Thomas Lewiner

Constructing Discrete Morse Functions

MsC Thesis

DEPARTMENT OF MATHEMATICS

Post–graduate program in Applied Mathematics

Rio de Janeiro
July 2002

Pontifícia Universidade Católica
do Rio de Janeiro

Thomas Lewiner

Constructing Discrete Morse Functions

MsC 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 Master in Applied Mathematics

Advisor : Prof. Hélio Côrtes Vieira Lopes
Co–Advisor: Prof. Geovan Tavares dos Santos

Rio de Janeiro
July 2002

Pontifícia Universidade Católica
do Rio de Janeiro

Thomas Lewiner

Constructing Discrete Morse Functions

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 Master in Applied Mathematics. Approved by the Examinating Comission:

Prof. Hélio Côrtes Vieira Lopes
Advisor
Department of Mathematics — PUC–Rio

Prof. Geovan Tavares dos Santos
Co–Advisor
Department of Mathematics — PUC–Rio

Prof. Luis Carlos Pacheco R. Velho
IMPA

Prof. Jorge Stolfi
UNICAMP

Prof. Ney Augusto Dumont
Coordinator of the Science and Engineering Center — PUC–Rio

Rio de Janeiro, 10 de July de 2002

Thomas Lewiner

He graduated from the Ecole Polytechnique (Paris, France) in Algebra and Computer Science, and in Theoretical Physics. He specialized himself at the Ecole Supérieure des Télécommunications (Paris, France) in Signal and Image Processing, and in Project Management. He worked for Inventel in wireless telecommunication systems based on BlueTooth technology. During his master preparation, he developed with his advisors tools oc computational topology.

Ficha Catalográfica

Lewiner, Thomas

Constructing Discrete Morse Functions/ Thomas Lewiner; advisor: Hélio Côrtes Vieira Lopes; co–advisor: Geovan Tavares dos Santos. — Rio de Janeiro : PUC–Rio, Department of Mathematics, 2002.

v., 89 f: il. ; 29,7 cm

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

Bibliography included.

1. Mathematics – Thesis. 2. Morse Theory. 3. Forman Theory. 4. Computacional Topology. 5. Computacional Geometry. 6. Geometric Modelling. 6. 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

To my advisors Professors Hélio Lopes and Geovan Tavares for the support, the everyday kindness and the incentive for the realisation of this work.

To the CNPq and the PUC–Rio, for the financial support, without which this work would not have been realized.

To my grand–mothers, who suffered the most of my expatriation. To my mother and father, sisters and all my family.

To my colleagues of the PUC–Rio, who have me loved this place.

To the professors Marcos da Silvera, Jean–Marie Nicolas and Anne Germa who offered me the opportunity of this cooperation.

To the people of the Mathematic departament for the constant help, particularly to Ana Cristina, Creuza and Sinesio.

Abstract

Lewiner, Thomas; Lopes, Hélio Côrtes Vieira; Santos, Geovan Tavares dos. **Constructing Discrete Morse Functions**. Rio de Janeiro, 2002. 89p. MsC Thesis — Department of Mathematics, Pontifícia Universidade Católica do Rio de Janeiro.

Morse theory has been considered a powerful tool in its applications to computational topology, computer graphics and geometric modeling. It was originally formulated for smooth manifolds. Recently, Robin Forman formulated a version of this theory for discrete structures such as cell complexes. It opens up several categories of interesting objects (particularly meshes) to applications of Morse theory.

Once a Morse function has been defined on a manifold, then information about its topology can be deduced from its critical elements. The purpose of this work is to design an algorithm to define optimal discrete Morse functions on general cell complex, where optimality entails having the least number of critical elements. This problem is proven here to be MAX–SNP hard. However, we provide a linear algorithm that, for the case of 2–manifolds, always reaches optimality.

Moreover, we proved various results on the structure of a discrete Morse function. In particular, we provide an equivalent representation by hyperforests. From this point of view, we designed a construction of discrete Morse functions for general cell complexes of arbitrary finite dimension. The resulting algorithm is quadratic in time and, although not guaranteed to be optimal, gives optimal answers in most of the practical cases.

Keywords

Morse Theory; Forman Theory; Computational Topology; Computational Geometry; Solid Modeling; Discrete Mathematics.

Contents

List of figures

List of tables

Un abîme effrayant, une profusion de questions de toutes sortes où ma responsabilité était en jeu se présentaient à moi. Et la plus importante: qu'est–ce qui doit remplacer l'objet manquant? Le danger d'un art ornemental m'apparaissait clairement, la morte existence illusoire des formes stylisées ne pouvait que me rebuter.

C'est seulement après de nombreuses années d'un travail patient, d'une réflexion intense, d'essais nombreux et prudents où je développais toujours plus la capacité de vivre purement, abstraitement les formes picturales et de m'absorber toujours plus profondément dans ces profondeurs insondables, que j'arrivais à ces formes picturales avec lesquelles je travaille aujourd'hui et qui, comme je l'espère et le veux, se développeront bien plus encore.

Il a fallu beaucoup de temps avant que cette question: 'qu'est ce qui doit remplacer l'objet ?' trouve en moi une véritable réponse. Souvent je me retourne vers mon passé et je suis desespéré de voir combien de temps il m'a fallu pour arriver à cette solution.

Wassily Kandinsky, *Regards sur le passé.*

Foreword

This whole work was originally motivated by 3D mesh compression. When it begun, my teacher Hélio Lopes had just proved an extension of the EdgeBreaker's compression scheme for orientable surfaces with genus. He did what I always admired: joining two fields of research (computer graphics and algebraic topology) to improve both of them. The mixture of algebra and topology already attracted me for the same reason, although I have had no occasion to learn it seriously. The mixing of topology with computer graphics has now its *lettres de noblesse*: the growing field of computational topology. The need of this discipline seems now obvious to me. First, topology and geometry always had a tight relation in mathematics, and geometrical proofs and properties frequently required some topological analysis. Second, the computational topology approach is a good way of avoiding dirty programming. Moreover, it helps improving algorithms, extending them to broader applications. Therefore, I was eager to learn more on that field, when my teacher proposed me a "simple" problem.

From my other teacher, Geovan Tavares, there has been a tradition at the MatMídia laboratory to use Morse theory. This tool has contributed both to strong mathematical theorems (as the Poincaré's conjecture in dimension above 4) and to computer graphics (for mesh generation and compression). However, this theory was formulated for smooth manifolds, which require some work before leading to rigorous implementation. The approximation of smooth properties by discrete structures can be done efficiently mainly for specific applications. However, leading discrete applications have usually emerged from discrete theories. This approach has given at least two complementary combinatorial differential geometries. One of them emerged from computer graphics, led by Mathieu Desbrun [Mey02]. The other one, grounded in topology and combinatorics, was stated by Robin Forman [For98].

The "simple" problem of my teacher was to build a discrete Morse function as defined by Forman. The problem was actually not obvious at all, and a mysterious paper of Ömer Eğecioğlu [Ege95] seemed to prove that it was more difficult than NP–hard. This impression turned out to be true a few months later.

I first oriented my research on applications to mesh compression, and I looked at related algorithms for some ideas. Compression schemes based on topological surgery [Tau98] offered an attracting trail, and rapidly allowed me to formulate a small algorithm. This algorithm had many implementation difficulties, but worked well on small models. However, it failed on non–manifolds, and gave bad results for non–spheres. Trying to understand the error, I went back to my lecture notes of the Ecole Polytechnique. There was a very simple demonstration of the Euler characteristic for spheres, based on a spanning–tree of the dual graph and its complement. The main idea was now obvious: define discrete Morse functions on forests, leaving the circuits as critical.

