



Thomas Lewiner

Geometric Discrete Morse Complexes

PhD Thesis

Thesis presented to the Post-graduate Program in Applied Mathematics of the Mathematics Department, PUC-Rio as partial fulfillment of the requirements for the degree of Doctor in Philosophy in Applied Mathematics

Adviser : Prof. Hélio Côrtes Vieira Lopes
Co-Adviser: Prof. Geovan Tavares dos Santos

Rio de Janeiro
July 2005



Thomas Lewiner

Geometric Discrete Morse Complexes

Thesis presented to the Post-graduate Program in Applied Mathematics of the Mathematics Department, PUC-Rio as partial fulfillment of the requirements for the degree of Doctor in Philosophy in Applied Mathematics. Approved by the following commission:

Prof. Hélio Côrtes Vieira Lopes

Adviser

Department of Mathematics — PUC-Rio

Prof. Geovan Tavares dos Santos

Co-Adviser

Department of Mathematics — PUC-Rio

Prof. Jean-Daniel Boissonnat

Géométrie Project — INRIA – Sophia Antipolis

Prof. Jean-Marie Morvan

C. Jordan Institute — Claude Bernard University – Lyon

Prof. Jorge Stolfi

Institute of Computing – UNICAMP

Prof. Luiz Carlos Pacheco Rodrigues Velho

Visgraf Laboratory — IMPA

Prof. Marcos Craizer

Department of Mathematics — PUC-Rio

Prof. José Eugenio Leal

Head of the Science and Engineering Center — PUC-Rio

Rio de Janeiro — July 29th, 2005

All rights reserved.

Thomas Lewiner

graduated from the Ecole Polytechnique (Paris, France) in Algebra and Computer Science, and in Theoretical Physics. He then specialized at the Ecole Supérieure des Télécommunications (Paris, France) in Signal and Image Processing, and in Project Management, while working for Inventel in wireless telecommunication systems based on Bluetooth technology. He then obtained a Master degree at the PUC-Rio in computational topology, and actively participated to the department's work for Petrobras.

Bibliographic data

Lewiner, Thomas

Geometric Discrete Morse Complexes / Thomas Lewiner; adviser: Hélio Côrtes Vieira Lopes; co-adviser: Geovan Tavares dos Santos. — Rio de Janeiro : PUC-Rio, Department of Mathematics, 2005.

v., 131 f: il. ; 29,7 cm

1. PhD Thesis - Pontifícia Universidade Católica do Rio de Janeiro, Department of Mathematics.

Bibliography included.

1. Mathematics – Thesis. 2. Morse Theory. 3. Forman Theory. 4. Homology. 5. Morse–Smale decomposition. 6. Gradient vector fields. 7. Computational Topology. 8. Computational Geometry. 9. Geometric Modeling. 10. Discrete Mathematics. I. Lopes, Hélio Côrtes Vieira. II. Santos, Geovan Tavares dos. III. Pontifícia Universidade Católica do Rio de Janeiro. Department of Mathematics. IV. Title.

CDD: 510

Acknowledgments

This section appears at the beginning of the work, but I believe we only realize the huge number of people who contributed to this thesis at the end. Of course, my family and friends, my advisers, the colleagues, professors and staff of the department of Mathematics and the INRIA gave me a real support, with the Fondation de l'École Polytechnique and the PUC who funded me during my thesis. But this mention is too short to state all they did for me during these three years. I will try to tell more of some of them, even if I would need the whole paper of this work not to forget anyone.

Ladies first, I begin with my mother, who achieved being there wherever I needed her, crossing the oceans just for a few days, as she knows how happy I am when she is close. I was actually brought up by more ladies: my grand mothers Fanny and Zaza, Simone, Elisabeth and France, Maggy, Muriel and Florence, Judith, Deborah, Myriam, Sodam, Scarlett and Nathalie. But I continue being educated by girls: Debora (sans accent) and Esther, Golda, Sarah, Salomé, Jasmine, Lio, Matis, Gabrielle, Emilie and the next nephews that are to come... And those who took care of me far from my family, including Ana Cristina, Agnès, Juliana(s), Silvana, Tania, Cynthia, Jessica, Christina, Marie, Creuza, Tereza, Kátia, and Albane, JA, Anne-Laure...

Gentlemen's names are less poetic to me, but they helped me a lot too. My father of course, by his experiences and expertises. My brothers Noam and William (in law and more) for making my sisters happy. My cousins David and Raphaël, and Gabriel, Daniel, Stéphane, and friends whose company is a home all over the world. Even if this work is of scientific concern, the professors and students I worked with gave me much more than a technical help. My adviser Hélio became one of my best friends, together with Marcos, Sinésio, Luiz, Shaoul, Gabriel. But also Jean-Daniel, Pierre, Alban, Olivier, Carlos, Nicolau, Geovan, Sergio, Lorenzo, Fred... My colleagues also gave me an everyday support and much more, including Bernardo, Wilson, Marcos, Sergio(s), João, Renner, Alex, Afonso, Francisco, Fabiano, Marcos and Luca, David, Marc, Steve, Philippe, Camille, Christophe, Abdelkrim, Laurent and Benjamin, Nicolas, David(s), Eric, Mathieu...

Abstract

Lewiner, Thomas; Lopes, Hélio Côrtes Vieira; Santos, Geovan Tavares dos. **Geometric Discrete Morse Complexes**. Rio de Janeiro, 2005. 131p. PhD Thesis — Department of Mathematics, Pontifícia Universidade Católica do Rio de Janeiro.

Differential geometry provides an intuitive way of understanding smooth objects in the space. However, with the evolution of geometric modeling by computer, this tool became both necessary and difficult to transpose to the discrete setting. The power of Morse theory relies on the link it created between differential topology and geometry. Starting from a combinatorial point of view, Forman's discrete Morse theory relates rigorously discrete objects to their topology, opening Morse theory to discrete structures.

This work proposes a constructive definition of geometric discrete Morse functions and their corresponding discrete Morse–Smale complexes, where the geometry is defined as a smooth function sampled on the vertices of the discrete structure. This construction required some homology computations that turned out to be a significant improvement over existing methods by itself. The resulting Morse–Smale decomposition can then be efficiently computed, and used for applications to persistence computation, Reeb graph generation, noise removal. . .

Keywords

Morse Theory. Forman Theory. Homology. Morse–Smale decomposition. Gradient vector fields. Computational Topology. Computational Geometry. Geometric Modeling. Discrete Mathematics.

Contents

I	Introduction	17
I.1	Main results	18
	<i>Flow and hypergraph components</i>	18
	<i>Complete homology computation</i>	20
	<i>Geometric discrete Morse functions and Morse–Smale decompositions</i>	20
I.2	Outline	22
II	Morse theories	25
II.1	Topological spaces	25
	<i>Smooth manifolds</i>	26
	<i>Cell complexes</i>	26
II.2	Vector fields	30
	<i>Tangent vector fields</i>	31
	<i>Cell matchings</i>	31
II.3	Morse functions	32
	<i>Smooth functions</i>	32
	<i>Acyclic matchings</i>	33
II.4	Critical sets	34
	<i>Critical points</i>	34
	<i>Critical cells</i>	35
II.5	Topological properties	36
	<i>Homotopy and Handle decomposition</i>	37
	<i>Simple homotopy and Morse inequalities</i>	38
II.6	Cancellations	42
	<i>Inversion</i>	42
	<i>Unique gradient path</i>	42
II.7	Flows and basins	43
	<i>Stable and unstable manifolds</i>	43
	<i>Invariant chains</i>	44
II.8	Morse complexes	45
	<i>Smale decomposition</i>	45
	<i>Witten–Morse homology</i>	46
III	Structure of a discrete Morse function	49
III.1	Layers and hypergraphs	50
	<i>Cell classification</i>	50
	<i>Layer hypergraph</i>	51
	<i>Hyperforest</i>	52
III.2	Greedy construction	56
	<i>Data structure and basic algorithms</i>	56
	<i>Greedy algorithm</i>	60

Contents	12
<i>Heuristic for optimal Morse functions</i>	66
<i>Complexity</i>	67
<i>Application to volumetric compression</i>	69
III.3 Flow basins and hypergraph components	70
<i>Flow image of a critical or dual cell</i>	70
<i>Discrete stable and unstable basins</i>	72
<i>Flow image of a primal cell</i>	74
IV Homology computation	77
IV.1 Algebraic and combinatorial tools	77
<i>Smith normal form</i>	78
<i>Boundary operator on the Morse complex</i>	80
IV.2 Complete homology calculus	84
<i>Betti numbers and torsion</i>	84
<i>Generators</i>	85
<i>Cycles decomposition</i>	87
<i>Computational results</i>	89
V Geometric discrete Morse complex	93
V.1 Geometric critical points	94
<i>Banchoff's definition and limits</i>	94
<i>Critical points in high-dimension</i>	96
<i>From critical points to critical cells</i>	96
V.2 Computation	99
<i>Critical points tracking</i>	99
<i>Critical cells selection</i>	99
<i>Main construction</i>	100
<i>Cancellation optimization</i>	102
<i>Basins identifications</i>	102
V.3 Properties and proof of the algorithm	104
<i>Regularity of the gradient</i>	104
<i>Minima positioning</i>	106
<i>Surfaces' maxima positioning</i>	107
<i>Surfaces' saddles positioning</i>	110
V.4 Results and applications	115
<i>Reeb graph</i>	115
<i>Persistence</i>	118
VI Conclusions and future works	121
Bibliography	123
Index	127
Summary of notations	129

L'idée d'intensité est donc située au point de jonction de deux courants, dont l'un nous apporte du dehors l'idée de grandeur extensive, et dont l'autre est allé chercher dans les profondeurs de la conscience, pour l'amener à la surface, l'image d'une multiplicité interne. Reste à savoir en quoi cette dernière image consiste, si elle se confond avec celle du nombre, ou si elle en diffère radicalement. (...) Et de même que nous nous sommes demandé ce que serait l'intensité d'une sensation représentative si nous n'y introduisions l'idée de sa cause, ainsi nous devons rechercher maintenant ce que devient la multiplicité de nos états internes, quelle forme affecte la durée, quand on fait abstraction de l'espace où elle se développe. Cette seconde question est autrement importante que la première. Car si la confusion de la qualité avec la quantité se limitait à chacun des faits de conscience pris isolément, elle créerait des obscurités, comme nous venons de le voir, plutôt que des problèmes. Mais en envahissant la série de nos états psychologiques, en introduisant l'espace dans notre conception de la durée, elle corrompt, à leur source même, nos représentations du changement extérieur et du changement interne, du mouvement et de la liberté.

Henri Bergson,

Essai sur les données immédiates de la conscience.

Foreword

Usually in mathematics, we expose our works in a reverse order: we generally discover and understand concepts by examples, applications, generalizations of some intuition and we present these same examples and applications as corollary or exercises deduced from our work. This work will conform to that practice. However, I would like first to summarize how the concepts of this thesis emerged.

This work is the realization of a slow process that took place during the last three years. The main problems were stated already from the end of my Master degree at PUC–Rio, and I thought that the main results would come out shortly after. At that time, I had no real experience of long lasting problems, since my Master’s thesis went quickly on a good direction, thanks to the feeling of my adviser. I ended my Master discovering that some of our results could actually be deduced more directly from Forman’s original work, although our formulation of the main problems was more efficient for deducing algorithms. At the same time, I noticed the relation between the connected components of the hypergraphs representing a discrete Morse function and the Morse–Smale decomposition, which plays a fundamental role in this work.

This relation helped at the beginning of my PhD at the INRIA in a specific application to molecular docking, and this work has been quickly accepted to a conference considered important. Nevertheless, the conference did not motivate further discussions and three years later, the biological problem is still not well understood. Although we spent almost one year working on this topic, our geometric approach of docking did not contribute so much to the area. Only after that year, I realized that the discrete Morse–Smale decomposition that motivated this paper was in itself an important topic that I had left aside.

I then began officially this separate PhD at PUC–Rio, spending the entire second year of this PhD in Brazil. While I worked mainly on other topics, from Petrobras projects to mesh compression and approximation of differential quantities of curves, the ground concepts of this work emerged slowly. First, the relation between the hypergraphs and the Morse–Smale decomposition became more intuitive by considering the flow more carefully. Then, the first

calculus of the flow of a dual cell made me more confident in the possibility of defining a discrete Morse–Smale complex, although I discovered afterward that this first calculus was again a translation of Forman’s calculus to our hypergraph representation. At that time, I completed the implementation of a complex algorithm for the Morse–Smale decomposition, and it became clear that the Banchoff’s definition of critical points I was using was not sufficient for Morse–Smale decompositions in dimension greater than 3.

Therefore, I focused on a definition of critical points, using local homology. For coherence, I tried to design an algorithm for computing Betti numbers using discrete Morse theory. This algorithm is in fact not only as efficient as the classical incremental algorithm for simplicial sub–manifolds of the 3–sphere, but it also worked on *any* regular cellular complex. The implementation of the algorithm was easy, then I extended it gradually to first compute the whole homology group with torsion, and then to derivate a basis of generating cycles for the homology. The next extension was to compute the decomposition of any cycle on the basis, and this problem opened many questions, not all of them being solved. That is the main reason why I completed the algorithm only recently, thanks to the help I received from the professors of the Mathematical department, include my adviser Hélio Lopes, Marcos Craizer, Geovan Tavares, Nicolau Saldanha and Carlos Tomei, and also the great help of David Cohen–Steiner, and Alban Quadrat from the INRIA. This community gives me hopes of answering the other questions from a theoretical point of view.

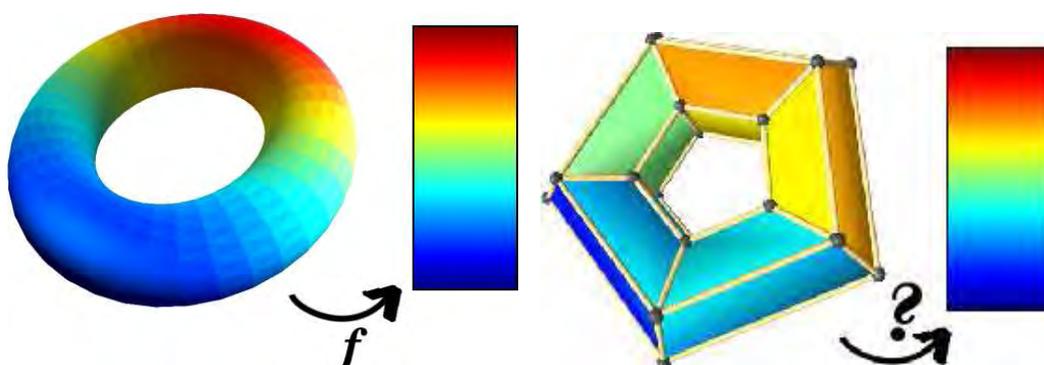
In parallel, I looked for applications and simplifications of the discrete Morse–Smale decomposition. Most of the papers on computational topology were referring to the “persistence” as a possible tool for their applications. This is a very nice concept built on top of homology, that was actually first used by Smale for his proof of Poincaré’s conjecture for high dimensions, but that took another name for obscure reasons. Persistence relies again on Banchoff’s definition of critical points, and I knew that this definition was incomplete. Moreover, since I knew that this notion actually came from Smale’s work, I believed that more insight on Morse–Smale theory would give a better definition and a more robust computation of persistence. I therefore worked back on my algorithm for geometric discrete Morse function. I clarified and simplified it, and the simplest formulation worked so well, that it seemed to be too simple to avoid basic problems, such as encompassing Banchoff’s definition critical points. These problems are addressed and solved in this thesis by small procedures around the core algorithm, although I observed that the core algorithm alone usually performs better, avoiding critical points that look more like noise and giving results that are more intuitive.

I

Introduction

The main object of both differential geometry and topology are smooth manifolds. However, these two fields have been studied almost separately until the early works of **Marton Morse** [Morse, 1925]. Since then, Morse theory has become a powerful tool to study the topology of smooth manifolds with differential geometry tools. This theory applied to various complex problems, from the Gauß-Bonnet theorem, the Poincaré-Hopf index theorem, the determination of the geodesic structure of a manifold, the Lefschetz singularity theory of hypersurfaces, to Milnor's exotic spheres, Yang-Mills theory on vector bundles, the geometry of Hamiltonian dynamical systems, Floer homology...

On one side, Morse theory describes part of the topology of a smooth manifold from a single scalar function defined on it, and in particular from the critical points of that function. On the other side, it gives a simple description of such a function from the topological decompositions it generates. This tool, mainly due to the work of **Stephen Smale** [Smale, 1960], allows using techniques from the whole topology to study the dynamics of smooth scalar functions, in particular algebraic and combinatorial topology. These extensions of Morse theory will then apply to a wider range of objects.



I.1(a): Height function $f(\mathbf{x}) = \mathbf{x}_z$ on a smooth torus. I.1(b): Height function on the faces of a discrete torus.

Figure I.1: The classical Morse function on a smooth torus: what would it be in the discrete setting?

This evolution encompasses the contributions of **Robin Forman** [Forman, 1995], who built an entirely discrete Morse theory. This theory studies the topology of discrete objects, among which are discrete manifolds, from the study of special functions (figure I.1). Contrarily to other attempts to formulate discrete versions of Morse theory, these functions are not easily described with differential geometry tools. However, this theory is the only discrete Morse theory that, to our knowledge, provides topological results, which is the main point of Morse theory.

The set of tools of differential geometry corresponds to the intuitive way of understanding an object in the space. However, with the evolution of geometric modelling by computer, these tools are at the same time necessary and difficult to transpose to the discrete setting used by these machines. Although the problem of sampling signals, i.e. defining their continuity, has been solved, the definition of derivatives on the discrete setting has not reached a consensus yet. Moreover, these notions are not properly defined on discrete geometric objects. In that perspective, discrete Morse theory is a hope to reproduce the link between differential topology and geometry starting from combinatorial topology. This work aims at contributing to this project by defining a discrete Morse function from a scalar function representing geometrical properties.

I.1 Main results

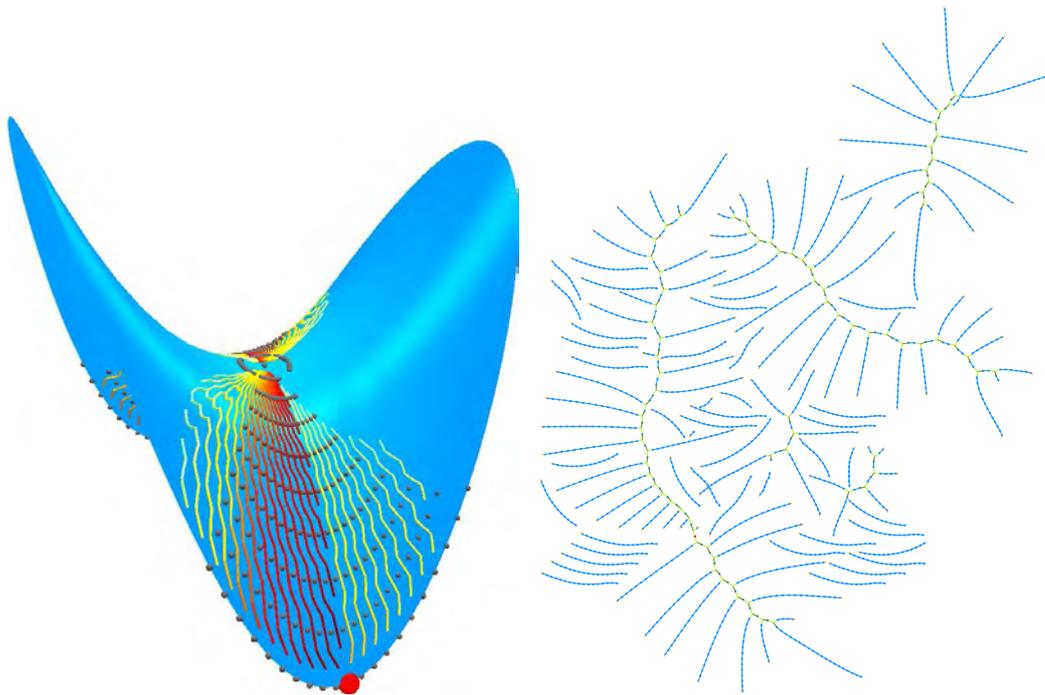
The presentation of this work emphasizes the construction of geometric discrete Morse decompositions. In order to describe this construction, each chapter introduces some of the concepts and results required. Most of these results are new and could be considered independently of the final construction as a contribution in itself. In particular, the extension of the hypergraph representation of discrete gradient [Lewiner, 2002] and its relations to the flow and the homology computation based on discrete Morse theory are promising new results.

(a) Flow and hypergraph components

Forman defines a combinatorial vector field on a cell complex as a matching of cells in [Forman, 1998]. We translated this definition in terms of *oriented hypergraphs* in [Lewiner, 2002], showing the layered structure it induced on the cell complex. When this vector field is a gradient, as defined in [Forman, 1995], the acyclicity of this gradient can be transposed on these hypergraphs, and in that case, each layer hypergraph is actually a *hyperforest*.

The structure of a hyperforest is relatively simple: each of its connected components is composed of connected regions containing only regular links, called *regular components*, which are connected with non-regular hyperlinks. Each of these regular components is a tree, and has at most one outgoing non-regular hyperlink.

In chapter III **Structure of a discrete Morse function**, we use these definitions to clarify the construction of discrete Morse functions and detail the corresponding algorithm (algorithm III.5). This core algorithm can be used either to build optimal discrete Morse functions, as in [Lewiner *et al.*, 2003b], or to build geometric Morse functions as in chapter V **Geometric discrete Morse complex**.



I.2(a): The flow lines from around the critical saddle, going towards the minima (red balls).

I.2(b): The flow on the layer structure, from the red dots towards the leaves of the hyperforest.

Figure I.2: The flow of a geometric discrete Morse function corresponding to $f(\mathbf{x}) = -\mathbf{x}_z$ on a saddle-shaped surface.

But the main benefit of this layered structure is the efficiency to compute the discrete flow, defined by Forman in [Forman, 1995] (figure I.2). This flow actually allows a direct mapping from the original complex to the Morse complex. This Morse complex can be seen in terms of Witten homology, as in chapter IV **Homology computation**, or as a Morse–Smale decomposition as in chapter V **Geometric discrete Morse complex**.

We show that the flow conquers each layer hyperforest from its roots towards its leaves. Roughly speaking, the invariant elements of the flow would

be the connected components of these hyperforests. However, this is not true in general, as in the discrete setting, gradient paths can merge and this merging can be destructive. Theorems 4 and 5 of section III.3(b) *Discrete stable and unstable basins* state that the regular component of a critical cell is part of the flow invariant chain, which is contained in the connected component.

(b) Complete homology computation

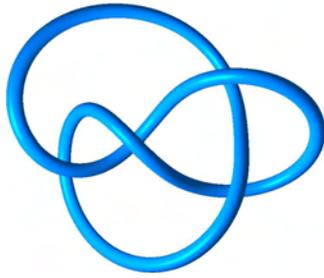
The flow behaves differently for a *primal*, *dual* or *critical* cell, following a classification close to [Forman, 1995]. Translated onto hypergraphs, this gives a very efficient way of computing the iterated flow, mapping the original complex onto the Witten–Morse complex. This complex has the same homology as the original one [Forman, 2002], but having much less elements. Therefore, the homology computation on that complex can be easily performed even by simple algorithms such as the Smith normal transform.

This idea is intensively used in chapter IV **Homology computation** to compute homology (figure I.3). Although the idea of simplifying the complex before computing homology is quite natural, this work is the first one, to our knowledge, to complete efficiently this task. In particular, we provide detailed algorithm to compute discrete Morse functions with a small number of critical cells (algorithm III.8), to compute the boundary operator on the resulting Witten–Morse complex (algorithm IV.1) using the properties of the flow we stated in section III.3 *Flow basins and hypergraph components*. Those algorithms are then used to compute the homology groups with torsion on any field (algorithm IV.2) in average almost linear time.

The flow actually maps the original complex and the Morse complex in both directions. We first use one way to compute a basis of cycle for the homology groups on the original complex (algorithm IV.3). In addition, we use the reverse direction to compute the decomposition of any cycle onto that basis (algorithm IV.4). This last step requires a pre-computation linear in the size of the complex, and then gives the decomposition in time linear in the size of the input cycle.

(c) Geometric discrete Morse functions and Morse–Smale decompositions

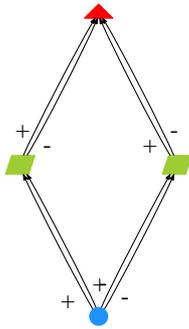
This efficiency in mapping the original complex to its Morse complex in both directions solves part of the Morse–Smale decomposition. It remains to construct the discrete Morse function. In the case of homology computation, the discrete Morse function should have the smallest possible number of critical



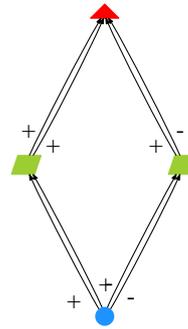
I.3(a): A knotted torus model with 27600 quadrangles.



I.3(b): A Klein bottle model with 8240 triangles.



I.3(c): The Morse complex of an optimal discrete gradient on the knotted torus has only 4 cells.



I.3(d): Idem for the Klein bottle, where the orientation of each cell marks the non-orientability.

Figure I.3: The homology can be efficiently computed on the Witten–Morse complex: for example on a knotted torus ($H_0 = \mathbb{Z}$, $H_1 = \mathbb{Z}^2$, $H_2 = \mathbb{Z}$) and a Klein bottle ($H_0 = \mathbb{Z}$, $H_1 = \mathbb{Z} \times \mathbb{Z}_2$, $H_2 = \mathbb{Z}_2$).

cells in order to accelerate the most expensive parts of the algorithms, mostly linear algebra. The case of general Morse decompositions is more delicate. Contrarily to smooth Morse theory, the definition of a discrete Morse function is not very intuitive, and there are very few examples where a discrete Morse function comes from another domain than discrete Morse theory itself.

The objective in chapter V **Geometric discrete Morse complex** is to define a discrete Morse function \mathfrak{f} that comes from a scalar function f defined on the vertices of the cell complex. This function will be called “geometry” in this work, although any scalar function could serve here. Since Morse–Smale decomposition is a discrete structure deduced from smooth properties of manifolds, it creates a link between smooth and discrete structure. In order to preserve this link, we will require our discrete Morse function \mathfrak{f} to have the same Morse–Smale decomposition as the smooth function f .

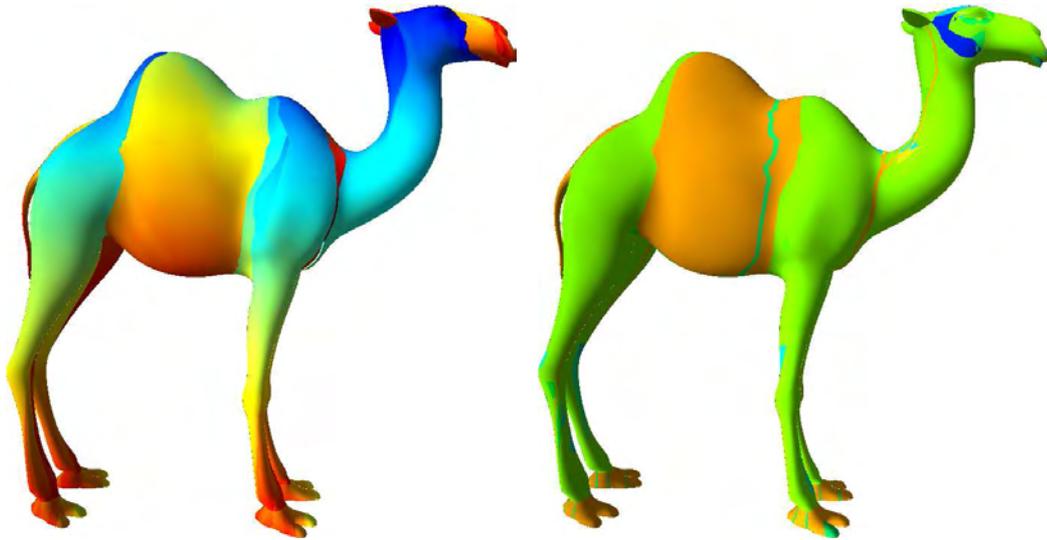


Figure I.4: A geometric discrete Morse function obtained with the only greedy algorithm, and the corresponding Morse–Smale decomposition.

Algorithm V.2 gives almost directly a constructive definition of such geometric discrete Morse function f (figure I.4). In particular, we prove that this definition achieves the desired Morse–Smale decomposition for the second barycentric subdivision of surfaces in section V.3 *Properties and proof of the algorithm*. However, if the cell complex is not adapted to the smooth function, this definition alone could miss some critical cells. This construction can then be complemented by an explicit critical cell selection (algorithm V.3) and a cancellation of unselected critical cells (algorithm V.4). This cancellation corresponds to the one used in Smale’s proof of the Poincaré’s conjecture in high dimension. Since the definition of persistence is another denomination for this technique, this construction gives an explicit and rigorous way of defining and computing persistence. We conclude this work by this application and by showing the relation between Morse–Smale decompositions and Reeb graphs, with an explicit algorithmic construction of those Reeb graphs on any cell complex.

I.2 Outline

This thesis is organized as follows. In chapter II **Morse theories**, we review the theoretical background that we use all along this work. Although all the notions are described there, the elements of smooth Morse theory are only mentioned, whereas their equivalent definitions in discrete Morse theory are more detailed.

Then, chapter III **Structure of a discrete Morse function** introduces the layer representation of a discrete Morse function, as a simplification

of [Lewiner, 2002]. The main algorithms to build discrete Morse functions are detailed there in a formulation that will be used in all the algorithms of this thesis. That chapter ends on the calculus of the flow on each layer, which will be a key element for the rest of this work.

Next, chapter IV **Homology computation** details our method to compute homology groups, generators and decompositions, which is a nice result in itself, but that will be used to define critical points in chapter V **Geometric discrete Morse complex**.

This last chapter proposes an algorithm to compute geometric discrete gradients directly derived from the algorithm of chapter III. The proof that this simple algorithm detects all critical points in the nice cases is given at the end of chapter V. However, in the general case, this algorithm can be completed by an explicit critical cell selection and cancellation in order to generate the only required critical cells. The corresponding algorithms are also part of chapter V, which ends on some direct applications of these constructions to Reeb graphs and persistence.

II

Morse theories

This chapter contains the fundamental notions that we will be using along this thesis. Since the matter of this work deals with Forman's discrete version of Morse theory, each concept will be introduced in the smooth setting (subsections (a)) and in its discrete version (subsections (b)). We assume that the reader is more familiar with smooth Morse theory than its discrete version. Therefore, the classical notions of topology are explained rather quickly, while discrete ones are defined and connected to the smooth ones with more details. This chapter is oriented towards the notion of Morse complexes. Further references on algebraic topology can be found in [Fomenko, 1987, Hatcher, 2002]. The classical description of smooth Morse theory relies in the first chapters of [Milnor, 1963], and the complete presentation of Forman's discrete Morse theory is detailed in [Forman, 1995].

Classical Morse theory deals with smooth functions on smooth manifolds, and connects the critical sets of those functions with the topology of their domain. Discrete Morse theory intends to provide a similar tool, although the functions to study are much less natural. Two clues will already give a better intuition on those notions. On one hand, the main information contained in a discrete Morse function is to be read in its gradient. Ideally, this gradient is aligned with the smooth gradient of a smooth Morse function. On the other hand, the smooth and discrete Morse functions coincide on both the Handle decomposition and Morse–Smale complex: those are discrete structures, namely cell complexes, which can also be described from the gradient of a discrete Morse function. The second construction is not straightforward from the smooth to the discrete case, and this is the main objective of this work.

II.1 Topological spaces

A topological space is a set of points X with a definition for the open subsets of X , usually called neighbourhoods. Two topological spaces X and Y are considered equivalent if there exist a homeomorphism between them, i.e. a continuous bijective function $f : X \rightarrow Y$ whose inverse is continuous.

This very general definition entails most of the classical spaces: subsets of \mathbb{R}^n , discrete spaces, subsets of functions... Morse theories apply on specific topological spaces, namely smooth manifolds for the classical one, and cell complexes for its discrete version. Formal definition of each one can be found in [do Carmo, 1976] and [Hatcher, 2002] respectively.

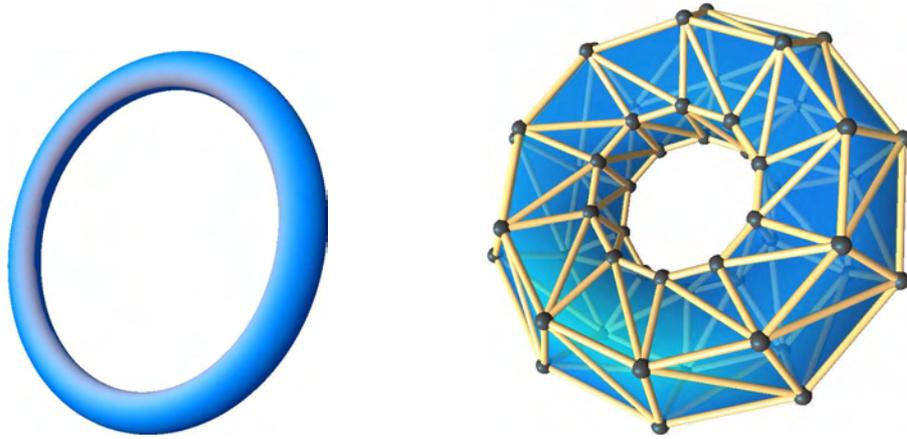


Figure II.1: A smooth and a discrete torus.

(a) Smooth manifolds

A *topological manifold* \mathcal{M} of dimension n is a topological space where the neighbourhood of each point is homeomorphic to \mathbb{R}^n , the dimension n being the same for all point of \mathcal{M} . This homeomorphism actually defines a local parameterization of \mathcal{M} . A manifold will be called *smooth* when this parameterization is smooth (i.e. many times differentiable) and when local parameterizations agree from neighbourhood of one point to the neighbourhood of close-by ones. For the smooth Morse theory, we will consider here only compact manifolds without boundary, such as the torus of figure II.1.

(b) Cell complexes



II.2(a): A 1-complex: collection of vertices $|K^0|$ and edges $|K^1| \setminus |K^0|$.
 II.2(b): Invalid topological space K if the intersections of edges are not cells of K

Figure II.2: Examples of a 1-complex and of an invalid discrete space.

A *cell* σ of dimension p is a topological space homeomorphic to the open ball $\mathbb{B}^p = \{\mathbf{x} \in \mathbb{R}^p : \|\mathbf{x}\| < 1\}$. The simplest example of p -cell is the interior of a p -simplex, which is the convex hull of $(p + 1)$ affine independent points in \mathbb{R}^p . A *cell complex* K is a collection of cells (figure II.2) that can be defined in two equivalent ways: by construction [Hatcher, 2002] or by decomposition [Cooke & Finney, 1967].

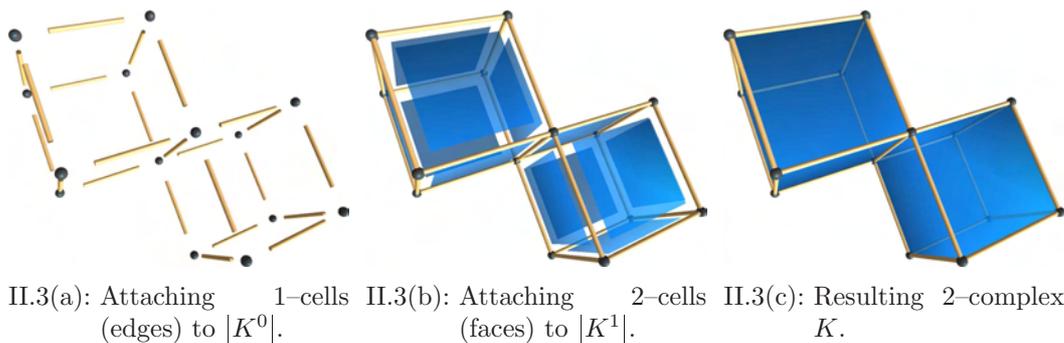


Figure II.3: Construction of a double cube as a 2-complex.

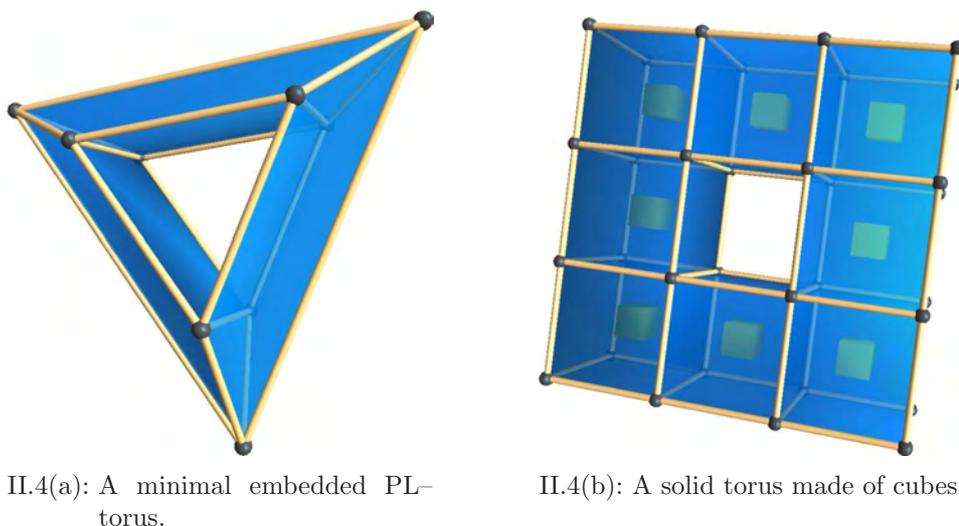


Figure II.4: Examples of a 2- and 3-complexes.

Construction. A cell complex is constructed by adding cells of increasing dimensions (figure II.3). Starting with a discrete set K^0 containing the 0-cells, we inductively attach p -cells to K^{p-1} to obtain K^p . Each cell σ^p is attached by identifying its geometric boundary (homeomorphic to the sphere \mathbb{S}^{p-1}) to a subset of K^{p-1} . This is not the case for example on figure II.2(b). The *geometric realization* $|K|$ of K is the geometric union of all its cells (figure II.4).

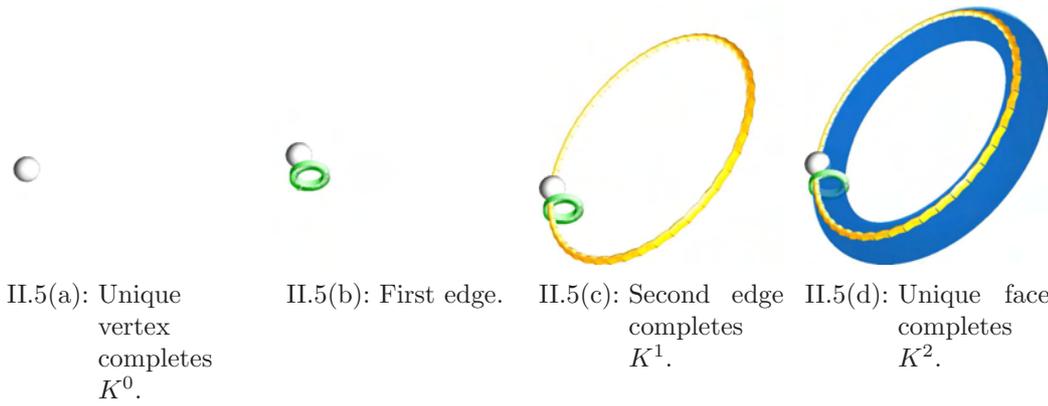


Figure II.5: Decomposition of a smooth torus into a cell complex: this decomposition is the same as the knotted torus of figure I.3(c).

Decomposition. A cell complex can also be defined by its decomposition $|K| = \bigcup |K^p|$, with $|K^p|$ being a closed subspace of $|K|$ and $|K^p|$ included in $|K^{p+1}|$ (figure II.5). This decomposition must satisfy the two following properties. First, the connected components σ_i of $|K^p| \setminus |K^{p-1}|$ are open sets of $|K^p|$: those are the cells. Second, there exists, for each cell σ_i , a homeomorphism $h_i : \mathbb{S}^{p-1} \rightarrow |\partial\sigma_i|$ from \mathbb{S}^{p-1} to the geometric boundary of σ_i that extends to a continuous map $\bar{h}_i : \mathbb{B}^p \rightarrow \sigma_i$ on \mathbb{B}^p . In both cases construction and decomposition, the topology of $|K|$ is the weak topology: a subset of $|K|$ is open if its intersection with each cell of K is open.

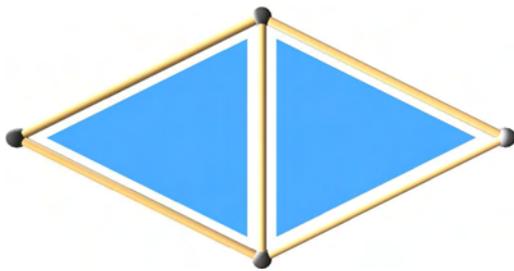


Figure II.6: Each triangle is incident to two external edges, and both are incident to the central edge.

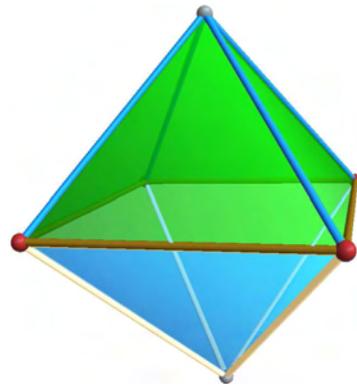


Figure II.7: The open star of the top vertex is made of the green triangles and the blue edges. Its link is made of the red vertices and the brown edges.

Incidences. From the definition above, the geometric boundary $|\partial\sigma|$ of a cell σ is the union of cells $|\tau_i|$ of lower dimension. Each cell τ_i is a *face* of σ and σ and τ_i are said to be *incident*, which is denoted $\tau_i < \sigma$ (figure II.6).

