnetic Theory and Computation*. Cambridge University Press, 2004. Cambridge Books Online.

[63] V. K. A. M. Gugenheim and H. J. Munkholm. On the extended functoriality of Tor and Cotor. *Journal of Pure and Applied Algebra*, 4(1):9–29, 1974.

[64] Masahiro Hachimori. Simplicial complex library. http://infoshako.sk.tsukuba.ac.jp/~hachi/math/library/index_eng.html.

[65] James L. Hafner and Kevin S. McCurley. Asymptotically fast triangularization of matrices over rings. *SIAM J. Comput.*, 20(6):1068–1083, 1991.

[66] Shaun Harker, Konstantin Mischaikow, Marian Mrozek, and Vidit Nanda. Discrete Morse theoretic algorithms for computing homology of complexes and maps. *Foundations of Computational Mathematics*, 14(1):151–184, 2014.

[67] Allen Hatcher. *Algebraic topology*. Cambridge University Press, Cambridge, New York, 2002.

[68] Daniel S. Hirschberg, Ashok K. Chandra, and Dilip V. Sarwate. Computing connected components on parallel computers. *Commun. ACM*, 22(8):461–464, 1979.

[69] John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

[70] Intel. Threading Building Blocks. https://www.threadingbuildingblocks.org/, 2016.

[71] Marcin Janaszewski, Michel Couprie, and Laurent Babout. Geometric approach to hole segmentation and hole closing in 3D volumetric objects. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, 14th Iberoamerican Conference on Pattern Recognition, CIARP 2009, Guadalajara, Jalisco, Mexico, November 15-18, 2009. Proceedings*, pages 255–262, 2009.

[72] Marcin Janaszewski, Michel Couprie, and Laurent Babout. Hole filling in 3D volumetric objects. *Pattern Recogn.*, 43(10):3548–3559, October 2010.

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

[74] Calvin R. Maurer Jr., Rensheng Qi, and Vijay Raghavan. A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(2):265–270, 2003.

[75] Mateusz Juda and Marian Mrozek. $\mathbb{Z}_2$-homology of weak $(p-2)$-faceless p-pseudomanifolds may be computed in $o(n)$ time. *Topol. Methods Nonlinear Anal.*, 40(1):137–159, 2012.

[76] Mateusz Juda and Marian Mrozek. CAPD::RedHom v2 - Homology software based on reduction algorithms. In *Mathematical Software - ICMS 2014 - 4th International Congress, Seoul, South Korea, August 5-9, 2014. Proceedings*, pages 160–166, 2014.

[77] Mateusz Juda, Marian Mrozek, Piotr Brendel, Hubert Wagner, et al. CAPD::RedHom. http://redhom.ii.uj.edu.pl, 2010–2016.

[78] Tomasz Kaczyński, Marian Mrozek, and Maciej Ślusarek. Homology computation by reduction of chain complexes. *Computers and Mathematics with Applications*, 35(4):59–70, 1998.

[79] T. Yung Kong and Azriel Rosenfeld. Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48(3):357–393, 1989.

[80] Dmitry N. Kozlov. *Combinatorial Algebraic Topology*, volume 21 of *Algorithms and computation in mathematics*. Springer, 2008.

[81] Daniel Kraft. Computing the Hausdorff distance of two sets from their signed distance functions. *Preprint IGDK-2015-03*, 2015. https://igdk1754.ma.tum.de/foswiki/pub/IGDK1754/Preprints/Kraft_2015.pdf.

[82] Larry Lambe and Jim Stasheff. Applications of perturbation theory to iterated fibrations. *Manuscripta Mathematica*, 58(3):363–376, 1987.

[83] Thomas Lewiner, Helio Lopes, and Geovan Tavares. Toward optimality in discrete Morse theory. *Experimental Mathematics*, 12(3):271–285, 2003.

[84] L. Liu, Erin W. Chambers, David Letscher, and Tao Ju. A simple and robust thinning algorithm on cell complexes. *Comput. Graph. Forum*, 29(7):2253–2260, 2010.

[85] Ronald Lumia, Linda G. Shapiro, and Oscar A. Zuniga. A new connected components algorithm for virtual memory computers. *Computer Vision, Graphics, and Image Processing*, 22(2):287–300, 1983.

[86] Albert T. Lundell and Stephen Weubgram. *The Topology of CW Complexes*. Springer New York, 1969.

[87] John Milnor. *Morse Theory*. Princeton University Press, 1962.

[88] Nikola Milosavljevic, Dmitriy Morozov, and Primoz Skraba. Zigzag persistent homology in matrix multiplication time. In *Proceedings of the 27th ACM Symposium on Computational Geometry, Paris, France, June 13-15, 2011*, pages 216–225, 2011.

[89] Konstantin Mischaikow. Conley index theory. In Russell Johnson, editor, *Dynamical Systems*, volume 1609 of *Lecture Notes in Mathematics*, pages 119–207. Springer Berlin Heidelberg, 1995.

[90] Helena Molina-Abril. *Homological Spanning Forests for Discrete Objects*. PhD thesis, Universidad de Sevilla, 2012.

[91] Helena Molina-Abril and Pedro Real. Homological spanning forest framework for 2D image analysis. *Annals of Mathematics and Artificial Intelligence*, 64(4):385–409, 2012.