Along this work, I discovered that this idea was actually the intrinsic structure of a discrete Morse function. A remark that did not appear in Forman's work. The explication was not obvious. The general representation of a dual of a cell complex is, in general, much more complex than a graph. Looking in the literature, this structure was defined as a *hypergraph*. Nevertheless, there was no extension of the notion of forest for hypergraph. It took me quite a long time to have an elegant formulation for this structure, which I called hyperforest. Once the mathematics was clear, the algorithm followed directly. I knew the problem was worse than NP–hard, but I proved the first algorithm was optimal for 2–manifolds, and the results I obtained in the general case seem very close to the optimum.

My teachers enhanced computer graphics from their knowledge of topology. This work has perhaps a complementary approach. From computational problems arose mathematical questions. I am still surprised of how fast we encountered opened problems of mathematics, such as the Poincaré's conjecture. But due to the existence of those computational problems we also found some answers.

I
Introduction

I.1 Motivations and applications

Computational geometry [Boi98] has led to major improvements in computer graphics, robotics, and computer–aided design. This field focuses mainly on discrete problems involving point sets, polygons, and polyhedrons, and uses combinatorial techniques to solve them, with emphasis on provable correctness, efficiency, and robustness. Its applications now involve information visualization, advanced scientific and engineering computation, and computational algorithms and methods.

Need for topological considerations. Earlier research in computational geometry has led to inextricable connections with combinatorial geometry, to the great benefit of both fields. Nowadays, some of the most difficult and least understood issues in geometric computing involve topology: when the emphasis lies on connectivity, continuity, on space, and on maps. This does not mean that the more geometric notions are absent from those problems, but rather that focusing on topological properties (i.e. separating global shape properties from local geometric attributes) leads to better and more elegant results. Many simple algorithms got stuck on topological singularities. Trying to detect those problems by geometric notions required expensive geomtric primitives, loosing robustness. For example, 3d mesh compressions scheme as the famous EdgeBreaker has been considerably improved at a very low computational cost by topological considerations [Lop02].

Computational Topology and its applications. The emerging field of computational topology [Veg97] deals actually with a broader scope of problems. Dey & al. [Dey99] have introduced six areas of applications: image processing, cartography, computer graphics, solid modeling, mesh generation and molecular modeling. The first Workshop on Computational Topology [Ber99] identified five other areas of applications: shape acquisition, shape representation, physical simulation, configuration spaces and topological computation. They delimited this field to encompass both algorithmic questions in topology (for example, recognizing knots) and topological questions in algorithms (for example, whether a discrete construction preserves the topology of the underlying continuous domain).

Scope of this work. For the matter of this work, the applications of computational topology include meshing, morphing, feature extraction, data compression, surface coding and more, in areas such as computer graphics, solid modeling, bio–informatics and computational medicine. With those objectives in mind, we added a new theory of combinatorial topology, named *discrete Morse theory*, [For98] to the toolbox of computational topology.

Morse theory [Mil63] is a fundamental tool for investigating the topology of smooth manifolds. Particularly for computer graphics, many applications have been deduced from the smooth case [Ede01, Har98, Lop96, Shi91]. Morse proved that the topology of a manifold is very closely related to the critical points of a real smooth map defined on it. The simplest example of this relationship is the fact that if the manifold is compact, then any continuous function defined on it must have a maximum and a minimum. Morse theory provides a significant refinement of this observation.

Forman's discrete Morse theory. The recent insights in Morse theory by Forman [For95, For98] extended several aspects of this fundamental tool to discrete structures. Its combinatorial aspect allows computation completely independent of a geometric realization: the algorithms we designed do not require any coordinate or floating–point calculation, and geometrical constraints can be applied independently. Forman proved several results and provided many applications of his theory [For00, For01].

I.2 Results

Once a Morse function has been defined on a smooth manifold, then information about its topology can be partly deduced from its critical points (i.e. the points where the gradient vanishes). Similarly to the differential case, Forman proved that the topology of a CW–complex can be partly read out of the critical cells of a discrete Morse function defined on it.

Therefore, the starting point of direct applications of this theory is to build a discrete Morse function. The topological information will be concise if the discrete Morse function has few critical cells. Thus, we will say that a Morse function is *optimal* if it has the minimum possible number of critical cells. The main contribution of this work is to build such functions.

Algorithm for optimal discrete Morse functions. We provide in chapter IV a linear algorithm to construct discrete Morse functions on 2–cell complexes. This algorithm is shown to be optimal for 2–manifolds in section IV.4. We extend this algorithm in chapter VI to build discrete Morse functions and discrete gradient vector field on general cell complexes of arbitrary dimension. This algorithm is quadratic in execution time. It is not guaranteed to be optimal, but it gives optimal results in most of the cases (cf section VI.5).

Theoretical results. From the theoretical point of view, we proved that constructing an optimal discrete Morse function is MAX–SNP hard (theorem III.19). We proved that the problem of finding a maximal hyperforest in a hypergraph is also MAX-SNP hard (section V.2.3). By the way of exposing discrete Morse theory, we provide a construction of optimal discrete Morse function on a cartesian product, given optimal discrete Morse function on each of its factor (cf section III.2.3). In chapter V, we develop a hypergraph representation of discrete Morse functions. We introduce the notion of hyperforest (section V.2.1) and prove the equivalence between discrete gradient vector fields and hyperforests in theorem V.8. We stated the equivalent of a critical cell for a hyperforest in proposition V.10. The optimality of a hyperforest is discussed in section VI.3. We finally proved for the case of 3–manifolds that discrete Morse numbers are topological invariants related to the simple–homotopy type (theorem V.14).

I.3 Outline

This work is organized as follow. Topological preliminaries and graph structures, particularly hypergraphs and the Hasse diagram, are discussed in chapter II. Forman's discrete Morse theory is introduced in chapter III, together with a proof of the complexity of finding an optimum discrete Morse function. Chapter IV gives a linear construction of discrete Morse function on cell complexes of dimension 2, as exposed in [Lew01]. This construction is optimal in the case of 2–manifolds. Chapter V discusses most of the theoretical results of this work, particularly the notions of hyperforest and its critical elements, and the proof of topological invariance of discrete Morse numbers for 3–manifolds. Finally, we give a construction of a discrete Morse function on general cell complexes of arbitrary dimension in chapter VI. This is a complete presentation of the algorithm partly introduced in [Lew02].

II
Preliminaries

II.1 Discrete structures

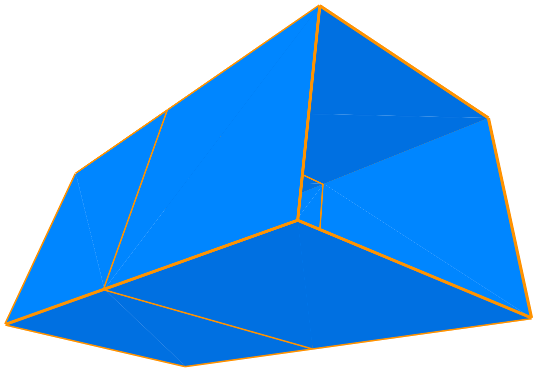


Figure II.1: An 8-squares cylinder model.