With this notation, we can write the *combinatorial boundary* as a formal sum of cells: $\partial\sigma = \sum_{\tau_i < \sigma} \tau_i$. The open star $\text{st}\tau$ of a cell τ is the set of all cells whose closure contains τ : $\text{st}\tau = \sum_{\tau < \sigma_i} \sigma_i$. The *star* is the closure of the open star (figure II.7). The *link* of τ is the set of cells belonging to the star but not to the open star: $\text{lk}\tau = \text{st}\tau \setminus \text{st}\tau$.

Regularity. In this work, we will consider only finite cell complexes, i.e. with a finite number of cells, and regular. In a regular cell complex, given two incident cells ρ and τ with $\dim(\tau) = \dim(\rho) - 2$, there exist exactly two cells σ_1 and σ_2 such that $\tau < \sigma_1 < \rho$ and $\tau < \sigma_2 < \rho$ (figure II.8).

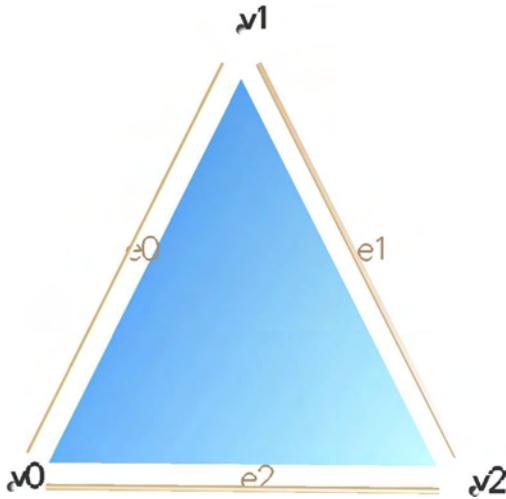


Figure II.8: Regularity of a complex made of only one triangle: there are exactly two edges e_i and e_{i-1} between the triangle and vertex v_i .

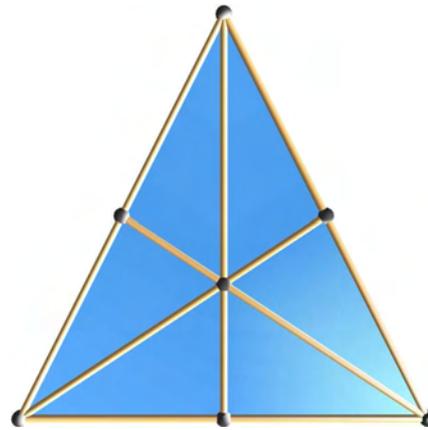


Figure II.9: Barycentric subdivision of a triangle.

Barycentric subdivision. If the geometric realization of each cell is convex, or if one can deform all those simultaneously to convex cells, it is possible to subdivide a cell complex into a finer, simplicial complex. For each sequence of cells $\sigma_0^{i_0} < \sigma_1^{i_1} < \dots < \sigma_k^{i_k}$ of increasing dimension ($0 \leq i_0 < i_1 < \dots < i_k \leq n$), each cell being a face of the next one, corresponds in the subdivided complex the k -simplex spanning $\mathbf{z}_0\mathbf{z}_1 \dots \mathbf{z}_k$, where \mathbf{z}_j is the barycentre of $\sigma_j^{i_j}$ (figure II.9). In particular, the classical proof of handle decomposition of piecewise-linear manifolds requires two successive barycentric subdivisions in order to separate handles properly [Rourke & Sanderson, 1972]. Similar considerations will be useful for the proof of our construction of the Morse complex.

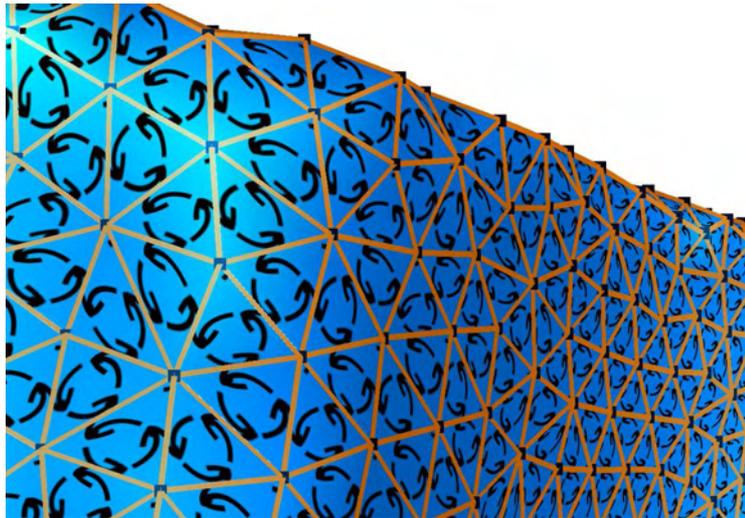


Figure II.10: Oriented surface: note that here, all the triangles have the same orientation, although this is not necessary for defining $[:]$.

Orientation. A cell complex K can be *oriented* defining $[:] : K \times K \rightarrow \{-1, 0, 1\}$ with the following three restrictions [Cooke & Finney, 1967]. First, $[\sigma : \tau] \neq 0$ if and only if τ is a face of σ and $\dim(\tau) = \dim(\sigma) - 1$. Second, if σ is an edge (1-cell) incident to the vertices (0-cells) τ_1 and τ_2 , then $[\sigma : \tau_1] + [\sigma : \tau_2] = 0$. Last, if ρ and τ are incident with $\dim(\tau) = \dim(\rho) - 2$, and σ_1 and σ_2 such that $\tau < \sigma_1 < \rho$ and $\tau < \sigma_2 < \rho$, then $[:]$ must satisfy $[\rho : \sigma_1] \cdot [\sigma_1 : \tau] + [\rho : \sigma_2] \cdot [\sigma_2 : \tau] = 0$. This orientation will be useful for defining the boundary operator and orienting the gradient vector field (figure II.10). For example on figure II.8, the orientation of the triangle on each edge can be 1, and $[e_i : v_i] = -1$, $[e_i : v_{i+1}] = +1$, $[e_i : v_{i+2}] = 0$.

A smooth manifold with a smooth Morse function can be decomposed into a cell complex where each cell corresponds to homogeneous parts of its gradient vector field, which is defined in the next section.

II.2 Vector fields

Until now, both smooth and discrete cases are intuitively coherent: a smooth manifold can be decomposed into a cell complex, and the geometric realization of a cell complex can be a manifold. However, the notions of smooth and discrete vector fields differ and they will coincide only when considering their flow. The main reason for this resides in the non-differentiability of discrete structures. In order to work with vector fields on combinatorial spaces, Forman actually interprets a cell as more than just a piece of its geometric realization: the dimension of a cell is related to the differential properties of its neighbourhood. Intuitively, a cell should correspond to a homogeneous part of

the vector field it supports. The fact that this vector field interpolates smoothly between the cell and a cell in its border will be represented by a matching between these cells.

(a) Tangent vector fields

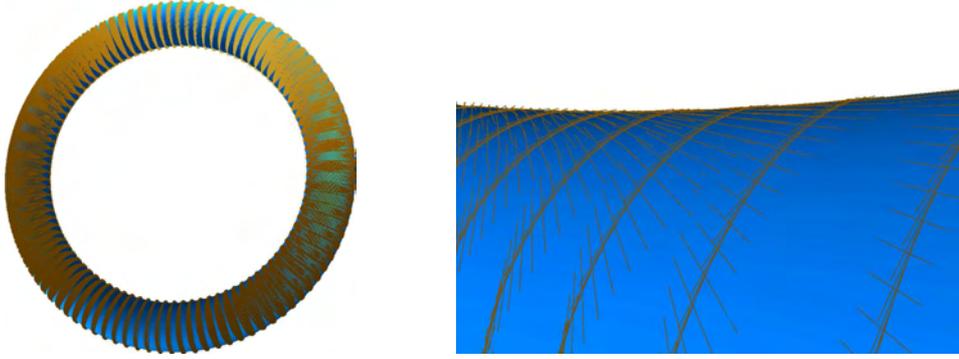


Figure II.11: The tangent manifold of a smooth torus: the lines represent tangent directions generating, at each point, the tangent plane.

Given a smooth curve $\gamma :]-1, 1[\rightarrow \mathcal{M}$ on a manifold \mathcal{M} , one can define the *tangent* of γ at $\gamma(0)$ as $\gamma'(0)$. The tangent space $T\mathcal{M}$ of \mathcal{M} is the collection of pairs $(\gamma(0), \gamma'(0))$ for all possible smooth curves γ (figure II.11). A *vector field* is a smooth cross section of this tangent space, i.e. a mapping each point \mathbf{x} of \mathcal{M} to a vector of $T\mathcal{M}$ tangent at \mathbf{x} [do Carmo, 1976].

(b) Cell matchings



II.12(a): Every cell is matched.

II.12(b): There are two vertices and two edges unmatched.

Figure II.12: Two combinatorial vector fields.

A *combinatorial vector field* \mathcal{V} is a collection $\{(\tau^p < \sigma^{p+1})\}$ of disjoint pairs of incident cells [Forman, 1995] (figure II.12). It can be defined as a function $\mathbf{V} : K \rightarrow K \cup \{0\}$ by $\mathbf{V}(\tau) = \pm\sigma$ and $\mathbf{V}(\sigma) = 0$. In particular, $\mathbf{V} \circ \mathbf{V} = 0$. If K is oriented, the sign of $\mathbf{V}(\tau)$ is determined by $[\mathbf{V}(\tau) : \tau] = -1$. The condition that the pairs are disjoint means that a cell can belong to at most one pair. If a cell σ does not belong to any pair, then $\mathbf{V}(\sigma) = 0$. This functional definition will be useful to formalize the flow. Conforming to [Forman, 1995],

we will represent a matching ($\tau^p < \sigma^{p+1}$) by an arrow from τ^p to σ^{p+1} . The disjointness condition implies that a cell can be the source or the destination of at most one arrow.

II.3 Morse functions

The key idea of Morse theory is to link the topology of a space to the critical sets of a scalar function defined on that space. The vector field to be considered will be the gradient of that function. This means that the vector field will not curl, and in the discrete setting, this implies that following the gradient one cannot go back to a previously visited cell.

(a) Smooth functions

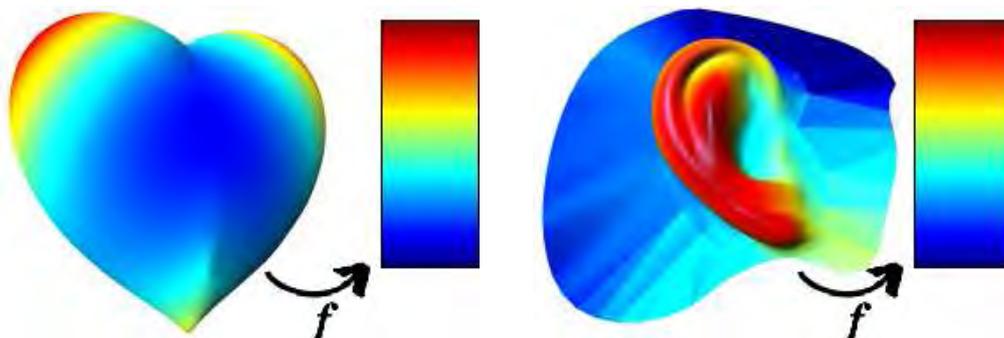


Figure II.13: Two smooth Morse functions: distance to the origin and projection on a given direction.

Given a smooth scalar function $f : \mathcal{M} \rightarrow \mathbb{R}$, its *gradient* ∇f is the vector field expressing the first derivatives of f . In a local parameterization of the manifold (x_i) around \mathbf{x} , the gradient can be written as $\frac{\partial f}{\partial x_i}$. An *integral curve* is the solution of $\dot{\mathbf{x}} = \nabla f(\mathbf{x})$. A point is *critical* for f if its gradient vanishes at that point, i.e. the end points of an integral curve. The *Hessian* $\text{Hess } f$ contains the second derivatives, and can be written via a local parameterization by the $n \times n$ matrix $\text{Hess } f = [\frac{\partial^2 f}{\partial x_i \partial x_j}]_{ij}$ [Milnor, 1963]. Then, f is an admissible Morse function if it has no degenerated critical points, which means that the Hessian matrix of f is invertible at any critical point. The classical examples of smooth Morse functions are the distance to a fixed point and the projection on a fixed direction (figure II.13). In the Morse–Smale decomposition, we will impose that the integral curves cross transversally, which is automatically satisfied by the following discrete formulation.

(b) Acyclic matchings

The discrete definition of a Morse function can be better understood as the integral of a discrete vector field. This integration requires the gradient to be acyclic [Forman, 1995]. In the smooth setting, this means that the integral curves are open.

Integral paths. An *integral path* for \mathcal{V} is a concatenation of steps $\blacktriangleleft \tau_i^p \sigma_i^{p+1} \tau_{i+1}^p \blacktriangleright$ where τ_i and τ_{i+1} are distinct faces of σ_i and $\mathbf{V}(\tau_i) = \pm \sigma_i$. Two steps $\blacktriangleleft \tau_i \sigma_i \tau_{i+1} \blacktriangleright$ and $\blacktriangleleft \tau_{i+1} \sigma_{i+1} \tau_{i+2} \blacktriangleright$ concatenate if the second one begins where the first one ends. Observe that an integral path contains cells of only two different dimensions. The *multiplicity* of a step is the product $\mu(\blacktriangleleft \tau_i \sigma_i \tau_{i+1} \blacktriangleright) = -[\sigma_i : \tau_i] \cdot [\sigma_i : \tau_{i+1}]$. The multiplicity of a path is the product of the multiplicities of its steps.

Discrete gradient. An integral path is *closed* when the last cell of the last step equals the first cell of its first step. For example, the vector field of figure II.12(a) has a closed integral path of length 4 on the left. A *discrete gradient vector field* is a combinatorial vector field with no closed integral path. For example, the vector field of figure II.12(a) is not a discrete gradient, while the one of figure II.12(b) is. This definition actually implies an ordering of the cells of an integral path. This ordering can be extended to the whole cell complex, which is the definition of a discrete Morse function.

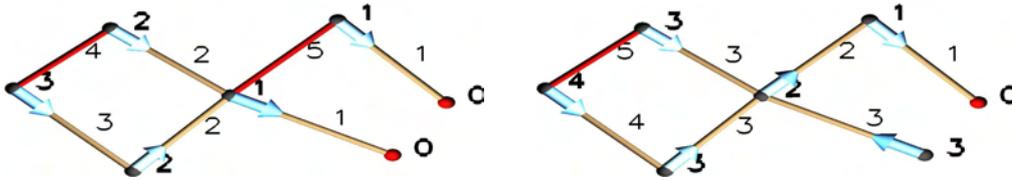


Figure II.14: Two discrete Morse functions. The first one corresponds to the discrete gradient of figure II.12(b).

Discrete Morse function. Formally, a *discrete Morse function* on a cell complex K is a real valued function $f : K \rightarrow \mathbb{R}$ satisfying:

$$\begin{cases} \forall \sigma^p \in K, & \text{card} \{ \rho^{p+1} > \sigma^p : f(\rho) \leq f(\sigma) \} \leq 1 & \text{and} \\ \forall \sigma^p \in K, & \text{card} \{ \tau^{p-1} < \sigma^p : f(\tau) \geq f(\sigma) \} \leq 1 \end{cases}$$

The corresponding discrete gradient \mathcal{V} is then the set of pairs $(\tau^p < \sigma^{p+1})$ satisfying $(f(\tau) \geq f(\sigma))$. From the regularity of the complex, both inequalities cannot be simultaneously equalities, which ensures the correctness of \mathcal{V} as a

discrete vector field (figure II.14). With the arrow representation of \mathcal{V} , the arrows are pointing from high values to low ones, therefore \mathcal{V} would rather correspond to $-\nabla f$ (figure II.15). This gradient contains the topological



Figure II.15: A discrete gradient and the corresponding discrete Morse function on a double cube.

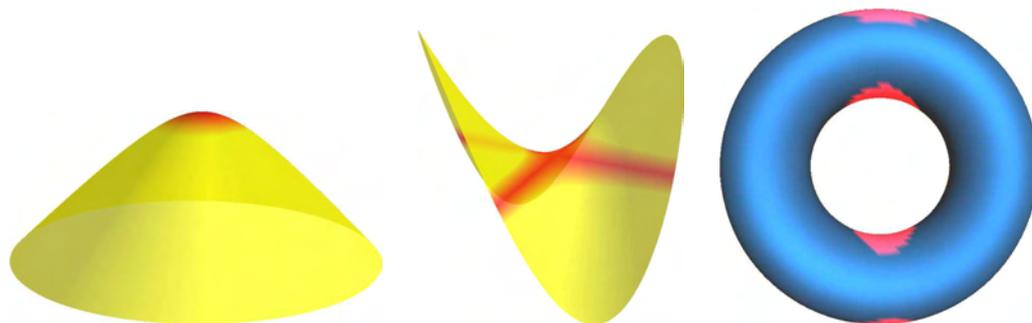
information of f and we will mainly work with the gradient in place of the function. However, the discrete Morse function contains additional information on relative heights of parts of the complex, since two different Morse functions have the same gradient only if they induce the same order inside each integral path.

II.4 Critical sets

The main interest for those Morse functions is their critical elements. Actually, the topology of the space controls those critical elements, in their nature and number, and the function reflects this. In particular, a smooth function on a complex topological space must have a complex geometry, i.e. many critical points. This nature of a critical element will be characterized in the smooth setting by the index, while it will directly be the dimension of the critical cell in the discrete setting.

(a) Critical points

As we said earlier, a point \mathbf{x} is *critical* for a smooth Morse function f if the gradient of f vanishes at that point [Milnor, 1963]. If f is a Morse function, the Hessian of f at \mathbf{x} has no zero eigenvalue. If the domain of f is a manifold of dimension n , the Hessian at \mathbf{x} will have p negative eigenvalues and $n - p$ positive ones. This integer p is called the *index* of \mathbf{x} . The index actually characterizes the critical point. For example, an index 0 means \mathbf{x} is a



II.16(a): Maximum (minimum for $-f$).

II.16(b): Saddle.

II.16(c): Critical points of a torus: one minimum, two saddles and one maximum.

Figure II.16: Critical points of the projection onto the vertical axis: the critical points are in the middle of the red region.

local minimum, an index n that \mathbf{x} is a local maximum. Critical points of index $0 < p < n$ will be called p -saddles (figure II.16).

(b) Critical cells

In the discrete setting, the definition still catches the idea of end-point of an integral path. A cell σ^p is critical for a discrete gradient \mathcal{V} if it does not belong to any pair of \mathcal{V} . The critical cells are drawn in red on figures II.12(b), II.14 and II.15. This definition can be translated in the functional formulation of the gradient: σ is critical if $\mathbf{V}(\sigma) = 0$ and $\sigma \notin \text{Im } \mathbf{V}$. It can be also written in terms of a discrete Morse function [Forman, 1995]:

$$\begin{cases} \text{card } \{\rho^{p+1} > \sigma^p : f(\rho) \leq f(\sigma)\} = 0 & \text{and} \\ \text{card } \{\tau^{p-1} < \sigma^p : f(\tau) \geq f(\sigma)\} = 0 \end{cases}$$

The classification for smooth critical point remains, defining the index of σ as its dimension.

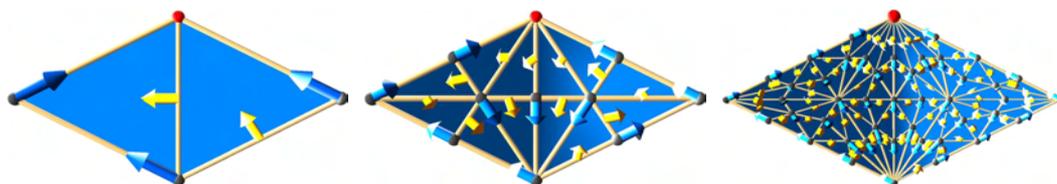


Figure II.17: Invariance under subdivision of the discrete gradient.

Discrete Morse functions have the nice property of invariance through subdivision: given a discrete Morse function on a complex K , it is easy to extend it on the barycentric subdivision of K , while preserving its critical cells

(figure II.17). This nice behavior with topological operations also encompasses the Cartesian product: given two discrete Morse functions on two complexes, we can construct a discrete Morse function on the Cartesian product of the complexes whose only critical cells are the products of the original critical cells [Lewiner, 2002] (figure II.18).

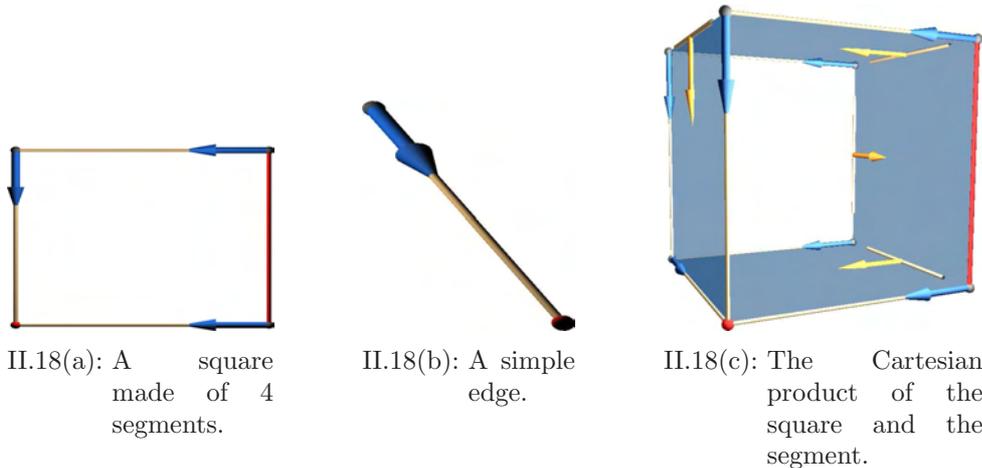


Figure II.18: The critical cells of the Cartesian product are the Cartesian products of critical cells.

We touched the deepest difference between smooth and discrete Morse theory. While a cell is *a priori* defined as a piece of the geometric realization of a complex, in Forman's theory a cell carries additional information on the local differential properties around that cell. In that perspective, a cell can be interpreted as a union of parts of adjacent integral curves of the gradient. That justifies why a minimum of the function is a vertex, since no integral curve goes out of a minimum for $-\nabla f$. Similarly, a maximum is a cell of full dimension as integral curves go out of it in all directions. Near a saddle, the integral curves go outward of the saddle for some directions, and inward for the others, which corresponds to cells of intermediate dimensions. This dimensionality becomes more precise in the handle decomposition described in the next section.

II.5 Topological properties

The relation between the critical set and the topology of the space will now become more precise. In the smooth setting, the main result of Morse theory states that a smooth manifold with a smooth Morse function f has the same homotopy type as a finite cell complex K such that each cell of dimension p of K is on one-to-one correspondence to a critical point of index p of f . This result is still valid in the discrete setting, using *simple homotopy* [Cohen, 1973] instead of homotopy. In this context, smooth manifolds can be decomposed into discrete structures, and tools of the smooth setting (such as the homotopy

type) and of the discrete setting (such as homology, Euler characteristic) both apply. Therefore, the two parts of this section are actually valid in both settings.

(a) Homotopy and Handle decomposition

The Morse inequalities can be easily deduced from the *handle decomposition* of a manifold through a smooth Morse function. A handle H^p of dimension n and index p is the Cartesian product of two balls: $H^p = \mathbb{B}^p \times \mathbb{B}^{n-p}$. Attaching a handle to a manifold by identifying $\mathbb{S}^{p-1} \times \mathbb{B}^{n-p} \subset H^p$ to a part of the manifold changes the topology of the manifold, and most of the topological changes can be interpreted as a handle attachment. This operation actually changes the *homotopy type*, defined as follows [Hatcher, 2002].

Homotopy type. Two topological spaces X and Y are homotopy equivalent if they can be continuously deformed one into the other. Formally, X and Y have the same homotopy type if there exists four continuous functions $f : X \rightarrow Y$, $g : Y \rightarrow X$, $h_X : X \times [0, 1] \rightarrow X$ and $h_Y : Y \times [0, 1] \rightarrow Y$ such that $h_X(\mathbf{x}, 0) = f \circ g(\mathbf{x})$, $h_X(\mathbf{x}, 1) = \mathbf{x}$ and $h_Y(\mathbf{y}, 0) = g \circ f(\mathbf{y})$, $h_Y(\mathbf{y}, 1) = \mathbf{y}$. For example each \approx symbol on figure II.19 corresponds to a homotopy equivalence, while there is no homotopy equivalence on the \rightarrow symbols.

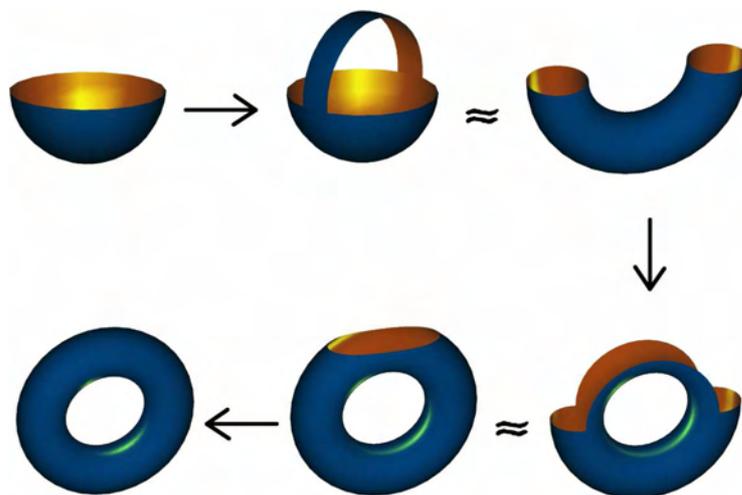


Figure II.19: Handlebody decomposition of a torus.

Handle decomposition. The Morse theorem states that there is no change in homotopy between cuts of a manifold below and above a level without critical point, and that the change in homotopy type between cuts below and above a level with only one critical point of index p corresponds to attaching

a handle of index p [Fomenko, 1987] (figure II.20). A cell complex $H^{\mathcal{M}}$ can then be constructed from a Morse function by successively attaching a handle for each critical point: a handle H^0 of index 0 for the absolute minimum f_0 of f , then a handle is attached to H^0 for the critical point with the smallest value $f_1 > f_0$ and so forth. . . A smooth manifold is thus homotopy equivalent to a finite cell complex such that to each critical point of index p corresponds one handle–cell of dimension p : this is the handle decomposition. This decomposition can also be described from the Smale decomposition introduced at the end of this chapter. The following tools of the discrete setting can then be applied to K via H^K , in particular the Morse inequalities.

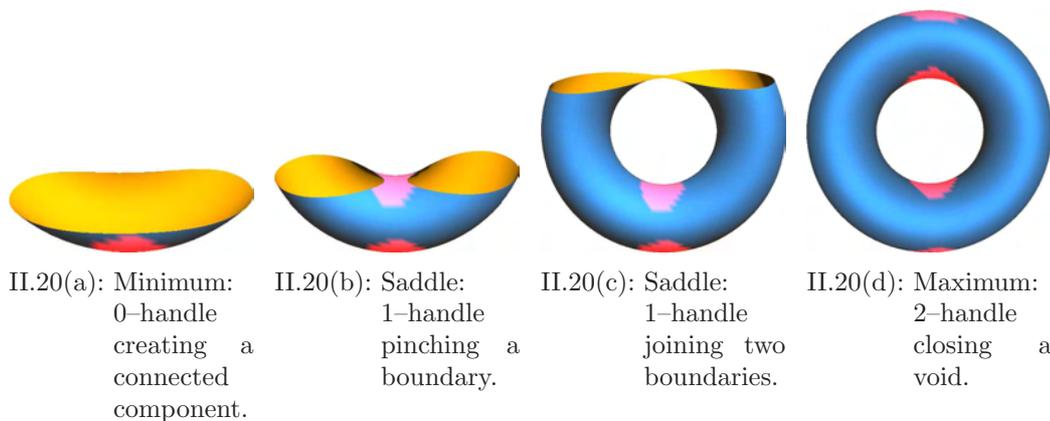
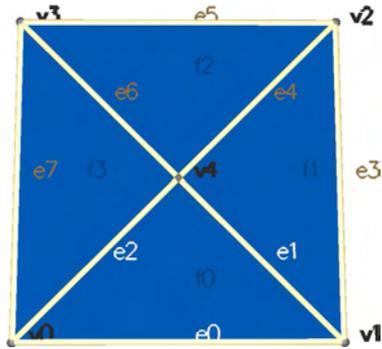


Figure II.20: Each one of the four critical points of a torus corresponds to a handle.

(b) Simple homotopy and Morse inequalities

The usual tools to characterize objects rely on invariants. For example, the homotopy type of a smooth manifold is a topological invariant, i.e. if two manifolds have different homotopy types, then they cannot be equal (homeomorphic). Whereas the handle decomposition $H^{\mathcal{M}}$ have the same homotopy type as \mathcal{M} , it is not in itself an invariant, since the same manifold with two different Morse functions will have two different handle decompositions. However, the handle decomposition of a manifold is a cell complex, on which invariants of algebraic topology apply, at least for characterizing the PL–topology of a manifold [Rourke & Sanderson, 1972]. In particular, topological invariants such as *singular homology* and the *Euler–Poincaré characteristic* can be related to the number of critical cells of any Morse function through the Morse inequalities, using the handle decomposition in the smooth setting, or *simple homotopy* in the discrete one.

Boundary operator. The objects considered by the homology are formal sums of cells having the same dimension called *chains* $\mathbf{c}_p = \sum_{\sigma^p \in K} c_\sigma \sigma^p$, c_σ being coefficients of a ring \mathbb{K} . The collection C_p of chains of dimension p is then a free module generated by the cells of dimension p . The central object of homology is the *boundary operator* $\partial_p : C_p \rightarrow C_{p-1}$, which is simply defined from the orientation $[\ :]$ of the complex: $\partial_p(\sigma^p) = \sum_{\tau^{p-1} \in K} [\sigma^p : \tau^{p-1}] \tau^{p-1}$ [Cooke & Finney, 1967]. For example, figure II.21 shows the boundary operator on a square made of 4 triangles. We can check for example that the boundary of the whole square is made of the four external edges: $\partial_2(f_0 + f_1 + f_2 + f_3) = e_0 + e_3 + e_5 - e_7$. The definition of $[\ :]$ implies directly that $\partial_p \circ \partial_{p+1} = 0$, i.e. $\text{Im } \partial_{p+1} \subseteq \text{ker } \partial_p$.



$$\begin{array}{l|l}
 \partial_2(f_0) = e_0 + e_1 - e_2 & \partial_1(e_0) = v_1 - v_0 \\
 \partial_2(f_1) = e_3 + e_4 - e_1 & \partial_1(e_1) = v_4 - v_1 \\
 \partial_2(f_2) = e_5 + e_6 - e_4 & \partial_1(e_2) = v_4 - v_0 \\
 \partial_2(f_3) = e_2 - e_6 - e_7 & \partial_1(e_3) = v_2 - v_1 \\
 \partial_0(v_0) = 0 & \partial_1(e_4) = v_4 - v_2 \\
 \partial_0(v_1) = 0 & \partial_1(e_5) = v_3 - v_2 \\
 \partial_0(v_2) = 0 & \partial_1(e_6) = v_4 - v_3 \\
 \partial_0(v_3) = 0 & \partial_1(e_7) = v_3 - v_0
 \end{array}$$

Figure II.21: The boundary operator on a small square model.

Homology. This can be written as an exact sequence [Hatcher, 2002]:

$$\{0\} \xleftarrow{\partial_0} C_0 \xleftarrow{\partial_1} C_1 \xleftarrow{\partial_2} C_2 \xleftarrow{\partial_3} \dots \xleftarrow{\partial_n} C_n \xleftarrow{\partial_{n+1}} \{0\}$$

The direction of the arrows in the diagram comes from the fact that the boundary operator decreases the dimension of the chains. The p -th homology group $H_p(K)$ is defined by $\text{ker } \partial_p / \text{Im } \partial_{p+1}$. Those groups are topological invariants: homeomorphic spaces have the same homology groups. Their ranks $\beta_p(K)$ are called the *Betti numbers* of K . For example on figure II.21, we have $\text{ker } \partial_2 = \{0\}$, $\text{Im } \partial_2 = \text{ker } \partial_1 \approx \mathbb{K}^4$, $\text{Im } \partial_1 \approx \mathbb{K}^4$, $\text{ker } \partial_0 = C_0 \approx \mathbb{K}^5$. We get $H_0 = \mathbb{K}$, $H_1 = H_2 = \{0\}$, i.e. $\beta_0 = 1$, $\beta_1 = \beta_2 = 0$.

Euler–Poincaré characteristic. Denoting by $\#_p$ the number of cells of dimension p of K , the *Euler–Poincaré characteristic* is the alternate sum of these quantities: $\chi(K) = \sum_{p \in \mathbb{Z}} (-1)^p \#_p(K)$. In that sense, it can be defined

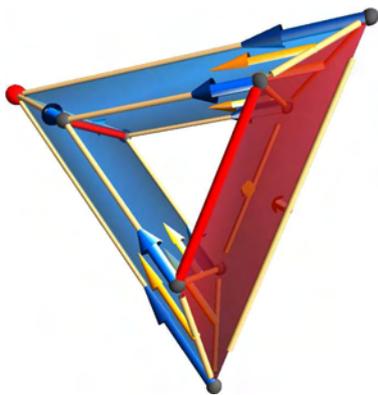
on the Handle decomposition of a smooth manifold. The Euler–Poincaré characteristic is also a topological invariant, since it can be written in terms of the Betti numbers: $\chi(K) = \sum_{p \in \mathbb{Z}} (-1)^p \beta_p(K)$. For example on figure II.21 we had $\chi = \# 5 - 8 + 4 = \beta 1 - 0 + 0 = 1$.



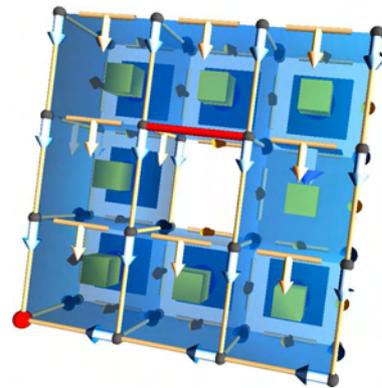
Figure II.22: Collapse of a heptagon: collapses do not alter the Euler characteristic.

Simple homotopy Homotopy of cell complexes is usually defined over their geometric realization. However, a weaker version of homotopy equivalence, called *simple homotopy*, can be defined combinatorially by successive collapses and extensions [Cohen, 1973]. If $\tau^{p-1} < \sigma^p$ are two cells of a cell complex K and τ is not the face of any other cell of K , then K collapses onto $K \setminus \{\tau, \sigma\}$ (figure II.22). The reverse operation of a collapse is called an *extension*. If one complex can be obtained from K by a sequence of collapses and extensions, it is said to have the same simple homotopy type of K .

Discrete decomposition. The main theorem of Forman’s discrete Morse theory states that a cell complex with a discrete Morse function f is simple homotopy equivalent to another cell complex having exactly one cell of



II.23(a): Simple torus, with 1 critical vertex, 2 critical edges and 1 critical face.



II.23(b): Solid torus with 1 critical vertex and 1 critical edge.

Figure II.23: Optimal Morse functions characterize the complex, for example to distinguish between a toric surface and a solid torus.

dimension p for each critical cell of f of dimension p [Forman, 1995]. For example, figure II.5 shows the discrete decomposition of a torus with only four critical points, which differs from the decomposition of a solid torus (figure II.23). This theorem actually puts, to our knowledge, Forman's Morse theory as the only discrete version of Morse theory that proves this homotopy equivalence.

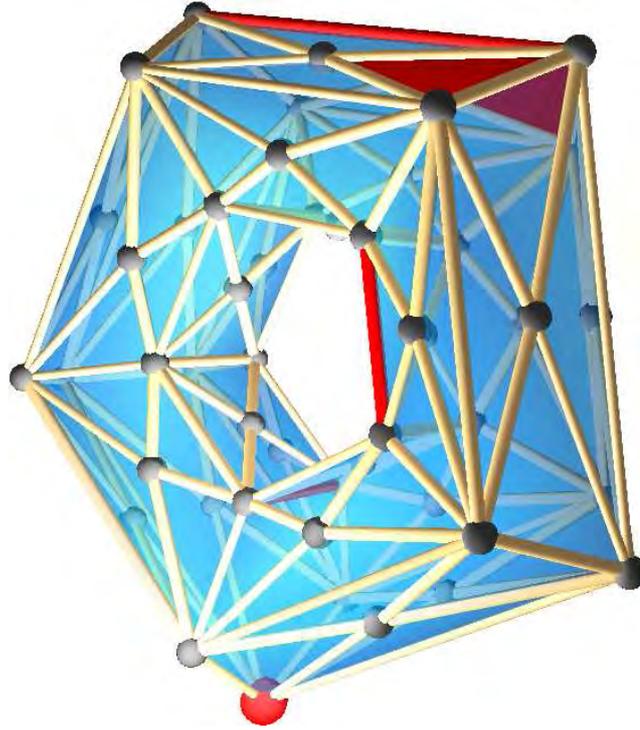


Figure II.24: A non-optimal discrete Morse function on a torus: $m_0 = 1 = \beta_0$, $m_1 = 3 \geq 2 = \beta_1$ and $m_2 = 2 \geq 1 = \beta_2$.

Morse inequalities. The number of p -cells in the decomposition of K by a Morse function f is the number of critical elements of index p , which we will denote $m_p(f)$. Since the Euler–Poincaré characteristic is a topological invariant, the characteristic of this decomposition is the same as the characteristic of K , and can thus be written: $\chi(\mathcal{M}) = \sum_{p \in \mathbb{Z}} (-1)^p m_p(f) = \sum_{p \in \mathbb{Z}} (-1)^p \beta_p$. This equality is weak, since it can be deduced from a stronger set of inequalities, called the Morse inequalities: $\sum_{p \leq k} (-1)^{k-p} \beta_p \leq \sum_{p \leq k} (-1)^{k-p} m_p(f)$. By summing these inequalities, we get $\beta_p \leq m_p(f)$. There are two obstructions for these inequalities to become equalities: either the Morse function is not optimal (figure II.24), the worst case being all cells critical ($f(\sigma^p) = p$, $\mathcal{V} = \emptyset$ and $\mathbf{V} \equiv 0$), or the cell complex has some finer topological characteristics that homology does not detect, for example homotopy-only features or collapsibility [Crowley *et al.*, 2005].

II.6 Cancellations

Morse functions are related to the topology of a space, but still in a weak manner. A given Morse function provides only an upper bound to the complexity of the topology, and a complex Morse function does not imply that the topology is not trivial. In particular, it is easy to produce Morse function critical everywhere (set $f(\sigma) = \dim \sigma$, $\mathcal{V} = \emptyset$ or $\mathbf{V} \equiv 0$ for the discrete case), which does not say much about topology. However, the minimum possible number of critical elements gives a better characterization of the topology. In the simple cases, this minimum corresponds to the Betti number of the space. Moreover, we proved in [Lewiner, 2002] that it is a topological invariant for cell complexes whose realization is a 2- or a 3-manifold. In order to reach that minimum, a possible strategy is to compute a reasonable Morse function and then to cancel pairs of critical elements.

(a) Inversion

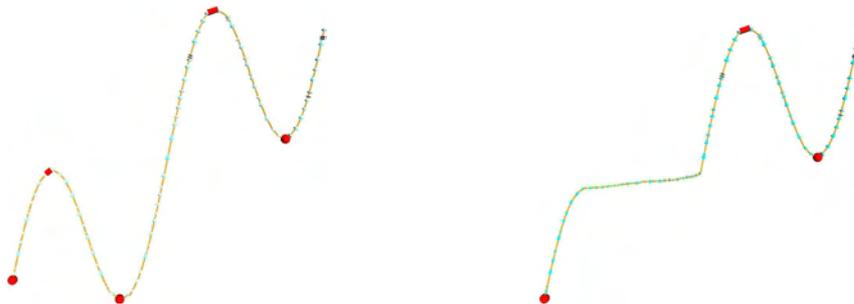


Figure II.25: Inversion of a smooth gradient path on the left side of the complex.

Given a smooth Morse function f , two critical points \mathbf{x} and \mathbf{y} of respective index p and $p - 1$ can be cancelled if the integral curves pointing to \mathbf{y} and the integral curves going out of \mathbf{x} meet transversally at exactly one point \mathbf{z} . In that case, there is a new Morse function f' for which \mathbf{x} and \mathbf{y} are no more critical. Moreover, f' coincides with f except on an arbitrary small neighbourhood of the integral curve \mathbf{xzy} [Fomenko, 1987] (figure II.25). This cancellation is performed by reversing the sign of the gradient on the integral curve, and interpolating on neighbourhood to preserve the smoothness of the gradient. This property has been extensively used for the demonstration of the Poincaré's conjecture in high dimensions.

(b) Unique gradient path

The big picture is the same in the discrete setting, although the construction is much simpler. For a given discrete gradient \mathcal{V} and two critical cells σ^p

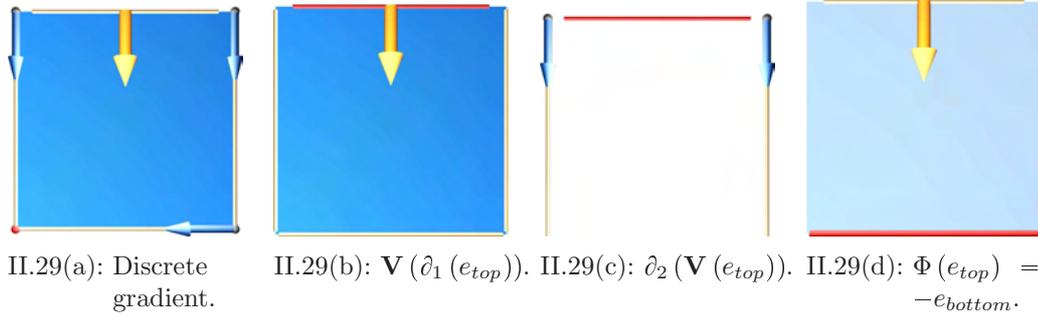
maps the state $\phi(\mathbf{x}, t)$ obtained at time t respecting $\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = \nabla f(\phi(\mathbf{x}, t))$. We get $\phi(\phi(\mathbf{x}, t), t') = \phi(\mathbf{x}, t + t')$, and an integral curve passing through \mathbf{x} can be parameterized by the flow as $\{\phi(\mathbf{x}, t), t \in \mathbb{R}\}$.