[92] Ugo Montanari. A method for obtaining skeletons using a quasi-Euclidean distance. *J. ACM*, 15(4):600–624, 1968.

[93] Marian Mrozek. Index pairs algorithms. *Foundations of Computational Mathematics*, 6(4):457–493, 2006.

[94] Marian Mrozek and Bogdan Batko. Coreduction homology algorithm. *Discrete & Computational Geometry*, 41(1):96–118, 2009.

[95] Marian Mrozek and Thomas Wanner. Coreduction homology algorithm for inclusions and persistent homology. *Computers & Mathematics with Applications*, 60(10):2812–2833, 2010.

[96] Marian Mrozek, Marcin Zelawski, A. Gryglewski, S. Han, and A. Krajniak. Homological methods for extraction and analysis of linear features in multidimensional images. *Pattern Recognition*, 45(1):285–298, 2012.

[97] Jayanta Mukherjee, Partha Pratim Das, M. Aswatha Kumar, and B. N. Chatterji. On approximating Euclidean metrics by digital distances in 2D and 3D. *Pattern Recognition Letters*, 21(6-7):573–582, 2000.

[98] Hans J. Munkholm. The Eilenberg-Moore spectral sequence and strongly homotopy multiplicative maps. *Journal of Pure and Applied Algebra*, 5(1):1–50, 1974.

[99] James R. Munkres. *Elements of algebraic topology*. Addison-Wesley, 1984.

[100] Akira Nakamura and Kunio Aizawa. On the recognition of properties of three-dimensional pictures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 7(6):708–713, 1985.

[101] Nicolas Normand and Pierre Évenou. Medial axis lookup table and test neighborhood computation for 3D chamfer norms. *Pattern Recognition*, 42(10):2288–2296, 2009.

[102] Samuel Peltier, Sylvie Alayrangues, Laurent Fuchs, and Jacques-Olivier Lachaud. Computation of homology groups and generators. In *Discrete Geometry for Computer Imagery, 12th International Conference, DGCI 2005, Poitiers, France, April 13-15, 2005, Proceedings*, pages 195–205, 2005.

[103] Pawel Pilarczyk and Pedro Real. Computation of cubical homology, cohomology, and (co)homological operations via chain contraction. *Adv. Comput. Math.*, 41(1):253–275, 2015.

[104] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.

[105] Azriel Rosenfeld and John L. Pfaltz. Sequential operations in digital picture processing. *J. ACM*, 13(4):471–494, October 1966.

[106] Francis Sergeraert. Effective homology, a survey. http://www-fourier.ujf-grenoble.fr/~sergerar/Papers/Survey.pdf, 1992. [Online; accessed 11-June-2014].

[107] Yossi Shiloach and Uzi Vishkin. An $O(\log n)$ parallel connectivity algorithm. *J. Algorithms*, 3(1):57–67, 1982.

[108] Thierry Sousbie. The persistent cosmic web and its filamentary structure - I. theory and implementation. *Monthly Notices of the Royal Astronomical Society*, 414(1):350–383, 2011.

[109] Thierry Sousbie, Christophe Pichon, and Hajime Kawahara. The persistent cosmic web and its filamentary structure - II. illustrations. *Monthly Notices of the Royal Astronomical Society*, 414(1):384–403, 2011.

[110] J. R. Stallings. Lectures on polyhedral topology. Tata Institute of Fundamental Research, Bombay, 1968.

[111] Arne Storjohann. Near optimal algorithms for computing Smith normal forms of integer matrices. In *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation*, ISSAC '96, pages 267–274, New York, NY, USA, 1996. ACM.

[112] Takashi Teramoto and Yasumasa Nishiura. Morphological characterization of the diblock copolymer problem with topological computation. *Japan Journal of Industrial and Applied Mathematics*, 27(2):175–190, 2010.

[113] Rien van de Weygaert, Gert Vegter, Herbert Edelsbrunner, Bernard J. T. Jones, Pratyush Pranav, Changbom Park, Wojciech A. Hellwing, Bob Eldering, Nico Kruithof, E. G. P. (Patrick) Bos, Johan Hidding, Job Feldbrugge, Eline ten Have, Matti van Engelen, Manuel Caroli, and Monique Teillaud. Transactions on computational science XIV. chapter Alpha, Betti and the Megaparsec Universe: On the Topology of the Cosmic Web, pages 60–101. Springer-Verlag, Berlin, Heidelberg, 2011.

[114] Uzi Vishkin. An optimal parallel connectivity algorithm. *Discrete Applied Mathematics*, 9(2):197–207, 1984.

[115] Hubert Wagner, Chao Chen, and Erald Vuçini. Efficient computation of persistent homology for cubical data. In Ronald Peikert, Helwig Hauser, Hamish Carr, and Raphael Fuchs, editors, *Topological Methods in Data Analysis and Visualization II*, Mathematics and Visualization, pages 91–106. Springer Berlin Heidelberg, 2012.

[116] Michael Werman and Matthew L. Wright. Intrinsic volumes of random cubical complexes. *Discrete & Computational Geometry*, 56(1):93–113, 2016.

[117] E.C. Zeeman. On the dunce hat. *Topology*, 2(4):341–358, 1963.

[118] Afra Zomorodian and Gunnar Carlsson. Localized homology. *Comput. Geom. Theory Appl.*, 41(3):126–148, November 2008.

[119] Afra Zomorodian and Gunnar E. Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.