Morse theory was originally formulated for continuous structures (smooth manifolds). Implementing tools related to this theory involves approximation of those structures to more directly computable ones. One of the main advantages of Forman's theory is that it is directly formulated for discrete structures such as cell complexes (cf figure II.1). A complete introduction to graphs and hypergraphs structure can be found in [Ber70].

II.1.1 Graphs

The simplest discrete structures used in computation are simple graphs (cf figure II.2. All the graphs of this work were drawn by GraphViz dot [GFZdot] and neato [GFZneato] softwares.

Definition II.1 (Simple graph) *A* simple graph *is a pair* (N, L)*, where* N *is a set of objects called* nodes *and* L *is a set of pairs of nodes. The elements of* L *will be called* links*.*

We say that a link *joins* its two end nodes, and that those nodes are *adjacent*. A graph can be oriented by distinguishing for each link one of its end nodes as its *source node*.

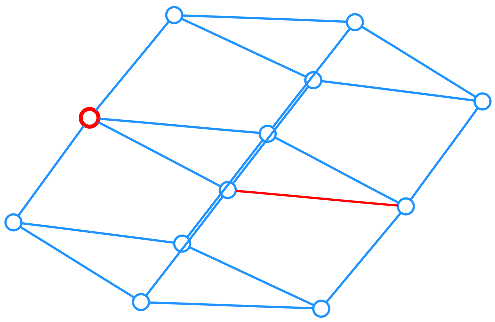


Figure II.2: The 0/1 graph of an 8-squares cylinder (figure II.1): nodes represents vertices, links represents edges.

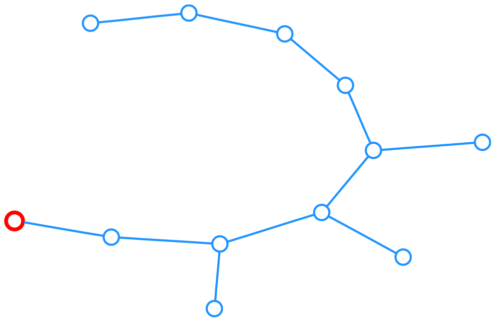


Figure II.3: A tree extracted from the graph of figure II.2.

A *path* in a graph is a sequence of pairwise distinct links, successively adjacent. Such a path is a *circuit* if the first and the last node of the sequence are identical. An *oriented path* in an oriented graph is a path where each link between two consecutive nodes has the first node as its source node. A graph with no circuit is called a *forest*, i.e. a union of trees (cf figure II.3):

Definition II.2 (Tree) *A simple graph is a* tree *if it is connected and contains no circuit.*

A *leaf* is a node that is incident to aat most one link. A graph (N', L') is a *subgraph* of a graph (N, L) (or a graph *extracted* of it) if $N' \subset N$ and $L' \subset L$. This subgraph is a *spanning graph* if $N' = N$.

II.1.2 Matching and bipartite graphs

Bipartite graphs became famous for matching problems [Lov86]:

Definition II.3 (Matching) *A* matching *on a graph is a subset of its links of such that no two of them have a vertex in common.*

For example, pairing boys and girls at school could be represented by a matching in a bipartite graph: the boys are one class of nodes, girls the other one, and the affinity between boys and girls are represented by a link. [Ber70] gives a complete introduction to bipartite graphs. Figure II.5 shows an example of a (partial) matching on graph II.4.

Definition II.4 (Bipartite graph) *A simple graph is a* bipartite graph *when its set of node can be decomposed into two disjoint sets, called here the* $\boldsymbol{N}$ *and* $\boldsymbol{L}$ *classes, such that no two nodes within the same set are adjacent.*

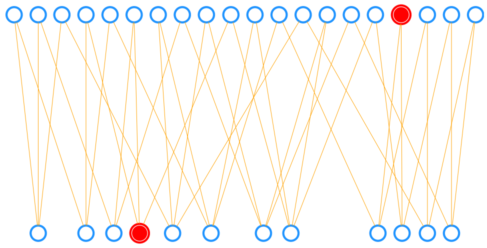


Figure II.4: A bipartite representation of the graph of figure II.2.

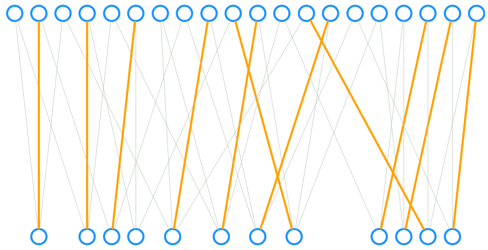


Figure II.5: A matching on the graph of figure II.4.

II.1.3 Pseudographs

We will see in section II.2.4 that the dual graph of a manifold without boundary is a simple graph. However, if the manifold has a boundary, the links representing (n-1)–cells of the boundary would be incident to only one node. For example, figure II.6 shows the dual graph of the 8-squares cylinder model (figure II.1), and figure II.7 shows a (pseudo–) tree extracted out of it. This would not fit in definition II.1, but in the following one:

Definition II.5 (Pseudograph) *A* pseudograph *is a pair* (N, L)*, where* L *is a family of subsets of* N *each of which having either 1 or 2 nodes.*

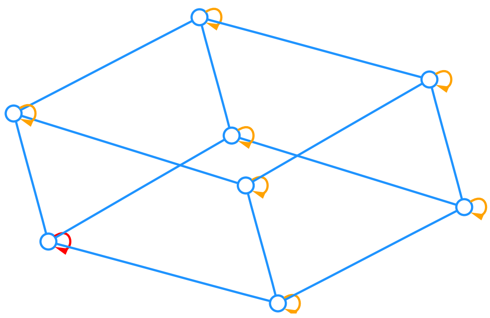


Figure II.6: The dual graph of figure II.1.

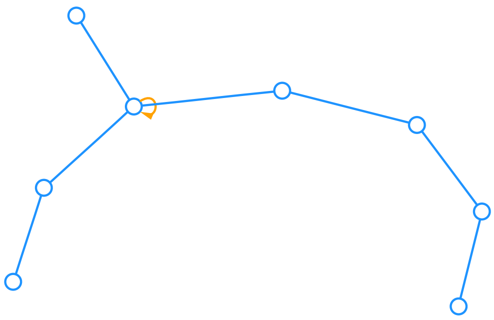


Figure II.7: A tree extracted from the graph of figure II.6.

II.1.4 Simply oriented hypergraphs

In the dual graph of a non–manifold cell complex, links that join more than two nodes may appear (cf figure II.8). This would not fit in definition II.5, but in the following one:

Definition II.6 (Hypergraph) *A* hypergraph *is a pair* (N, L)*, where* L *is a family of families of* N*. The elements of* L *are called* hyperlinks*.*

We will classify non–empty hyperlinks into the *regular hyperlinks* (or shortly, *link*), which join two distinct nodes as in simple graphs, the *loops*, which are incident to only one node, and the *non–regular hyperlinks*, which either join three nodes or more or are multiply incident to one node. We can extract the simple graph part of a hypergraph by considering its *regular components*:

Definition II.7 (Regular components) *The* regular components *of a hypergraph* (N, L) *are the connected components of the simple graph* (N, R)*, where* R *is the set of the regular hyperlinks of* (N, L)*.*

We will give a hypergraph a simple orientation by distinguishes one node of each hyperlink as its *source node*.