The *stable basin* $W^s(\mathbf{x})$ of critical point \mathbf{x} is the set of all points $\mathbf{y} \in \mathcal{M}$ such that $\phi(\mathbf{y}, t) \xrightarrow{t \rightarrow +\infty} \mathbf{x}$. Similarly, the *unstable basin* $W^u(\mathbf{x})$ of critical point \mathbf{x} is the set of all points $\mathbf{y} \in \mathcal{M}$ such that $\phi(\mathbf{y}, t) \xrightarrow{t \rightarrow -\infty} \mathbf{x}$. Actually, the stable and unstable basins are open manifolds. Moreover, if f is a Morse–Smale function, i.e. if its integral curves meet transversally, the intersections of those basins are topological balls.

(b) Invariant chains



Figure II.28: The flow of a vertex is the vertex pointed by the gradient: $\Phi(v_0) = v_0 + \partial_1(\mathbf{V}(v_0)) = v_0 + \partial_1(e) = v_0 + (v_1 - v_0) = v_1$.



II.29(a): Discrete gradient. II.29(b): $\mathbf{V}(\partial_1(e_{top}))$. II.29(c): $\partial_2(\mathbf{V}(e_{top}))$. II.29(d): $\Phi(e_{top}) = -e_{bottom}$.

Figure II.29: Combinatorial definition of the flow, the square edges being counter-clockwise oriented: $\Phi(e_{top}) = e_{top} + \mathbf{V}(\partial_1(e_{top})) + \partial_2(\mathbf{V}(e_{top})) = e_{top} + \mathbf{V}(v_{left} - v_{right}) + \partial_2(-face) = e_{top} + ((-e_{left}) - e_{right}) - (e_{top} + e_{left} + e_{right} + e_{bottom}) = -e_{bottom}$.

The idea of the flow is to go along with the gradient. Therefore, the flow of a vertex should be the vertex pointed by the gradient (figure II.28). This can be defined in a combinatorial way by $\Phi(\sigma^0) = \sigma^0 + \partial_1(\mathbf{V}(\sigma^0))$. One can imagine the discrete flow of a cell σ^p of higher dimension as the collection of adjacent cells that contain an integral curve passing through σ . Its image is thus a p -chain. The above definition can then be extended for general cells: $\Phi(\sigma^p) = \sigma^p + \mathbf{V}(\partial_p(\sigma^p)) + \partial_{p+1}(\mathbf{V}(\sigma^p))$ (figure II.29). The flow then extends to the chain modules: $\Phi : C_p \rightarrow C_p$ with $\Phi = \text{Id} + \mathbf{V}\partial_p + \partial_{p+1}\mathbf{V}$. Observe that the flow commutes with the boundary operator [Forman, 1995], and this announces the fact that the flow preserves the homology.

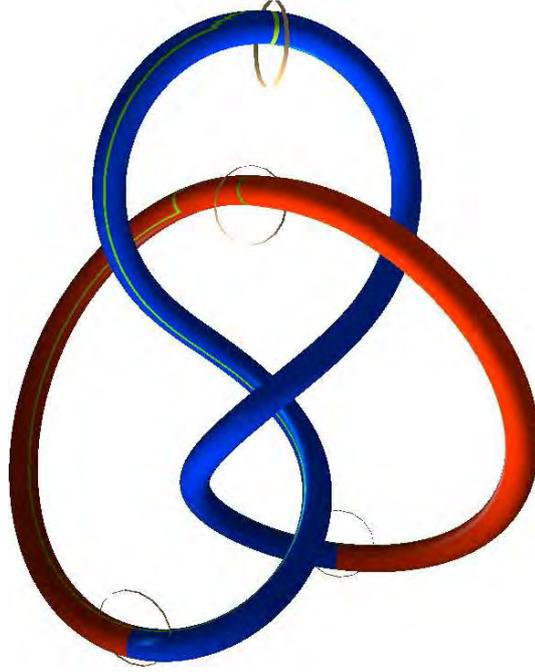


Figure II.30: Invariant chains for the flow on a geometric discrete Morse function: the saddles are outlined.

This flow can be iterated, and by the gradient structure of \mathbf{V} (that does not curl, and therefore has no closed orbit) this iterated flow will have some fixed point: the *invariant chains* $\Phi^\infty(c) = c$ (figure II.30). Observe that those invariant chains correspond to the unstable manifold, while the stable manifold is not directly considered since there is no direct definition of the flow inverse on K (but on the algebraic dual of K [Forman, 2002]). Moreover, there is a one-to-one correspondence between the set C_p^Φ of invariant chains of dimension p and the free module M_p generated by the critical cells of dimension p [Forman, 1995]. More precisely, M_p and C_p^Φ are isomorphic: $M_p \xrightarrow{\Phi^\infty} C_p^\Phi$, and this isomorphism is the building block of the Morse complex.

II.8 Morse complexes

The purpose of this work is to build Morse complexes and to apply it to two specific cases: the Morse–Smale decomposition, which is the decomposition into stable and unstable manifolds, and the Witten-Morse homology, which will lead to fast computation of the homology group and efficient decomposition of cycles into its generators.

(a) Smale decomposition

Given a Morse–Smale function f , the Morse–Smale decomposition of \mathcal{M} by f is a cell complex K whose geometric realization is $|K| = \mathcal{M}$ [Smale, 1960,

commute [Forman, 1995]:

$$\begin{array}{cccccccccccc}
 \{0\} & \xleftarrow{\partial_0} & C_0 & \xleftarrow{\partial_1} & C_1 & \xleftarrow{\partial_2} & C_2 & \xleftarrow{\partial_3} & \dots & \xleftarrow{\partial_n} & C_n & \xleftarrow{\partial_{n+1}} & \{0\} \\
 & & \Phi^\infty \downarrow & \uparrow \text{Id} & \Phi^\infty \downarrow & \uparrow \text{Id} & \Phi^\infty \downarrow & \uparrow \text{Id} & & & \Phi^\infty \downarrow & \uparrow \text{Id} & \\
 \{0\} & \xleftarrow{\tilde{\partial}_0} & C_0^\Phi & \xleftarrow{\tilde{\partial}_1} & C_1^\Phi & \xleftarrow{\tilde{\partial}_2} & C_2^\Phi & \xleftarrow{\tilde{\partial}_3} & \dots & \xleftarrow{\tilde{\partial}_n} & C_n^\Phi & \xleftarrow{\tilde{\partial}_{n+1}} & \{0\}
 \end{array}$$

Then, the isomorphism $M_p \xrightarrow{\Phi^\infty} C_p^\Phi$ extends this boundary operator $\tilde{\partial}_p$ to M_p . This gives a new chain complex:

$$\{0\} \xleftarrow{\tilde{\partial}_0} M_0 \xleftarrow{\tilde{\partial}_1} M_1 \xleftarrow{\tilde{\partial}_2} M_2 \xleftarrow{\tilde{\partial}_3} \dots \xleftarrow{\tilde{\partial}_n} M_n \xleftarrow{\tilde{\partial}_{n+1}} \{0\}$$

This complex actually has the same homology as the original complex, but contains much fewer cells (only the critical ones). For example on figure I.3 and figure II.32, the homology can be computed using only the 4 cells of the Morse complex and the boundary operator represented on the diagrams. This leads to efficient computation of homology groups, if we can compute $\tilde{\partial}$. Forman proved that the boundary of a cell σ of M_p is a formal sum of cells τ of M_{p-1} , where the coefficient of τ is the sum of the multiplicity of all gradient paths from a face of σ to τ [Forman, 1995]. The collection of these gradient paths can be easily formalized in terms of layers and graphs, as detailed in the next chapter.

III

Structure of a discrete Morse function

Before constructing a discrete Morse complex from a geometrical function, we will need some detailed notions about the structure of a discrete Morse function. We saw already two different representations for discrete Morse structures: a function $f : K \rightarrow \mathbb{R}$ and a discrete gradient \mathcal{V} which is an acyclic matching. We will now detail a third one in terms of graphs, that was first introduced in [Lewiner, 2002].

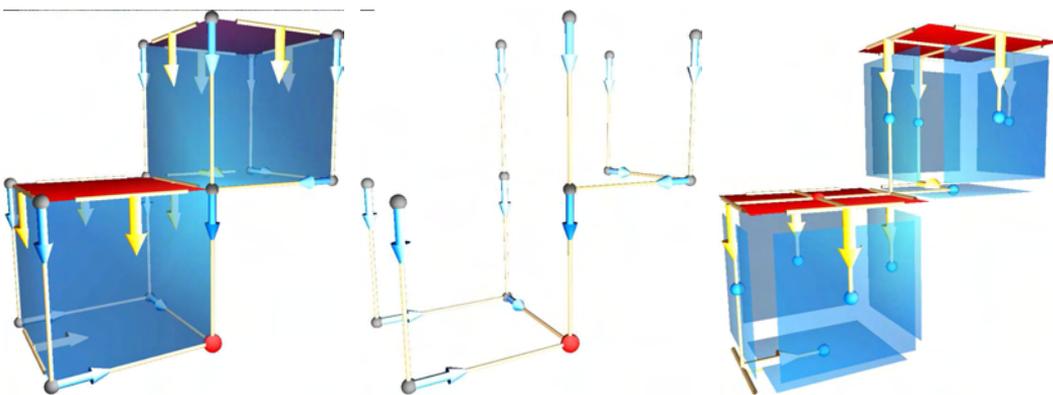


Figure III.1: The discrete gradient of figure II.15, decomposed on the primal layer \mathcal{L}_{01} and the dual layer \mathcal{L}_{21}

The first section of this chapter is a simplified (and clarified) presentation of some results of [Lewiner, 2002], namely the layered structure of a Morse function and the corresponding hypergraph. The second section presents an adaptable algorithmic construction of Morse functions, from which the optimal constructions of [Lewiner, 2002, Lewiner *et al.*, 2003b, Lewiner *et al.*, 2004] can be described concisely. The last section is a simple evaluation of the flow on this graph structure. On one hand, this construction identifies graph connectivity properties with the stable parts of the flow, enforcing the graph point-of-view. On the other hand, this calculus can be written using equations (3), (5) and (9), which allows efficient computations of the flow. This section also introduces simple algorithms and data structures that will be the building blocks of our main constructions. They are purposely mixed with the text to

show their proximity with the theory. Since these algorithms are efficient, it will enforce the idea that Forman's discrete Morse theory is very well suited for the discrete applications.

III.1 Layers and hypergraphs

A finite cell complex K can be seen as a graph \mathcal{H} , called the *Hasse diagram*, that represents explicitly every incidence relation inside K . A discrete gradient \mathcal{V} corresponds to a subgraph of \mathcal{H} that has a particular layered structure that we will describe now. We will see that, for a given discrete gradient, each pair of dimensions $(p, p + 1)$ or $(p, p - 1)$ corresponds to the hypergraph $\mathcal{L}_{p(p+1)}$ or $\mathcal{L}_{p(p-1)}$ (figure III.1). Actually, the converse is also true: there is a one-to-one correspondence between discrete gradients and particular collections of layer hypergraphs. A particular layer represents the incidence relation and the gradient of certain class of cells that we will introduce now.

(a) Cell classification

Once a combinatorial vector field \mathcal{V} has been defined on a cell complex K , the cells of K are naturally classified between the image and the kernel of \mathbf{V} , observing that $\mathbf{V} \circ \mathbf{V} = 0$, i.e. $\text{Im } \mathbf{V} \subset \ker \mathbf{V}$. The first class of cells is then the set Crit_p of *critical cells* of dimension p : $\text{Crit}_p(\mathcal{V}) = \{\sigma^p \in \ker \mathbf{V} \setminus \text{Im } \mathbf{V}\}$ (section II.4(b) *Critical cells*). Recall that each regular (i.e. non-critical) p -cell σ^p belongs to a pair of \mathcal{V} (section II.2(b) *Cell matchings*). Therefore σ^p is either the tail of an arrow ($\sigma^p < \rho^{p+1}$) or the head of an arrow ($\tau^{p-1} < \sigma^p$) (figure III.2). In the first case, σ will be designated as *primal*, and *dual* in the last case. This can be expressed in terms of \mathbf{V} : $\text{Prim}_p(\mathcal{V}) = \{\sigma^p \notin \ker \mathbf{V}\}$ and $\text{Dual}_p(\mathcal{V}) = \{\sigma^p \in \text{Im } \mathcal{V}\}$.

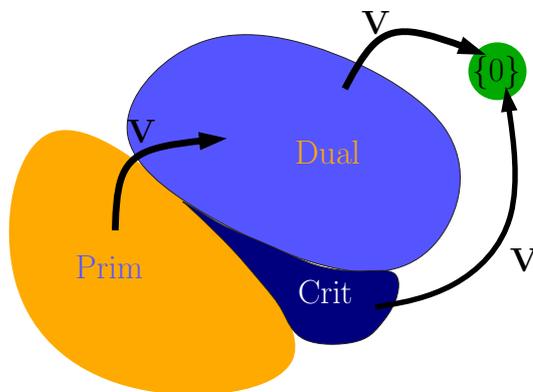


Figure III.2: \mathbf{V} operator on each class of cells.

This cell classification corresponds to the fact that each gradient path

contains cells of only two dimensions. It has been also used in [Forman, 1995], but for flow computation, and in [Lewiner, 2002] for the construction optimal discrete Morse functions. This construction is based on hypergraphs, whose structure actually represents in a simple way the iterated flow Φ^∞ of \mathcal{V} .

(b) Layer hypergraph

Forman's definition of an integral path (section II.3(b) *Acyclic matchings*) $\langle \tau_0^p \sigma_0^{p+1} \tau_1^p \sigma_1^{p+1} \dots \sigma_{r-1}^{p+1} \tau_r^p \rangle$ forces cells σ_i and τ_i to belong to only two consecutive dimensions p and $p+1$. All these paths between dimensions p and $p+1$ can be represented by a graph, which we called the *layer hypergraph* $\mathcal{L}_{p(p+1)}$ of \mathcal{V} . The nodes of this graph are the primal and critical cells of dimension p , and its links are the dual cells of dimension $p+1$. A link is incident to a node if the cell represented by the link is incident to the cell represented by the node.

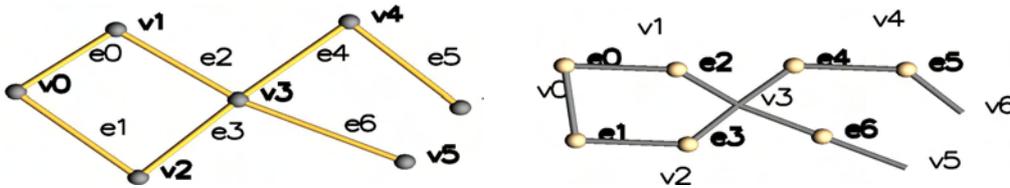


Figure III.3: The dual of a graph, obtained by inverting vertices (nodes) and edges (links), is not generally a graph but a hypergraph: for example v_3 is not a regular link.

Hypergraph. As we can observe, this graph is not a regular graph, as links can be incident to only one node (for example a gradient step $\langle \tau_0 \sigma_0 \tau_1 \rangle$ with $\tau_1 = \tau_0$) or to more than 2 nodes. In the last case, the link will be called a *hyperlink*, and the graph a *hypergraph* (figure III.3). Formally, a hypergraph is a set of nodes and links, where the links are collections of nodes [Berge, 1970]. Observe that a link can be incident more than once to the same node. We will denote a hypergraph by $\text{hg}(Nodes, Links, incidence)$, where $incidence : Links \times Nodes \rightarrow \mathbb{Z}$ indicates the incidence of a link to a node.

Orientation. This representation of hypergraph indicates the orientation $[:]$ of K by the sign of $incidence = [:]$. However, this representation still does not describe completely the discrete vector field, for it does not indicate which node is matched with a given link. This information can be included by considering *oriented hypergraphs*: $\vec{\text{hg}}(Nodes, Links, incidence, orientation)$.

Formally, a hypergraph is oriented by choosing for each link one *source-node* such that a node can be the source of at most one link (figure III.4). This definition is more restrictive than the one of [Berge, 1970], and corresponds to the definition of a discrete Morse function (section II.3(b) *Acyclic matchings*).

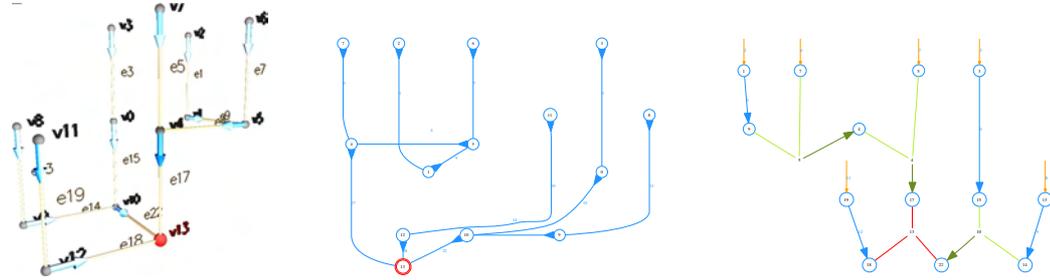


Figure III.4: First layers \mathcal{L}_{01} and \mathcal{L}_{10} of the discrete gradient of figure II.15

Primal and dual layers. The layer hypergraph $\mathcal{L}_{p(p+1)}$ of \mathcal{V} can be written as $\mathcal{L}_{p(p+1)} = \vec{\text{hg}}(\text{Prim}_p \cup \text{Crit}_p, \text{Dual}_{p+1}, <, \mathcal{V})$ (figures III.4 and III.5). As its nodes are the primal nodes of \mathcal{V} , it will be called a *primal layer hypergraph*. We can construct its dual, in the usual meaning of graph theory [Berge, 1970] by reversing the role of the links and nodes (figure III.3). Then removing the critical links and adding the critical $(p + 1)$ -cells (now nodes), we obtain the *dual layer hypergraph*: $\mathcal{L}_{(p+1)p} = \vec{\text{hg}}(\text{Dual}_{p+1} \cup \text{Crit}_{p+1}, \text{Prim}_p, >, \mathcal{V})$.

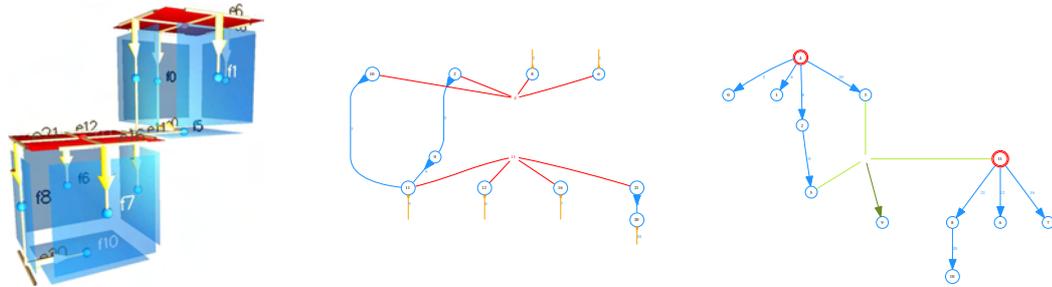


Figure III.5: Last layers \mathcal{L}_{12} and \mathcal{L}_{21} of the discrete gradient of figure II.15

(c) Hyperforest

The above definition of layer hypergraph is valid for any combinatorial vector field. The acyclicity of the discrete gradient corresponds directly to the acyclicity of the layer hypergraphs. As the general vector fields correspond to oriented hypergraphs, gradient vector fields correspond to *hyperforest*.

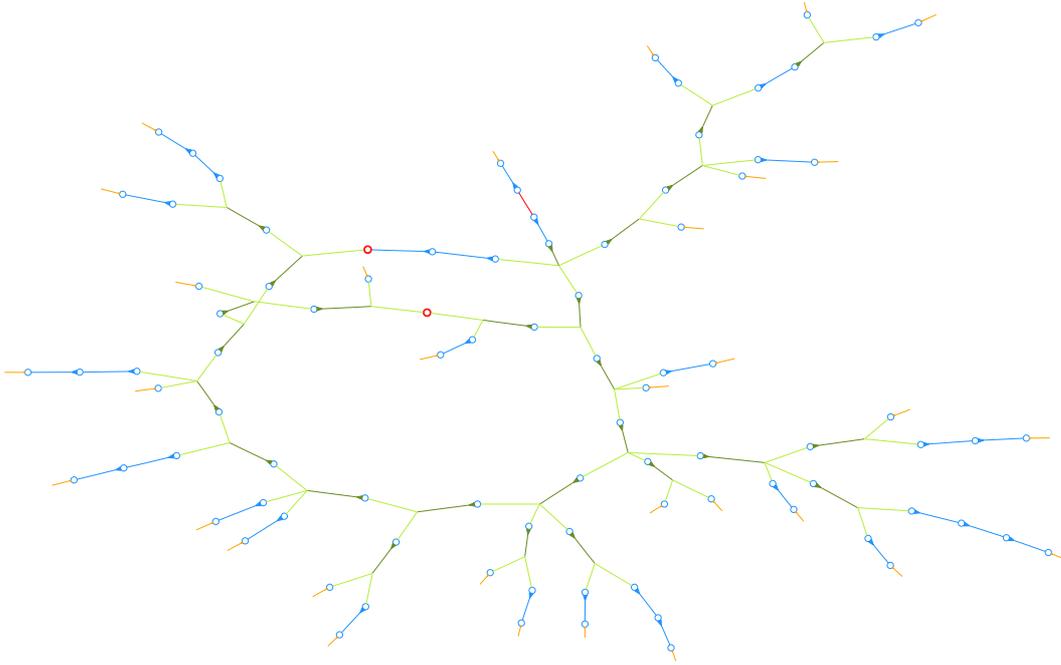


Figure III.6: Layer \mathcal{L}_{23} of an optimal Morse function on Poincaré's homological sphere: the oriented hypergraph is acyclic, although without orientation it contains cycles.

Acyclicity. We defined the layer of a discrete vector field as a representation of its integral paths. We can formalize this notion by defining a *hyperpath* $\langle n_0 l_0 n_1 l_1 \dots l_{r-1} n_r \rangle$ as a sequence of an initial node n_0 , distinct step nodes n_i and links l_i where n_i is the source of l_i and l_i is incident to n_{i+1} . This definition is just a reformulation of an integral path (section II.3(b) *Acyclic matchings*). A hypergraph corresponds to a layer of a discrete gradient if it does not contain any closed hyperpath. In that case, it will be called a *hyperforest* (figure III.6).

Regular components. Looking closer at the \mathcal{L}_{01} layer of a discrete vector field, we observe that there are only regular links, i.e. links incident to exactly two nodes. These layers are thus regular graphs. Now, we can extract a similar structure from a general hypergraph: the *regular part*, which consists of the same set of nodes and only the regular links. The connected components of this regular part will be called the *regular components* of the hypergraph (figure III.7).

Roots. In [Lewiner, 2002], we proved some simple properties of the layer hyperforest of a discrete gradient.

Proposition 1 *Considering a layer of a discrete gradient or equivalently a hyperforest, the following claims are true:*

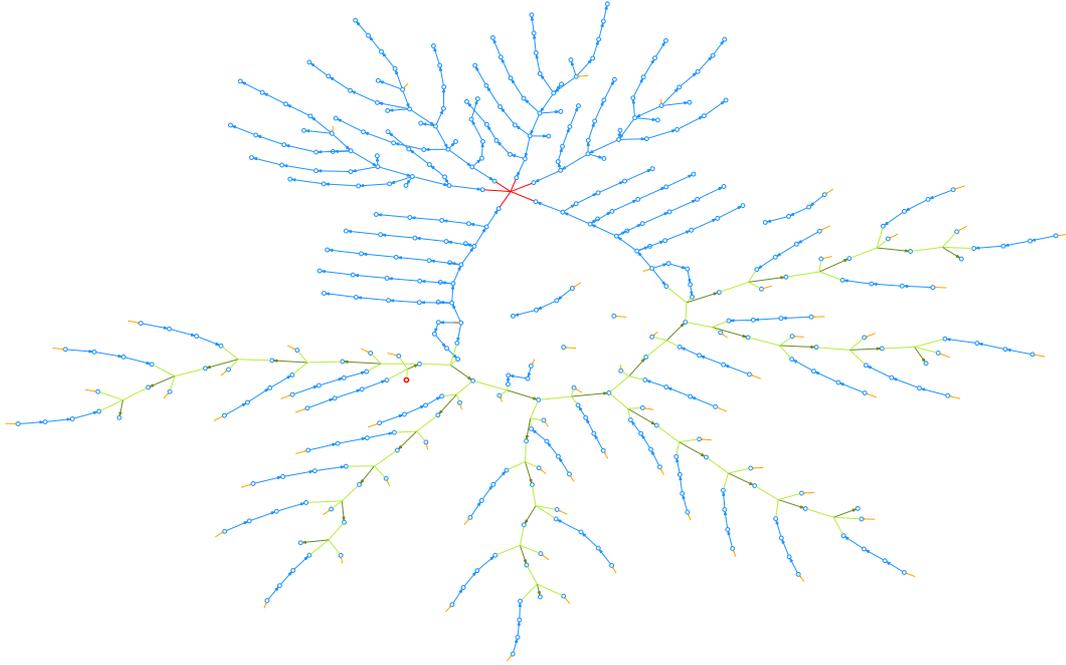


Figure III.7: Layer \mathcal{L}_{21} of an optimal Morse function on $\mathbb{S}^2 \times \mathbb{S}^1$: the hyperlinks are drawn in green and orange, and the regular components (in blue) are trees.

1. *The dual of a hyperforest is a hyperforest.*
2. *The regular components of a hyperforest are regular non-oriented trees.*
3. *For each regular component of a hyperforest, there is exactly one node, called its root, that is not the source of a regular link (i.e. containing exactly two nodes). The root is thus either critical or the source of a non-regular hyperlink. In a primal layer, all the links are pointing towards the root, and outwards in a dual layer.*

Proof: The first part of the proposition is immediate, once noticing that a closed hyperpath in a hypergraph is also a closed hyperpath in the dual hypergraph. As the critical nodes and links are not oriented, the assertion is also true for layers and dual layers. This ensures the direct equivalence between a discrete gradient and its hyperforest representation.

The second part can be proved easily by contradiction. Suppose a regular component is not a non-oriented tree, i.e. it contains a cycle $n_0 l_0 n_1 \dots n_r l_r n_{r+1} = n_0$ with l_i incident to n_i and n_{i+1} . Since the links are regular, the source of the link l_k in the whole hypergraph is either n_k or n_{k+1} . Suppose, without loss of generality, that the source of l_0 is n_1 . Then the source of l_1 must be n_2 , as a node, in particular n_1 , is the source of at most one link. Following the cycle, we get that the source of l_k is always n_{k+1} . Therefore, the cycle considered is in fact a closed hyperpath, in contradiction with the definition of a hyperforest.

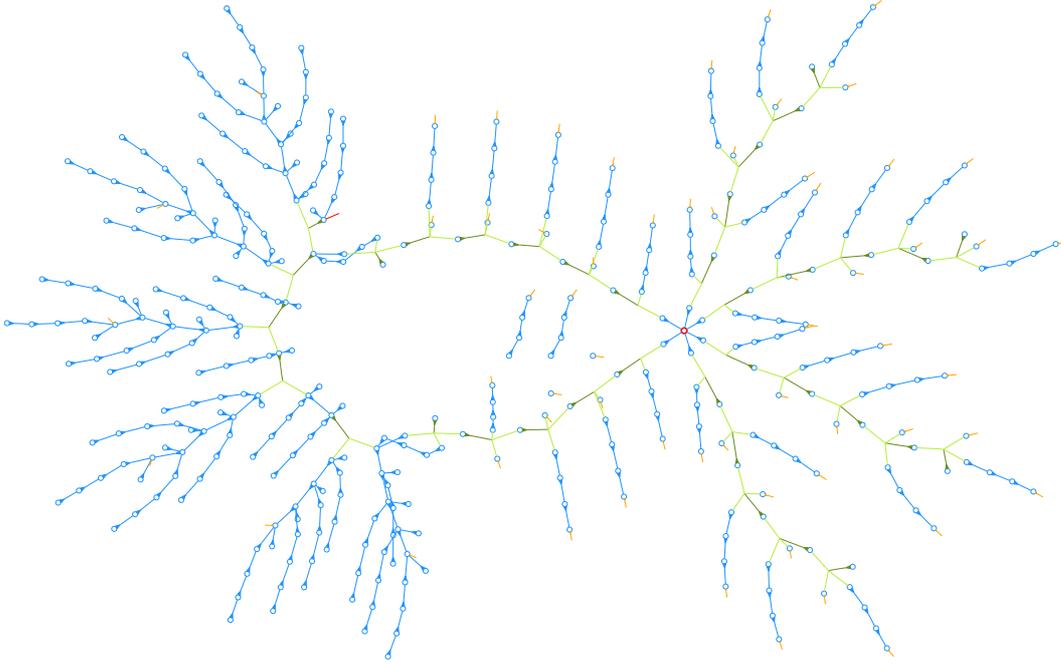


Figure III.8: Layer \mathcal{L}_{12} of an optimal Morse function on $\mathbb{S}^2 \times \mathbb{S}^1$: there is at most one non-regular hyperlink (green or orange) entering a regular component.

We will now prove that there is only one root per regular component (figure III.8). Since each link of a layer hyperforest is oriented, each link of a regular component has its source in the regular component. Recalling that a tree with k nodes has $k - 1$ links, there is exactly one node that is not the source of a link of the regular component. Either this node is critical, or the link must be non-regular, otherwise it would belong to the regular component. The orientation towards or outwards the root is a consequence of the limit of one sourced link per node. We can then follow the orientation of the links in a unique way inside the regular component, similarly to the proof of the second part. The only node where this path ends or goes out of the regular component is the root. ■

III.2 Greedy construction

The definition of the gradient as an acyclic matching recalls two classical algorithm of graph theory. The first one is the perfect matching algorithm [Lovasz & Plummer, 1986], which tries to extract disjoint pairs of adjacent nodes in a graph. This corresponds to a general combinatorial vector field, and is half-way to the definition of the gradient. The other half deals with acyclicity. Since an acyclic graph is a tree, we will use classical spanning tree extraction on graphs [Berge, 1970] (figure III.13), which becomes more efficient with appropriate data structures such as the Union & Find [Tarjan, 1975].

In this section, we will introduce a simple greedy construction of discrete Morse function, which can actually generate *any* discrete gradient. We will describe it mainly through pseudo-code, since the algorithms are simple. They can be adapted for different purposes. For example the constructions of optimal discrete Morse functions [Lewiner, 2002, Lewiner *et al.*, 2003b, Lewiner *et al.*, 2004] can be derived from it. The greedy algorithm is the central step of our construction of discrete Morse complexes.

(a) Data structure and basic algorithms

The overall formulation we adopted for discrete Morse theory strengthens the tree representation, and therefore our data structure completes this point of view by representing more the matching of each cell than the elements for the acyclicity test.

Cell complex. A cell complex is represented as a collection of cells, and each cell contains a reference to its incident cells (boundary and co-boundary) and the representation of the discrete gradient with its value through the discrete Morse function (structure III.1). For example, the `.star` of vertex 3 of figure III.3 contains the edge identifiers 2,3,4 and 6, while the `.bdry` of edge 6 contains the vertex identifiers 3 and 5.

Example: function to gradient conversion. The above data structure is easy to manipulate. As an illustration, algorithm III.2 computes the discrete gradient \mathcal{V} from a discrete Morse function: $\mathcal{V} = \{(\tau \geq \sigma)\}$ (figure III.9). This is the reverse operation of algorithm III.7, which is the final step of the construction exposed in this section.

Union & Find implementation. The `.basin` variable actually contains the Union & Find data structure, and will be used to optimize the gradient

Structure III.1 Cell σ_i^p *Cell complex representation*

dim	$\sigma_i^p = p$	// Dimension of the cell
id	$\sigma_i^p = i$	// Identifier of the cell
bdry	$\sigma_i^p = \text{list} \{ [\sigma_i^p : \tau_j^{p-1}] j \}$	// Boundary $\partial_p \sigma_i^p$
star	$\sigma_i^p = \text{list} \{ [\rho_j^{p+1} : \sigma_i^p] j \}$	// Co-boundary σ_i^p

Morse structure representation

match	$\sigma_i^p = \mathbf{V}(\sigma_i^p)$	// Discrete gradient image or pre-image
val	$\sigma_i^p = \mathbf{f}(\sigma_i^p)$	// Discrete function image
basin	$\sigma_i^p = \rho^p \in K$	// Root of the regular component
homo	$\sigma_i^p = \rho^p \in M_p$	// Homology decomposition
weight	$\sigma_i^p \in \mathbb{R}$	// Weight for the greedy order

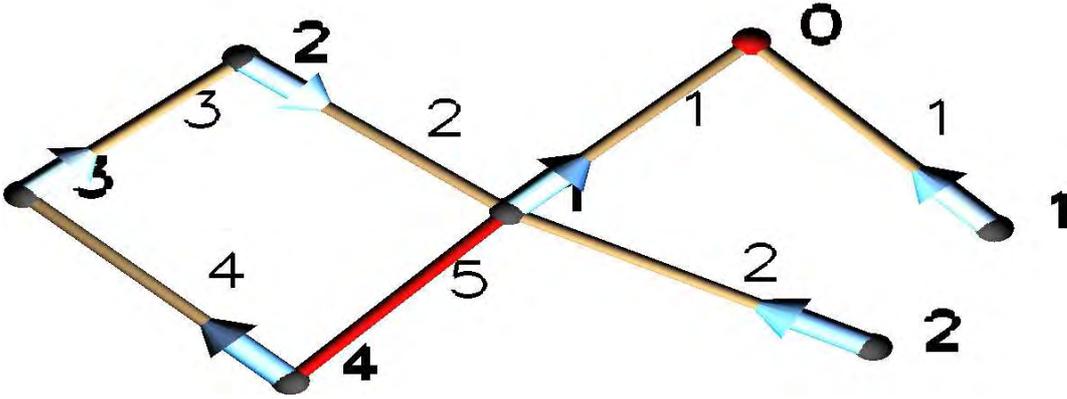


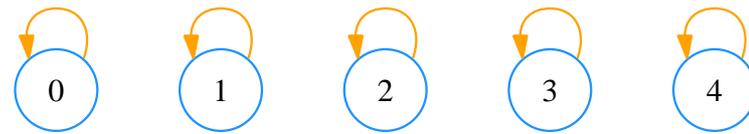
Figure III.9: Example of fun2grad (algorithm III.2) execution, with the labels of figure III.3: vertex v_0 has a value inferior or equal to the one of edge e_0 in its co-boundary, and so the algorithm matches them. Then v_1 and e_2 , v_2 and e_1 , v_3 and e_4 , v_4 remains critical, v_5 and e_6 , v_6 and e_5 . The edges have empty co-boundaries, thus the algorithm will not do anything more.

Algorithm III.2 fun2grad: compute \mathcal{V} from \mathbf{f}

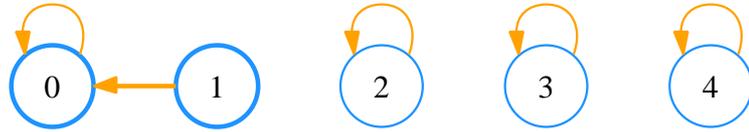
```

1: for  $\sigma \in K$  do // For each cell of  $K$ 
2:   for  $\rho \in \sigma.\text{star}$  do // For each cell of the co-boundary
3:     if  $\mathbf{f}(\rho) \leq \mathbf{f}(\sigma)$  then // Cardinality condition
4:        $\sigma.\text{match} \leftarrow \rho$  // Set the gradient
5:        $\rho.\text{match} \leftarrow \sigma$  // Set the gradient
6:     end if
7:   end for
8: end for

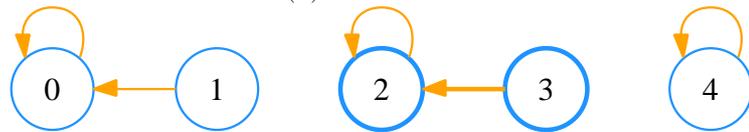
```



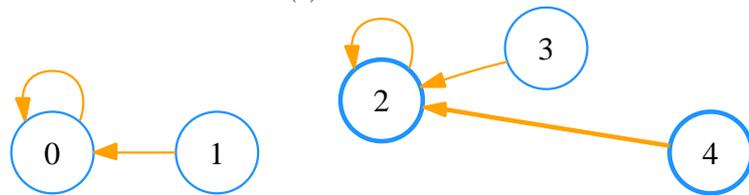
III.10(a): Original state.



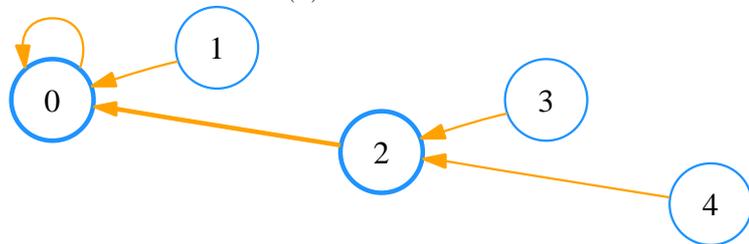
III.10(b): Union of 0 and 1.



III.10(c): Union of 2 and 3.



III.10(d): Union of 2 and 4.



III.10(e): Union of 0 and 2.

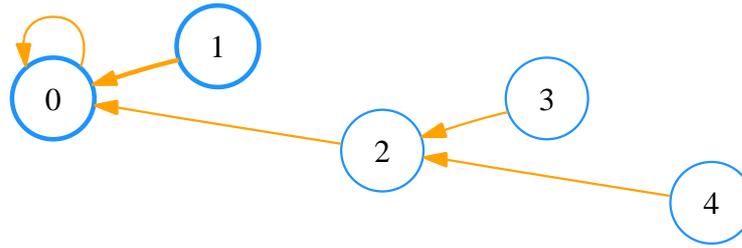
Figure III.10: Example of union (algorithm III.3) executions.

Algorithm III.3 $\text{union}(\sigma, \tau)$: union of the basins of σ and τ

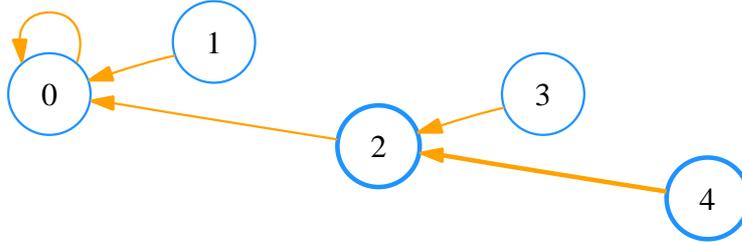
```

1:  $\sigma' \leftarrow \text{find}(\sigma)$  // Get the basin of  $\sigma$ 
2:  $\tau' \leftarrow \text{find}(\tau)$  // Get the basin of  $\tau$ 
3: if  $\sigma'.\text{weight} < \tau'.\text{weight}$  then //  $\sigma'$  is lower than  $\tau'$ 
4:    $\tau'.\text{basin} \leftarrow \sigma'.\text{basin}$  // Assigns to the lower basin
5: else //  $\tau'$  is lower than  $\sigma'$ 
6:    $\sigma'.\text{basin} \leftarrow \tau'.\text{basin}$  // Assigns to the lower basin
7: end if

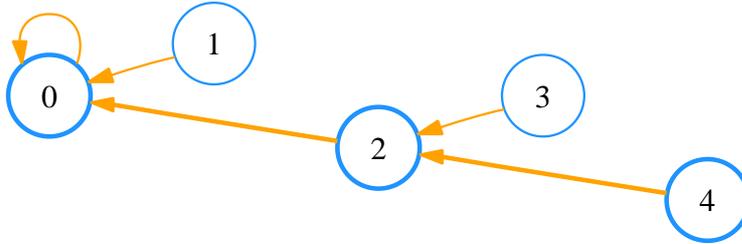
```



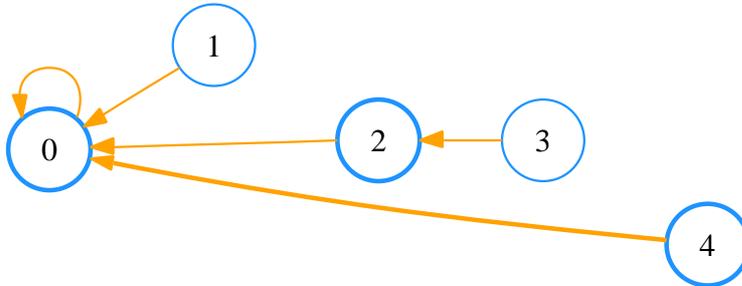
III.11(a): Find 1 returns 0.



III.11(b): Find 4 calls find 2.



III.11(c): Find 2 returns 0.



III.11(d): Find 4 updates 4.basin \leftarrow 0 and returns 0.

Figure III.11: Example of find (algorithm III.4) executions: each call updates the structure to optimize the next call.