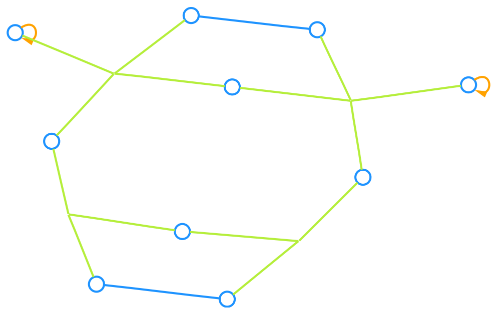


Figure II.8: A hypergraph example.

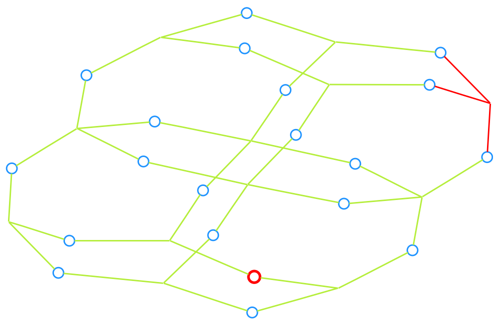


Figure II.9: The dual hypergraph of the graph of figure II.2.

II.1.5 Dual of a hypergraph

The dual of a hypergraph [Ber70] is obtained by reading the nodes as hyperlinks and the hyperlinks as nodes. For example, figure II.9 shows the dual hypergraph of the vertex/edge graph of the 8-squares cylinder (figure II.2).

Definition II.8 (Dual of a hypergraph) *The dual* $\mathcal{D}(H)$ *of a hypergraph* H *is a hypergraph whose nodes are the hyperlinks of* H *and whose hyperlinks joins the hyperlinks of* H *that share a node, loop for each leaf of* H *or are empty for each isolated node.*

If each node of an oriented hypergraph H is the source of at most one hyperlink, then the dual $\mathcal{D}(H)$ of H can be partly oriented as follow: if a node n is the source of a hyperlink l in H, the node representing l in $\mathcal{D}(H)$ is the source of the hyperlink representing n. We notice that the dual operation is an involution: $\mathcal{D} \circ \mathcal{D}(H) = H$.

II.1.6 Bipartite representation of a hypergraph

A hypergraph can be represented by a bipartite graph [Ber70]. For example, figure II.4 gives a representation of the hypergraph of figure II.2. This gives a simple but expensive representation of hypergraphs:

Definition II.9 (Bipartite graph of a hypergraph) *The* bipartite graph $\mathcal{B}(H)$ *of a hypergraph* H *is the simple graph whose* $\boldsymbol{N}$ *and* $\boldsymbol{L}$ *nodes are the nodes and the links of* H *respectively. For every hyperlink* l *of* H*, there are* $\#l$ *links of* $\mathcal{B}(H)$ *joining the node representing* l *in* $\mathcal{B}(H)$ *to each representing node, in* $\mathcal{B}(H)$*, incident to* l *in* H*.*

When H is oriented, $\mathcal{B}(H)$ will be oriented the following way:

> If a node n of H is the source of a hyperlink l, then the node representing l will be the source of the link of $\mathcal{B}(H)$ joining n to l.
>
> If a node n of H is the not source of an incident hyperlink l, then its representing node in $\mathcal{B}(H)$ is the source node of the link joining it to the representing node of l in $\mathcal{B}(H)$

II.1.7 Hypergraph representations of a bipartite graph

The operation of taking the bipartite graph of a hypergraph can be reversed. Depending on which class of nodes becomes the links of the hypergraph, we can obtain a hypergraph or its dual. For example, figure II.4 can be represented by both figures II.2 and II.9. The bipartite graph is not supposed to have a consistent orientation in the general case. Therefore, the hypergraph representing a bipartite graph will not always be oriented.

Definition II.10 (Hypergraphs of a bipartite graph) *A bipartite graph* B *admits two representations by hypergraphs:* $\mathcal{B}^{-1}(B)$ *and its dual* $\mathcal{D}(\mathcal{B}^{-1}(B))$*. The nodes of* $\mathcal{B}^{-1}(B)$ *are the* $\boldsymbol{N}$ *class of nodes of* B*. For every node* l *of the* $\boldsymbol{L}$ *class, there is a hyperlink of* $\mathcal{B}^{-1}(B)$ *joining all the nodes adjacent to* l*.*

II.2 Basic concepts of algebraic topology

The intuition behind topology is the study of the properties of shapes, which remain unchanged under deformation. Some of those can be expressed algebraically as structures defined on topological spaces. This part of topology is called *algebraic topology.*

II.2.1 Point set topology

A natural way to investigate the space we are living in consists of analyzing what lies around us. We get information about our space by considering local neighborhoods. This is the art of topology, which deduces from those local observations some global properties, as living on a sphere rather than on a disc.

Definition II.11 (Topological space) *A* topological space *is a set of points X with a collection V of subsets of X called* open sets*, with the following restrictions:*

The union of open sets is an open set.

The intersection of two open sets is an open set.

The empty set and the set X are open sets.

Those open sets define the notion of *neighborhood*: a neighborhood of a point is any open set containing it. A simple way of defining a topology on a set consists in using a metric d on that space. In that case, the open sets can be generated as unions and intersections, of open balls $\mathrm{B}_{c,r} = \{x \in X : d(x, c) < r\}$.

In this work, we will consider *Hausdorff spaces of finite dimension*, i.e. bounded spaces where there always exist disjoint neighborhoods for distinct points. We would first notice that there are infinitely many different such spaces, and topology aims at describing and classifying those topological spaces. Among the tools for describing such spaces, one of the most important is the notion of map.

Definition II.12 (Continuous function) *Let X and Y be two topological spaces. A function $f : X \to Y$ is a* continuous function *(or* map*) if the inverse image by f of every set open in Y is open in X.*

Definition II.13 (Topological equivalence) *Two spaces X and Y are said to be* topologically equivalent*, or* homeomorphic*, if there exists a continuous function $f : X \to Y$, invertible, whose inverse $f^{-1} : Y \to X$ is continuous.*

A genuine way for classifying topological spaces would require being able to generate all of them and then to determine if two spaces are topologically equivalent. Unfortunately, this problem is much too complex. For dimension 4 and above, this cannot be computed even with an ideal computer [Mar58]. Even from dimension 2 on, we will encounter in this work some NP hard problems (cf section III.4.3).

However, there exist other computable tools that can describe topological properties [Del93, Dey01] and that can prove, in some cases, that two spaces are not homeomorphic. For example, if there is no *homotopy* (i.e. continuous deformation) between two spaces, they cannot be topologically equivalent. Morse theory [Mil63] unifies many of those tools.

II.2.2 Cell complexes

A cell complex is, roughly speaking, a generalization of the structures used to represent solid models: it is a consistent collection of cells (vertices, edges, faces...). In particular, triangulations of topological spaces or 3D meshes are cell complexes (cf figure II.10). Figure II.11 gives a minimal construction of a torus cell complex. A complete introduction to cell complexes can be found in [Lun69].

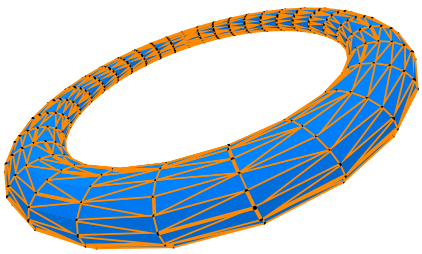


Figure II.10: A triangulated torus.