Algorithm III.4 find(σ) : find the basin (root) of σ

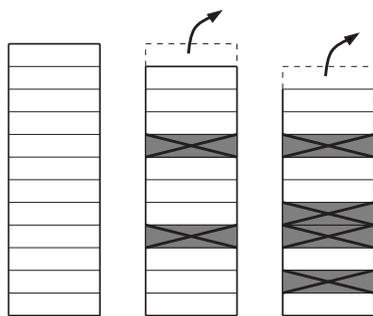
- 1: **if** σ .basin \neq σ **then** *// σ is not the root of its basin*
 - 2: σ .basin \leftarrow find(σ .basin) *// Recursion*
 - 3: **end if**
 - 4: **return** σ
-

construction and the Morse complex representation, since both are partitions of the original cell complex constructed by cancellations, i.e. union of classes in that partition. The **Union & Find** [Tarjan, 1975] data structure is an efficient representation for these operations on such a partition. It is described in algorithms III.3 (figure III.10) and III.4 (figure III.11).

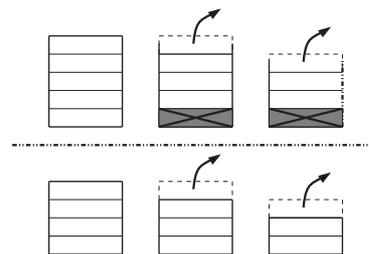
(b) Greedy algorithm

The greedy construction of discrete Morse function starts with an empty discrete gradient \mathcal{V} (with all the cells critical) and adds pair per pair \mathcal{V} , performing local cancellations of incident cells. A pair $(\tau < \sigma)$ is added if neither τ nor σ already belong to a pair (discrete vector field condition) and if this pair does not create any cycle inside \mathcal{V} (gradient condition). The second test (algorithm III.6) is the only complex part of the whole algorithm (algorithm III.5). The implementation of the greedy construction is similar to the spanning tree extraction (figure III.13): the links of the graphs are tested sequentially, and if a link joins two distinct components, it is added to the spanning tree. More precisely, the two nodes of a link belong to the same component if the find algorithm returns the same root for both. In that case, the link would create a cycle. Otherwise, the two components are joined by the addition of the link, which is performed by a **union** operation.

Actually any discrete gradient \mathcal{V} can be constructed with this algorithm, by defining the weight of a cancellation such that only the pairs of \mathcal{V} appear in the priority queue.



III.12(a): The whole priority queue: cancelling a pair invalidates other elements of the queue.



III.12(b): The priority queue divided by half will have stored less pairs at the end of the algorithm.

Figure III.12: The priority queue of the greedy algorithm can be optimized.

Greedy algorithm. Since the greedy construction can generate any discrete gradient, it depends heavily on its unique parameter: the **cancellation_weight** function, which induces the order for considering local cancellations. The

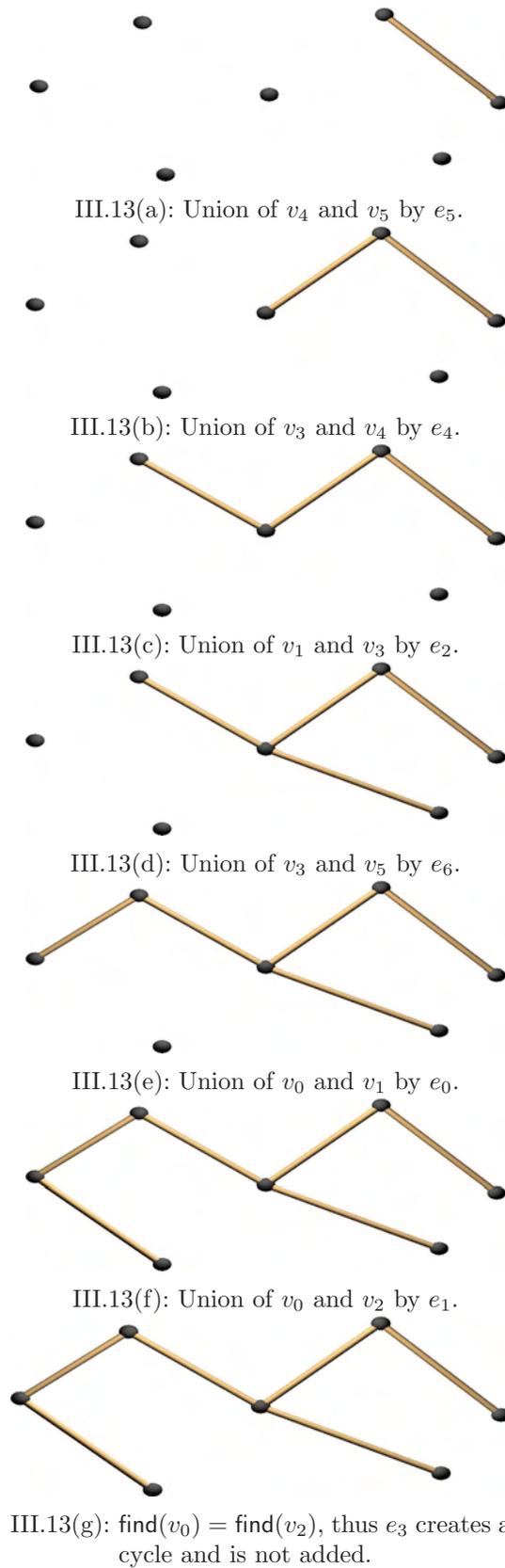
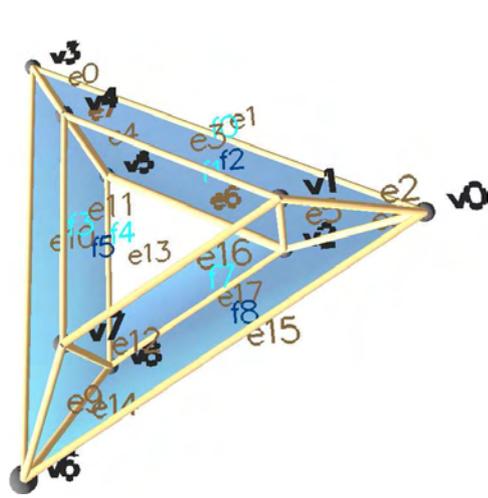
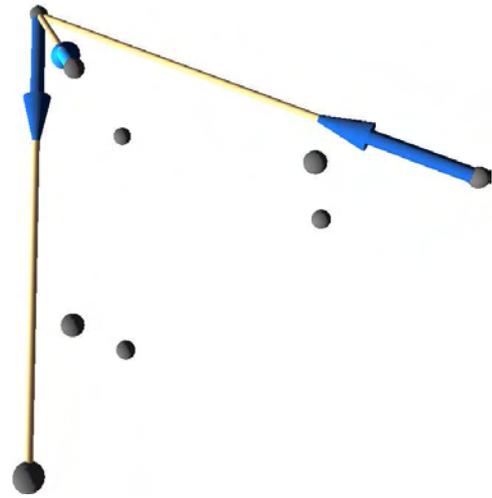


Figure III.13: The spanning tree algorithm on the example of figure III.3.



III.14(a): Labeling of the torus.



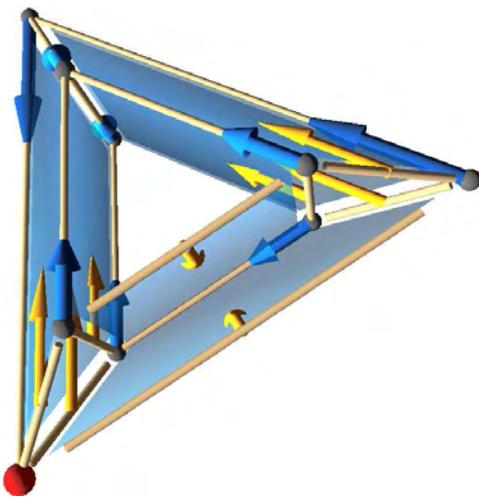
III.14(b): First 3 cancellations: v_0 and e_1 , v_4 and e_0 , v_3 and e_{10} .



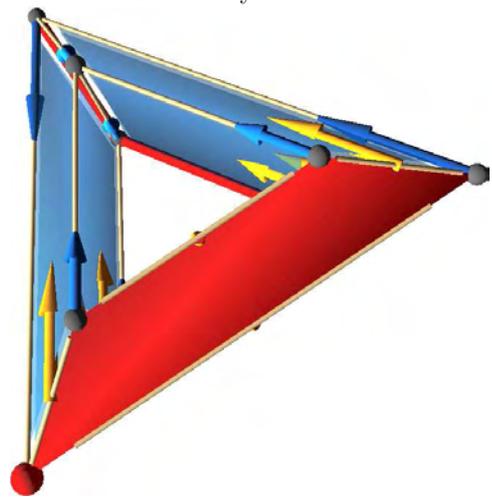
III.14(c): 4 cancellations: e_5 and f_1 , e_8 and f_2 , e_{12} and f_4 , e_{14} and f_5 .



III.14(d): 4 cancellations: v_1 and e_3 , v_2 and e_{17} , v_7 and e_{10} , v_8 and e_{13} . Cancellation v_2 and e_6 would create a cycle.



III.14(e): 4 cancellations: e_2 and f_0 , e_9 and f_3 , e_{15} and f_8 , e_{16} and f_7 .



III.14(f): The remaining cancellations would create cycles, and v_6 , e_6 , e_7 and f_6 remain critical.

Figure III.14: The greedy algorithm on a torus model.

construction simply orders all the local cancellations with finite weight and tests each of them sequentially (algorithm III.5 and figure III.14). Since the cancellation (τ, σ) prohibits any cancellation involving τ or σ , the algorithm can be improved by considering in a first stage only the cancellations of weight lower than a given threshold, then considering the valid cancellations of weight below a second threshold and so forth. This improvement reduces the cost of sorting the cancellations since the priority queue does not need to sort pairs invalidated in previous stages (figure III.12).

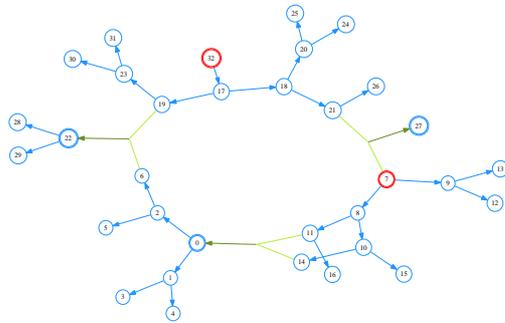
Algorithm III.5 `greedy(cancellation_weight)` : greedy construction of \mathcal{V}

```

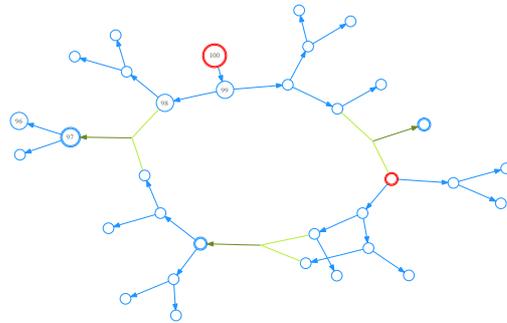
1: priority_queue  $\{\mathbb{R} \times K \times K\} \leftarrow \emptyset$  // Initialize the priority queue
2: for  $\sigma \in K$  do // For each cell of  $K$ 
3:    $\sigma.\text{basin} \leftarrow \sigma$  // Reset the Union & Find data structure
4:    $\sigma.\text{match} \leftarrow 0$  // Reset  $\mathcal{V}$ 
5:   for  $\tau \in \sigma.\text{bdry}$  do // For each cell of the boundary of  $\sigma$ 
6:      $w \leftarrow \text{cancellation\_weight}(K, (\tau, \sigma))$  // Weight of the local
       cancellation
7:     if  $w \neq \infty$  then // Allow selection of matchings
8:       priority_queue.push  $((w, \tau, \sigma))$  // Sort the local cancellations
9:     end if
10:  end for
11: end for
12:
13: while  $(w, \tau, \sigma) \leftarrow \text{priority\_queue.top}$  do // Traverses the priority queue
14:   if  $\sigma.\text{match} = 0$  and  $\tau.\text{match} = 0$  and not create_cycle  $((\tau, \sigma))$  then //
     the cancellation is valid
15:      $\sigma.\text{match} \leftarrow \tau$  // Set the gradient
16:      $\tau.\text{match} \leftarrow \sigma$  // Set the gradient
17:     for  $\tau' \in \sigma.\text{bdry}$  do // For each cell of the boundary of  $\sigma$ 
18:       union  $(\tau', \tau)$  // updates the union & find data structure
19:     end for
20:     for  $\sigma' \in \tau.\text{star}$  do // For each cell of the co-boundary of  $\tau$ 
21:       if  $\sigma'.\text{match.dim} = \tau.\text{dim}$  then //  $\sigma'$  belongs to the layer
22:         union  $(\sigma'.\text{match}, \tau)$  // updates the union & find data structure
23:       end if
24:     end for
25:   end if
26: end while

```

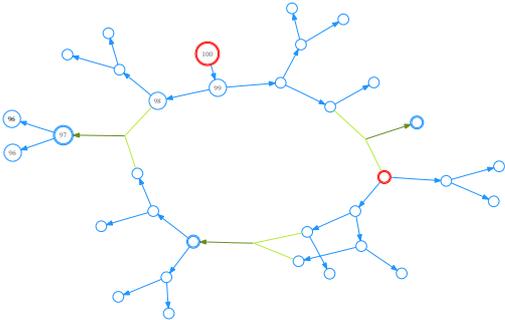
Acyclicity test. To test if the cancellation $(\tau < \sigma)$ would create a closed hyperpath, the algorithm checks if the basin that would contain τ after the cancellation already contains a node incident to σ (figure III.15). If the layer contains only regular links, the test is the same as the spanning tree algorithm.



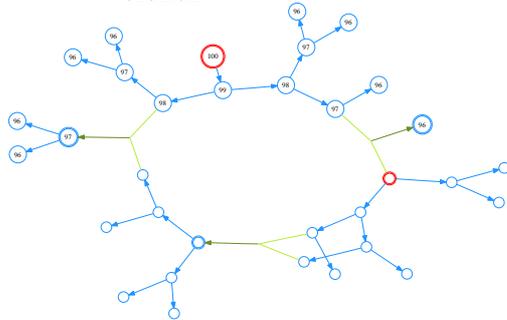
III.16(a): Labels: there are 4 roots: 7 and 32, 22 and 27.



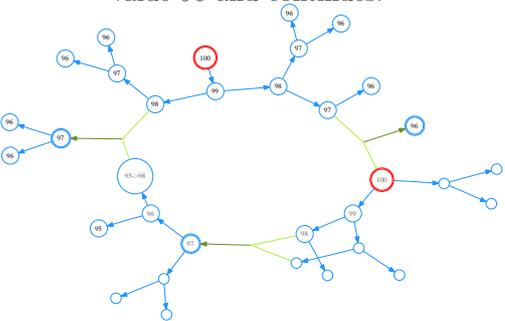
III.16(b): Starting from node 32: the stack, until line III.7, contains one branch.



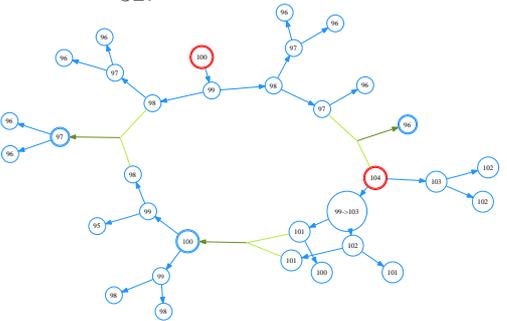
III.16(c): Then pops the stack, confirms value 96 and continues.



III.16(d): Completes the component of node 32.



III.16(e): Starts from node 7: the branch reaches nodes 6 which is already assigned, and increases the stored value of m .



III.16(f): Completes the component of node 7: the value needs to be increased again from node 10 to node 8, but not from node 27 to node 7.

Figure III.16: Example of `grad2fun` execution (algorithm III.7).

the construction of algorithm III.7 is the simplest one, assigning to each cell its depth in its layer hyperforest (figure III.16). The algorithm assigns first a Morse value to the nodes of each regular component and assigns the same value to the destination link. If it encounters a hyperlink entering to a regular component, it follows that link to assign first the lower regular component. Therefore, the assignment must be done from the leaves to the roots in order to maintain a node with a higher value than its sons.

Algorithm III.7 grad2fun : compute f from \mathcal{V}

```

1: for  $p \in [1 \dots \dim K]$  do // For each layer hyperforest
2:    $m \leftarrow 2 \cdot \text{card } K^p$  // Value big enough to separate layers
3:   for  $\sigma \in \{\text{roots of } \mathcal{L}_{p(p-1)}, <\}$  do // Get the minimal root
4:      $\text{stack} \{\text{Dual}_p \cup \text{Crit}_p \times \text{Prim}_{p-1} \cup \{0\}\} \leftarrow \{(\sigma, 0)\}$  // Traversal stack
5:     while  $\text{stack} \neq \emptyset$  do // Regular component traversal
6:        $\sigma \leftarrow \text{stack.top}$  // Get the first cell of the path
7:       while  $\exists \tau < \sigma, \text{unmarked}(\tau)$  do // Still not the leaf
8:          $\sigma \leftarrow \tau.\text{match}$  // Completes the gradient step
9:          $\text{stack.push}((\sigma, \tau))$  // Stores the gradient step
10:         $\text{mark}(\sigma); \text{mark}(\tau)$  // Marks step as visited
11:         $m \leftarrow m - 1$  // Down one depth level in the tree
12:      end while
13:      // Reached a leaf
14:       $(\sigma, \tau) \leftarrow \text{stack.top}$  // Get the last step of the path
15:       $M = \max \text{val} \{\tau' < \sigma, \text{marked}(\tau'), \tau' \in \mathcal{L}_{p(p-1)}, \tau' \text{ non regular}\}$ 
// Maximum of assigned non-regular links
16:       $m \leftarrow \max(m, M + 2)$  // Maintain increasing  $f$ 
17:      if  $\tau \neq 0$  then // Non-critical cell
18:         $\sigma.\text{val} \leftarrow m; \tau.\text{val} \leftarrow m$  // Assign matched cells
19:      end if
20:       $m \leftarrow m + 1$  // Up one depth level in the tree
21:    end while
22:  end for
23: end for
24:
25: for  $\rho \in K$  do // For the unassigned critical cells
26:   if  $\rho.\text{val} = \text{unassigned}$  then //  $\rho$  unassigned
27:      $\rho.\text{val} = \max \text{val} \{\text{Dual}_{\dim \rho} \cup \text{Crit}_{\dim \rho}\} + 1$  // Assigns a critical value
28:   end if
29: end for

```

(c) Heuristic for optimal Morse functions

In [Lewiner, 2002], our goal was to define optimal discrete Morse functions, i.e. Morse functions having the minimal possible number of critical cells. This problem has been proved NP-hard [Lewiner *et al.*, 2003b,

[Joswig & Pfetsch, 2005]. An exponential algorithm can be simply implemented testing all possible matchings, or in a more elegant way using integer programming [Joswig & Pfetsch, 2005]. However, the greedy algorithm above can be used to compute discrete Morse functions with a very small number of critical cells (algorithm III.8). On the models of [Hachimori Models], this algorithm always reached the optimum in less than quadratic time. Moreover, it has been proved to reach the optimum for surfaces in almost linear time [Lewiner *et al.*, 2003a].

Since the regular part of the hypergraphs is easier to process, the algorithm calls first the **greedy** procedure for each layer separately, considering only the regular links (figure III.17). Then it prunes the layer hypergraph. If there is no more leaves in a non-empty layer, there is a cycle of valid cancellations. This cycle is broken arbitrarily (here is the heuristic), using the matching of higher **cancellation_weight**. To do so, we call “**once**” the greedy algorithm, where **once** command on line III.8 of the algorithm means that **greedy** performs only one valid matching and returns.

Algorithm III.8 `optimal(cancellation_weight)` : Construction of an optimal \mathcal{V}

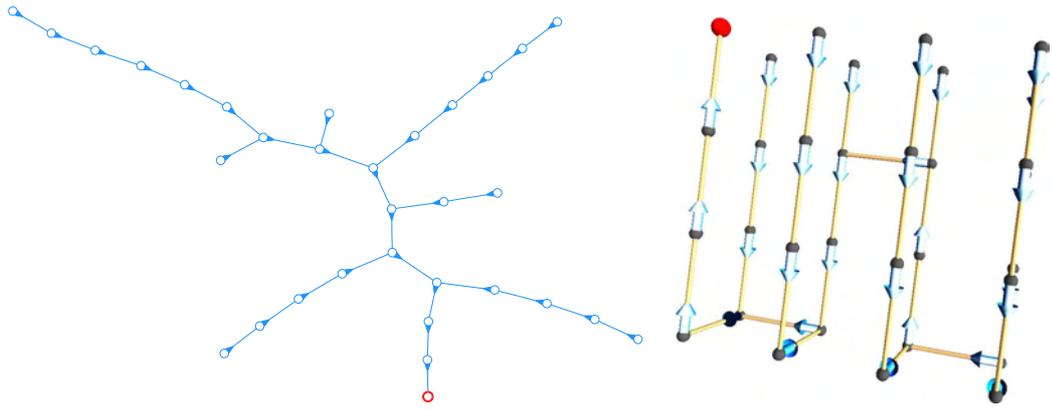
```

1: for  $d \in \{1, \dots, \dim K\}$  do                                     // Process layers separately
2:   define weight ( $K, (\tau, \sigma)$ )  $\leftarrow$                  // Discards non-regular links
       $\begin{cases} \infty & \text{if } \sigma.\dim \neq d \\ \infty & \text{if } \sigma \text{ is not a regular link of } \mathcal{L}_{(d-1)d} \\ \text{cancellation\_weight}(K, (\tau, \sigma)) & \text{otherwise} \end{cases}$ 
3:   greedy(weight) // Construction of the regular components of  $\mathcal{L}_{(d-1)d}$ 
4: end for
5:
6: repeat // Pruning and greedy step
7:   define weight ( $\tau^{d-1}, \sigma^d$ )  $\leftarrow$  // Pruning
       $\begin{cases} \text{cancellation\_weight}(K, (\tau, \sigma)) & \text{if } \tau \text{ is a leaf of } \mathcal{L}_{(d-1)d} \\ \text{cancellation\_weight}(K, (\tau, \sigma)) & \text{if } \sigma \text{ is a leaf of } \mathcal{L}_{d(d-1)} \\ \infty & \text{otherwise} \end{cases}$ 
8:   greedy(weight) // Primal and dual pruning
9:   once greedy(cancellation_weight) // Unlock cancellation cycles
10: until no more valid local cancellation

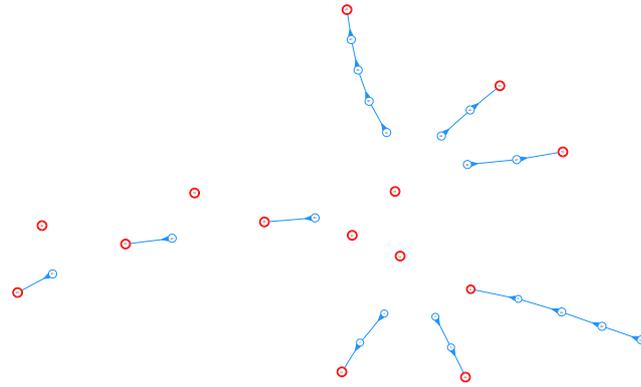
```

(d) Complexity

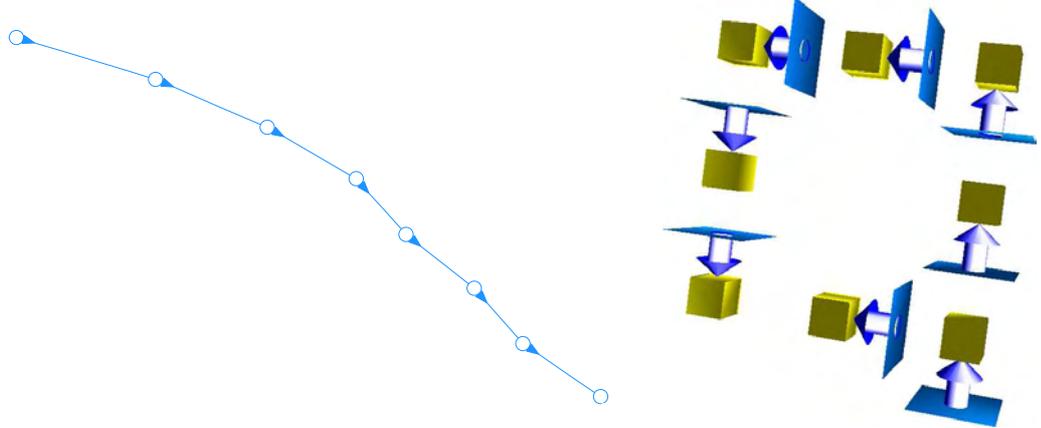
The acyclicity test of algorithm III.6 performs in time $O(s \cdot \log s)$, where s is the average size of the boundary and co-boundary of a cell. We will suppose that for a complex K , this size is in average bounded by a small constant. This implies that algorithm III.6 has in average a constant complexity.



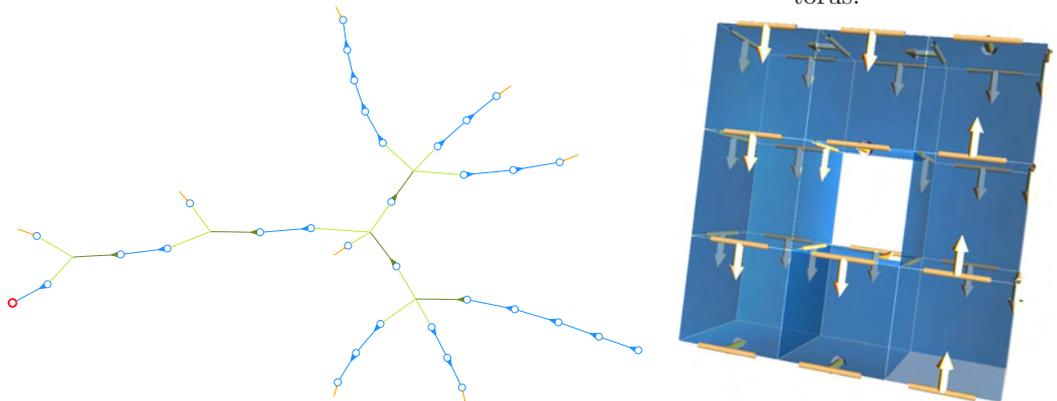
III.17(a): First call to greedy with $d = 1$ (layer \mathcal{L}_{01}). III.17(b): Layer \mathcal{L}_{01} on the torus.



III.17(c): Second call to greedy with $d = 2$ (layer \mathcal{L}_{12}).



III.17(d): Third call to greedy with $d = 3$ (layer \mathcal{L}_{23}). III.17(e): Layer \mathcal{L}_{23} on the torus.



III.17(f): Last call to greedy: pruning layer \mathcal{L}_{12} . III.17(g): Layer \mathcal{L}_{12} on the torus.

Figure III.17: Example of optimal execution (algorithm III.8) on a solid torus.

The Union & Find structure of algorithms III.3 and III.4 has complexity $O(\alpha(n))$, where $\alpha(n)$ is the inverse Ackermann function [Tarjan, 1975]. In fact, this function has a value less than 5 for any conceivable input size (table III.18). Since then, the complexity of the Union & Find structure is often called *almost constant*.

The greedy construction of algorithm III.5 first creates the priority queue, and then calls for each candidate matching the acyclicity test of algorithm III.6, and eventually $O(s)$ times the union function. Since we considered that s is in average constant, the second part of the algorithm is almost linear. If the `cancellation_weight` does not matter, for example if `cancellation_weight`($K, (\tau, \sigma)$) is constant, the priority queue is created in a time linear to the number of matchings $O(s \cdot \#K) \approx O(\#K)$. If the `cancellation_weight` is significant, the priority queue has complexity $O(\#K \cdot \log \#K)$.

$\mathbf{n} \in$	$\alpha(n)$
{1, 2, 3}	1
{4, 5, 6, 7}	2
{8, ..., 61}	3
$\left\{ 62, \dots, 2^{2^{2^{2^{2^2}}}} - 3 \right\}$	4

Table III.18: Inverse Ackermann function values.

Finally, the complexity of the optimal construction is *a priori* the same as the greedy construction, i.e. either $O(\#K \cdot \log \#K)$ if the cancellation weight must be respected, or $O(\#K \cdot \alpha(\#K))$ on the contrary. However, the second loop of the algorithm can require a time quadratic in the number of remaining cancellations. Observe that there is at most h remaining cancellations, where h is the number of non-regular hyperlinks. The final complexity is therefore usually linear, and quadratic in the worst case [Lewiner, 2002].

(e) Application to volumetric compression

Among the different strategies to compress polyhedral meshes, most of the state-of-the-art approaches [Rossignac, 1999, Lopes *et al.*, 2002, Lewiner *et al.*, 2004, Touma & Gotsman, 1998, Alliez & Desbrun, 2001, Kälberer *et al.*, 2005] for surfaces encode the connectivity of the mesh by a traversal equivalent to a dual layer of a discrete gradient \mathcal{V} . The roots of the layer are the most expensive elements to encode. In order to minimize those elements, the algorithm tends to optimize \mathcal{V} , since each critical cell is a root. In the case of surfaces, we proved in [Lewiner *et al.*, 2003a] that any spanning forest reaches the optimum.



Figure III.19: Because of the critical edge (in red), the green triangle must be a dual cell, isolated inside the dual hyperforest. This triangle will be expensive to encode.

However, for 3-manifolds, the problem is NP-hard and requires some heuristics to be quickly resolved. The strategy of algorithm III.8 is similar for that special case to the one used by Grow & Fold [Szymczak & Rossignac, 2000], as described in [Lewiner *et al.*, 2004]. This heuristic actually generalizes to higher dimensions and to non-manifold, non-simplicial and non-pure cell complexes, which can be useful for further compression algorithms.

III.3 Flow basins and hypergraph components

The representation of a discrete gradient as hyperforests simplified the algorithmic construction of discrete Morse functions. Actually, this representation turned out to be more meaningful than just an efficient data structure. This section shows the close relation between the regular components of a layer $\mathcal{L}_{p(p-1)}(\mathcal{V})$ (section III.1(c) *Hyperforest*) and the unstable basin of the flow $W_{\mathcal{V}}^u(\sigma^p)$ (section II.7(b) *Invariant chains*). Actually, the Union & Find data structure represents almost directly those unstable basins $W_{\mathcal{V}}^u$. This will enhance the representation of both the discrete Morse–Smale decomposition and the Morse homology computation. In order to compute the unstable basins, we will consider only dual layers. The stable basins can be obtained in a similar way using the primal layers instead, although those were not explicitly defined in Forman’s original works.

(a) Flow image of a critical or dual cell

Consider a node n of the dual layer $\mathcal{L}_{p(p-1)}(\mathcal{V})$. This node represents a cell σ^p that is either dual or critical. The *sons* of n are the nodes n' of $\mathcal{L}_{p(p-1)}$ that are sources of links incident to n : $n \rightsquigarrow n'$ if $\exists l : n > l$ and $(l, n') \in \mathcal{V}$. We will say that n is the *father* of the n' . The *iterated son* of a node n is either the son of n or the son of an iterated son of n . We will prove now the following proposition:

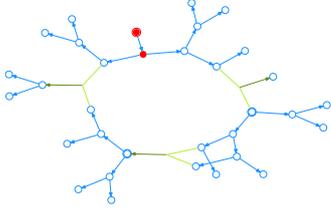
Proposition 2 *The flow image $\Phi(\sigma^p)$ of σ^p is composed of the sons of n plus n itself if σ^p is critical.*

Proof: The flow of σ^p is defined as (section II.7(b) *Invariant chains*):

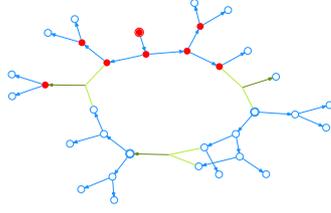
$$\Phi(\sigma^p) = \sigma^p + \mathbf{V}(\partial_p(\sigma^p)) + \partial_{p+1}(\mathbf{V}(\sigma^p)) \quad (1)$$

Independently if σ^p is dual or critical, $\mathbf{V}(\sigma^p) = 0$, which simplifies the right term of the equation. Now the boundary $\partial_p(\sigma^p)$ of σ^p is composed of cells τ^{p-1} that belongs either to $\mathcal{L}_{p(p-1)}$ or to $\mathcal{L}_{(p-1)(p-2)}$. The latter case implies that τ^{p-1} is either dual or critical, which means its image through \mathbf{V} is zero. Therefore, we get:

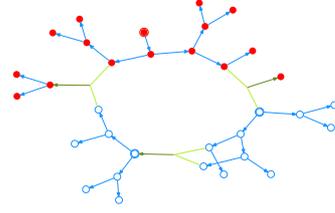
$$\Phi(n) = n + \mathbf{V} \left(\sum_{l \in \text{Prim}_{(p-1)}, l < n} [n : l] \cdot l \right) \quad (2)$$



III.20(a): 1 iteration.



III.20(b): 3 iterations.



III.20(c): 4 iterations.

Figure III.20: Iterated flow image of a critical cell on a dual layer: the critical cell is included in its image.

Image of a critical node. From the definition of son, each son of n is the image by \mathbf{V} of a term of the sum of equation (2). On one hand, if n is critical, it is not the image of any link, and will therefore be part of its image through the flow (figure III.20). On the other hand, each term of the sum of equation (2) is mapped by \mathbf{V} to a son of n . Therefore, we get:

$$\Phi(n) = n + \sum_{n' \rightsquigarrow n} [n : l] \cdot [n' : l] \cdot n' \quad (3)$$

The coefficient of each son n' actually corresponds to the multiplicity of the gradient path $\mu(\triangleleft n l n' \triangleright)$.

Image of a dual node. If n is not critical, it is the image of a link l_n which appears in the sum of equation (2). The other terms of the sum behave like the critical case:

$$\Phi(n) = n + [n : l_n] \cdot \mathbf{V}(l_n) + \sum_{n' \rightsquigarrow n} \mu(\triangleleft n l n' \triangleright) \cdot n' \quad (4)$$

From the orientation of the gradient, the terms with n cancels (figure III.21), and we get:

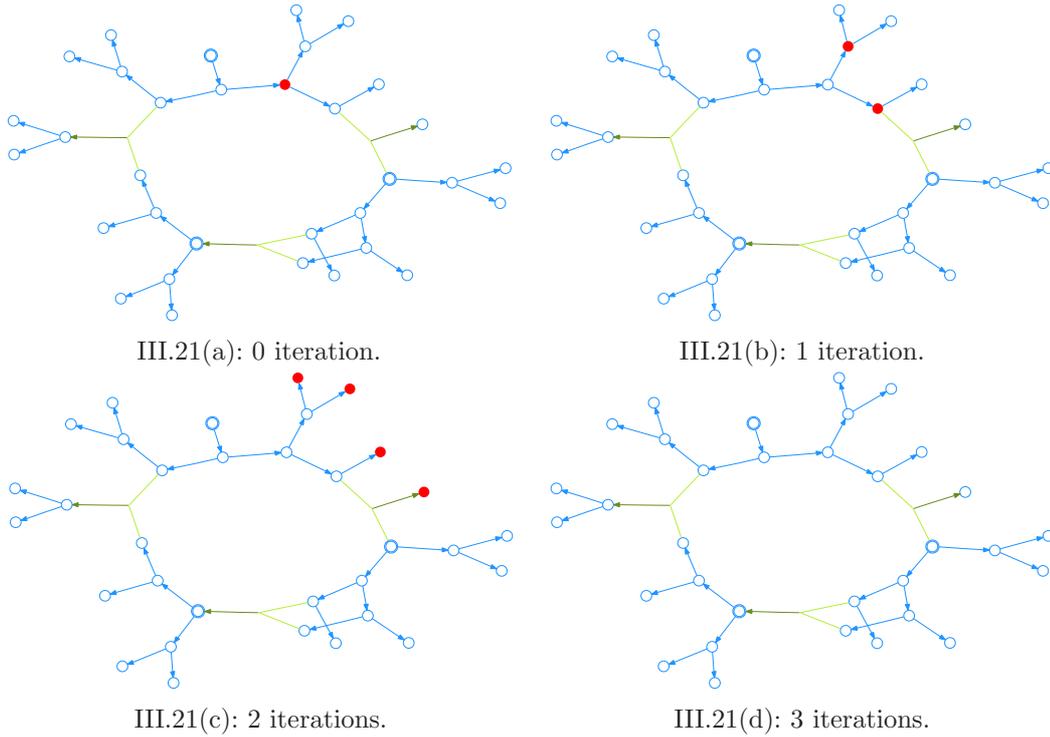


Figure III.21: Iterated flow image of a dual cell: the image vanishes after a number of iterations equal to the depth of the connected component.

$$\Phi(n) = \sum_{n' \leftarrow n} \mu(\langle n \mid n' \rangle) \cdot n' \quad (5)$$

■

We can deduce from this calculus the following result of [Forman, 1995]:

Corollary 3 *The iterated flow of a dual node vanishes:*

$$\sigma \in \text{Dual}(\mathcal{V}) \Rightarrow \Phi^\infty(\sigma) = 0 .$$

Proof: Iterating equation (5), we get:

$$\Phi^k(\sigma^p) = \sum_{n_0 \leftarrow \sigma^p} \sum_{n_1 \leftarrow n_0} \dots \sum_{n_k \leftarrow n_{k-1}} \mu(\langle \sigma^p \mid n_0 \mid n_1 \dots \mid n_k \rangle) \cdot n_k \quad (6)$$

Since there is no closed hyperpath in a hyperforest, the length of the hyperpath starting from σ^p is bounded by a particular k . With the notation of equation (6), there is no n_{k+1} son of n_k , and there is no term in the sum of $\Phi^{k+1}(\sigma^p)$. ■

(b) Discrete stable and unstable basins

From the above result, we can already compute the unstable basins directly from the layer representation.

Theorem 4 *The unstable basin $\Phi^\infty(\sigma^p) \in C_p^\Phi$ of a critical cell σ^p contains the regular component of σ^p in $\mathcal{L}_{p(p-1)}$ and is contained in the connected component of σ^p in $\mathcal{L}_{p(p-1)}$.*

Proof: The theorem is a simple consequence of propositions 1 and 2. From proposition 2, the flow image of a critical cell contains the critical cells. Therefore the elements of the image accumulate, and we get that:

$$\Phi^k(\sigma^p) = \sigma^p + \sum_{n_0 \leftarrow \sigma^p} \mu(\langle \sigma^p \dots n_0 \rangle) \cdot \left(n_0 + \sum_{n_1 \leftarrow n_0} \mu(\langle n_0 n_1 \rangle) \cdot \left(n_1 + \dots \sum_{n_k \leftarrow n_{k-1}} \mu(\langle n_{k-1} n_k \rangle) \cdot n_k \right) \right) \quad (7)$$

From the last part of proposition 1, the fathers of the nodes of the regular component of σ^p belongs to the regular component of σ^p . There is thus no cancellation inside the sum of terms belonging to the regular component of σ^p . From equation (7), the last part of the theorem is obvious. ■

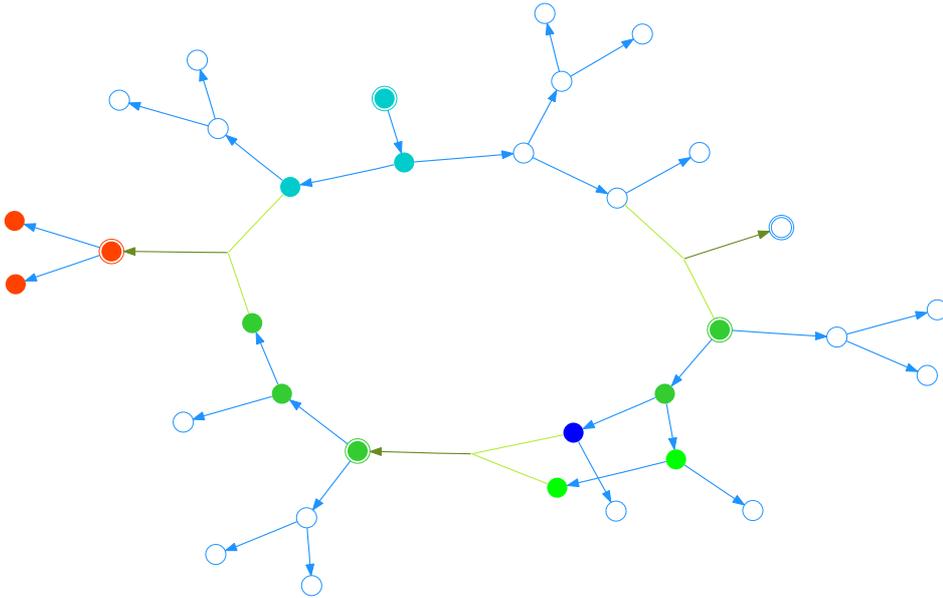


Figure III.22: Integral paths from both critical nodes 7 (in green) and 32 (in blue) merge on root 22 (red).

Integral path merging. Observe that a strong difference between smooth and discrete Morse theories is that a cell can belong to several stable basins at the same time (figure III.22). This is due to the definition of integral path that allow two paths to merge. In the smooth setting, the uniqueness of the solution of ordinary differential equations ensures that two integral curves can become arbitrarily close, but not merge. In the discrete setting, integral path can obviously not become arbitrarily close. In particular, this induces

that a node can be the iterated son of a critical cell and not belongs to its unstable basin. This occurs for example when a node is at the junction of two integral path, and when the multiplicity of the first one is the opposite of the multiplicity of the second one. However, this will be solved gracefully the case of homology cycles that share cells.

Stable basins. The considerations of this chapter apply directly for the stable basins, when considering primal layers instead of dual ones. For manifolds, this can be also been proved by considering the *dual manifold* with the discrete Morse function $-\mathfrak{f}$ (i.e. reverting the pairs of \mathcal{V}). The dual layers become primal layers, and the unstable basins become stable basins.