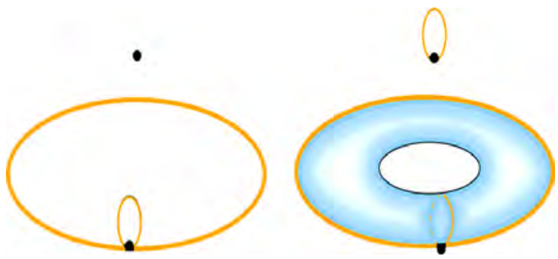


Figure II.11: A construction of a torus with 4 cells.

Definition II.14 (Cell) *A* cell $\alpha^{(p)}$ *of dimension* p *is a set homeomorphic to the open* p*–ball* $\{x \in \mathbb{R}^p : \|x\| < 1\}$.

When the dimension p of the cell is obvious, we will simply denote α instead of $\alpha^{(p)}$.

Definition II.15 (CW–complex) *A* CW–complex K *is built by starting off with a discrete collection of 0–cells (vertices) called* K^0*, then attaching 1–cells (edges) to* K^0 *along their boundaries, obtaining* K^1*, then attaching 2–cells (faces) to* K^1 *along their boundaries, writing* K^2 *for the new space, and so on, giving spaces* K^n *for every* n.

A CW–complex will be said to be finite when it is built out of a finite number of cells. In this work, we will consider only finite (and thus regular) CW–complexes. This permits to compute them.

A p–cell $\alpha^{(p)}$ is a face of a q–cell $\beta^{(q)}$ $(p < q)$ if $\alpha \subset \mathit{closure}\,(\beta)$. If $q = p - 1$, we will use the notation $\alpha^{(p)} \prec \beta^{(q)}$, and say that α and β are *incident*.

II.2.3 Hasse diagram of a cell complex

Cell complexes and graphs are discrete structures. In a sense, a cell complex is a generalization of a graph, as a graph can be seen as a cell complex of dimension 1. Nevertheless, we can also represent a cell complex by a pseudograph, called the *Hasse diagram*.

Definition II.16 (Hasse diagram) *The* Hasse diagram *of a cell complex K is the oriented pseudograph H:*

Each node of H represents a cell of K.

The links of H joins nodes representing incident cells of K. The source node of each link is the one of highest dimension.

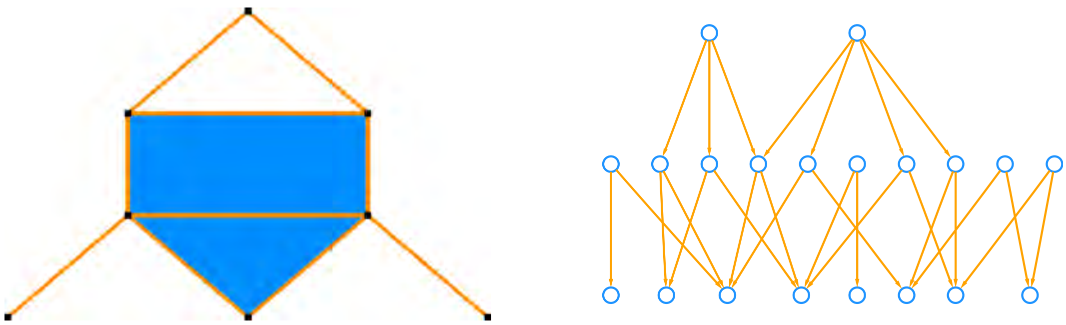


Figure II.12: The Hasse diagram of a simple cell complex.

The Hasse diagram is usually drawn with the nodes ranked by their dimension. On figures II.12 and II.13, the faces (2–cells) are aligned on top rank, the edges (1–cells) on the middle one and the vertices (0–cells) on the bottom rank. A link between two nodes symbolizes that the corresponding cells are incident.

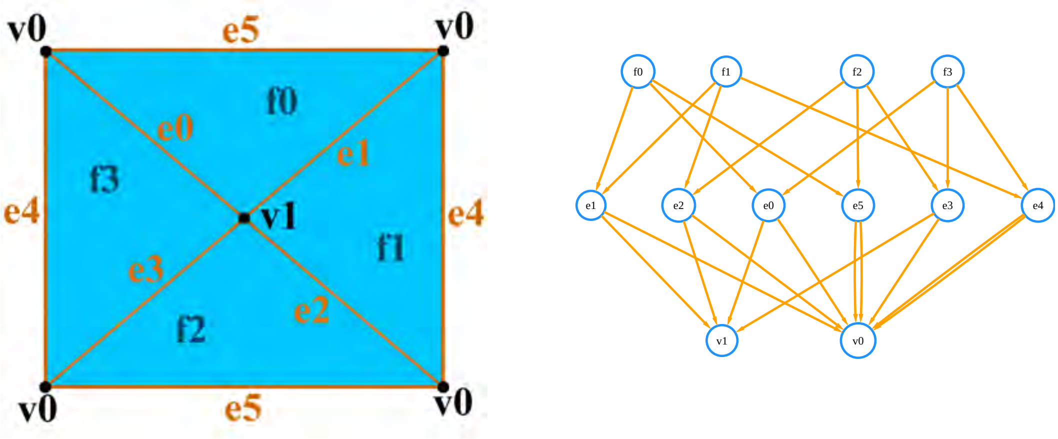


Figure II.13: The Hasse diagram of a non–PL torus.

II.2.4 Manifolds

Definition II.17 (Manifold) *An* n–manifold *is a topological space in which every point has a neighborhood homeomorphic to either* $\mathbb{R}^n$ *or* $\mathbb{R}_+ \times \mathbb{R}^{n-1}$.

In this work, the only topological space we will consider are cell complexes, thus *manifold* will denote a cell complex which has the topology of a manifold. Figures II.14 and II.15 gives example of non–manifold cell complexes.

The set of points whose neighborhood is $\mathbb{R}_+ \times \mathbb{R}^{n-1}$ is called the *boundary* of the manifold. Notice that the boundary of a compact n–manifold is an (n-1)–manifold without boundary. It can be shown [Lun69] that if a finite cell complex is an n–manifold, then each (n-1)–cell is incident to either one or two n–cells.

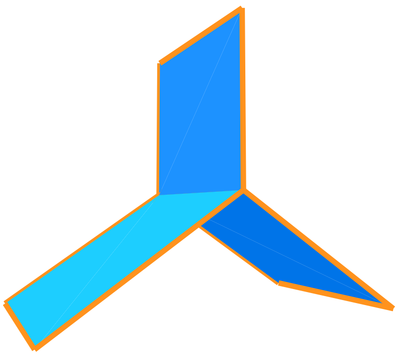


Figure II.14: A non–manifold example: non–regular edge.

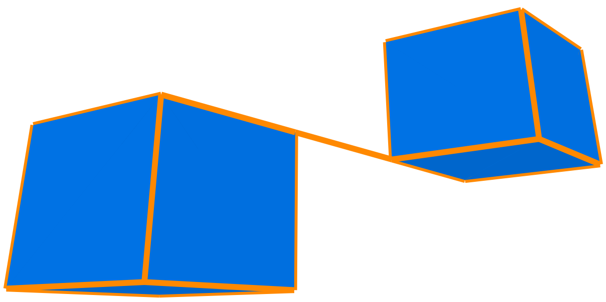


Figure II.15: Another non–manifold example: the neighborhood of the points of the dangling edge is homeomorphic to $\mathbb{R}$.

The dual graph of a manifold is the pseudograph whose nodes are the n–cells and where each (n-1)–cell σ is represented by a link l in this pseudograph: l joins the nodes representing n–cells incident to σ (cf figure II.16).