Theorem 5 *The stable basin of a critical cell σ^p contains the regular component of σ^p in $\mathcal{L}_{p(p+1)}$ and is contained in the connected component of σ^p in $\mathcal{L}_{p(p+1)}$.*

(c) Flow image of a primal cell

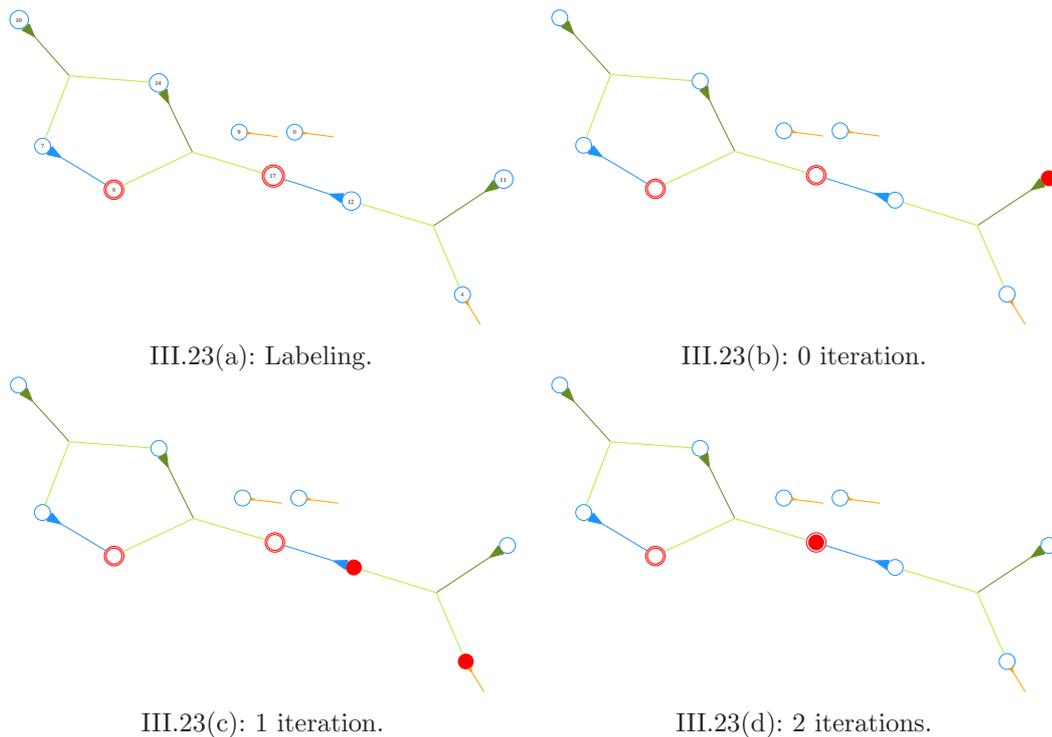


Figure III.23: Flow iteration on layer \mathcal{L}_{12} of figure III.14: while the flow of 4 vanishes, the flow of 11 reaches critical edge 17.

The dual layer has simplified the expression of the flow of a dual cell, since the image of a dual node is the sum its sons in the dual layer $\mathcal{L}_{p(p-1)}$. We reach a similar result for the primal cells, and we will therefore use the

primal layer $\mathcal{L}_{(p-1)p}$. The primal and dual layers differ in one point, noted in proposition 1: the orientation of the dual layer goes towards the roots, while the orientation goes outwards in a primal layer (figure III.23). This means in particular that the son of a primal cell can be a critical cell, and therefore its image under the iterated flow cannot vanish.

Since the image of a cell by \mathbf{V} is always a dual cell (section III.1(a) *Cell classification*), we can write equation (1) as follow:

$$\Phi(\tau^{p-1}) \cong \tau^{p-1} + \partial_p(\mathbf{V}(\tau^{p-1})) + \text{Dual}_{p-1} \quad (8)$$

From corollary 3, the terms of Dual_{p-1} will vanish when iterating the flow. Observe that $\mathbf{V}(\tau^{p-1})$ is the link l_n having the node n representing τ^{p-1} as source. The boundary of l_n is then composed of dual cells, n itself and the sons of n in $\mathcal{L}_{(p-1)p}$:

$$\Phi(n) = n - [l_n : n] \cdot [l_n : n] \cdot n + \sum_{n' \rightsquigarrow n} \mu(\langle n \ l_n \ n' \rangle) \cdot n' \quad (9)$$

The first two terms cancels. If n has no iterated son that is critical, its iterated flow will vanish:

Theorem 6 *The iterated flow of a primal $(p-1)$ -cell n vanishes if it has a no critical iterated son. More generally, the iterated flow of n is the sum of the iterated flow of the critical iterated sons of n :*

$$\Phi^\infty(n) = \sum_{\rho \in \text{Crit}_{p-1}} \sum_{\langle n \rightsquigarrow n_0 \dots n_k \rightsquigarrow \rho \rangle} \mu(\langle n \rightsquigarrow n_0 \dots n_k \rightsquigarrow \rho \rangle) \cdot \Phi^\infty(\rho) \quad (10)$$

This result allows an enhanced computation of the decomposition of a cycle on the Morse–Witten homology base. In particular, since $\mathcal{L}_{p(p-1)}$ is a hyperforest, the nodes leading to a critical node are easily identified. The only problem could come from two hyperpath leading to the same critical node, and cancelling it. This requires a path merge that only occur on non-regular link. That is why we will first concentrate on the regular parts of the layers, in particular for the proofs of the last chapter.

IV

Homology computation

Homology is a simple combinatorial invariant of geometrical objects. Particularly for high-dimensional cell complexes, it offers a direct classification that helps distinguishing models, checking their validity and even correcting them. However, the cost of homology computation is still a problem for big amount of data. The classical method for computing homology relies on the Smith normal form [Dumas *et al.*, 2003]. This method is essentially a Gaussian pivoting, which becomes very costly when the size of the complex increases. In the specific case of sub-triangulations of \mathbb{S}^3 , the Betti numbers on \mathbb{Z}_2 can be computed using Mayer-Vietoris sequences in time $O(\#K \cdot \alpha(\#K))$ [Delfinado & Edelsbrunner, 1993, Dey & Guha, 1998]. A basis of generator of the homology can be obtained for this very restrictive case with the same complexity.

Homology computation was at first an attempt to define properly the critical cells of the geometric Morse complex of the next chapter. However, the algorithms introduced here outperform most of the known algorithm. In particular, it computes with the same complexity as [Delfinado & Edelsbrunner, 1993, Dey & Guha, 1998] the Betti numbers, but without restriction on the cell complex or on the field of coefficients. They further compute the complete homology groups on any field, provide an explicitly basis for these groups on the complex and compute the decomposition of any cycle on that basis. The first part of this chapter introduces the tools used by the second part, which contains the detailed description of these algorithms.

IV.1 Algebraic and combinatorial tools

Homology can be completely computed from the Smith normal forms. However, even with the optimizations of this normalization (for example using [Dumas *et al.*, 2003]), it is still an expensive operation. In particular when computing the homology of real geometrical models, that have usually millions of cells, this computation can be prohibitive, even if the topology of those models is trivial! We propose here to reduce drastically the complexity of

this calculus, by computing the homology on the Morse complex instead of the whole complex. We know from section II.8(b) *Witten–Morse homology* that both are equivalent, but the Morse complex is usually much smaller. Particularly for usual geometrical models with millions of cells, if the topology is trivial then the Morse complex will have only a few cells.

(a) Smith normal form

Consider a cell complex K and a ring \mathbb{K} for the coefficients of the chains. To clarify the presentation, we can think that $\mathbb{K} = \mathbb{Z}_2$ or $\mathbb{K} = \mathbb{Z}$, but the whole chapter is valid on any computable ring. Since the chain modules are freely generated by the cells of K , the boundary operator $\partial_p : C_p \rightarrow C_{p-1}$ can be represented by a rectangular matrix \mathbf{B}_p of size $\#_{p-1} \times \#_p$ with coefficient in \mathbb{K} . For example the boundary of an edge σ is the difference of its end vertices τ_1 and τ_2 . This means that in the matrix of ∂_1 , the line indexed by σ will have exactly two non-null entries: a $+1$ for τ_2 and a -1 for τ_1 .

Recall that the homology groups of K are $H_p = \ker \partial_p / \text{Im } \partial_{p+1}$. We will try to find a basis for C_p on which both $\ker \partial_p$ and $\text{Im } \partial_{p+1}$ will be diagonal. The main tool to do so is the Smith normal form, which is a conjugation of matrices obtained by the Gauß operations on lines and columns in order to let the considered matrix diagonal. The coefficients of the diagonal will be either 0 or other elements of \mathbb{K} totally ordered by division. More precisely, the Smith normal form of a matrix $\mathbf{M}_{m \times n}$ is given by the two invertible matrices $\mathbf{P}_{m \times m}$ and $\mathbf{Q}_{n \times n}$ such that:

$$\mathbf{P}_{m \times m} \cdot \mathbf{M}_{m \times n} \cdot \mathbf{Q}_{n \times n} = \begin{bmatrix} k_1 & 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & k_2 & \dots & 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & k_r & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 \end{bmatrix} \quad \text{with } 0 \neq k_1 \mid k_2 \mid \dots \mid k_r$$

With this Smith normal form, we will build a base $\{\mathbf{c}_1 \cdots \mathbf{c}_{\#_p}\}$ of C_p such that $\ker \partial_p$ will be generated from $\{\mathbf{c}_1 \cdots \mathbf{c}_r\}$ and $\text{Im } \partial_{p+1}$ will be generated from $\{\mathbf{c}_{r+1} \cdots \mathbf{c}_{\#_p}\}$. Let $\mathbf{P}_{\#_{p-1} \times \#_{p-1}}$ and $\mathbf{Q}_{\#_p \times \#_p}$ be the matrices of the Smith normal form of \mathbf{B}_p :

$$\mathbf{P} \cdot \mathbf{B}_p \cdot \mathbf{Q} = \begin{bmatrix} \mathbf{K}_{r \times r} & \mathbf{0}_{r \times (\#_p - r)} \\ \mathbf{0}_{(\#_{p-1} - r) \times r} & \mathbf{0}_{(\#_{p-1} - r) \times (\#_p - r)} \end{bmatrix}.$$

Observe that \mathbf{K} is diagonal. Since $\partial_p \circ \partial_{p+1} = 0$, we have $\mathbf{B}_p \cdot \mathbf{B}_{p+1} = 0$.

Therefore $\mathbf{P} \cdot \mathbf{B}_p \cdot \mathbf{B}_{p+1} = 0$:

$$\begin{bmatrix} \mathbf{K}_{r \times r} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \cdot \mathbf{Q}^{-1} \cdot \mathbf{B}_{p+1} = \mathbf{0} \quad \text{and thus} \quad \mathbf{Q}^{-1} \cdot \mathbf{B}_{p+1} = \begin{bmatrix} \mathbf{0}_{r \times \#_{p+1}} \\ \tilde{\mathbf{B}}_{p+1} \end{bmatrix}$$

We can now apply the Smith normal form on $\tilde{\mathbf{B}}_{p+1}$, which is of size $(\#_p - r) \times \#_{p+1}$, using matrices $\tilde{\mathbf{P}}_{(\#_p - r) \times (\#_p - r)}$ and $\tilde{\mathbf{Q}}_{\#_{p+1} \times \#_{p+1}}$:

$$\tilde{\mathbf{P}} \cdot \tilde{\mathbf{B}}_{p+1} \cdot \tilde{\mathbf{Q}} = \begin{bmatrix} \tilde{\mathbf{K}}_{\tilde{r} \times \tilde{r}} & \mathbf{0}_{\tilde{r} \times (\#_{p+1} - \tilde{r})} \\ \mathbf{0}_{(\#_p - r - \tilde{r}) \times \tilde{r}} & \mathbf{0}_{(\#_p - r - \tilde{r}) \times (\#_{p+1} - \tilde{r})} \end{bmatrix}.$$

Observe that $\tilde{\mathbf{K}}$ is also diagonal. Let $\mathbf{R}_{\#_p \times \#_p} = \begin{bmatrix} \mathbf{1}_{r \times r} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{P}} \end{bmatrix}$. We can then write:

$$\mathbf{R} \cdot \mathbf{Q}^{-1} \cdot \mathbf{B}_{p+1} \cdot \tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{0}_{r \times \tilde{r}} & \mathbf{0}_{r \times (\#_{p+1} - \tilde{r})} \\ \tilde{\mathbf{K}}_{\tilde{r} \times \tilde{r}} & \mathbf{0}_{\tilde{r} \times (\#_{p+1} - \tilde{r})} \\ \mathbf{0}_{(\#_p - r - \tilde{r}) \times \tilde{r}} & \mathbf{0}_{(\#_p - r - \tilde{r}) \times (\#_{p+1} - \tilde{r})} \end{bmatrix}.$$

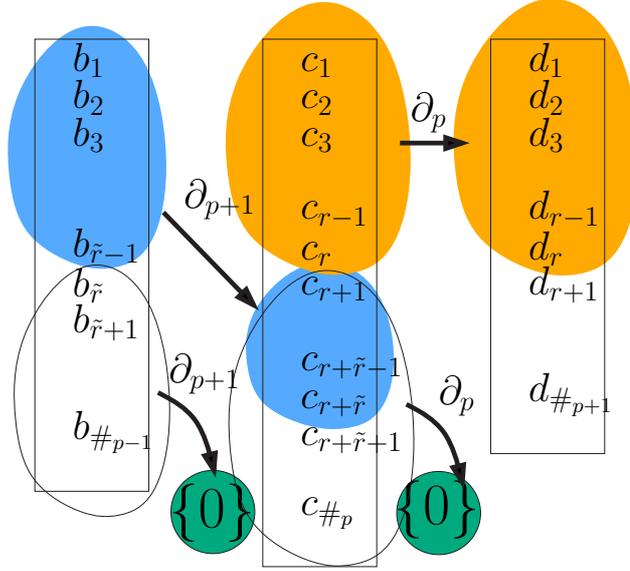


Figure IV.1: Two Smith normal forms are used to decompose the boundary operator.

Now we can build the bases $\{\mathbf{d}_i\}$, $\{\mathbf{c}_i\}$ and $\{\mathbf{b}_i\}$ for C_{p-1} , C_p and C_{p+1} respectively (figure IV.1). Let $\{\tau_i\} = \{\tau^{p-1} \in K\}$, $\{\sigma_i\} = \{\sigma^p \in K\}$ and $\{\rho_i\} = \{\rho^{p+1} \in K\}$ be the canonical basis of C_{p-1} , C_p and C_{p+1} . Then $\mathbf{d}_i = \mathbf{P}^{-1} \cdot [\tau_j]_j$, $\mathbf{c}_i = \mathbf{Q}\mathbf{R}^{-1} \cdot [\sigma_j]_j$ and $\mathbf{b}_i = \tilde{\mathbf{Q}} \cdot [\rho_j]_j$.

We can then express ∂_p and ∂_{p+1} on these bases:

$$\begin{aligned}\partial_p(\mathbf{c}_i) &= (\mathbf{P}^{-1}\mathbf{P}) \cdot \mathbf{B}_p \cdot (\mathbf{QR}^{-1}\sigma_i) = \mathbf{P}^{-1} \cdot \begin{bmatrix} \mathbf{K}_{r \times r} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{1}_{r \times r} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{P}}^{-1} \end{bmatrix} \cdot \sigma_i \\ &= \begin{cases} k_i \mathbf{d}_i & i \leq r \\ 0 & i > r \end{cases} \\ \partial_{p+1}(\mathbf{b}_i) &= (\mathbf{QR}^{-1} \cdot \mathbf{RQ}^{-1}) \cdot \mathbf{B}_{p+1} \cdot \tilde{\mathbf{Q}} \tau_i = \mathbf{QR}^{-1} \cdot \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \tilde{\mathbf{K}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \cdot \tau_i \\ &= \begin{cases} \tilde{k}_i \mathbf{c}_{r+i} & i \leq \tilde{r} \\ 0 & i > \tilde{r} \end{cases}.\end{aligned}$$

The homology group $H_p = \ker \partial_p / \text{Im } \partial_{p+1}$ can then be written in terms of the coefficient of \mathbf{K} and $\tilde{\mathbf{K}}$: $H_p = \langle \mathbb{K}_{\tilde{k}_1} \mathbf{c}_{r+1}, \mathbb{K}_{\tilde{k}_2} \mathbf{c}_{r+2}, \dots, \mathbb{K}_{\tilde{k}_{\tilde{r}}} \mathbf{c}_{r+\tilde{r}} \rangle$. Observe that since $\partial_p(\mathbf{c}_i) = 0$ for $i > r$, the basis chains \mathbf{c}_i are cycles for $i > r$.

(b) Boundary operator on the Morse complex

Consider now a cell complex K , with a discrete gradient \mathcal{V} defined on it. \mathcal{V} is not necessarily optimal, but the homology computation will be more efficient if it has less critical cells. We would like to compute explicitly the boundary operator $\tilde{\partial}_p : M_p \rightarrow M_{p-1}$ on the Morse complex, where M_p is the free module generated by the critical p -cells of \mathcal{V} . Recalling the diagram of section II.8(b) *Witten–Morse homology*:

$$\begin{array}{ccccccccccc} \{0\} & \xleftarrow{\partial_0} & C_0 & \xleftarrow{\partial_1} & C_1 & \xleftarrow{\partial_2} & C_2 & \xleftarrow{\partial_3} & \dots & \xleftarrow{\partial_n} & C_n & \xleftarrow{\partial_{n+1}} & \{0\} \\ & & \Phi^\infty \downarrow & & \text{Id} \uparrow & \Phi^\infty \downarrow & \text{Id} \uparrow & \Phi^\infty \downarrow & & \text{Id} \uparrow & \Phi^\infty \downarrow & & \text{Id} \uparrow \\ \{0\} & \xleftarrow{\tilde{\partial}_0} & C_0^\Phi & \xleftarrow{\tilde{\partial}_1} & C_1^\Phi & \xleftarrow{\tilde{\partial}_2} & C_2^\Phi & \xleftarrow{\tilde{\partial}_3} & \dots & \xleftarrow{\tilde{\partial}_n} & C_n^\Phi & \xleftarrow{\tilde{\partial}_{n+1}} & \{0\} \\ & & \cap \downarrow & \Phi^\infty \uparrow & \cap \downarrow & \Phi^\infty \uparrow & \cap \downarrow & \Phi^\infty \uparrow & & \cap \downarrow & \Phi^\infty \uparrow & & \cap \downarrow & \Phi^\infty \uparrow \\ \{0\} & \xleftarrow{\tilde{\partial}_0} & M_0 & \xleftarrow{\tilde{\partial}_1} & M_1 & \xleftarrow{\tilde{\partial}_2} & M_2 & \xleftarrow{\tilde{\partial}_3} & \dots & \xleftarrow{\tilde{\partial}_n} & M_n & \xleftarrow{\tilde{\partial}_{n+1}} & \{0\} \end{array}$$

we can define $\tilde{\partial}$ by $\tilde{\partial}_p(\sigma^p) = \partial_p(\Phi^\infty(\sigma^p)) \cap M_{p-1}$. Since the boundary operator commutes with the flow ($\partial_p \circ \Phi = \Phi \circ \partial_p$), we will use the following formula for the boundary operator on M_p : $\tilde{\partial}_p(\sigma^p) = \Phi^\infty(\partial_p(\sigma^p)) \cap M_{p-1}$.

In terms of layers of \mathcal{V} , the boundary of a critical p -cell is made of dual, primal and critical $(p-1)$ -cells. From corollary 3, the iterated flow of a dual cell vanishes. From equation (3) of section III.3(a) *Flow image of a critical or dual cell*, the only critical cell in the iterated flow of a critical cell τ^{p-1} is τ^{p-1} itself. We can thus compute the boundary operator of a critical cell σ^p directly

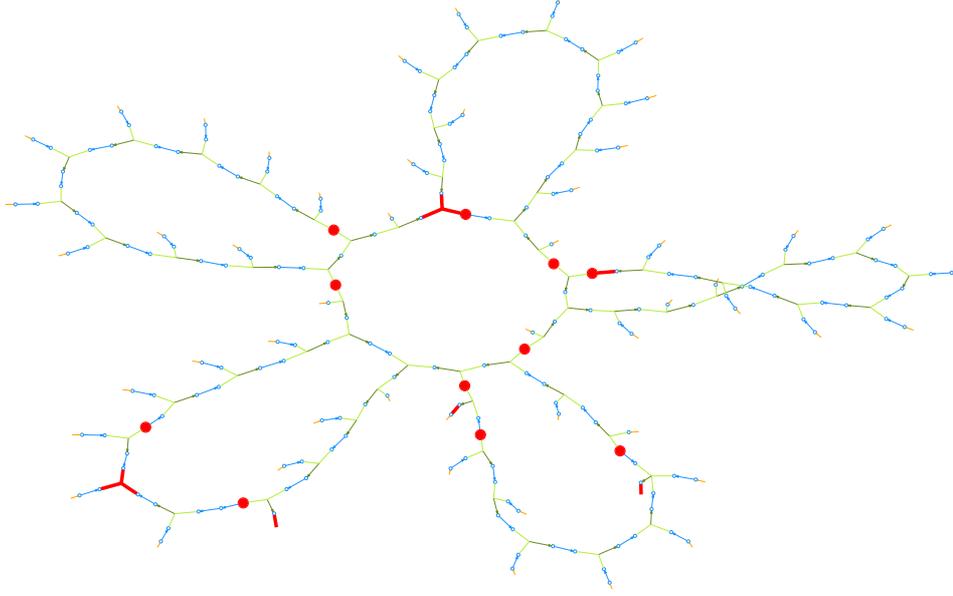


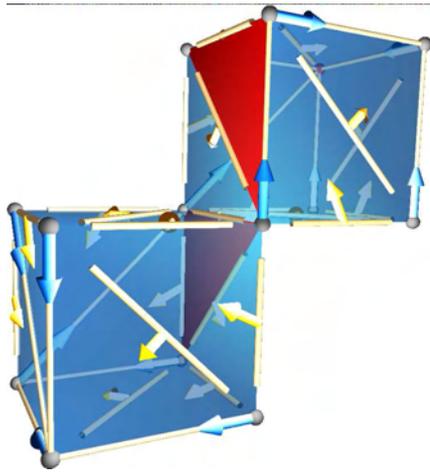
Figure IV.2: The boundary operator can be read directly on the layer looking at the connection between the critical cells as nodes (red nodes) and the critical cells as links (red links).

on layer $\mathcal{L}_{(p-1)p}$, using theorem 6 as:

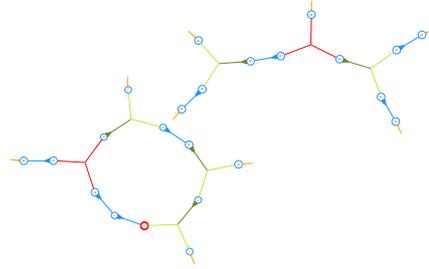
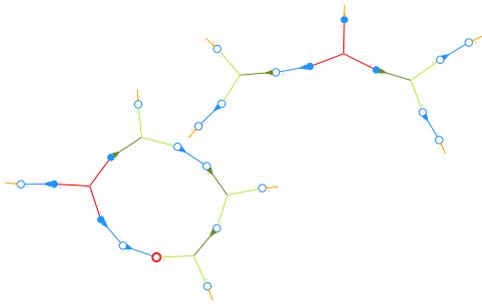
$$\begin{aligned} \tilde{\partial}_p(\sigma^p) = & \sum_{\tau^{p-1} \in \partial_p(\sigma^p) \cap \text{Crit}_{p-1}} [\sigma^p : \tau^{p-1}] \tau^{p-1} + \\ & \sum_{n \in \partial_p(\sigma^p) \cap \text{Prim}_{p-1}} \sum_{\rho \in \text{Crit}_{p-1}} \sum_{\langle n \rightsquigarrow \dots \rightsquigarrow \rho \rangle} \mu(\langle n \rightsquigarrow \dots \rightsquigarrow \rho \rangle) \cdot \rho \end{aligned} \quad (1)$$

Although equation (1) seems complex, it is easily implemented by algorithm IV.1 (figure IV.3). Since our representation of \mathcal{V} on structure III.1 maps a cell to its root, it is easier to get the iterated father of a cell than its iterated son. We will thus compute the boundary matrix \mathbf{B}_p by the co-boundary ∂^p , whose matrix is the transposed \mathbf{B}_p^T of \mathbf{B}_p . We need to modify the union procedure (algorithm III.3) to store directly at the gradient creation the multiplicity of the unique path of a cell to its root:

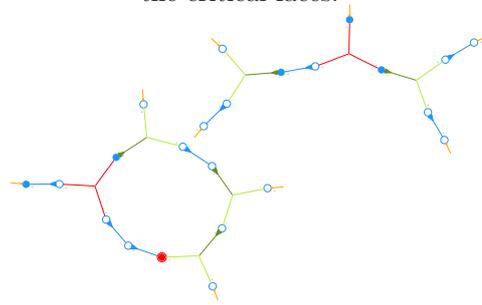
$$(\sigma.\text{basin})_\mu = \sum_{\langle \sigma \rightsquigarrow \dots \rightsquigarrow \sigma.\text{basin} \rangle} \mu(\langle \sigma \rightsquigarrow \dots \rightsquigarrow \sigma.\text{basin} \rangle) \quad .$$



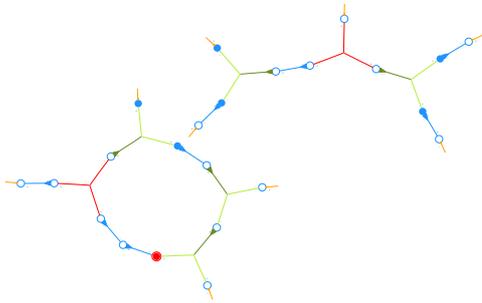
IV.3(a): Original model.

IV.3(b): Labeling of layer \mathcal{L}_{12} with the critical faces.

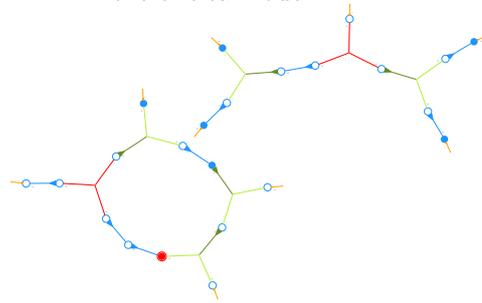
IV.3(c): Looks in the star of the critical links.



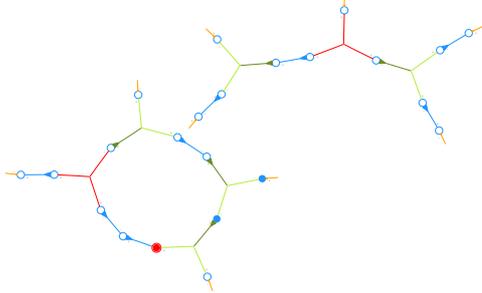
IV.3(d): Takes the root of the active cell: there is a negative incidence on the critical node.



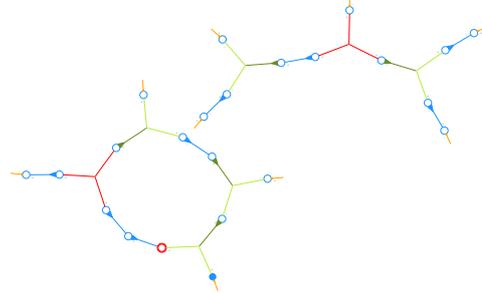
IV.3(e): If the active cell is not critical, iterates on its matched cell.



IV.3(f): Takes the root again.



IV.3(g): Iterate: the top critical link has an empty boundary on the Morse complex.



IV.3(h): The matched hyperlink cancels the critical node with a positive incidence.

Figure IV.3: Example of boundary (algorithm IV.1) execution: the execution looks synchronous but a stack allows computing each part separately.

IV.2 Complete homology calculus

The algorithms we will introduce now are direct applications of section II.8 *Morse complexes*, using the optimal construction of section III.2(c) *Heuristic for optimal Morse functions* to reduce the computational costs. They first compute an optimal discrete gradient, and then compute the homology on the Morse complex generated. Since this Morse complex is generally composed of only a few cells, any algorithm will be almost instantaneous. These algorithms actually work with any discrete gradient, but they will be more efficient on optimal ones. For example, to use the Smith normal form method, all we need is the matrix representation of the boundary operator we just computed. The iterated flow then links back the homology of the Morse complex onto the original complex. It can be efficiently computed on the hypergraph structure we introduced in the last chapter, using the results of equation (3), corollary 3 and theorem 6 of section III.3 *Flow basins and hypergraph components*.

(a) Betti numbers and torsion

If we only need the Betti numbers over \mathbb{Z}_2 or \mathbb{Q} or \mathbb{R} , we do not need to compute the whole Smith normal form, since there is no torsion. In that case, it is sufficient to compute the ranks of \mathbf{B}_p and \mathbf{B}_{p+1} to get $\beta_p = \text{rank } \mathbf{B}_p - \text{rank } \mathbf{B}_{p+1}$. In the general case, the complete homology group is composed of the free part of degree β_p plus the torsion. It can be computed by algorithm IV.2. Observe that although the algorithm calls the **optimal** procedure, it works even if the discrete gradient is not optimal.

Algorithm IV.2 homology(\mathbb{K}) : compute the homology groups

```

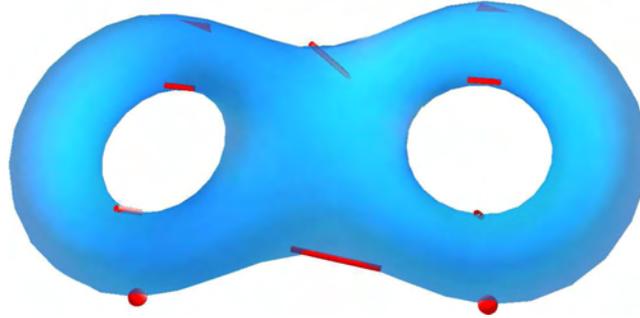
1: optimal(0) // Computes an optimal discrete gradient
2:  $\mathbf{B}_{-1} \leftarrow 0_{\mathbb{K}}$  // For joint reduction
3: for  $p \in \{0, \dots, \dim K\}$  do // Get the matrices of the boundary operator
4:    $\mathbf{B}_p \leftarrow \text{boundary}(p)$  // Original matrix
5:    $(\tilde{\mathbf{B}}_p, \mathbf{P}_p, \mathbf{Q}_p) \leftarrow \text{smith}(\mathbf{B}_p, \mathbf{B}_{p-1})$  // Normal form on the basis of  $\mathbf{B}_{p-1}$ 
6:    $r_p = \text{rank } \tilde{\mathbf{B}}_p$  // Rank of the boundary operator
7:   for  $i \in \{r_{p-1}, \dots, r_{p-1} + r_p\}$  do // Homology group
8:      $\mathbf{c}_{p-1,i} = \mathbf{Q}_{p-1} \mathbf{P}_p^{-1} \cdot [\sigma_j]_j$  // Compute the basis
9:      $k_{p-1,i} = \tilde{\mathbf{B}}_p[i][i]$  // Compute the basis
10:  end for
11:   $\beta_{p-1} = \text{card} \{k_{p-1,i} = 1_{\mathbb{K}}, i \in \{r_{p-1}, \dots, r_{p-1} + r_p\}\}$  // Betti number
12: end for
13: return  $\left\{ H_p = \langle \mathbb{K}_{k_{p,1}} \mathbf{c}_{p,r_p+1}, \dots, \mathbb{K}_{k_{p,r_{p+1}}} \mathbf{c}_{p,r_p+r_{p+1}} \rangle, p \in \{0, \dots, \dim K\} \right\}$ 

```

The complexity of algorithm IV.2 is the one of the **optimal** procedure,

$O(\#K \cdot \alpha(\#K))$ (section III.2(d) *Complexity*) plus the complexity of the Smith normal form on the critical cells, which is in the worst case $O(m_p^3)$. Observe that this algorithm computes the Betti numbers, the torsion and a set of generators for the homology on the Morse complex.

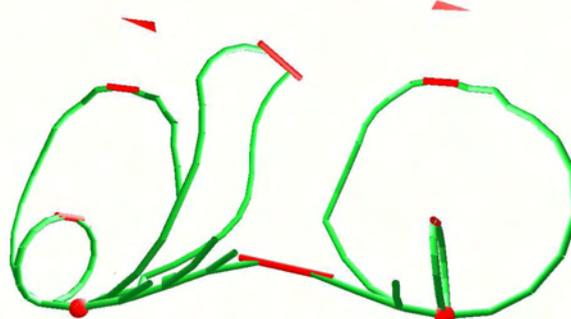
(b) Generators



IV.4(a): Critical cells.



IV.4(b): Pruned iterated flow of the two critical faces.



IV.4(c): Pruned iterated flow of the six critical edges.

Figure IV.4: Almost pruned iterated flow images of the critical faces and edges on a double torus.

The elements $\mathbf{c}_{p-1,i}$ are the generators of the homology on the Morse complex: in the simple cases, $\mathbf{c}_{p-1,i}$ is simply one critical cell. In order to obtain a set of generators on the original complex, we need to map them back onto the original complex K . We know from section II.8(b) *Witten–Morse homology* that this mapping involves the iterated flow Φ^∞ . The invariant chain $\Phi^\infty(\mathbf{c}_{p-1,i})$ always contains a generator of the homology groups on the original complex (figure IV.4), but it generally contains other cells and is therefore not

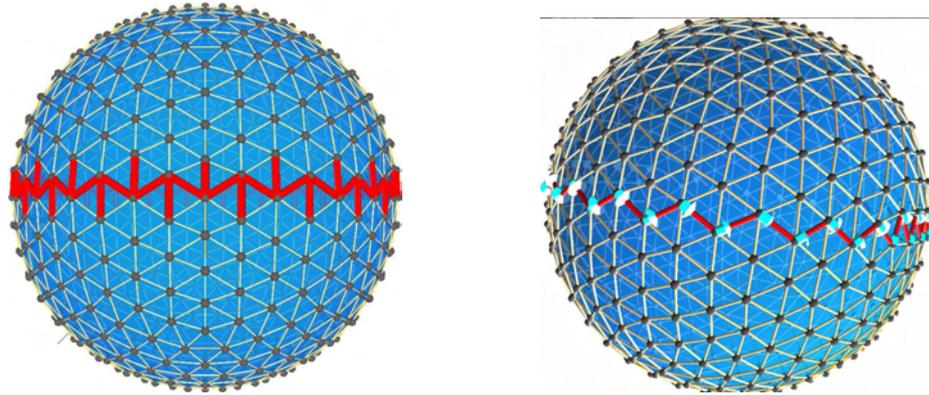
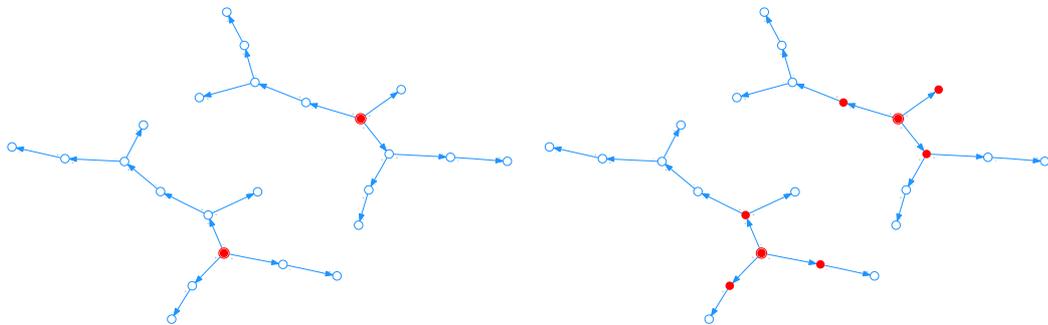
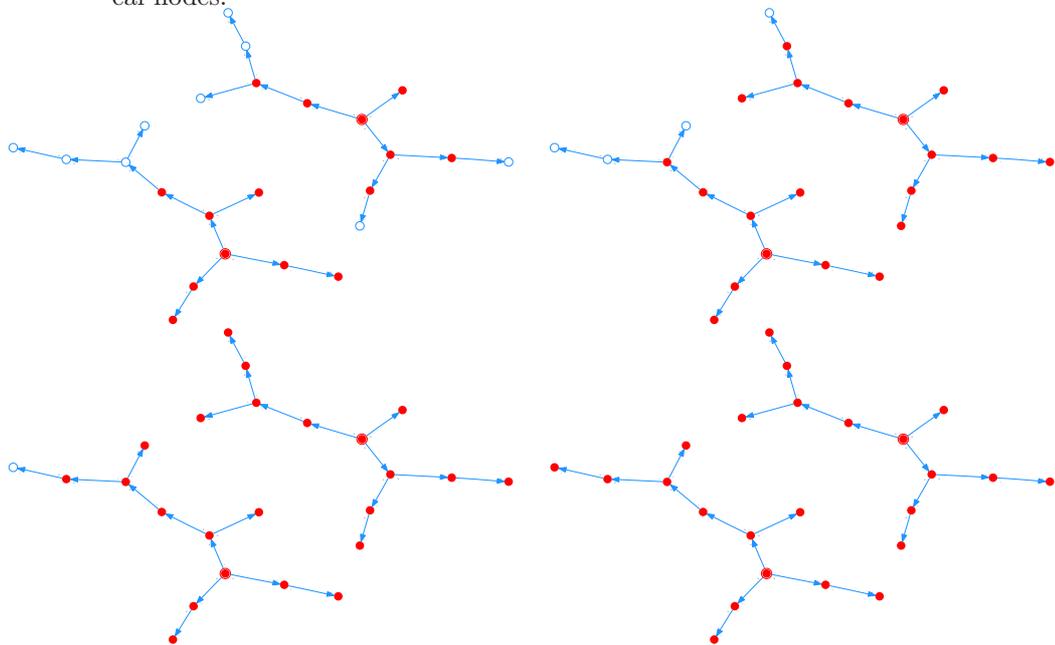


Figure IV.5: Pruning a chain to get a cycle.



IV.6(a): On layer \mathcal{L}_{21} of figure IV.3(a), the invariant chains begins at the critical nodes.

IV.6(b): Adding the sons (with their sign) to the invariant chain.



IV.6(f): The resulting invariant chain is already pruned on that simple example.

Figure IV.6: Example of generators (algorithm IV.3) execution.

a cycle: $\partial_p(\Phi^\infty(\mathbf{c}_{p-1,i})) \neq 0$. In order to get a set of generators on the original complex, the simplest method is to prune the invariant chains $\Phi^\infty(\mathbf{c}_{p-1,i})$ until they become cycle (figure IV.5). At each step of the pruning, a cell is removed from the chain $\Phi^\infty(\mathbf{c}_{p-1,i})$ if it is adjacent to only one other cell of $\Phi^\infty(\mathbf{c}_{p-1,i})$. Algorithm IV.3 and figure IV.6 illustrate this procedure.

Algorithm IV.3 $\text{generators}(\mathbf{c}_{p,i})$: computes a set of generators on the original cell complex

```

1: // Compute the iterated flow
2: stack {Dualp ∪ Critp × ℚ} ← ∅ // Stores flow front
3: for (σ, μ) ∈ cp,i do // For each cell of the chain
4:   stack.push((σ, μ)) // Initial chain
5: end for
6: b ← 0 // Iterated flow of cp,i
7: while stack ≠ ∅ do // Iterate the flow
8:   (σ, μ) ← stack.top // Get a cell of the chain
9:   b ← b + μ · σ // Adds the cell to the invariant chain
10:  for τ ∈ σ.bdry ∩ Primp do // Links of σ in Lp(p-1)
11:    σ' ← τ.match // Unique outgoing node of the hyperlink
12:    μ ← [τ : σ] · [τ : σ'] · μ // Path multiplicity
13:    stack.push((σ', μ)) // Stores for path extension
14:  end for
15: end while
16:
17: // Prune the invariant chain
18: boolean invariant ← false // Invariance marker
19: repeat
20:   boolean invariant ← true // Not changed yet
21:   for τ ∈ ∂b do // cells that prevent b from being a cycle
22:     b ← b \ τ.star // Prune the generator of the spurious boundary
23:     invariant ← false // Invariant when ∂b = 0
24:   end for
25: until invariant // Pruned

```

(c) Cycles decomposition

With this efficient computation of homology groups, we can give a first classification of arbitrary cell complexes. The next step would be to characterize elements on these complexes. For example, if we would like to match two shapes, we would first match their topological features, and then other features such as geometrical ones. To maintain the geometric matching coherent with the topological one, it is necessary to check the topology of the geometrical feature. If the feature is a closed curve on a surface, such as sharp edges, we cannot map a curve that makes 2 turns around a handle with a curve

that completes only one turn. This kind of characterization can be achieved through homology, by computing the decomposition of those curves.

The method for computing the decomposition of cycles is based on the same idea as the cycle generation. Given a chain \mathbf{c}_p , we apply the boundary operator to check if it is a cycle: $\partial_p(\mathbf{c}_p) = 0$. If so, we map that chain onto the Morse complex through Φ^∞ , and compute its homology by applying the matrices of the Smith normal form. Since each of these operations is linear on the chain module, we can actually pre-compute for each cell of the cell complex its image through Φ^∞ applying the transformations of the Smith normal form. From corollary 3 the image of dual cells by Φ^∞ vanishes. Therefore, once the chain is identified as a cycle, we only need to look at the primal and critical cells of the cycle. The only critical cell in the iterated flow image of a critical cell σ is σ itself, and thus the case of critical cell is trivial. Finally, from theorem 6, the flow of a primal cell does not vanish if there is a critical iterated son of this primal cell in $\mathcal{L}_{(p-1)p}$. Therefore, it is sufficient to check only the ancestors of critical nodes in $\mathcal{L}_{(p-1)p}$.