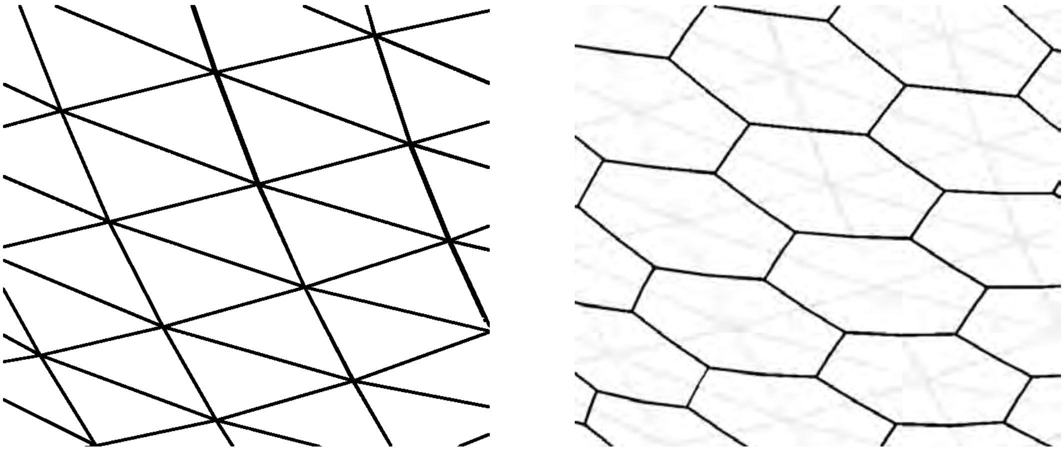


Figure II.16: A part of a triangulation and its dual.

II.2.5 Homology theory and the Betti numbers

Homology theory is an efficient way of describing some connectivity properties of cell complexes, as the number of connected components, of holes, of voids... Its major invariants, the Betti numbers, appears in Morse theory as a lower bound to the number of critical elements (cf section III.3.2). We will here introduce the $\mathbb{Z}_2$ homology, as it leads to higher values of the Betti numbers, and thus tighter bounds in the Morse inequalities (cf section III.3.2).

Definition II.18 (Chain) *A p–*chain *$c^{(p)}$ is a subset of p–cells of a cell complex K:*

$$c^{(p)} = \sum_{\sigma^{(p)} \in K} c_\sigma . \sigma^{(p)}, \qquad c_\sigma \in \{0, 1\}.$$

The coefficients $c_\sigma \in \mathbb{Z}_2$ indicate whether the cell σ belongs to the chain c or not. The addition of two p–chains is trivially defined element–wise on each cell. In other terms, the addition of two p–chains is the symmetric difference of the two sets $c + d = (c \cup d) \setminus (c \cap d)$. The group C_p of all p–chains is called the *chain group* of order p of a given cell complex. The empty set is the zero element of C_p.

The boundary $\partial_p\left(\sigma^{(p)}\right)$ of a p–cell σ is the collection of its (p-1)–dimensional faces, which is a (p-1)–chain. The *boundary operator* ∂_p is extended to p–chain by linearity:

$$\partial_p\left(\sum_{\sigma^{(p)} \in K} c_\sigma . \sigma^{(p)}\right) = \sum_{\sigma^{(p)} \in K} c_\sigma . \partial_p\left(\sigma^{(p)}\right)$$

Definition II.19 (Cycles and boundaries) *A p–*cycle *$z^{(p)}$ is a p–chain of K whose boundary is null: $\partial_p\left(z^{(p)}\right) = 0$. A p–*boundary *$b^{(p)} = \partial_{p+1}\left(c^{(p+1)}\right)$ is the boundary of a (p+1)–chain.*

Since the boundary operator ∂_p preserves the addition from C_p to C_{p-1}, the set of the p–boundaries $\mathrm{Im}\partial_{p+1}$ and the set of the p–cycles $\ker\partial_p$ are subgroups of C_p.

An essential property of the boundary operators is that the boundary of a boundary is always empty ($\partial_p \circ \partial_{p+1} = 0$). So every p–boundary is a p–cycle and $\mathrm{Im}\partial_{p+1} \subseteq \ker\partial_p$.

Definition II.20 (Homology groups) *For each p, the p–th* homology group *$H_p = \ker\partial_p \;/\; \mathrm{Im}\partial_{p+1}$ (with coefficients in $\mathbb{Z}_2$) is obtained by equating two p–cycles that only differ by a p–boundary:*

$$\forall z^{(p)}, t^{(p)} \in \ker\partial_p, \qquad z^{(p)} \equiv t^{(p)} \Leftrightarrow z^{(p)} - t^{(p)} \in \mathrm{Im}\partial_{p+1}.$$

Those homology groups are commutative and finitely generated (the cell complex is finite). Thus, they can be written as $H_p = \mathbb{Z}_2^{\beta_p}$, where β_p is called the p–th *Betti number* with coefficients in $\mathbb{Z}_2$:

Definition II.21 (Betti numbers) *The* Betti numbers *are the ranks of the Homology groups.*

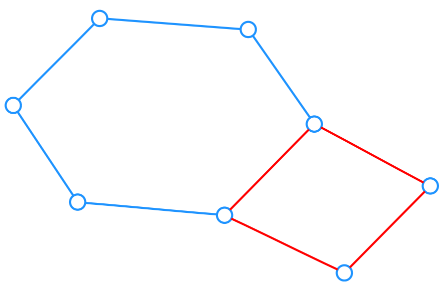


Figure II.17: Cycles in a 1–cell complex are circuits.

Figure II.18: The cycles of a map are around seas (holes), where lands are faces and frontiers are edges.

The basic interpretation for Betti numbers is a way of counting "holes" in a given complex: β_0 counts the number of connected component, β_1 counts the tunnels of a surface, β_2 the voids of a solid... The cycles of a graph (i.e. 1–cell complex) are the independent circuits (cf figure II.17). Thus, the homology of a graph is its connectivity (the minimal number of edges to remove to obtain a forest). On figure II.18, we can compare equated cycles (in yellow and red), and null cycles (in green).

II.2.6 Homotopy and simple homotopy

If two spaces are topologically equivalent, then they have isomorphic homology groups. But the homology is not sufficient to distinguish two spaces. A more refined tool to describe the topology of space is its *simple homotopy type* [Coh73]. If two spaces have the same simple homotopy type, they have the equivalent homology groups. But the contrary is not true. A famous counter–example is the homological sphere of Poincaré. The question whether the simple homotopy is sufficient is not solved yet, and part of it relies in the Poincaré's conjecture. Simple homotopy is a weak version of homotopy, but is more directly linked to discrete Morse theory.

Definition II.22 (Homotopic functions) *Two continuous functions f and g from X to Y are said to be* homotopic *if there exists a continuous function H from $X \times [0,1]$ to Y such that $\forall x \in X, H(x,0) = f(x)$ and $H(x,1) = g(x)$.*

Intuitively, the second argument of H can be viewed as time, and then the homotopy describes a continuous deformation from f to g.

Definition II.23 (Homotopy type) *Two topological spaces X and Y have the same homotopy type if there exists two continuous functions $f : X \to Y$ and $g : Y \to X$ such that $f \circ g$ is homotopic to the identity map of X and $g \circ f$ is homotopic to the identity map of Y.*

A topological space is *contractible* if it has the same homotopy type of a point. A way of showing that two spaces have the same homotopy type is to show that they retract by deformation on homotopic spaces.