Algorithm IV.4 precomputation : pre-computes the mapping of each cell onto the homology basis of the cell complex

```

1: for  $p \in \{0, \dots, \dim K\}$  do                                     // For all the cells of  $K$ 
2:   for  $\sigma^p \in \text{Dual}_p$  do                                       // Dual cells
3:      $\sigma^p.\text{homo} \leftarrow 0$                                        // Null on the Morse complex
4:   end for
5:   for  $\sigma^p \in \text{Crit}_p$  do                                         // Critical and primal cells
6:      $\sigma^p.\text{homo} \leftarrow \mathbf{Q}_p \mathbf{P}_{p+1}^{-1} \cdot \sigma^p$          // Decomposed on  $\mathbf{c}_{i,p}$ 
7:      $\text{stack} \{ \text{Prim}_p \cup \text{Crit}_p \times \mathbb{K} \} \leftarrow \{ (\sigma^p, 1_{\mathbb{K}}) \}$  // Stores the ancestors of  $\sigma^p$ 
8:     while  $\text{stack} \neq \emptyset$  do                                       // Hyperpath traversals
9:        $(\sigma', \mu) \leftarrow \text{stack.top}$                                // Get the first cell of the path
10:      for  $\tau^{p+1} \in \sigma'.\text{star} \cap \text{Dual}_p$  do                       // Links of  $\sigma'$  in  $\mathcal{L}_{p(p+1)}$ 
11:         $\rho^p \leftarrow \tau^{p+1}.\text{match}$                                // Unique outgoing node of the hyperlink
12:         $\mu \leftarrow [\tau^{p+1} : \rho^p] \cdot [\tau^{p+1} : \sigma'] \cdot \mu$  // Path multiplicity
13:         $\rho^p.\text{homo} \leftarrow \mu \cdot \sigma^p.\text{homo} + \rho^p.\text{homo}$        // Decomposition
14:         $\text{stack.push}((\rho^p, \mu))$                                        // Stores the path extension
15:      end for
16:    end while
17:  end for
18: end for

```

The cycle decomposition is then obtained by summing all the pre-computed terms of the cells of the cycle. The whole algorithm is thus linear with respect to the size of the chain, and the pre-computation is linear in terms of the size of the complex.

All together, this gives a fast method to pre-compute the mapping from the original complex to the homology basis of the Morse complex. This

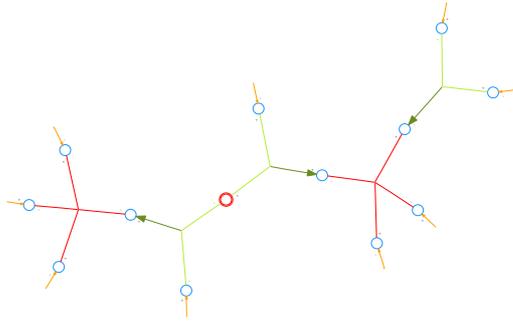
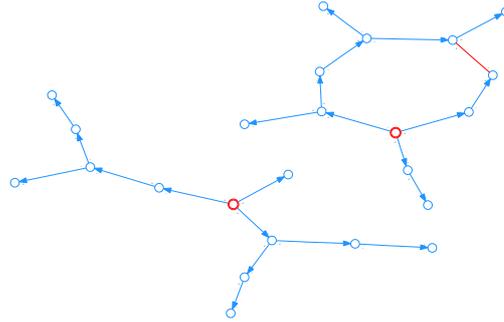
IV.7(a): \mathcal{L}_{10} with critical linksIV.7(b): \mathcal{L}_{21} with critical links

Figure IV.7: The pre-computation of a critical cell is simply itself. On the layers \mathcal{L}_{10} and \mathcal{L}_{21} of figure IV.3(a) with the critical links, we see clearly that each non-critical primal 0-cell will have as homology class exactly one of the two critical 0-cells, while only some critical faces could create a non-trivial cycle.

procedure is illustrated by algorithm IV.4 (figure IV.7). Note that the central part of algorithm IV.4 is similar to algorithm IV.1, but the primal and dual layers are reversed. This reflects the fact that this pre-computation looks like a co-cycle computation, although this theoretical property is still not completely clear to us.

(d) Computational results

Tables IV.8 and IV.9 and IV.10 show some of the results of algorithm IV.2, used to compute the only Betti numbers. The computational times include the whole process, from the optimal Morse function generation (without the priority queue optimization) to the boundary operator computation and the matrix transforms. Particularly for the 3D case, the number of locks corresponds to the number of heuristic choices performed during the optimal Morse function computation, i.e. the number of times line III.8 of algorithm III.8 is called. The 2D models were chosen to be representative and of quite big size. On the contrary, none of the 3D models is of very big size, but they have a complex topology. In particular, Poincaré's homological sphere has a more complex topology than what is noticed by the homology. Trickier, the non-shellable 3-sphere (NC Sphere) is a delicate model since no discrete Morse function can reach the minimal number of critical points for smooth homotopy. Nevertheless, the homology computation performs well even on those models.

Model	(# ₀ , # ₁ , # ₂)	χ	(m_0, m_1, m_2)	($\beta_0, \beta_1, \beta_2$)	time
alien 20kv	(19198, 57552, 38368)	14	(7, 0, 7)	(7, 0, 7)	1.002
alien 20kv r	(17954, 53725, 35772)	1	(1, 0, 0)	(1, 0, 0)	0.929
alien 40kv	(38256, 114658, 76403)	1	(1, 0, 0)	(1, 0, 0)	2.037
ant31850	(1147, 2783, 1850)	214	(108, 0, 106)	(108, 0, 106)	0.038
bear	(29642, 88270, 58630)	2	(1, 0, 1)	(1, 0, 1)	1.225
boris	(11034, 33141, 22095)	-12	(1, 14, 1)	(1, 14, 1)	0.448
bunny-closed	(15005, 44973, 29969)	1	(1, 0, 0)	(1, 0, 0)	0.708
bunny	(15000, 44786, 29783)	-3	(1, 4, 0)	(1, 4, 0)	0.701
bunnyR	(34834, 104288, 69451)	-3	(1, 4, 0)	(1, 4, 0)	1.689
bunny r	(99999, 299320, 199322)	1	(1, 0, 0)	(1, 0, 0)	4.608
camel1	(2579, 7458, 4884)	5	(14, 9, 0)	(14, 9, 0)	0.101
camel2	(1865, 5440, 3576)	1	(9, 8, 0)	(9, 8, 0)	0.074
cow	(2904, 8706, 5804)	2	(1, 0, 1)	(1, 0, 1)	0.113
dilo	(27174, 81516, 54356)	14	(1, 6, 19)	(1, 0, 19)	1.188
dino7400	(3703, 11102, 7400)	1	(1, 0, 0)	(1, 0, 0)	0.160
dinosaur	(14070, 42204, 28136)	2	(1, 0, 1)	(1, 0, 1)	0.606
feline	(49864, 149598, 99732)	-2	(1, 4, 1)	(1, 4, 1)	2.168
hammerhead	(2564, 7688, 5150)	26	(1, 1, 26)	(1, 1, 26)	0.106
head-dragon	(19119, 57107, 37989)	1	(1, 0, 0)	(1, 0, 0)	0.750
hcamel2 sl2	(12333, 36940, 24608)	1	(1, 0, 0)	(1, 0, 0)	0.521
hcamel2 tri	(10033, 30008, 19976)	1	(1, 0, 0)	(1, 0, 0)	0.444
horse	(48485, 145449, 96966)	2	(1, 0, 1)	(1, 0, 1)	2.126
hound	(12515, 37531, 25030)	14	(6, 19, 27)	(6, 19, 27)	0.528
jurassic	(5000, 13276, 8284)	8	(27, 19, 0)	(27, 19, 0)	0.181
mondo	(2769, 6712, 4084)	141	(137, 15, 19)	(137, 15, 19)	0.099
octopus	(16944, 50808, 33872)	8	(4, 0, 4)	(4, 0, 4)	0.745
open bunny	(30344, 90652, 60309)	1	(1, 0, 0)	(1, 0, 0)	1.658
pig	(1843, 5408, 3560)	-5	(1, 6, 0)	(1, 6, 0)	0.073
pig2	(3522, 10560, 7040)	2	(1, 0, 1)	(1, 0, 1)	0.157
pig r1	(20004, 59965, 39962)	1	(1, 0, 0)	(1, 0, 0)	0.872
pig r2	(20006, 59984, 39979)	1	(1, 0, 0)	(1, 0, 0)	0.864
pig r3	(4999, 14970, 9972)	1	(1, 0, 0)	(1, 0, 0)	0.212
pig r4	(50005, 149949, 99945)	1	(1, 0, 0)	(1, 0, 0)	2.216
pig r5	(100001, 299915, 199915)	1	(1, 0, 0)	(1, 0, 0)	4.508
rhino	(8071, 24058, 16031)	44	(26, 6, 24)	(26, 6, 24)	0.331
snail1850	(1013, 2879, 1849)	-17	(6, 34, 11)	(6, 34, 11)	0.041
triceratops	(2832, 8490, 5660)	2	(1, 0, 1)	(1, 0, 1)	0.113
triceratops2	(2832, 8490, 5660)	2	(1, 0, 1)	(1, 0, 1)	0.105

Table IV.8: Computational results on classical models of computer graphics. The computational time are expressed in seconds.

Model	(# ₀ , # ₁ , # ₂)	χ	(m_0, m_1, m_2)	($\beta_0, \beta_1, \beta_2$)	time
beethoven	(2655, 5461, 2812)	6	(8, 4, 2)	(8, 4, 2)	0.077
david	(24085, 71837, 47753)	1	(1, 0, 0)	(1, 0, 0)	1.159
david100kf	(50059,150042,100000)	17	(27, 11, 1)	(27,11, 1)	2.815
david24k	(24085, 71837, 47753)	1	(1, 0, 0)	(1, 0, 0)	1.158
david50kf	(24988, 74985, 49988)	-9	(1, 10, 0)	(1,10, 0)	1.372
david r1	(30058, 90013, 59956)	1	(1, 0, 0)	(1, 0, 0)	1.322
david r2	(3107, 9302, 6196)	1	(1, 0, 0)	(1, 0, 0)	0.130
david r3	(10022, 30039, 20018)	1	(1, 0, 0)	(1, 0, 0)	0.432
david r4	(29768, 89228, 59461)	1	(1, 0, 0)	(1, 0, 0)	1.305
david r5	(49672,148923, 99252)	1	(1, 0, 0)	(1, 0, 0)	2.161
david r7	(5118, 15287, 10170)	1	(1, 0, 0)	(1, 0, 0)	0.218
david r8	(315, 927, 613)	1	(1, 0, 0)	(1, 0, 0)	0.012
david r9	(30, 82, 53)	1	(1, 0, 0)	(1, 0, 0)	0.003
david r10	(50327,150780,100454)	1	(1, 0, 0)	(1, 0, 0)	2.258
david r11	(50327,150780,100454)	1	(1, 0, 0)	(1, 0, 0)	2.259
david r12	(50054,150129,100076)	1	(1, 0, 0)	(1, 0, 0)	2.224
david r13	(50352,150826,100475)	1	(1, 0, 0)	(1, 0, 0)	2.261
egea	(8268, 24798, 16532)	2	(1, 0, 1)	(1, 0, 1)	0.356
egea r1	(3133, 9259, 6127)	1	(1, 0, 0)	(1, 0, 0)	0.129
face-o	(13746, 40895, 27150)	1	(1, 0, 0)	(1, 0, 0)	0.567
face	(17357, 51663, 34308)	2	(1, 0, 1)	(1, 0, 1)	0.725
gargoyle	(30059, 89998, 59940)	1	(1, 0, 0)	(1, 0, 0)	1.353
head1 o	(16374, 49116, 32744)	2	(1, 0, 1)	(1, 0, 1)	0.685
holes	(4291, 12584, 8288)	-5	(1, 6, 0)	(1, 6, 0)	0.181
holes r1	(3009, 8701, 5687)	-5	(1, 6, 0)	(1, 6, 0)	0.124
lion-dog	(24930, 74981, 50000)	-51	(1, 57, 5)	(1,57, 5)	1.208
mannequin	(1694, 5049, 3356)	1	(1, 0, 0)	(1, 0, 0)	0.071
mannequin r1	(8188, 24487, 16300)	1	(1, 0, 0)	(1, 0, 0)	0.345
mannequin r2	(7989, 23919, 15931)	1	(1, 0, 0)	(1, 0, 0)	0.336
mannequin r3	(7971, 23864, 15894)	1	(1, 0, 0)	(1, 0, 0)	0.336
mannequin r4	(3001, 8964, 5964)	1	(1, 0, 0)	(1, 0, 0)	0.126
maxplanck	(23609, 70690, 47082)	1	(1, 0, 0)	(1, 0, 0)	1.040
maxplanck5	(19767, 59199, 39433)	1	(1, 0, 0)	(1, 0, 0)	0.855
maxplanck r	(50018,149849, 99832)	1	(1, 0, 0)	(1, 0, 0)	2.271
nef-smooth	(2615, 7740, 5126)	1	(1, 0, 0)	(1, 0, 0)	0.108
pieta	(13940, 41856, 27904)	-12	(1, 14, 1)	(1,14, 1)	0.614
venus	(711, 2106, 1396)	1	(1, 0, 0)	(1, 0, 0)	0.027

Table IV.9: Computational results on scanned sculptures. The computational time are expressed in seconds.

Model	($\#_0, \#_1, \#_2, \#_3$)	χ	(m_0, m_1, m_2, m_3)	$(\beta_0, \beta_1, \beta_2, \beta_3)$	time	lock
1 cube	(8, 12, 6, 1)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.003	
1 cubeT	(8, 19, 18, 6)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.002	
4 cubes	(18, 33, 20, 4)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.003	
4 cubesT	(18, 57, 64, 24)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.004	
9 cubes	(32, 64, 42, 9)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.003	
9 cubesT	(32, 115, 138, 54)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.007	
36 cubes	(80, 184, 141, 36)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.007	
36 cubesT	(80, 361, 498, 216)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.025	
360 Cubes	(572,1477,1266, 360)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.263	
360 CubesT	(572,3103,4692,2160)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.911	
solid torus	(32, 64, 40, 8)	0	(1, 1, 0, 0)	(1, 1, 0, 0)	0.003	
solid torusT	(32, 112, 128, 48)	0	(1, 1, 0, 0)	(1, 1, 0, 0)	0.005	
solid \mathbb{S}^2	(64, 144, 108, 26)	2	(1, 0, 1, 0)	(1, 0, 1, 0)	0.007	2
solid \mathbb{S}^2 T	(64, 278, 372, 156)	2	(1, 0, 1, 0)	(1, 0, 1, 0)	0.017	1
$\mathbb{S}^2 \times \mathbb{S}^1$	(192, 588, 612, 216)	0	(1, 1, 1, 1)	(1, 1, 1, 1)	0.068	1
\mathbb{S}^3	(162, 522, 576, 216)	0	(1, 0, 0, 1)	(1, 0, 0, 1)	0.056	
$\mathbb{R}^3 \setminus \text{Furch}$	(148, 292, 181, 36)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.013	
$\mathbb{R}^3 \setminus \text{FurchT}$	(148, 509, 578, 216)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.033	
Furch	(600,1580,1350, 369)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.29	
FurchT	(600,3299,4914,2214)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	1	
Bing	(480,2511,3586,1554)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.123	
Björner	(6, 15, 11,)	2	(1, 0, 1,)	(1, 0, 1,)	0.001	
c ns 3	(10, 31, 22,)	1	(1, 1, 1,)	(1, 0, 0,)	0.002	
c ns2	(13, 39, 27,)	1	(1, 0, 0,)	(1, 0, 0,)	0.003	
c ns	(12, 37, 26,)	1	(1, 1, 1,)	(1, 0, 0,)	0.002	
Dunce hat	(8, 24, 17,)	1	(1, 1, 1,)	(1, 0, 0,)	0.003	
Gruenbaum	(14, 54, 70, 29)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.003	
knot	(380,1929,2722,1172)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.084	
Lockeberg	(12, 60, 96, 48)	0	(1, 0, 0, 1)	(1, 0, 0, 1)	0.005	
NC Sphere	(381,2309,3856,1928)	0	(1, 2, 2, 1)	(1, 0, 0, 1)	0.142	2
nonextend	(7, 19, 13,)	1	(1, 0, 0,)	(1, 0, 0,)	0.001	
Poincaré	(16, 106, 180, 90)	0	(1, 2, 2, 1)	(1, 0, 0, 1)	0.007	3
Projective	(6, 15, 10,)	1	(1, 1, 1,)	(1, 1, 1,)	0.003	
Rudin	(14, 66, 94, 41)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.004	
Simon 2	(6, 15, 10,)	1	(1, 0, 0,)	(1, 0, 0,)	0.002	
Simon	(7, 20, 14,)	1	(1, 0, 0,)	(1, 0, 0,)	0.002	
solid 2-torus	(6, 12, 6,)	0	(1, 1, 0,)	(1, 1, 0,)	0.002	
walkup C	(20, 126, 212, 106)	0	(1, 0, 0, 1)	(1, 0, 0, 1)	0.006	
walkup D	(16, 106, 180, 90)	0	(1, 0, 0, 1)	(1, 0, 0, 1)	0.006	
Ziegler	(10, 38, 50, 21)	1	(1, 0, 0, 0)	(1, 0, 0, 0)	0.002	

Table IV.10: Computational results on solid models and [Hachimori Models]. Models ending with a ‘T’ are simplicial versions of the model without ‘T’. The computational time are expressed in seconds.

V

Geometric discrete Morse complex

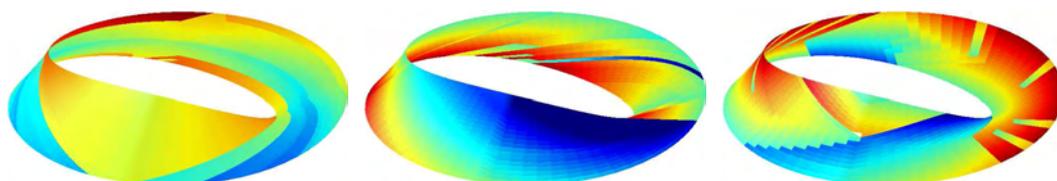
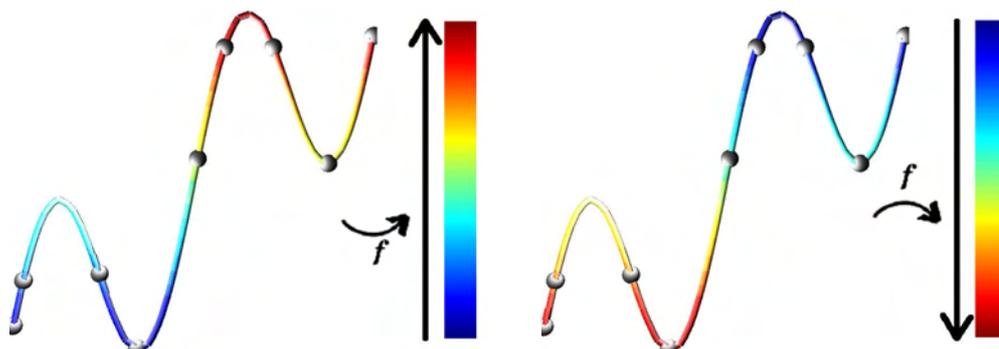


Figure V.1: Different geometries on a Möbius band: a geometric discrete Morse function must simultaneously capture the geometry and assert the topology.

The definition of a discrete Morse function (section II.3(b) *Acyclic matchings*) does not provide a straightforward way to transform a smooth Morse function $f : \mathcal{M} \rightarrow \mathbb{R}$ to a discrete one. Moreover, even if K is a cellular decomposition of a manifold \mathcal{M} : $|K| = \mathcal{M}$, a direct sampling $f|_K$ of a smooth Morse function f is not a discrete Morse function. Such a correspondence would allow using discrete Morse theory from topology to geometry and vice-versa, in a similar way to smooth Morse theory.



V.2(a): The complex is well adapted to f : the minima occur on vertices and almost all the maxima on edges.

V.2(b): The complex is not adapted to f : the minima occur out of the vertices and the maxima on vertices.

Figure V.2: If the complex is not adapted to the geometric function, the critical cells are difficult to define.

This section is an attempt to define a discrete gradient \mathcal{V} from a given scalar function on the vertices $f : K^0 \rightarrow \mathbb{R}$ that will be called “geometry”

along this chapter (figure V.1). We guarantee that our construction generates a valid discrete gradient and that, at least in the case of surfaces, the flow of \mathcal{V} is increasing with respect to f . Since the critical points of $f : \mathcal{M} \rightarrow \mathbb{R}$ may not be vertices of K , $|K| = \mathcal{M}$ (figure V.2), the notion of critical point is delicate in the discrete setting. The usual definition is due to Banchoff [Banchoff, 1967], and relies on the Euler characteristic. This definition can miss some essential critical points in high dimensions, and we will thus extend it naturally using either homology (section IV.2(a) *Betti numbers and torsion*) or directly our optimal discrete Morse function construction (section III.2(c) *Heuristic for optimal Morse functions*). The main advantages of this construction over other so-called discrete Morse–Smale decompositions [Edelsbrunner *et al.*, 2001, Edelsbrunner *et al.*, 2003] consist in its simplicity, the rigor of the construction and the generality, since the construction works on any finite cellular complex. Moreover, the construction can be complemented to guarantee the position of the critical cells and their shapes, even if this step is proved not necessary for refined surfaces. Finally, Forman’s theory ensures the homology of the resulting complex, guaranteeing a consistent result.

V.1 Geometric critical points

The definition of a critical point should follow the Morse lemma: a smooth function f near one of its critical points \mathbf{x} is locally similar to the quadratic form deduced from its Hessian matrix (section II.4(a) *Critical points*). In particular, close to a critical point \mathbf{x} of Morse index p , there is an orthonormal basis containing p directions for which \mathbf{x} is a local maximum and $n - p$ directions for which it is a local minimum. On the other hand, we cannot guarantee neither that \mathbf{x} is a vertex of K nor that the cells incident to \mathbf{x} represent each direction of the orthonormal basis. A critical point is thus defined as a break in the monotony of f (figure V.3). The Banchoff index of such critical point \mathbf{x} will be linked to the number of changes between increasing and decreasing directions around \mathbf{x} [Banchoff, 1967].

(a) Banchoff’s definition and limits

Given a function $f : K^0 \rightarrow \mathbb{R}$ defined on the vertices of a cell complex, such that f assigns different values for adjacent vertices, Banchoff gives the following definition of its critical points. Define the *lower star* $\text{lw}_f \tau$ of a vertex τ as the set of cells σ of the open star $\text{st}\tau$ of τ having all the images $f(\tau')$ of their vertices τ' lower or equal to $f(\tau)$ (figure V.4). A vertex τ is critical if

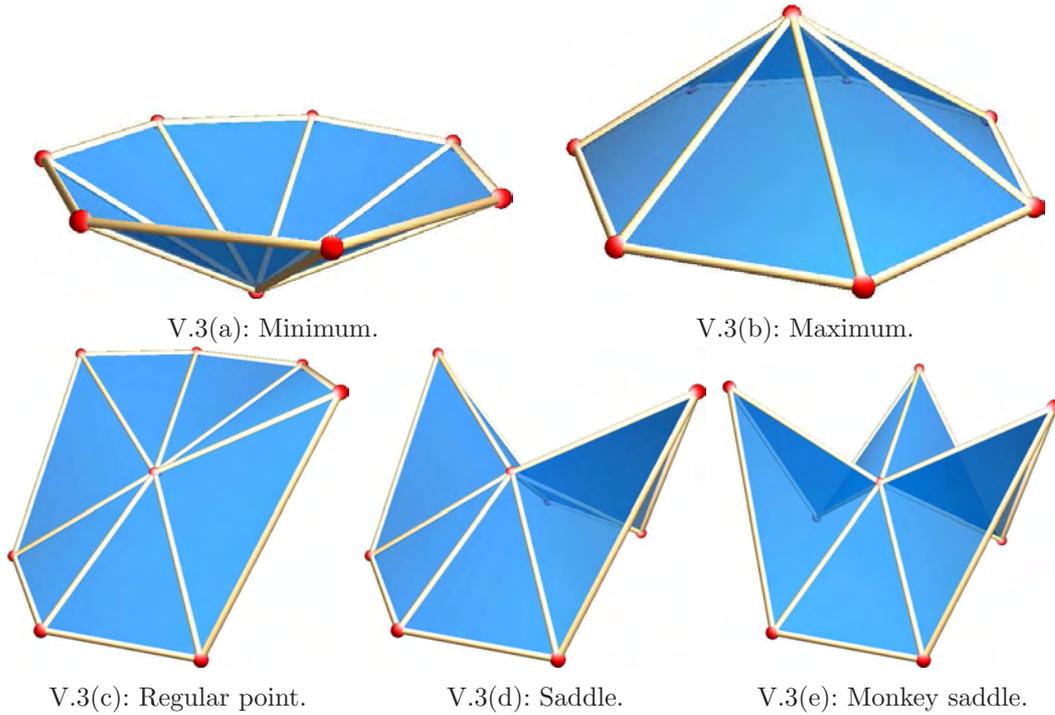


Figure V.3: Classification of generic points on surfaces, and a non-generic saddle (e).

the Euler characteristic of its lower star differs from the one of a semi-opened disk: $\chi(\text{lw}_f \tau) \neq 0$. The Banchoff index of a critical point is defined as this Euler characteristic:

$$\text{idx}(\tau) = \chi(\text{lw}_f \tau) = \sum_{p \in \mathbb{N}} (-1)^p \text{card} \{ \sigma^p > \tau, f(\sigma^p) < f(\tau) \}.$$

Observe that this definition differs from the Morse index $q(\tau)$ of a critical point τ , defined as the number of negative eigenvalues of the Hessian matrix. However, $\text{idx}(\tau) = (-1)^{q(\tau)}$ for non-degenerated critical points, and Banchoff proved in [Banchoff, 1967] that his index is still linked to the Euler characteristic of the manifold: $\chi(K) = \sum_{\tau \in K^0} \text{idx}(\tau)$.

Although this definition is simple and intuitive, we cannot use it to define rigorously a discrete Morse complex for the following reasons: it requires additional tests even to decide if a critical point is a maximum or a minimum, and in higher dimensions, the Euler characteristic is not sufficient to capture every critical point, as it does not determine whether a complex is homeomorphic to a disk or not. Therefore, there can be some critical points essential to compute the homology properly on the Morse complex that are missed by Banchoff's definition. In any case, our construction of discrete Morse complexes can use even an incomplete set of critical points, as it would automatically generate a complete set of critical cells from those critical points.

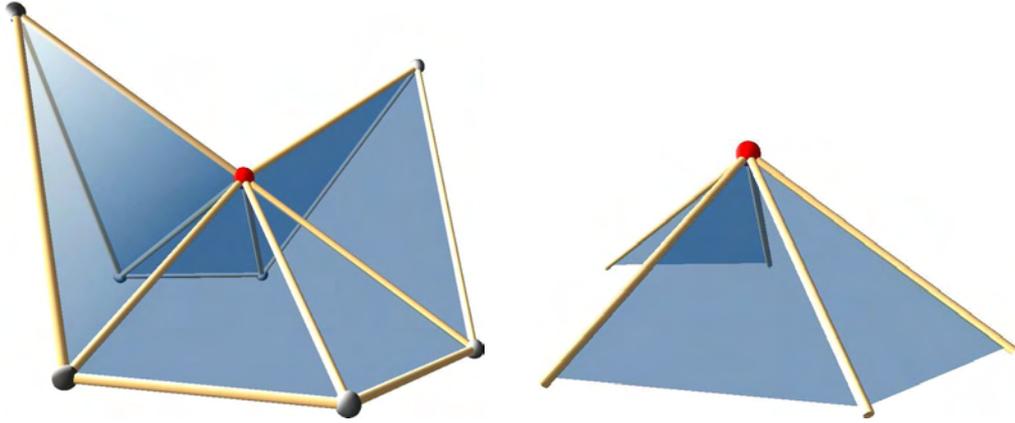


Figure V.4: Lower star of a saddle point.

(b) Critical points in high–dimension

We use two different definitions of critical points that coincide in all practical case. The first relies on homology computation and it is defined uniquely in all cases. If the homology of the lower star is $(\mathbb{K}, \{0\}, \{0\} \dots)$, the point is regular. Otherwise, it is a maximum if $\beta_0 = 0$, a minimum if $\beta_{n-1} = 1$, a 1–saddle if $\beta_0 = 2$ and a k –saddle if $\beta_{k-1} = 1$ (figure V.5). It can also be a degenerated saddle if $\beta_k > 1$ or $\beta_0 > 2$. In some particularly complex topological case, this definition can miss some critical points, for example if the lower star of a vertex is a homological disc. The homology is computed using the algorithm of section IV.2(a) *Betti numbers and torsion*.

The second definition uses our heuristic to define optimal Morse function (section III.2(c) *Heuristic for optimal Morse functions*). If the heuristic achieves defining a discrete gradient with no critical point on the lower star of a vertex, this vertex will be considered as regular. If not, the point will be critical, and its class is computed in the same way as in the first definition. Observe that even the lower star is not a cellular complex, the algorithm of section III.2 *Greedy construction* works directly. This definition is not as rigorous as the homology–based definition, but it allows considering as critical obstructions to cancelling cells such as non–shellability, non–simple homotopy features or homological discs.

Observe that both definitions coincides with Banchoff’s one for surfaces and solids, but the class of the critical point can be computed in an easier way.

(c) From critical points to critical cells

In Forman’s discrete Morse theory, the critical elements are cells in general, instead of only points. Although this point of view gave powerful results, the detection of critical cells *per se* is a little more complex than

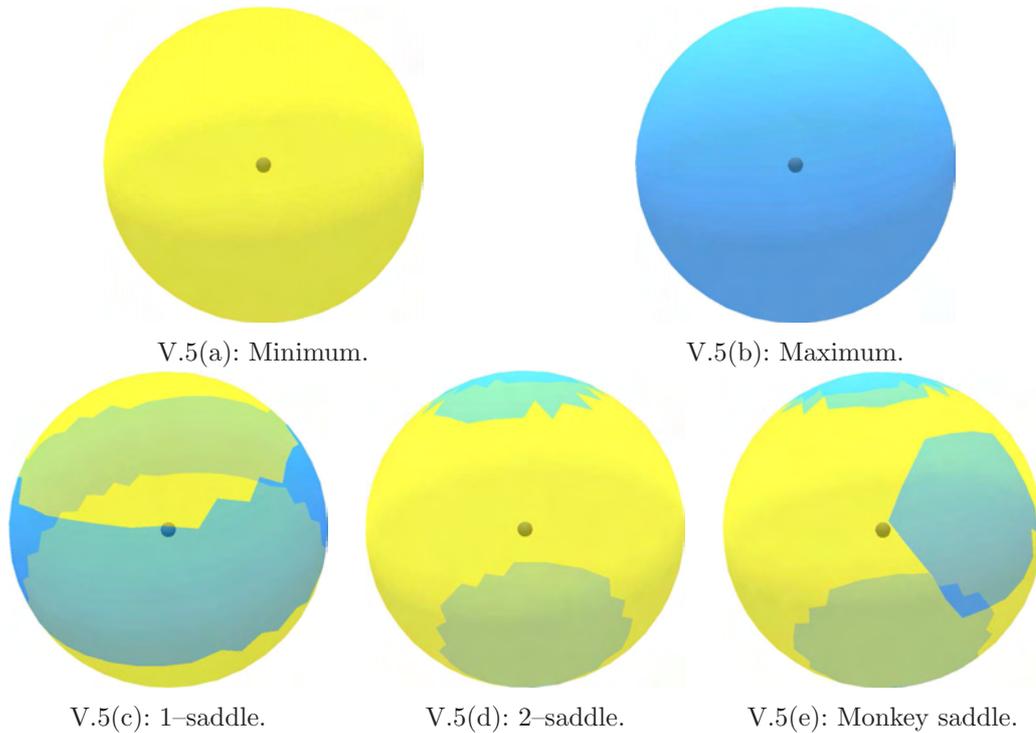


Figure V.5: Classification of generic critical points in solids, and a non-generic saddle (e): the link of the point is mapped onto a sphere, the blue parts represent the higher values, and the yellow ones the lower values.

detecting critical points. The main step of our algorithm actually does not require this detection, but it can be suited to control the output of the algorithm. There are two contexts to work on. In the first context, if the mesh contains more information than just the support for the geometry, for example when it has been generated to obtain some special features on the triangles, a critical cell has to be selected from the existing cells. In the second context where we are free to change the mesh, we will create a critical cell directly from a critical point.

The classification of critical points is essential for discrete Morse theory, since the index of a critical point determines the dimension of the corresponding critical cell. As degenerate critical points actually correspond to various critical points agglomerated, the procedure to generate a critical cell from a critical point can be repeated to identify the many critical cells of a degenerated critical point.

In both of the above contexts, the easiest part is the detection of a minimum since in Forman's theory, a minimum is a vertex which identifies easily with a point, and our algorithm will always place the minima at the right position. A maximum τ is also easily detected, but the corresponding critical cell σ must be a cell of maximal dimension n . We will choose σ as the one containing the vertices of the link of τ having the highest value

through f . Formally, if we denote $\tau_1 = \operatorname{argmax} \{f(\tau'), \tau' \in K^0 \cap \operatorname{lk} \tau\}$ and $\tau_{i+1} = \operatorname{argmax} \{f(\tau'), \tau' \in K^0 \cap \operatorname{lk} \tau_i \setminus \{\tau_j, j \leq i\}\}$, σ is the cell incident to the first τ_i 's. If the complex is simplicial, we can write $\sigma = \operatorname{span}(\tau, \tau_1, \tau_2, \dots, \tau_n)$.

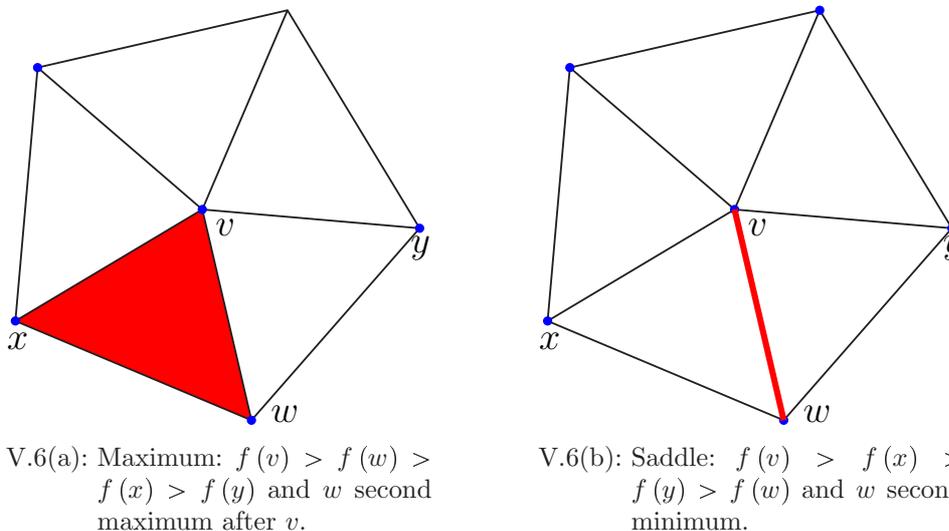


Figure V.6: Construction of critical cells from a maximum and a saddle.

The construction of a saddle relies on the following observation. In the smooth case, the flow elongates a small region around a saddle in the directions where the saddle point is a local maximum. This elongation extends on both sides of the saddle point, and the lowest side is a natural direction for the flow. The construction of a k -saddle, $k > 1$, is thus similar to the construction of a maximum, but considering the argmin instead of the argmax in the definition of τ_i .

The construction of a 1-saddle is slightly more complex, since the lower star is not connected. For each connected component i of the lower star, denote by σ_i the edge of the connected component incident to the lowest vertex. Let σ_0 be the edge incident to the lowest vertex of the whole star. Then each cell $\sigma_i, i \neq 0$ will be critical, where σ_0 is excluded because it would be the regular direction of the flow. This definition even entails degenerated saddles, such as monkey saddles.

In the second context, we can perform a vertex split operation to create the critical cell in the appropriate direction. For example a maximal point τ would be substituted by a cell of maximal dimension σ having the same combinatorics as the intersection of a small ball around τ with the star of τ . The vertices of σ (i.e. the intersections of the sphere with the edges of $\operatorname{st} \tau$) will be positioned all at the same height $f(\tau)$. A k -saddle is constructed in the same way, but considering a small k -cylinder instead of a ball.

V.2 Computation

Our algorithm computes a geometric discrete Morse complex using the greedy algorithm of section III.2 *Greedy construction*. This part of the algorithm is sufficient in most of the practical cases we tested. However, run alone, it can miss some critical cells, or generate some critical cells that do not correspond to the definition of section V.1 *Geometric critical points*. To guarantee that no critical point is missed, the construction can independently detect the critical points before the greedy part of the algorithm. Those critical points are transformed into critical cells following the description of section V.1(c) *From critical points to critical cells*. Some more critical cells can be generated by the greedy construction. They are usually induced by complex local geometry or local obstruction to shellability. Those spurious critical points can, in most cases, be cancelled using directly the results of [Forman, 1995] (section II.6 *Cancellations*). The construction is then composed of the four steps described in this section, but in practice, it only requires the third one. This actually constructs a discrete gradient aligned with f , and the discrete Morse complex can be read using the last algorithm described in this section.

(a) Critical points tracking

The critical points can be identified in three different ways, constructing the lower star of the candidate τ and computing either its Euler characteristic, or its homology or an optimal discrete Morse function on it. Algorithm V.1 details the second case, which we used for our experimental results. The third case can also be computed from algorithm V.1 considering $\beta_p = m_p$.

Although the lower star is not a cell complex, the homology algorithm and the optimal discrete Morse function algorithm work directly on these structures, since they only require the incidence operator. Or from another point of view, this corresponds to the calculus on the abstract closure of the lower star.

(b) Critical cells selection

In section V.1(c) *From critical points to critical cells*, we described two ways of creating critical cells from critical points. We will detail here the first one with the function `crittrack` (algorithm V.1), which chooses the critical cell from existing cells. The function f is pre-computed on each cell as the value of f on the barycentre $f(\sigma) = \text{mean}\{f(\tau), \tau \in K^0 \cap \sigma\}$. The case of 1-saddle is not described in algorithm V.3 since its description of last section and since it is more efficiently implemented directly from algorithm V.1.

Algorithm V.1 $\text{crittrack}(\tau, f)$: check if τ is critical

```

1:  $L = \{\sigma \succ \tau, f(\sigma) < f(\tau)\}$  // Get the lower star of  $\tau$ 
2: if  $L = \emptyset$  then // Maximum
3:   return maximum
4: end if
5:  $(\beta_i) \leftarrow \text{betti}(L)$  // Compute the Betti number of  $L$ 
6: if  $(\beta_i) = (1, 0, 0, \dots)$  then // Regular cell
7:   return regular
8: end if
9: if  $\beta = (1, 0, 0, \dots, 0, 1, 0)$ :  $\beta_{n-1} = 1$  then // Minimum
10:  return minimum
11: end if
12: if  $\beta = (2, 0, 0, \dots)$  then // 1-Saddle
13:  return saddle(1)
14: end if
15: if  $\beta = (1, 0, \dots, 0, 1, 0, \dots, 0)$ :  $\beta_{k-1} = 1$  then //  $k$ -Saddle
16:  return saddle( $k$ )
17: end if
18: return degenerated( $\beta_i$ )
```

(c) Main construction

Algorithm V.2 $\text{smale}(f)$: geometric gradient construction

```

1: define weight( $K, (\tau^{p-1}, \sigma^p)$ )  $\leftarrow$  // Geometrical function
   
$$\begin{cases} \infty & \text{if } \sigma^p.\text{val} = \infty \\ \infty & \text{if } \tau^{p-1}.\text{val} = \infty \\ f(\sigma^p) - f(\tau^{p-1}) & \text{otherwise} \end{cases}$$

2: greedy(weight) // Construction of the discrete gradient of  $f$ 
```

We will now describe the central part of the construction of geometric discrete Morse complexes. Since the greedy construction of section III.2 *Greedy construction* can generate *any* discrete gradient, the only part to define is the `cancellation_weight` function (algorithm V.2). We tried many of them, but the best results were obtained by the simple difference of height between the barycentres of the two cells to be cancelled. This weight also allowed the results of section V.3 *Properties and proof of the algorithm* to be proved, in particular the automatic detection of critical cells by the greedy construction in the regular cases. However, for special cases, the critical cells can be detected using `critcreate` (algorithm V.3). In that case, the selected critical cells are respected by the algorithm, but it can also generate more critical cells, that can eventually be cancelled using function `cancel` (algorithm V.4) described in the next section.

Algorithm V.3 $\text{critcreate}(f)$: select critical cells

```

1: for  $\tau \in K^0$  do                                     // For each cell of  $K$ 
2:    $\text{tag} \leftarrow \text{crittrack}(\tau, f)$                  // Check if  $\tau$  is critical
3:   if  $\text{tag} = \text{minimum}$  then                             // Minimum
4:      $\tau.\text{val} \leftarrow \infty$                          // Mark as critical
5:   end if
6:
7:   if  $\text{tag} = \text{maximum}$  then                             // Maximum
8:      $\text{max} \leftarrow 0$                                    // Inits the search
9:     for  $\sigma^1 \in \tau.\text{star}$  do                       // Get the maximal incident edge
10:      if  $\text{max} < f(\sigma^1)$  then                       // Greatest edge
11:         $\sigma = \sigma^1$  ;  $\text{max} = f(\sigma^1)$            // New maximum
12:      end if
13:    end for
14:    for  $i \in \{2, \dots, n\}$  do                         // Get the maximal incident cell
15:       $\text{max} \leftarrow 0$                                    // Inits the search
16:      for  $\sigma^i \in \sigma.\text{star}$  do                   // Get the maximal cell incident to  $\sigma$ 
17:        if  $\text{max} < f(\sigma^i)$  then                     // Greatest cell
18:           $\sigma = \sigma^i$  ;  $\text{max} = f(\sigma^i)$          // New maximum
19:        end if
20:      end for
21:    end for
22:     $\sigma.\text{val} \leftarrow \infty$                          // Mark as critical
23:  end if
24:
25:  if  $\text{tag} = \text{saddle}(k)$ ,  $k \neq 1$  then                 //  $k$ -Saddle
26:     $\text{min} \leftarrow \infty$                                // Inits the search
27:    for  $\sigma^1 \in \tau.\text{star}$  do                       // Get the minimal incident edge
28:      if  $\text{min} > f(\sigma^1)$  then                       // Lowest edge
29:         $\sigma = \sigma^1$  ;  $\text{min} = f(\sigma^1)$            // New minimum
30:      end if
31:    end for
32:    for  $i \in \{2, \dots, k\}$  do                         // Get the minimal incident cell
33:       $\text{min} \leftarrow \infty$                                // Inits the search
34:      for  $\sigma^i \in \sigma.\text{star}$  do                   // Get the minimal cell incident to  $\sigma$ 
35:        if  $\text{min} > f(\sigma^i)$  then                     // Lowest cell
36:           $\sigma = \sigma^i$  ;  $\text{min} = f(\sigma^i)$          // New minimum
37:        end if
38:      end for
39:    end for
40:     $\sigma.\text{val} \leftarrow \infty$                          // Mark as critical
41:  end if
42: end for
43: (...)