Definition II.24 (Deformation retract) *A subspace Y of a topological space X is a* deformation retract *of it if there exists a continuous map $H : X \times [0,1] \to X$ such that:*

- $\forall x \in X, H(x,0) = x$.
- $\forall x \in X, H(x,1) \in Y$.
- $\forall y \in Y, \forall t \in [0,1], H(y,t) = y$.

In the case of cell complexes, a succession of *cell collapses* (cf figure II.19) is a deformation retract:

Definition II.25 (Collapse) *If $\sigma^{(p)} \prec \upsilon^{(p-1)}$ are two cells of a cell complex K and σ is not the face of any other cell of K, then K* collapses *onto $K \setminus (\sigma \cup \upsilon)$.*

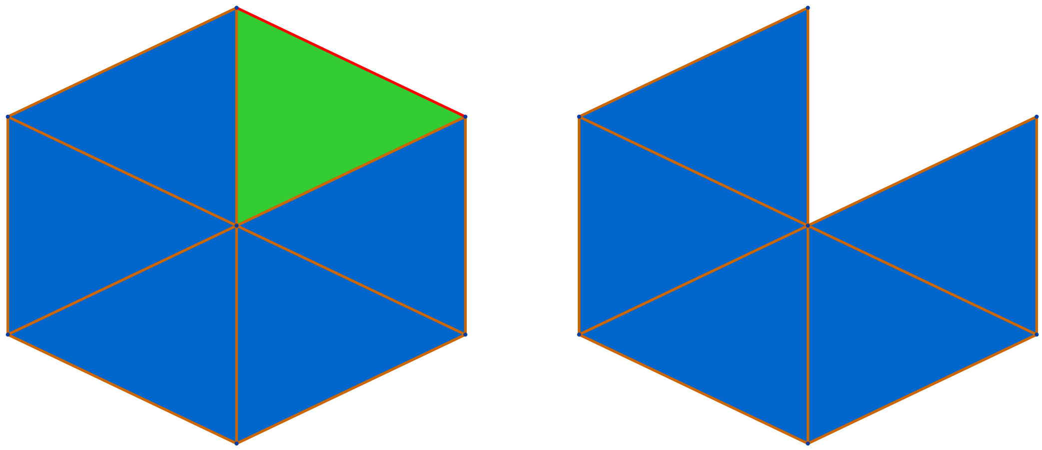


Figure II.19: An elementary collapse.

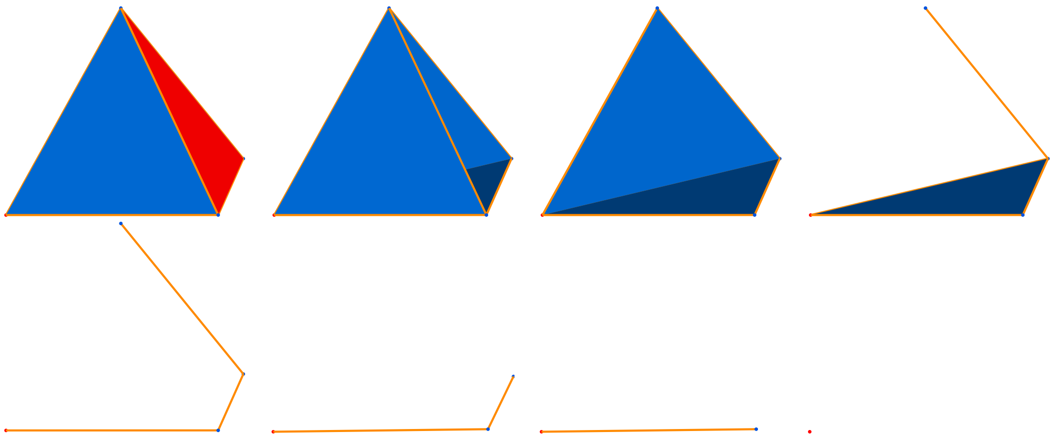


Figure II.20: The collapse of a tetrahedra onto a point.

If L can be obtained from K by successive elementary collapses, we also say that K *collapses* onto L and that K is an *extension* of L. We write $K \searrow L$. For example, figure II.20 shows the collapse of a tetrahedron onto a point. The equivalence relation generated by collapses is called simple homotopy equivalence [Coh73]:

Definition II.26 (Simple homotopy) *A* simple homotopy *is a succession of collapses and extensions. If two complexes can be obtidos one from the other by a simple homotopy, we say they have the same* simple homotopy type*.*

Discrete Morse theory characterizes the simple homotopy of a cell complex from the critical elements of a discrete Morse function defined on it.

III
Forman's discrete Morse theory

III.1 Discrete gradient vector field

Forman's theory relies either on admissible functions on a cell complex, called discrete Morse functions, or equivalently their gradient vector field. We chose here to introduce the theory first from the second point of view.

III.1.1 Combinatorial vector field

Definition III.1 (Combinatorial vector field) *A* combinatorial vector field $\mathcal{V}$ *defined on a cell complex* K *is a collection of disjoint pairs of incident cells* $\{\alpha^{(p)} \prec \beta^{(p+1)}\}$.

We can define the combinatorial vector field as a function $V : K \to K \cup \{0\}$:

$$\{\alpha^{(p)} \prec \beta^{(p+1)}\} \in \mathcal{V} \quad \Rightarrow \quad V(\alpha) = \beta \text{ and } V(\beta) = 0.$$

If a cell σ does not belong to any pair, then $V(\sigma) = 0$.

We will represent a combinatorial vector field by an arrow from the cell of lower dimension to its paired cell of higher dimension, i.e. from α to $V(\alpha)$ (cf figure III.1).

Definition III.2 (V–path) *A* V–path *is an alternating sequence of cells* $\alpha_0^{(p)}, \beta_0^{(p+1)}, \ldots, \alpha_r^{(p)}, \beta_r^{(p+1)}, \alpha_{r+1}^{(p)}$ *satisfying:*

$$V(\alpha_i^{(p)}) = \beta_i^{(p+1)} \quad \text{and} \quad \beta_i^{(p+1)} \succ \alpha_{i+1}^{(p)} \neq \alpha_i^{(p)}.$$

A V–path is non–trivial and closed if $r \geq 1$ and $\alpha_{r+1} = \alpha_0$. For example, figure III.2 shows in red the closed V–path of the combinatorial vector field of figure III.1.

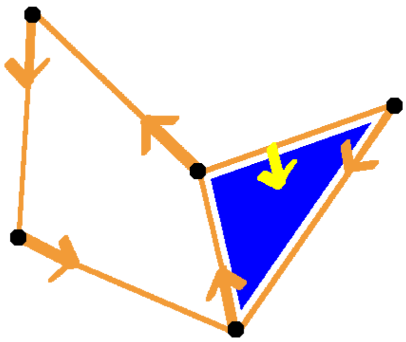


Figure III.1: An example of a combinatorial vector field.

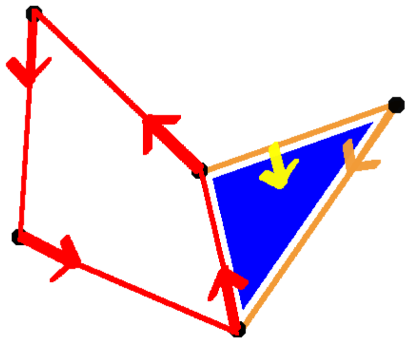


Figure III.2: The closed V–path of the combinatorial vector field of figure III.1 (in red).