```

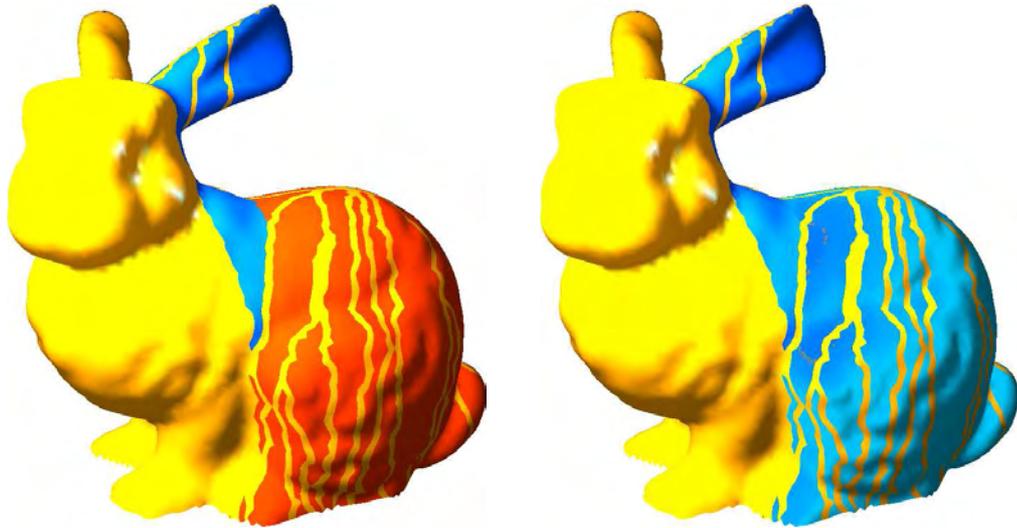
(d) Cancellation optimization

Figure V.7: Cancellation corresponds to region merging on the Morse–Smale decomposition.

The algorithm V.2 of the previous section can actually generate some more critical cells than expected, either because of noise or local irregularities of the complex, or because of topological singularities that were not detected. In the first case, it is often possible to correct this behavior by cancelling pairs of critical cells (figure V.7). This is efficiently performed on the Morse complex directly, using the boundary operator construction of section IV.1(b) *Boundary operator on the Morse complex*. A cancellation can also be performed on the original complex with algorithm V.4, which also checks if the cancellation is valid. The whole cancellation process simply calls `cancel` on all pairs (τ^{p-1}, σ^p) of critical cells that were not detected by algorithm V.3, ordered by `cancellation_weight`. The Morse–Smale complex can be further simplified in the same way, considering all pairs of critical cells, leading to a hierarchical representation of the Morse–Smale decomposition.

(e) Basins identifications

Once a discrete gradient has been defined, one can read the stable and unstable basins directly on our data structure (section III.2(a) *Data structure and basic algorithms* and [Lewiner, 2002]). This completes the construction of our Morse–Smale decomposition. Function `basin` (algorithm V.5) is almost identical to the boundary calculus of algorithm IV.1 (section IV.1(b) *Boundary operator on the Morse complex*), but working on $\mathcal{L}_{p(p+1)}$ instead of $\mathcal{L}_{p(p-1)}$. Observe that a cell can belong to several basins at the same time, as described in section III.3(b) *Discrete stable and unstable basins*.

Algorithm V.4 $\text{cancel}(\tau^{p-1}, \sigma^p)$: cancel a pair of critical cells

```

1:  $path \leftarrow \emptyset$ 
2:  $stack \leftarrow \{Dual_p \cup Crit_p \times (Dual_p \cup Crit_p \times Prim_{p-1})^{\mathbb{N}}\} \leftarrow \{\sigma^p, \emptyset\}$  //
   Gradient path storage
3: while  $stack \neq \emptyset$  do // Hyperpath traversals
4:    $(\sigma', path') \leftarrow stack.top$  // Get the first cell of the path
5:   for  $\tau' \in \sigma'.bdry \cap (Prim_{p-1} \cup Crit_{p-1})$  do // For each critical or primal
   cell bounding  $\sigma$ 
6:      $path' \leftarrow path' \cup (\sigma', \tau')$  // Extends the path
7:     if  $\tau' = \tau^{p-1}$  then // Reached the critical pair
8:       if  $path \neq \emptyset$  then // Already found a path
9:         error // Not a valid cancellation
10:      end if
11:       $path \leftarrow path'$  // Stores the path
12:    end if
13:     $\rho \leftarrow \tau'.match$  // Next gradient step
14:    if  $\rho \neq \sigma'$  then // Valid path extension
15:       $stack.push((\rho, path'))$  // Stores for the next traversal
16:    end if
17:  end for
18: end while
19:
20: for  $(\sigma', \tau') \in path$  do // Traverses the path
21:    $\sigma'.match \leftarrow \tau'$  // Eventually cancels  $\sigma^p$ 
22:    $\tau'.match \leftarrow \sigma'$  // Eventually cancels  $\tau^{p-1}$ 
23: end for

```

Algorithm V.5 $\text{basin}(\sigma^p)$: basin of σ

```

1: if  $\sigma^p \in Prim_p$  then // Primal cell
2:   return  $basin(\sigma^p.match)$  // Basin of the dual cell associated
3: end if
4:  $\mathbf{b} \leftarrow \emptyset$  // Result
5:  $stack \leftarrow \{Dual_p \cup Crit_p\} \leftarrow \{\sigma^p\}$  // Hyperpath storage
6: while  $stack \neq \emptyset$  do // Hyperpath traversals
7:    $\sigma \leftarrow stack.top$  // Get the first cell of the path
8:   for  $\tau \in \sigma.bdry \cap Prim_{p-1}$  do // For each link of  $\mathcal{L}_{p(p-1)}$ 
9:      $\rho \leftarrow (\tau.match).basin$  // Accesses the root
10:    if  $\rho \in Crit_p$  then // Critical root
11:       $\mathbf{b} \leftarrow \mathbf{b} \cup \rho$  // Adds the root
12:    continue // End of hyperpath
13:    end if
14:    if  $\rho \neq \sigma$  then // Valid path extension
15:       $stack.push(\rho)$  // Stores the path extension before next  $\sigma$ 
16:    end if
17:  end for
18: end while

```

V.3 Properties and proof of the algorithm

The construction above is composed of a core, the greedy construction of section III.2(b) *Greedy algorithm* with a specific cancellation weight, a pre-processing that selects the critical cells and a post-processing that cancels some of the critical cells. The complete algorithm always build a valid discrete gradient, since it relies on the acyclicity test (algorithm III.6) and since the cancellation of algorithm V.4 checks the validity of its operation before performing it. Moreover, the critical cells of the resulting gradient always contain the cells detected during the pre-processing. We will first prove that this gradient is aligned with the increasing directions of f in the regular parts, which will prove the validity of the whole algorithm. We then study the properties of the core, without the pre-processing. In particular, we will prove that for subdivided surfaces, the core alone constructs a discrete gradient with the right critical cells.

Before our work on optimal Morse functions, [Babson & Hersh, 2005] proposed a first scheme to build discrete Morse function that relies on a previous ordering of the cells. This construction roughly corresponds to considering the cancellation weight of a pair as the lowest image by f of the vertices of the pair. Although the objective differs from generating a geometrical decomposition, the proofs provided in this work are more general than our proofs, and uses advanced combinatorial tools to conclude elegantly their works. We hope to complete the following proofs in a more elegant way, possibly using [Babson & Hersh, 2005] as a model.

(a) Regularity of the gradient

Considering that K is simplicial, we will say that a gradient step $\blacktriangleleft \tau_0 \sigma_0 \tau_1 \blacktriangleright$ is *decreasing* if the value of f on the first vertex (i.e. the vertex of $\sigma_0 \setminus \tau_1$) is greater than the value of f on the last vertex of the gradient step (i.e. the vertex of $\sigma_0 \setminus \tau_0$). We would like first to prove that the geometrical function f is globally decreasing on each gradient path. We were able to prove it for *regular* \mathcal{V} -path, i.e. when the gradient path $\blacktriangleleft \tau_0^{p\pm 1} \sigma_0^p \tau_1^{p\pm 1} \sigma_1^p \dots \sigma_{r-1}^p \tau_r^{p\pm 1} \blacktriangleright$ is made of only regular links in layer $\mathcal{L}_{(p\pm 1)p}$ or in layer $\mathcal{L}_{p(p\pm 1)}$. This condition is necessary, since outside the regular parts can exists locks (figure V.9). It is sufficient for this gradient path to be regular on only one of these layers.

Theorem 7 *Let \mathcal{V} be the discrete gradient on simplicial complex K , constructed only by algorithm V.2 with geometric function f . Then the longest increasing regular gradient \mathcal{V} -path is of length 1.*

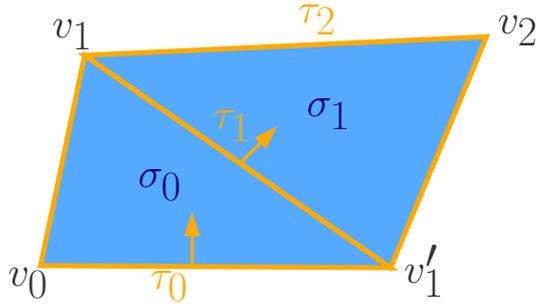


Figure V.8: Notation for the proof: in that case $\rho = \emptyset$.

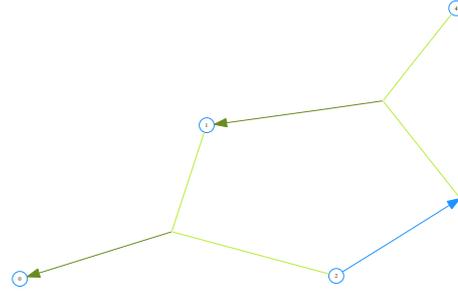


Figure V.9: Lock: hyperlink 012 cannot be reversed.

Proof: The proof of the theorem works equal with gradient path with p and $(p+1)$ -cells or p and $(p-1)$ -cells, and we will write the proof for p and $(p-1)$ -cells. Consider a gradient \mathcal{V} -path $\blacktriangleleft \tau_0 \sigma_0 \tau_1 \sigma_1 \tau_2 \blacktriangleright$, τ_0, τ_1, τ_2 being $(p-1)$ -cells of K and σ_0, σ_1 being regular links of $\mathcal{L}_{(p-1)p}$. Denote by v_0 and v_1 the first and last vertices of $\blacktriangleleft \tau_0 \sigma_0 \tau_1 \blacktriangleright$, v_1' and v_2 the first and last vertices of $\blacktriangleleft \tau_1 \sigma_1 \tau_2 \blacktriangleright$, and by ρ the $(p-3)$ -simplex common to τ_0, τ_1 and τ_2 (figure V.8). Observe that ρ can be empty. We will prove that either $f(v_0) > f(v_1)$ or $f(v_0) > f(v_2)$ and $f(v_1) > f(v_2)$. Defining f on each simplex of K as the mean of f on each vertex: $f(\sigma) = \text{mean}(|\partial\sigma| \cap K^0)$, as in algorithm V.2, we have:

$$\begin{aligned} f(\tau_0) &= \frac{p-3}{p-1}f(\rho) + \frac{1}{p-1}(f(v_0) + f(v_1')) \\ f(\tau_1) &= \frac{p-3}{p-1}f(\rho) + \frac{1}{p-1}(f(v_1) + f(v_1')) \\ f(\tau_2) &= \frac{p-3}{p-1}f(\rho) + \frac{1}{p-1}(f(v_1) + f(v_2)) \\ f(\sigma_0) &= \frac{p-3}{p}f(\rho) + \frac{1}{p}(f(v_0) + f(v_1) + f(v_1')) \\ f(\sigma_1) &= \frac{p-3}{p}f(\rho) + \frac{1}{p}(f(v_2) + f(v_1) + f(v_1')) \end{aligned}$$

The proof can then be simplified by distinguishing 2 cases:

case 1: (τ_0, σ_0) occurred before (τ_1, σ_1) . We know from proposition 1 that the regular components of a hyperforest are tree, even without orientation. Since (τ_0, σ_0) does not create cycle, as it has been cancelled, (τ_1, σ_0) does not create cycle either. Then, as cancellation (τ_1, σ_1) occurred after cancellation (τ_0, σ_0) , the cell τ_1 was not matched when cancellation (τ_0, σ_0) has been considered. This implies that cancellation (τ_1, σ_0) has been considered after cancellation (τ_0, σ_0) : $f(\sigma_0) - f(\tau_1) > f(\sigma_0) - f(\tau_0)$, which gives $f(\tau_0) > f(\tau_1)$ and then $f(v_0) > f(v_1)$.

case 2: (τ_1, σ_1) occurred before (τ_0, σ_0) . With same deductions, (τ_1, σ_0) is a valid cancellation that was considered after (τ_1, σ_1) , and (τ_2, σ_1) is a valid cancellation that was considered after (τ_1, σ_1) . Then $f(\sigma_0) - f(\tau_1) >$

$f(\sigma_1) - f(\tau_1)$ and $f(\sigma_1) - f(\tau_2) > f(\sigma_1) - f(\tau_1)$. The first inequality gives $f(\sigma_0) > f(\sigma_1)$, and then $f(v_0) > f(v_2)$. The second inequality gives $f(\tau_1) > f(\tau_2)$, and then $f(v_1) > f(v_2)$.

Therefore the gradient path $\blacktriangleleft \tau_0 \sigma_0 \tau_1 \sigma_1 \tau_2 \blacktriangleright$ cannot be increasing. \blacksquare

This theorem asserts the fact that the gradient is globally decreasing with f , even if one specific pair can be increasing.

(b) Minima positioning

The above theorem actually proves directly that the minima resulting of algorithm V.2 always correspond to the minima in Banchoff's definition. The converse is true if two minima are sufficiently far one from the other, which can be achieved by one barycentric subdivision.

Theorem 8 *Let \mathcal{V} be the discrete gradient on a simplicial complex K , constructed by the only algorithm V.2 with geometric function f . Then the critical vertices of \mathcal{V} are critical minima for Banchoff's definition. Conversely, if \mathcal{V} is the discrete gradient build by algorithm V.2 on the first barycentric subdivision K' of K , with geometric function f' corresponding to the subdivision of f , then the minima of \mathcal{V} are exactly the minima as defined by Banchoff.*

Proof: First note that the barycentric subdivision does not affect Banchoff's critical points, nor does it create any. In particular Banchoff's critical points always belong to K , and not to $K' \setminus K$. First, the edges around a critical vertex v_2 cannot be all matched with faces, otherwise it would create a cycle. Since the layer \mathcal{L}_{01} is a regular graph, if vertex v_2 is critical, any increasing gradient step $\blacktriangleleft v_1 e v_2 \blacktriangleright$, with $e = (v_2 v_1)$, would have been reversed. Therefore there are only decreasing gradient steps around v_2 , i.e. v_2 is a local minimum of \mathcal{L}_{01} . Now we have to prove that v_2 is a minimum over all its link, including the part on \mathcal{L}_{12} . If there is no edge incident to v_2 matched with a face, then this part of the proof is concluded. If edge $e = (v_2 v_1)$ is matched with triangle $t = (v_2 v_1 v_3)$, then since cancellation (e, t) has been considered before (v_2, e) , $2f(v_1) > f(v_2) + f(v_3)$. If edge $e' = (v_2 v_3)$ is matched with a triangle $t' = (v_2 v_3 v_4)$, then we are in a similar case to theorem 7, with $v'_1 = v_1$. This implies $v_3 > v_1$, and then $v_2 < v_1$. If edge e' is matched with v_3 , then v_3 belongs to \mathcal{L}_{01} and then $v_3 > v_2$, which implies $v_2 < v_1$. Therefore v_2 is a local minimum of K .

Now if v_2 is a local minimum, it must be critical for \mathcal{V} on K' . If not, there would be an edge $e = (v_2 v_1)$ matched with v_2 , with $f(v_2) < f(v_1)$. The vertex v_1 cannot be critical, since the collapse $e' = (v_2, e)$ is considered after the collapse (v_1, e) . Then there is another edge $(v_1 v_0)$ matched with v_1 . Since f' is the barycentric subdivision of f , we have $f(v_1) < f(v_0)$. Then,

$\langle v_2 e v_1 e' v_0 \rangle$ would be an increasing gradient path of length greater than 1, leading to a contradiction. ■

This property is valid for the minimum of any simplicial cell complex. However, the subdivision or equivalently a certain distance between minima is necessary. The other proofs are restricted to the case of surfaces, where all the layers or their dual are regular.

(c) Surfaces' maxima positioning

In order to position correctly the minima, we needed to be sure that two minima are sufficiently far one from the other in the mesh. The case of maxima is similar, but we need the same spacing between critical points as the handlebody decomposition [Rourke & Sanderson, 1972], i.e. two barycentric subdivisions.

Proposition 9 *Let \mathcal{V} be the discrete gradient on a cell complex K , the second barycentric subdivision of K , constructed by the only algorithm V.2 with geometric function subdivided from f . Then for each of Banchoff's maxima v , there is a critical triangle σ in the star of v . Moreover, σ is the triangle subdivided from vwx , where w is the vertex of $\text{st}(v)$ with the highest value, x and y are the vertices adjacent to v and w , and $f(x) > f(y)$.*

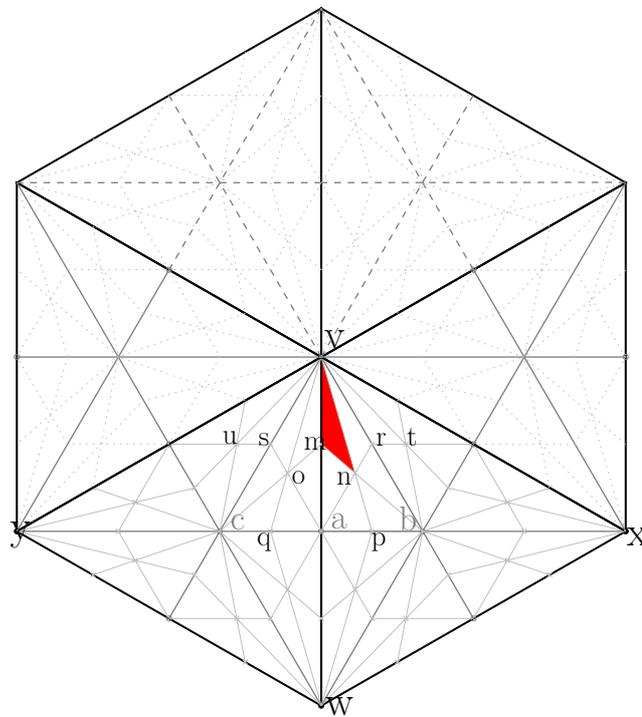


Figure V.10: Second barycentric subdivision around a maximum: v, w, x and y are original vertices, a, b and c are generated by the first subdivision and the other marked ones by the second one.

Proof: Since we consider barycentric subdivisions of K , the proof is equivalent for convex or simplicial cell complexes. The ordering $f(v) > f(w) > f(x) > f(y)$ generates a partial ordering on the cells of the second barycentric subdivision, and then the partial ordering P on the candidate cancellations considered by algorithm V.2. The goal now is to prove that this partial ordering guarantees that each cancellation involving σ is invalidated before it is considered. We will use the notation of figure V.10 for the vertices, and denote each cell by the vertices it spans.

Poset P partially represented on figure V.11 actually proves the theorem. The triangle mnv must not be matched to be critical. It could be matched with any of its three boundary edges: mv , mn or nv . First, we will ensure that v is not matched with any problematic edge. Let d be the vertex of minimum value in the star of v , in particular d is either not marked on figure V.10 or s , t , u or o . Then, $f((v < dv)) = f(dv) - f(v) \leq f(\xi v) - f(v) = f((v < \xi v))$, for ξ different from s , t , u or o .

$(mv < mnv)$: The cancellation of mv with mnv is considered after the cancellation of mv with mov . This cancellation is considered before $(m < mv)$, $(ov < mov)$ and $(mo < mov)$ (figure V.11). The only problem would be $(v < mv)$, but since m differs from s , t and o , this cancellation will not occur. And even if it occurred, it would also prevent the cancellation of mv with mnv .

$(mn < mnv)$: The cancellation of mn with mnv is considered after the cancellation of mn with amn . If $(m < mn)$ would be valid, it would also serve, but it is a priori invalidated by $(m < mo)$. Then cancellation $(mn < amn)$ is considered before $(n < mn)$ and $(an < amn)$ (figure V.11). The only problems would be $(m < mn)$ and $(am < amn)$. Since the first one either solves or is invalidated, it is sufficient to invalidate $(am < amn)$ by $(am < amo)$. This will occur since it appears first, and $(am < amo)$ is considered before $(ao < amo)$ and $(mo < amo)$ is invalidated by $(m < mo)$.

$(nv < mnv)$: Last, the cancellation of nv with mnv is considered after the cancellation of nv with nrv . This cancellation is considered before $(nr < nrv)$, $(n < nv)$. Then, the cancellation $(v < nv)$ will not occur, and if it would, this would prohibit the cancellation of nv with mnv . The only problem would be cancellation $(rv < nrv)$, but it is invalidated by $(rv < rtv)$, since the latter occurs before $(r < rv)$, $(rt < rtv)$ and $(tv < rtv)$ and we know that v will not match with rv .

■

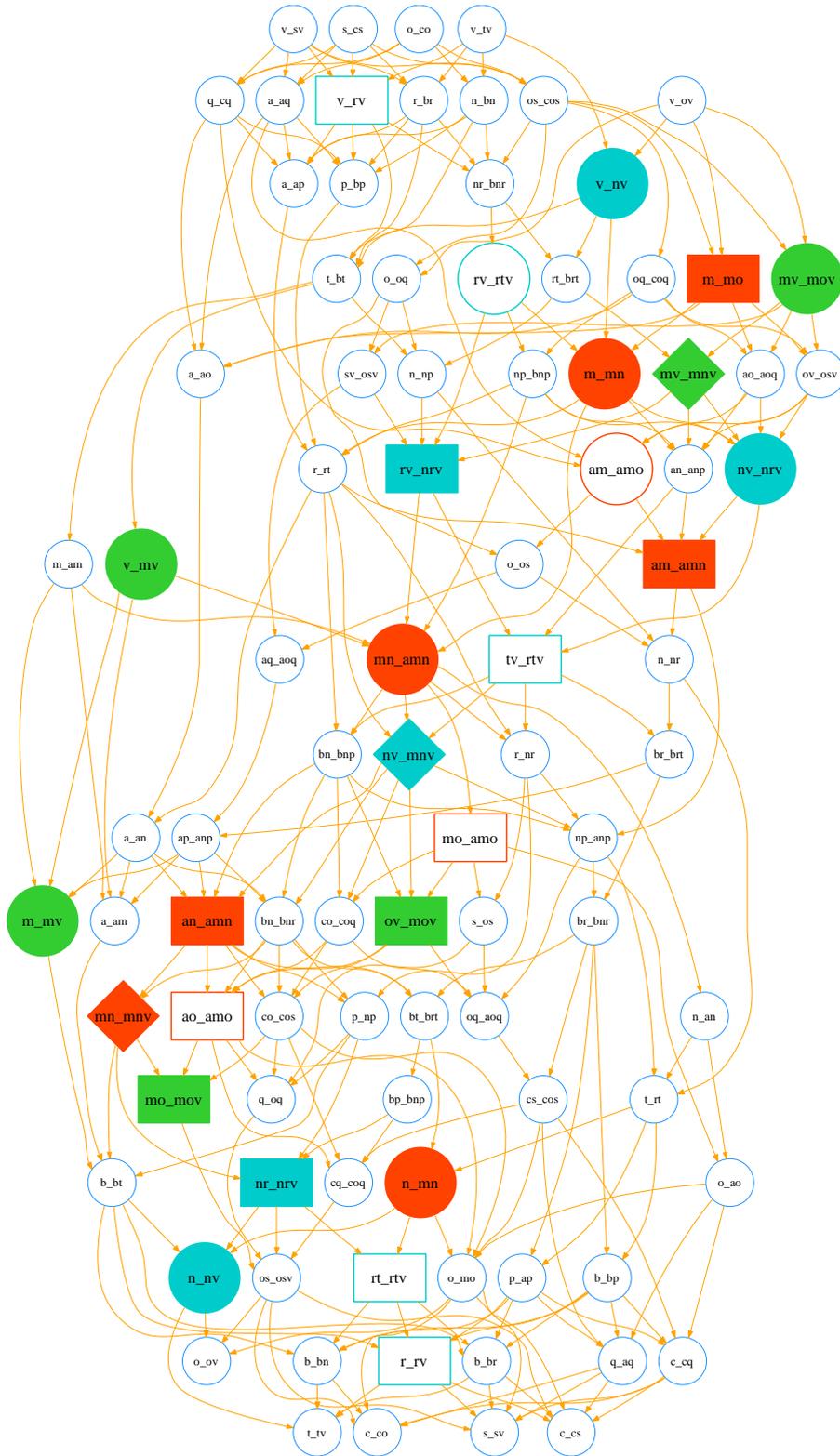


Figure V.11: Poset generated by $f(v) > f(w) > f(x) > f(y)$ onto the pairs of incident cells: the matchings that would forbid mnv to be critical are marked by coloured diamonds, the one that would prevent these diamonds are marked as circles of the same colour, and the ones that would prevent the circles are marked with rectangles.

Observe that there can be other critical triangles than just in the star of a Banchoff's maximum. However, in that case there is also a critical edge in the star of the greatest point of these triangles. Moreover, this case occurs usually where the triangulation is not well adapted to the geometrical function (figure V.2).

(d) Surfaces' saddles positioning

The strategy for positioning saddles is mainly the same one, although the proof is a little more difficult.

Proposition 10 *Let \mathcal{V} be the discrete gradient on a cell complex K ", the second barycentric subdivision of K , constructed by the only algorithm V.2 with geometric function subdivided from f . Then for each of Banchoff's saddle v , there is a critical edge τ in the star of v .*

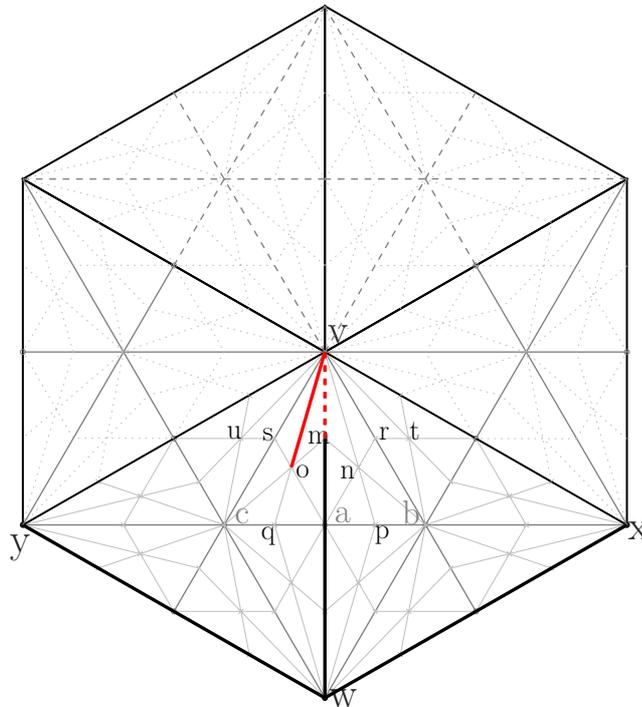


Figure V.12: Second barycentric subdivision around a saddle: v , w , x and y are original vertices, a , b and c are generated by the first subdivision and the other marked ones by the second one.

Proof: The proof of this proposition is mainly the same one as for proposition 9, with the notation of figure V.12. The ordering here is different, since w is a local minimum. In order to achieve a sufficiently dense partial ordering, we will consider 3 cases, depending if $f(v) > f(x) > f(y) > f(w)$, $f(x) > f(v) > f(y) > f(w)$ or $f(x) > f(y) > f(v) > f(w)$. The last case must be split in two sub-cases in order to identify exactly the critical edge, either ov or mv , depending on which of o and m is lower.

Since the proof is almost the same, we will go faster in the descriptions of each case. First of all, the vertex v will be matched in the other component of the lower link, and we can therefore consider any cancellation involving v as invalid. In the 3 first cases, we would like to prove that ov will remain critical, and mv in the last case.

$$f(v) > f(x) > f(y) > f(w)$$

$(o < ov)$: this cancellation is invalidated either by $(o < ao)$, $(o < co)$ or $(o < oq)$, which are guaranteed since they are considered before, respectively, $(c < co)$, $(co < cos)$ and $(co < coq)$; $(q < oq)$, $(oq < coq)$ and $(oq < aoq)$; $(a < ao)$, $(ao < aoq)$ and $(ao < aom)$ (figure V.13).

$(ov < osv)$: this cancellation is invalidated by $(sv < osv)$, which is guaranteed since $(v < sv)$ is invalidated and since it is considered before $(s < sv)$ and $(sv < svw)$ (figure V.13).

$(ov < mov)$: this cancellation is invalidated by $(mv < mov)$, which is guaranteed since $(v < mv)$ is invalidated and since it is considered before $(m < mv)$ and $(mv < mnv)$ (figure V.13).

$f(x) > f(v) > f(y) > f(w)$ The proof is exactly the same as before, although the poset is different. Observe that $(mv < mov)$ appears before $(ov < osv)$ because there is $(uv < suv)$ between them, even if it is difficult to read it on figure V.14.

$f(x) > f(y) > f(v) > f(w)$ **and** $f(m) > f(o)$ The proof is essentially the same again, with the poset of figure V.15. The only difficulty is to see that $f(osv) - f(os) < f(sv) - f(s)$. This can be proved as follows. $f(sv) - f(s) - (f(osv) + f(os)) = \frac{1}{2}f(s+v) - s - \frac{1}{3}f(o+s+v) + \frac{1}{2}f(o+s) = \frac{1}{6}f(o+v-2s)$. Decomposing onto x, y, v and w , we get $216f(o) = 132f(v) + 60f(w) + 24f(y)$, $216f(s) = 144f(v) + 36f(w) + 36f(y)$ and $216f(m) = 162f(v) + 54f(w)$. Therefore, $\frac{216}{6}f(o+v-2s) = \frac{216}{6}f(60v-12w-48y) = \frac{1}{3}f(m-o) > 0$.

$$f(x) > f(y) > f(v) > f(w) \text{ **and** } f(o) > f(m)$$

$(m < mv)$: this cancellation is invalidated by $(m < am)$, which is guaranteed since it is considered before $(a < am)$, $(am < amo)$ and $(am < amn)$ (figure V.16).

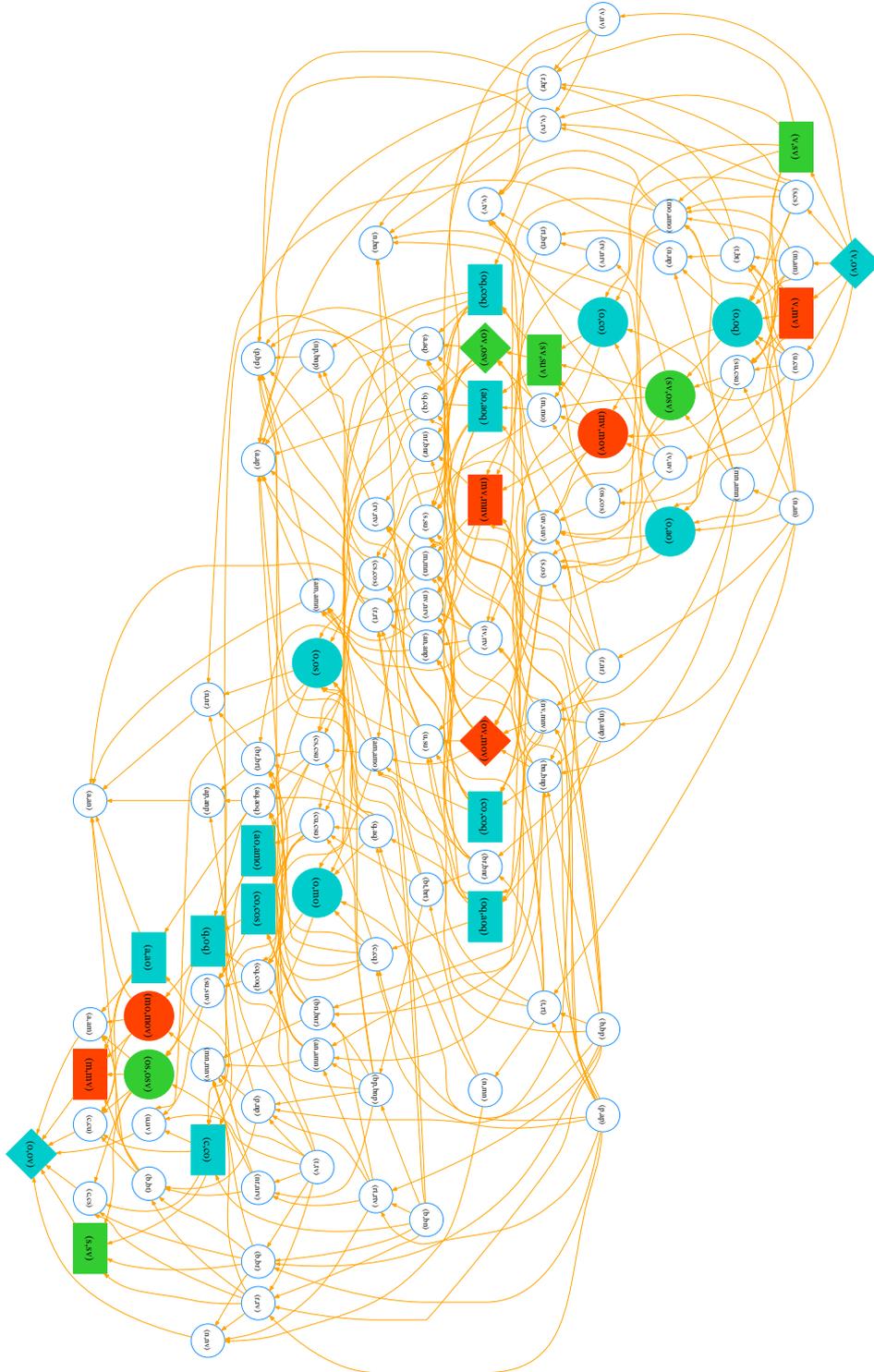


Figure V.14: Poset generated by $f(x) > f(v) > f(y) > f(w)$ onto the pairs of incident cells, marking the relations to ov .

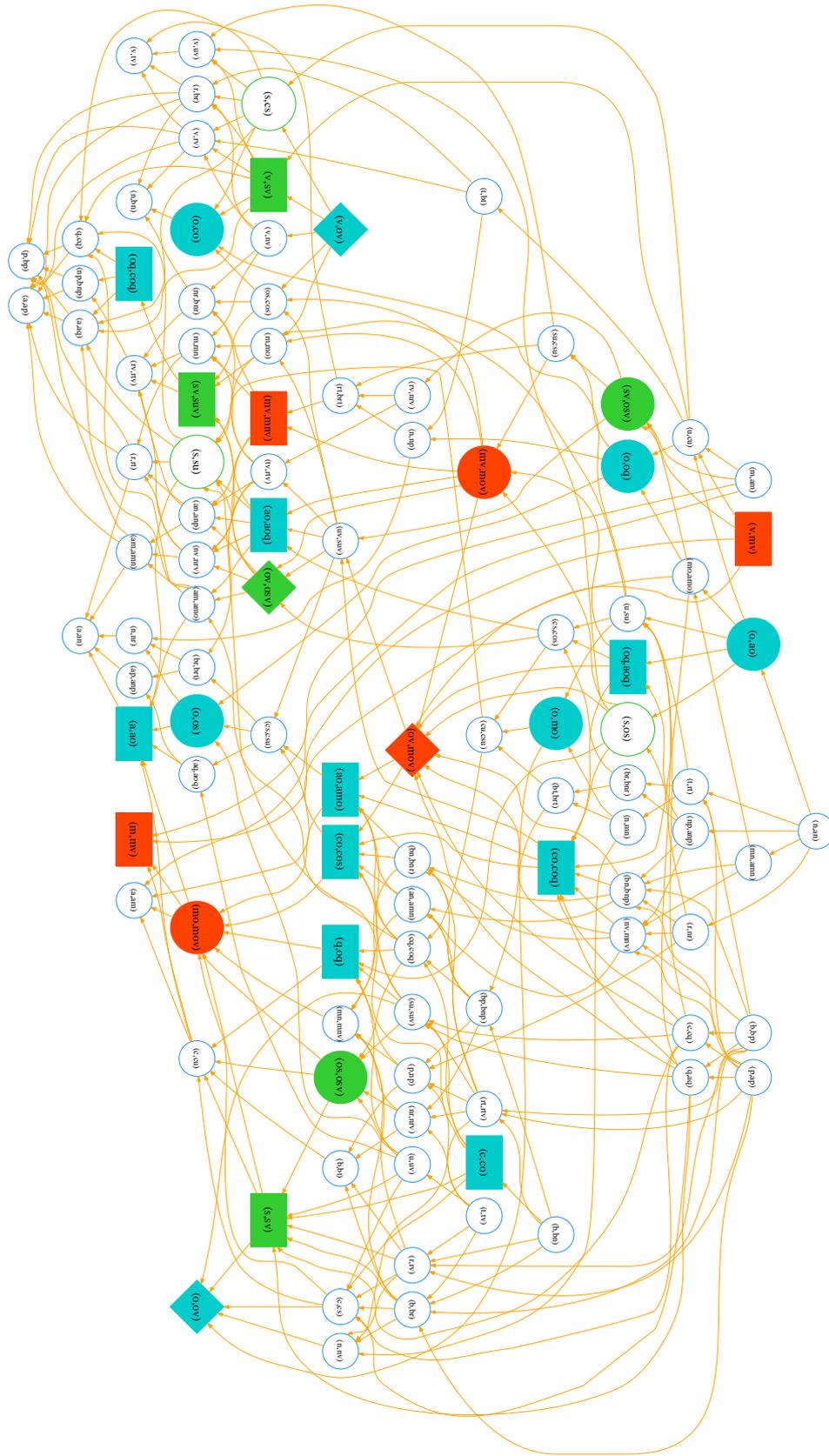


Figure V.15: Poset generated by $f(x) > f(y) > f(v) > f(w)$ and $f(m) > f(o)$ onto the pairs of incident cells, marking the relations to ov .

$(mv < mnv)$: this cancellation is invalidated by $(nv < mnv)$, which is guaranteed since it is considered before $(nv < nrv)$, and $(n < nv)$ is invalidated by $(n < an)$, which appears before $(a < an)$, $(an < amn)$ and $(an < anp)$ (figure V.16).

$(mv < mov)$: this cancellation is invalidated by $(ov < mov)$, which appears before since $f(m) > f(o)$. Cancellation $(ov < mov)$ is guaranteed since it appears before $(ov < osv)$, and that $(o < ov)$ is invalidated by $(o < ao)$, that appears before $(a < ao)$, $(ao < amo)$ and $(ao < aoq)$.

■

V.4 Results and applications

The construction of algorithm V.2 is efficient since it does not require any critical point detection and usually provides a nice result (figure V.17). For surfaces, it is possible to guarantee that all of Banchoff's critical points generate a critical cell. It is also possible to detect only the maxima and the minima, which are the easiest to detect. The Morse inequalities then guarantee the number of saddles. Moreover, it is possible to accelerate algorithm III.5 by slicing the priority queue. In particular, considering cancellations with only negative weights ensures that all critical cells are preserved. The search for critical cells can then be narrowed to non-matched cells.

In most of the practical cases, the core algorithm V.2 generates the required decomposition (figure V.19). In the presence of noise, Banchoff's definition of critical points can even generate too many critical points (figure V.18), which can be resolved by non-local cancellations (algorithm V.4).

(a) Reeb graph

The *Reeb graph* (also called *contour graph*) of a geometrical object \mathcal{M} with a Morse function f defined on it is a subset of \mathcal{M} where two points \mathbf{x} and \mathbf{y} are identified if they have the same value through f : $f(\mathbf{x}) = f(\mathbf{y})$, and if they lie in the same connected component of $f^{-1}(f(\mathbf{x}))$. Georges Reeb first used those diagrams to prove that if a manifold admits a Morse function with only two critical points, then it is a sphere. This theorem is valid in the discrete setting [Forman, 1995].

Since the connections of the Reeb graph correspond to the critical points of f , we can compute the Reeb graph directly from our Morse–Smale decomposition. This can be performed efficiently through one simple graph operation (figure V.20). This operation removes from the Morse–Smale

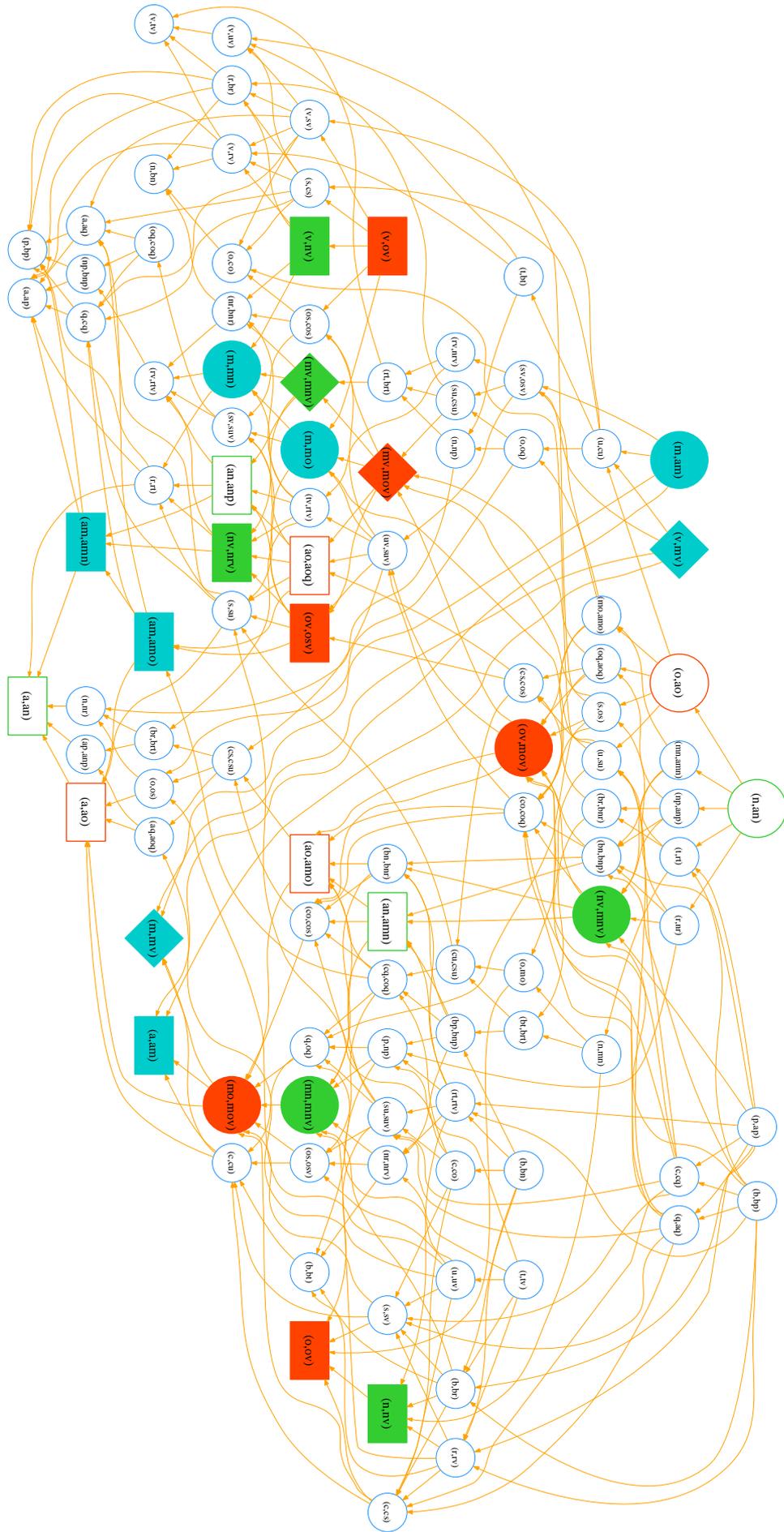


Figure V.16: Poset generated by $f(x) > f(y) > f(v) > f(w)$ and $f(o) > f(m)$ onto the pairs of incident cells, marking the relations to mv .

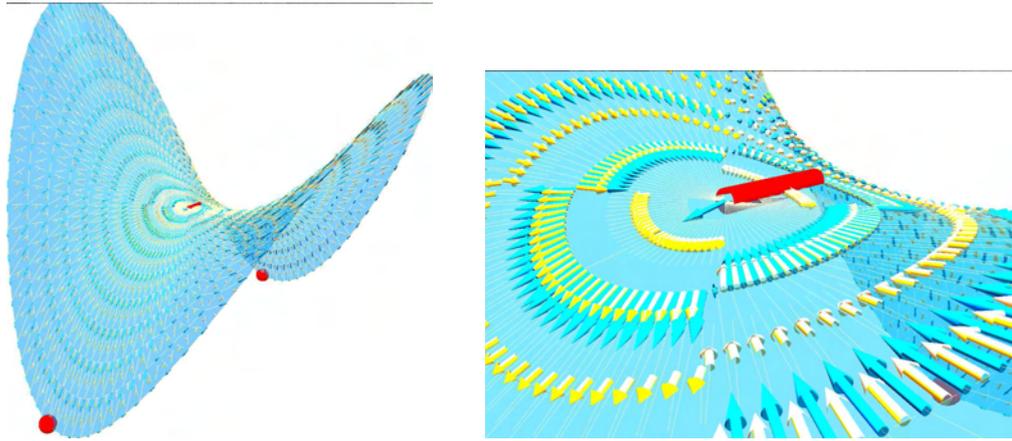


Figure V.17: The result of the only greedy algorithm on a saddle.

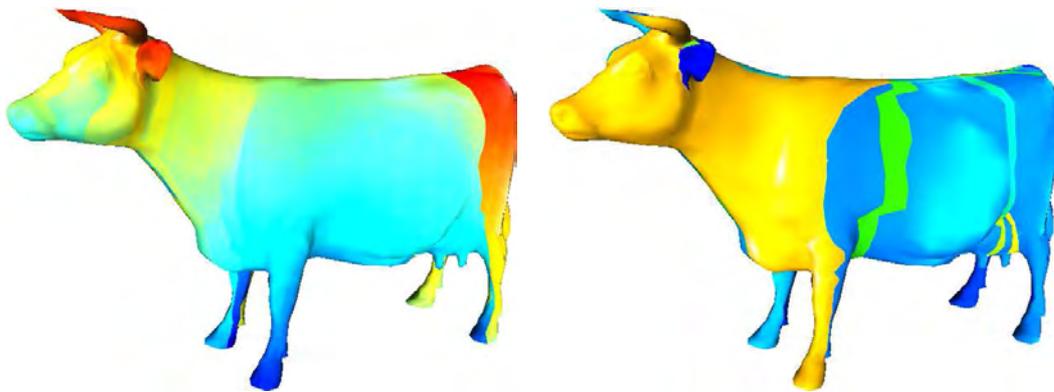
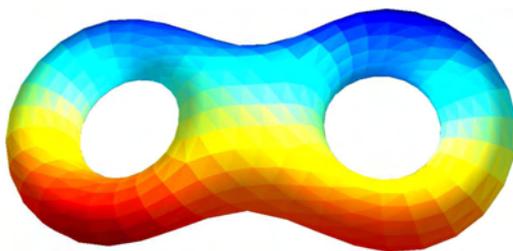
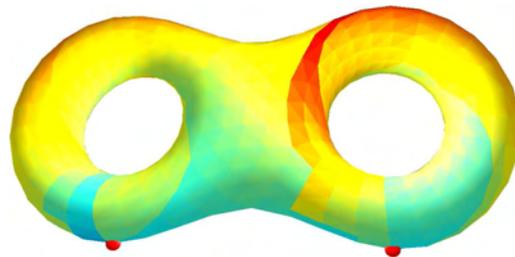


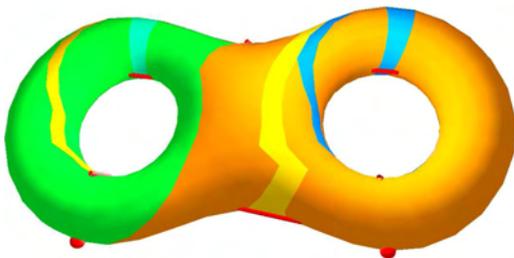
Figure V.18: A Morse–Smale complex generated from the height function on a cow model.



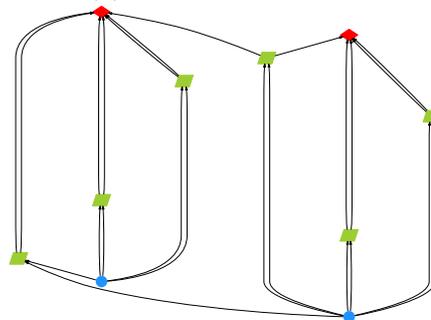
V.19(a): Geometric function.



V.19(b): Discrete Morse function.



V.19(c): Morse–Smale decomposition.



V.19(d): Morse–Smale diagram.

Figure V.19: A Morse–Smale complex generated from the height function $x + \frac{1}{100}y$ on double torus model.

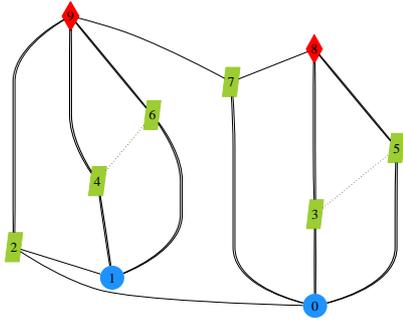
decomposition the information not contained in the Reeb graph. Consider two saddles s_1 and s_2 , $f(s_1) < f(s_2)$, incident in the Morse complex to the minima v_1 and v_2 for s_1 , and v_2 and v_3 for s_2 . The Reeb graph will only note that s_2 is incident to s_1 and v_3 , and does not detail whether s_2 is also incident to v_1 or v_2 . The procedure removes the link s_2v_2 from the Morse–Smale diagram and adds the link s_2s_1 to obtain the Reeb graph, in a technique similar to [Lazarus & Verroust, 1999]. The only delicate points are the loops, as usual, but those are detected at each step of the algorithm. The efficiency of our algorithm makes this result a significant improvement of [Carr *et al.*, 2000] and [Cole–McLaughlin *et al.*, 2003].

The first ideas of this process came from a discussion with Francis Lazarus at Poitiers, on February the 14th, 2003.

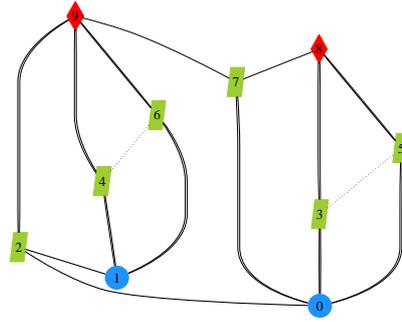
(b) Persistence

Persistence is a concept that came out from Smale’s work [Smale, 1961]. Smale used it as a strategy to cancel critical points in manifolds of high dimension. In the case of homotopical spheres, the cancellation always reached a minimal configuration, proving Poincaré’s conjecture for dimension from 5 and above as a particular case of the *h-cobordism* theorem. More recently, [Edelsbrunner *et al.*, 2000] renamed this technique “persistence”, and deduced it from the incremental computation of Betti numbers on \mathbb{Z}_2 for subtriangulations of \mathbb{S}^3 [Delfinado & Edelsbrunner, 1993]. The definition of persistence of [Edelsbrunner *et al.*, 2000] does not involve Morse theory, although it relies on a given filtration that is directly inspired of Morse concepts. It has recently been defined more rigorously in [Cohen–Steiner *et al.*, 2005], proving a stability with respect to noise.

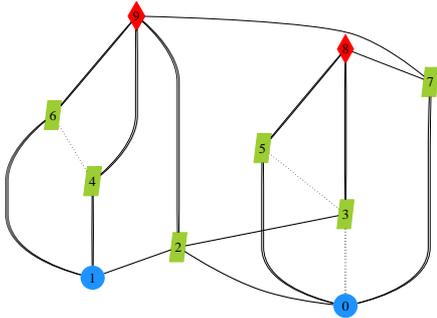
Persistence is essentially the difference of height $f(\sigma) - f(\tau)$ between pairs of critical cells (τ, σ) , when cancelling them from the lowest persistence to the highest one (figure V.21). The persistence can then be read directly on the Morse–Smale decomposition, by performing the cancellations only on the Morse complex. This computation is then more efficient than the original one, since the Morse complex generally has much fewer cells.



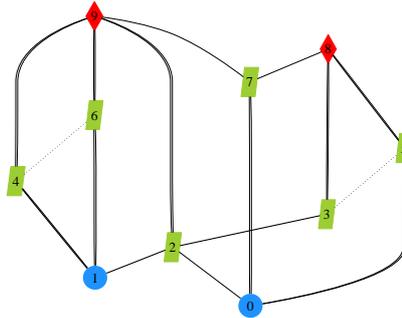
V.20(a): There are two loops made of double edges. Their diagonals are marked.



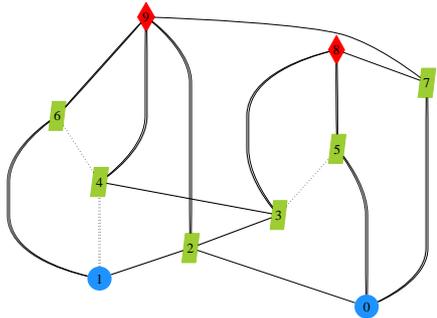
V.20(b): First rotation $0 \rightarrow 2$ with $0 \Rightarrow 3$.



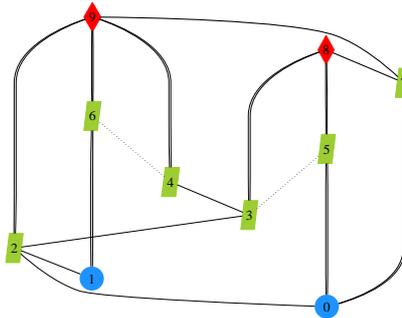
V.20(c): Rotation replaced $0 \Rightarrow 3$ by $2 \rightarrow 3$.



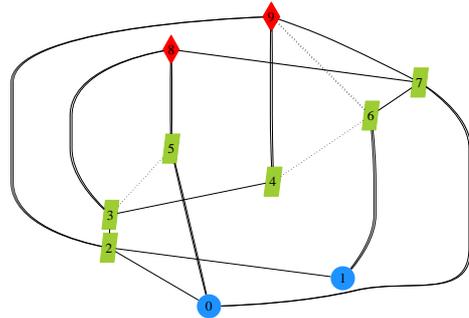
V.20(d): Second rotation $1 \rightarrow 2 \rightarrow 3$ with $1 \Rightarrow 4$.



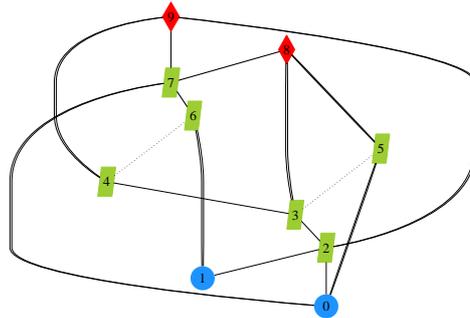
V.20(e): Rotation replaced $1 \Rightarrow 4$ by $3 \rightarrow 4$.



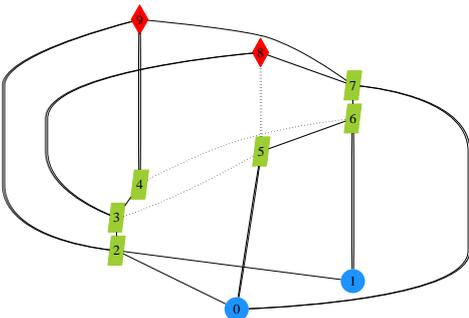
V.20(f): Third rotation $9 \leftarrow 7$ with $9 \Leftarrow 6$.



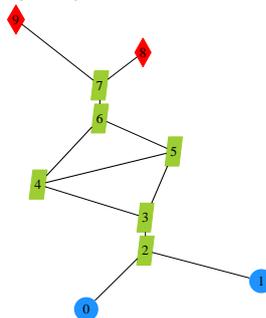
V.20(g): Rotation replaced $9 \Leftarrow 6$ by $7 \leftarrow 6$.



V.20(h): Fourth rotation $8 \leftarrow 7 \leftarrow 6$ with $8 \Leftarrow 5$.

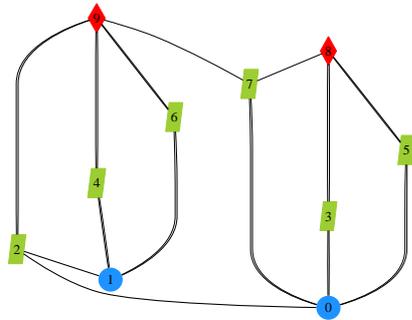


V.20(i): Rotation replaced $8 \Leftarrow 5$ by $6 \leftarrow 5$.

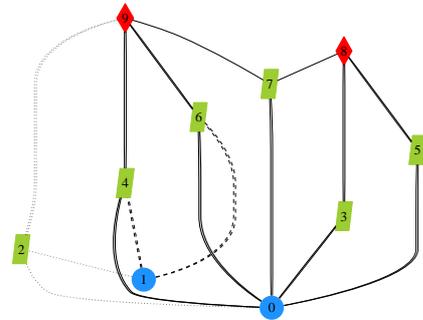


V.20(j): 4 possibilities $0 \Rightarrow 5$, $1 \Rightarrow 6$, $8 \Leftarrow 5$ and $9 \Leftarrow 4$ leading to $4 \rightarrow 5$.

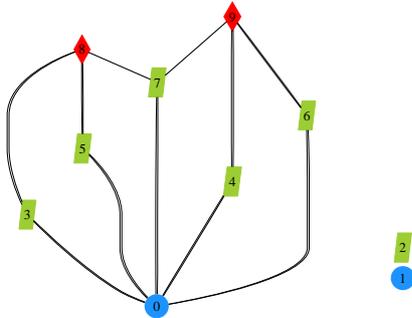
Figure V.20: Transforming a Morse–Smale decomposition in Reeb graph.



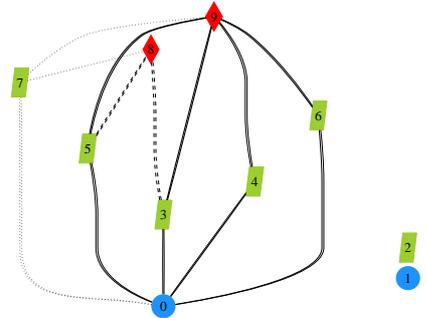
V.21(a): Original complex of figure V.19:



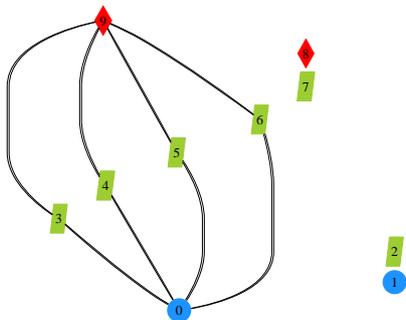
V.21(b): Cancellation shift links from vertex 1 to vertex 2, since it is primal.



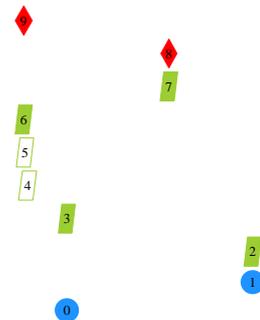
V.21(c): Second cancellation (7, 8).



V.21(d): Cancellation shift links from face 8 to face 9, since it is dual.



V.21(e): There is no more valid cancellation.



V.21(f): The persistence is computed as the valid cancellations $f(2) - f(1)$ and $f(8) - f(7)$ and the smallest pairing $f(3) - f(0)$ and $f(9) - f(6)$.

Figure V.21: Computing the persistence from the Morse–Smale complex.

VI

Conclusions and future works

This thesis introduced some new methods in computational topology and geometric modelling. Starting from the combinatorial structures we developed in [Lewiner, 2002], we formulated the flow calculus in both theoretical and algorithmical directions. The computation of the flow onto the hypergraph layers of a discrete Morse function led to concise formulations and efficient algorithms. In particular, it linked a cell complex with a discrete gradient defined on it to its Morse complex in both directions. On one hand, this correspondence gave a direct calculus of homology, including torsion on arbitrary field, generating cycles and cycle decompositions. The algorithms that perform these calculi are as efficient as and more general than previous methods. On the other hand, this Morse complex restores the relation between geometry and topology in the discrete setting. Starting from the Morse–Smale decomposition, we gave a rigorous constructive definition of a discrete Morse function from the sampling of a smooth Morse function, having the same Smale decomposition. This construction has been proved and applied to practical computations of the persistence and the Reeb graphs.

The format of this work emphasizes the relations between abstract smooth notions and their concrete computation. As usual in discrete mathematics, continuous notions have many properties that are not directly compatible with the discrete setting. More particularly for Morse theory, the topological and geometrical properties involved are difficult to define rigorously in the discrete setting. The work of Forman provides a complete and extensive theory that guarantees the topological properties of Morse theory, which are the essential point. The geometrical properties are not directly needed for the theory, but are necessary for practical applications. In this perspective, this work could illustrate the benefits of strong theories for concrete applications, since the rigor of Forman’s theory achieved more efficient and general applications than the previous poorly mathematical approximations of Morse theory.

In the reverse perspective, this work opened many questions from the theoretical point of view, and most of them have not been solved yet. In particular, the relation of our decomposition and the co-homology is not

yet clear, since Spanier's co-homology does not live in the same space and most of the other co-homologies do not live in the same dimension. The combinatorial proof of our construction of geometrical Morse complex seems to generalize to any dimension and to non-regular cases. However, the proof in itself is laborious and a generalization would require some more tools from combinatorics. I hope that this work will generate more exchange between theory and experiment, and between pure and applied mathematics.

Bibliography

- [Alliez & Desbrun, 2001] P. Alliez and M. Desbrun. **Valence-driven connectivity encoding of 3D meshes**. In *Computer Graphics Forum*, pages 480–489, 2001. III.2(e)
- [Babson & Hersh, 2005] E. Babson and P. Hersh. **Discrete Morse functions from lexicographic orders**. *Transactions of the American Mathematical Society*, 357:509–534, 2005. V.3
- [Banchoff, 1967] T. Banchoff. **Critical points and curvature for embedded polyhedra**. *Journal of Differential Geometry*, 1:257–268, 1967. V, V.1, V.1(a)
- [Berge, 1970] C. Berge. **Graphes et hypergraphes**. Dunod, Paris, 1970. III.1(b), III.1(b), III.1(b), III.2
- [do Carmo, 1976] M. do Carmo. **Differential geometry of curves and surfaces**. Prentice Hall, 1976. II.1, II.2(a)
- [Carr *et al.*, 2000] H. Carr, J. Snoeyink and U. Axen. **Computing contour trees in all dimensions**. *Computational Geometry: Theory and Applications*, 24:75–94, 2000. V.4(a)
- [Cohen, 1973] M. M. Cohen. **A course in simple homotopy theory**. *Graduate text in Mathematics*. Springer, New York, 1973. II.5, II.5(b)
- [Cohen–Steiner *et al.*, 2005] D. Cohen–Steiner, H. Edelsbrunner and J. L. Harer. **Stability of Persistence Diagrams**. In *Symposium on Computational Geometry*, pages 263–271. ACM, 2005. V.4(b)
- [Cole–McLaughlin *et al.*, 2003] K. Cole–McLaughlin, H. Edelsbrunner, J. L. Harer, V. Natarajan and V. Pascucci. **Loops in reeb graphs of 2-manifolds**. In *Symposium on Computational Geometry*, pages 344–350. ACM, 2003. V.4(a)
- [Cooke & Finney, 1967] G. E. Cooke and R. L. Finney. **Homology of cell complexes**. Princeton University Press, 1967. II.1(b), II.1(b), II.5(b)

- [Crowley *et al.*, 2005] K. Crowley, A. Ebin, H. Kahn, P. Reyfman, J. White and M. Xue. **Collapsing a simplex to a noncollapsible subcomplex**. *St. Olaf College and Columbia University*, 2005. II.5(b)
- [Delfinado & Edelsbrunner, 1993] C. J. A. Delfinado and H. Edelsbrunner. **An incremental algorithm for Betti numbers of simplicial complexes**. In *Symposium on Computational Geometry*, pages 232–239. ACM, 1993. IV, V.4(b)
- [Dey & Guha, 1998] T. K. Dey and S. Guha. **Computing homology groups of simplicial complexes in \mathbb{R}^3** . *Journal of ACM*, 45(2):266–287, 1998. IV
- [Dumas *et al.*, 2003] J.-G. Dumas, F. Heckenbach, B. D. Saunders and V. Welker. **Computing simplicial homology based on efficient smith normal form algorithms**. In M. Joswig and N. Takayama, editors, *Algebra, Geometry, and Software Systems*, pages 177–206. Springer, 2003. IV, IV.1
- [Edelsbrunner *et al.*, 2000] H. Edelsbrunner, D. Letscher and A. Zomorodian. **Topological persistence and simplification**. In *Symposium on Foundations of Computer Science*, pages 454–463, 2000. V.4(b)
- [Edelsbrunner *et al.*, 2001] H. Edelsbrunner, J. L. Harer and A. Zomorodian. **Hierarchical Morse complexes for piecewise linear 2-manifolds**. In *Symposium on Computational Geometry*, pages 70–79. ACM, 2001. V
- [Edelsbrunner *et al.*, 2003] H. Edelsbrunner, J. L. Harer, V. Natarajan and V. Pascucci. **Morse–Smale complexes for piecewise linear 3-manifolds**. In *Symposium on Computational Geometry*, pages 361–370. ACM, 2003. V
- [Fomenko, 1987] A. T. Fomenko. **Differential geometry and topology**. *Contemporary Soviet Mathematics*. Kluwer, 1987. II, II.5(a), II.6(a)
- [Forman, 1995] R. Forman. **A discrete Morse theory for cell complexes**. In S. T. Yau, editor, *Geometry, Topology and Physics for Raoul Bott*. International Press, 1995. I, I.1(a), I.1(a), I.1(b), II, II.2(b), II.3(b), II.4(b), II.5(b), II.6(b), II.7(b), II.7(b), II.8(b), III.1(a), III.3(a), V.2, V.4(a)
- [Forman, 1998] R. Forman. **combinatorial vector fields and dynamical systems**. *Mathematische Zeitschrift*, 228(4):629–681, 1998. I.1(a)
- [Forman, 2002] R. Forman. **Discrete morse theory and the cohomology ring**. *Transactions of the American Mathematical Society*, 354:5063–5085, 2002. I.1(b), II.7(b)

- [Hachimori Models] M. Hachimori. **Simplicial complex library**. infos-hako.sk.tsukuba.ac.jp/~hachi. III.2(c), IV.10
- [Hatcher, 2002] A. Hatcher. **Algebraic topology**. Cambridge University Press, 2002. II, II.1, II.1(b), II.5(a), II.5(b)
- [Joswig & Pfetsch, 2005] M. Joswig and M. Pfetsch. **Computing optimal discrete Morse functions**. *Electronic Notes in Discrete Mathematics*, 17:191–195, 2005. III.2(c)
- [Kälberer *et al.*, 2005] F. Kälberer, K. Polthier, U. Reitebuch and M. Wardetzky. **Freelence — coding with free valences**. In *Eurographics*, 2005. III.2(e)
- [Lazarus & Verroust, 1999] F. Lazarus and A. Verroust. **Level Set Diagrams of Polyhedral Objects**. In *Solid Modeling and Applications*, pages 130–140. ACM, 1999. V.4(a)
- [Lewiner, 2002] T. Lewiner. **Constructing discrete Morse functions**. Master’s thesis, *Department of Mathematics, PUC–Rio*, 2002. I.1, I.1(a), I.2, II.4(b), II.6, III, III, III.1(a), III.1(c), III.2, III.2(c), III.2(d), V.2(e), VI
- [Lewiner *et al.*, 2003b] T. Lewiner, H. Lopes and G. Tavares. **Towards optimality in discrete Morse theory**. *Experimental Mathematics*, 12(3):271–285, 2003. I.1(a), III, III.2, III.2(c)
- [Lewiner *et al.*, 2003a] T. Lewiner, H. Lopes and G. Tavares. **Optimal discrete Morse functions for 2–manifolds**. *Computational Geometry: Theory and Applications*, 26(3):221–233, 2003. III.2(c), III.2(e)
- [Lewiner *et al.*, 2004] T. Lewiner, H. Lopes and G. Tavares. **Applications of Forman’s discrete Morse theory to topology visualization and mesh compression**. *Transactions on Visualization and Computer Graphics*, 10(5):499–508, 2004. III, III.2, III.2(e)
- [Lewiner *et al.*, 2004] T. Lewiner, H. Lopes, J. Rossignac and A. W. Vieira. **Efficient Edgebreaker for surfaces of arbitrary topology**. In *Sibgrapi*, pages 218–225, Curitiba, Oct. 2004. IEEE. III.2(e)
- [Lopes *et al.*, 2002] H. Lopes, J. Rossignac, A. Safonova, A. Szymczak and G. Tavares. **Edgebreaker: a simple compression for surfaces with handles**. In C. Hoffman and W. Bronsvort, editors, *Solid Modeling and Applications*, pages 289–296, Saarbrücken, Germany, 2002. ACM. III.2(e)

- [Lovasz & Plummer, 1986] L. Lovász and M. D. Plummer. **Matching theory**. Van Nostrand Reinhold, Amsterdam, 1986. III.2
- [Milnor, 1963] J. W. Milnor. **Morse theory**. Number 51 in *Annals of Mathematics Study*. Princeton University Press, 1963. II, II.3(a), II.4(a)
- [Morse, 1925] M. Morse. **Relations between the critical points of a real function of n independent variables**. *Transactions of the American Mathematical Society*, 27:345–396, 1925. I
- [Palis & de Melo, 1982] J. Palis and W. de Melo. **Geometric theory of dynamical systems: an introduction**. Springer, New York, 1982. II.8(a)
- [Rossignac, 1999] J. Rossignac. **Edgebreaker: connectivity compression for triangle meshes**. *Transactions on Visualization and Computer Graphics*, 5(1):47–61, 1999. III.2(e)
- [Rourke & Sanderson, 1972] C. P. Rourke and B. J. Sanderson. **Introduction to piecewise-linear topology**. Springer, New York, 1972. II.1(b), II.5(b), V.3(c)
- [Smale, 1960] S. Smale. **On dynamical systems**. *Boletín de la Sociedad Matemática Mexicana*, 5(2):195–198, 1960. I, II.8(a)
- [Smale, 1961] S. Smale. **Generalized Poincaré’s conjecture in dimensions greater than four**. *Annals of Mathematics*, 74:391–406, 1961. V.4(b)
- [Szymczak & Rossignac, 2000] A. Szymczak and J. Rossignac. **Grow & Fold: compressing the connectivity of tetrahedral meshes**. *Computer-Aided Design*, 32(8/9):527–538, 2000. III.2(e)
- [Tarjan, 1975] R. E. Tarjan. **Efficiency of a good but not linear set union algorithm**. *Journal of the ACM*, 22(2):215–225, 1975. III.2, III.2(a), III.2(d)
- [Touma & Gotsman, 1998] C. Touma and C. Gotsman. **Triangle mesh compression**. In *Graphics Interface*, pages 26–34, 1998. III.2(e)

Index

- Algorithms, 60
 - acyclicity test, 63
 - basin identification, 102
 - cancellation, 102
 - complexity, 67
 - critical cells, 99
 - critical points, 99
 - data structure, 56
 - function to gradient, 56
 - geometric gradient, 100
 - gradient integration, 64
 - greedy construction, 60
 - homology generators, 85
 - homology group, 84
 - Morse boundary, 80
 - union & find, 56
- Cell, 26
 - critical cell, 50
 - dual cell, 50
 - primal cell, 50
- Cell complex, 26
 - cell, 26
 - Euler characteristic, 39
 - incident cells, 28
 - link, 28
 - orientation, 29
 - realization, 27
 - star, 28, 94
 - subdivision, 29
- Critical, 33
 - cancellation, 41
 - cells, 35, 96
 - points, 34, 94
 - saddle, 34, 96
- Flow, 43
 - image of a critical cell, 70
 - image of a dual cell, 70
 - image of a primal cell, 74
 - invariant chains, 43, 72, 80
 - stable basin, 74
 - stable manifold, 43
 - unstable basin, 72
 - unstable basins, 43
 - unstable manifold, 43
- Homology, 38, 77
 - Betti numbers, 38, 84
 - boundary operator, 38, 80
 - chain, 38
 - cycles, 87
 - generators, 85
 - Smith normal form, 78
 - torsion, 84
 - Witten homology, 45, 80, 96
- Homotopy, 36
 - collapse, 39
 - handle decomposition, 37
 - simple homotopy, 39
- Hypergraph, 51
 - dual hypergraph layer, 52
 - hyperforest, 52
 - hyperlink, 51
 - hyperpath, 53
 - hyperpath multiplicity, 71
 - layer hypergraph, 51

- orientation, 51
- primal hypergraph layer, 52
- regular component, 53, 72
- root, 53, 64

Morse

- complex, 44, 99
- inequalities, 40
- Smale decomposition, 45, 99
- theory, 25

Morse function, 31

- discrete gradient, 33
- discrete Morse function, 33
- geometrical function, 99
- gradient from function, 56
- increasing path, 104
- integral path, 32, 73
- optimal function, 66, 96
- path multiplicity, 32
- regular path, 104

Summary of notations

Spaces and functions

X, Y	topological spaces
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	points of a topological space
$\text{Im } g = \{\mathbf{y} : \exists \mathbf{x}, g(\mathbf{x}) = \mathbf{y}\}$	image of g
$\ker g = \{\mathbf{x} : g(\mathbf{x}) = 0\}$	kernel of g
$\text{Id}_{X \rightarrow Y} : \mathbf{x} \mapsto \mathbf{x}$	identity map from X to $Y \subset X$
\mathbb{R}	set of the real number
$\mathbb{R}^n = \mathbb{R} \times \mathbb{R} \times \dots \times \mathbb{R}$	Euclidean space of dimension n
$\ \mathbf{x}\ = \sqrt{\sum_i x_i^2}$	Euclidean norm of point \mathbf{x}
$\mathbb{B}^p = \{\mathbf{x} \in \mathbb{R}^p : \ \mathbf{x}\ < 1\}$	unit ball in \mathbb{R}^p
$\mathbb{S}^{p-1} = \{\mathbf{x} \in \mathbb{R}^p : \ \mathbf{x}\ = 1\}$	unit sphere in \mathbb{R}^p
\mathbb{Z}	set of the relative integers
\mathbb{K}	generic field

Smooth Morse theory

\mathcal{M}	manifold
$T\mathcal{M}$	tangent manifold
$\gamma :]-1, 1[\rightarrow \mathcal{M}$	curve on \mathcal{M}
$f : \mathcal{M} \rightarrow \mathbb{R}$	smooth Morse function on \mathcal{M}
$(x_i), x_i : \mathcal{M} \rightarrow \mathbb{R}$	local parameterization of \mathcal{M}
$\nabla f = \left(\frac{\partial f}{\partial x_i} \right)_i$	gradient of f
$\text{Hess } f = \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right]_{i,j}$	Hessian matrix of f
$H^p = \mathbb{B}^p \times \mathbb{B}^{n-p}$	handle of index p
$\phi : \mathcal{M} \times \mathbb{R} \rightarrow \mathcal{M}, \quad \frac{\partial \phi(\mathbf{x}, t)}{\partial t} = \nabla f(\phi(\mathbf{x}, t))$	flow of ∇f
$W^s(\mathbf{x}) = \left\{ \mathbf{y} \in \mathcal{M}, \phi(\mathbf{y}, t) \xrightarrow{t \rightarrow +\infty} \mathbf{x} \right\}$	stable basin of \mathbf{x}
$W^u(\mathbf{x}) = \left\{ \mathbf{y} \in \mathcal{M}, \phi(\mathbf{y}, t) \xrightarrow{t \rightarrow -\infty} \mathbf{x} \right\}$	unstable basin of \mathbf{x}

Cell complex

ρ, σ, τ	cells, homeomorphic to \mathbb{B}^p
$\rho^{p+1}, \sigma^p, \tau^{p-1}$	cells of dimension $(p+1), p, (p-1)$
$\dim \sigma^p = p$	dimension of σ^p
$ \partial\sigma^p $	geometric boundary of σ , homeo. to \mathbb{S}^{p-1}
$K = \{\sigma^p, p \leq n\}$	cell complex of dimension n
$K^p = \{\sigma^k \in K, k \leq p\}$	p skeleton of K
$ K $	geometric realization of K
$\tau < \sigma : \tau \subset \partial\sigma $	σ incident to τ , τ face of σ
$\partial\sigma = \{\tau_i < \sigma\}$	boundary of σ
$\dot{\text{st}}\tau = \sum_{\tau < \sigma_i} \sigma_i$	open star of τ
$\text{st}\tau = \sum_{\tau < \sigma_i} \sum_{\rho_{ij} < \sigma_i} \rho_{ij}$	star of τ
$\text{lk}\tau = \text{st}\tau - \dot{\text{st}}\tau$	link of τ
$\text{lw}_f\tau = \dot{\text{st}}\tau \cap f^{-1}([-\infty, f(\tau)])$	lower star of τ
$[\sigma^p : \tau^{p-1}] \in \{-1, 0, 1\}$	orientation of τ with respect to σ
$\partial\sigma = \sum_{\tau_i < \sigma} [\sigma : \tau_i] \tau_i$	combinatorial boundary of σ

Homology

$\mathbf{c}_p = \sum_{\sigma^p \in K} c_\sigma \sigma^p, c_\sigma \in \mathbb{K}$	cellular chain of dimension p on ring \mathbb{K}
$C_p = \{\mathbf{c}_p\}$	chain module of dimension p
$\partial_p : C_p \rightarrow C_{p-1}$	boundary operator of dimension p
$H_p = \ker \partial_p / \text{Im } \partial_{p+1}$	Homology group of dimension p
$\beta_p = \text{rank } H_p$	Betti number of index p
$\#_p = \text{card}(K^p \setminus K^{p-1})$	number of p -cells of K
$\chi = \sum_{p \in \mathbb{N}} (-1)^p \#_p$	Euler–Poincaré characteristic of K
$\mathbf{B}_p \in \mathbb{K}^{\#_{p-1} \times \#_p}$	matrix of the boundary operator ∂_p

Discrete Morse theory

$\mathcal{V} = \{(\tau^p < \sigma^{p+1})\}$	discrete vector field on K
$\mathbf{V} : K \rightarrow K \cup \{0\}$	functional of a vector field
$\blacktriangleleft \tau_i^p \sigma_i^{p+1} \tau_{i+1}^p \blacktriangleright, (\tau_i^p > \sigma_i^{p+1}) \in \mathcal{V}$	gradient step
$\blacktriangleleft \tau_0^p \sigma_0^{p+1} \tau_1^p \sigma_1^{p+1} \dots \sigma_{r-1}^{p+1} \tau_r^p \blacktriangleright$	gradient path
$f : K \rightarrow \mathbb{R}^n$	discrete Morse function
$\text{Crit}_p(\mathcal{V}) = \{\sigma^p \in \ker \mathbf{V} \setminus \text{Im } \mathbf{V}\}$	critical cells of \mathcal{V}
$M_p = \left\{ \sum_{\sigma^p \in \text{Crit}_p} c_\sigma \sigma^p, c_\sigma \in \mathbb{K} \right\}$	module of critical p -cells
$m_p = \text{card } \text{Crit}_p$	number of critical p -cells
$\mu(\blacktriangleleft \tau_i \sigma_i \tau_{i+1} \blacktriangleright) = [\mathbf{V}(\sigma_i) : \tau_i] [\mathbf{V}(\sigma_i) : \tau_{i+1}]$	multiplicity of a step
$\mu(\blacktriangleleft \tau_0 \sigma_0 \dots \tau_r \blacktriangleright) = \prod_i \mu(\blacktriangleleft \tau_i \sigma_i \tau_{i+1} \blacktriangleright)$	multiplicity of a path
$\Phi = \text{Id}_{C_p \rightarrow C_p} + \mathbf{V} \partial_p + \partial_{p+1} \mathbf{V}$	discrete flow of \mathcal{V}
$C_p^\Phi = \{\mathbf{c}_p = \Phi^\infty(\mathbf{c}_p)\}$	module of invariant p -chains
$\tilde{\partial}_p = \text{Id}_{C_{p-1}^\Phi \rightarrow C_{p-1}} \circ \partial_p \circ \Phi^\infty$	boundary operator on M_p

Hypergraphs

graph ($Nodes\ N, links : N \times N \rightarrow \{0, 1\}$)	general non-oriented graph
$\mathcal{H} = \text{graph}(K, <)$	Hasse diagram of K
$\text{Prim}_p(\mathcal{V}) = \{\sigma^p \notin \ker \mathbf{V}\}$	primal cells of \mathcal{V}
$\text{Dual}_p(\mathcal{V}) = \{\sigma^p \in \text{Im } \mathbf{V}\}$	dual cells of \mathcal{V}
$\text{hg}(N, Links\ L, incidence\ i : L \times N \rightarrow \mathbb{Z})$	general hypergraph
$\vec{\text{hg}}(N, L, i, orientation : L \rightarrow N \cup \{0\})$	general oriented hypergraph
$\mathcal{L}_{p(p+1)} = \vec{\text{hg}}(\text{Prim}_p \cup \text{Crit}_p, \text{Dual}_{p+1}, <, \mathcal{V})$	primal layer of \mathcal{V}
$\mathcal{L}_{p(p-1)} = \vec{\text{hg}}(\text{Dual}_p \cup \text{Crit}_p, \text{Prim}_{p-1}, >, \mathcal{V})$	dual layer of \mathcal{V}
$\triangleleft n_0 l_0 n_1 l_1 \dots l_r n_r \triangleright$	hyperpath
$n \rightsquigarrow n' \text{ if } \exists l \in \mathcal{L} : n > l, (l, n') \in \mathcal{V}$	n father of n' , n' son of n