III.1.2 Discrete gradient vector field and its critical cells

Definition III.3 (Discrete gradient vector field) *A* discrete gradient vector field *is a combinatorial vector field with no non–trivial closed V–path.*

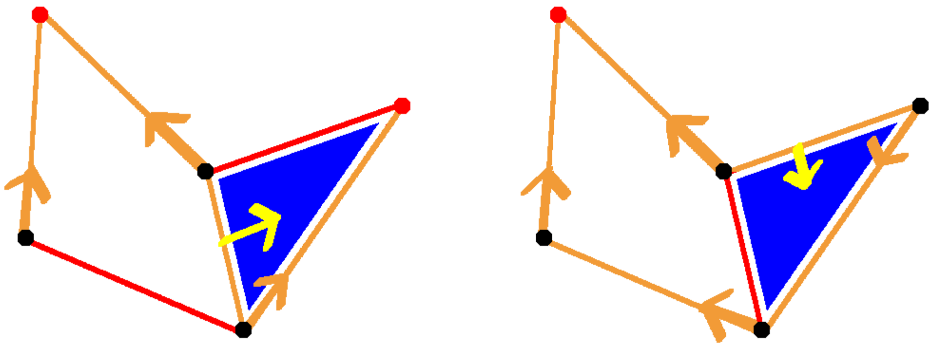
Morse proved that the topology of a manifold is related to the critical elements of a smooth function defined on it. Forman gave an analogous result, with the following definition for the critical cells:

Definition III.4 (Critical cells) *A cell α is* critical *if it is not paired with any other cell, i.e.:*

$$V(\alpha) = 0 \qquad \text{and} \qquad \alpha \notin Im(V)$$

The example of figure III.1 is not a discrete gradient vector field as it contains a closed V–path. On figure III.3, the critical cells of the discrete gradient vector field are drawn in red.

We will denote by $m_p(f)$ the number of critical cells of dimension p. The number of critical cells is not a topological invariant of the cell complex, as it depends on the discrete gradient vector field considered. For example, with an empty discrete vector field (i.e. no cells are paired) every cell is critical, which would be the maximal number of critical cells. In this work, we are more concerned in minimizing this number, as it would give a more concise description of the topology.

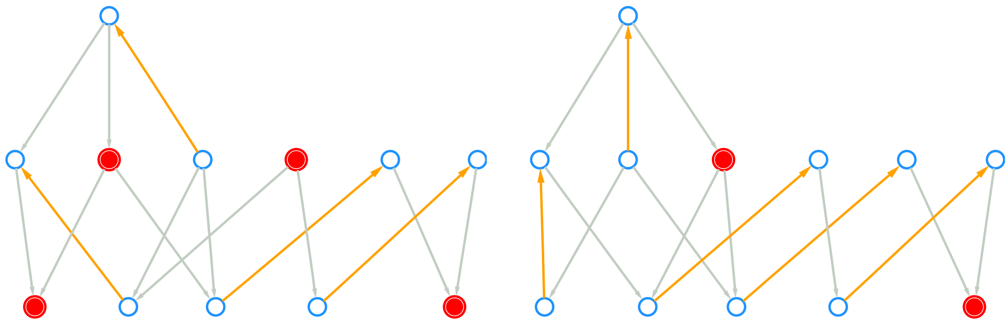


III.3(a): valid III.3(b): optimal

Figure III.3: Examples of discrete gradient vector field.

III.1.3 Hasse diagram of vector fields

A combinatorial vector field is a partial matching in the Hasse diagram: each pair of $\mathcal{V}$ corresponds to matched nodes in the Hasse diagram.



III.4(a): valid III.4(b): optimal

Figure III.4: Hasse diagram of the examples of figure III.3.

We will represent such a matching by inverting the orientation of the link between each pair in $\mathcal{V}$: the arrow's source node will be $\alpha^{(p)}$ for each $\{\alpha^{(p)} \prec \beta^{(p+1)}\} \in \mathcal{V}$. For example, figure III.4 shows the Hasse diagram of the discrete gradient vector fields of figure III.3.

With this modified orientation, a closed V−path is just an oriented circuit in the Hasse diagram (cf figure III.5). A discrete gradient vector field contains no closed V−path, and thus will be an *acyclic matching*.

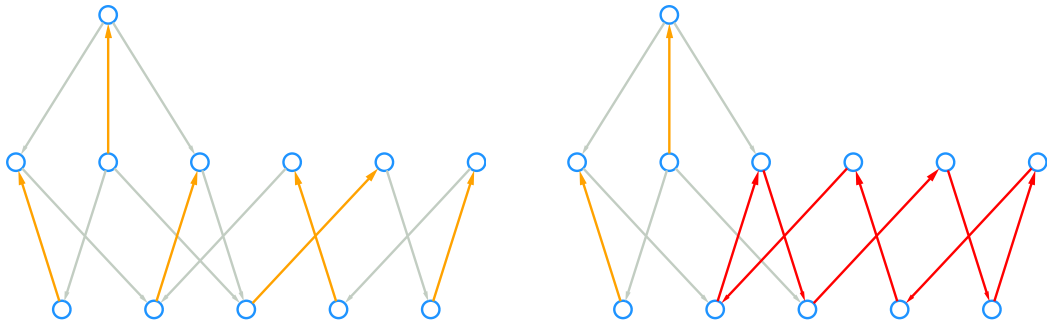
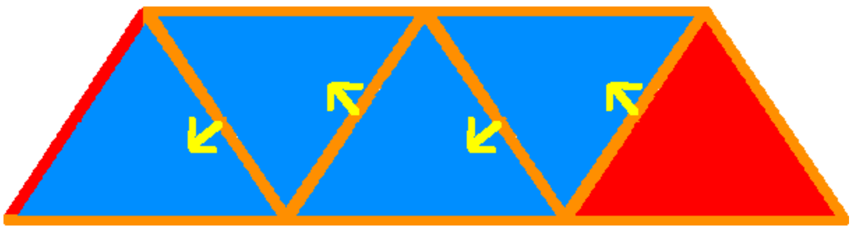


Figure III.5: The Hasse diagram of the combinatorial vector field of figure III.1, and the circuit of its closed V–path (in red).

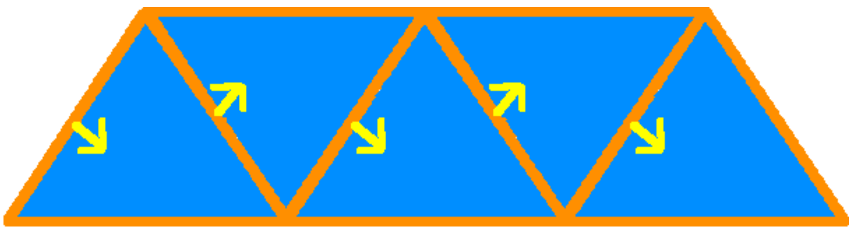
III.1.4 Canceling critical cells

Proposition III.5 *Suppose V is a discrete gradient vector field such that $\alpha^{(p)}$ and $\beta^{(p+1)}$ are critical and there is exactly one V–path from a face of β to α. Then there is another discrete gradient vector field W, equal to V away from this V–path, with the same critical cells except that α and β are no longer critical for W.*

Although the demonstration of the same theorem in the smooth case is rather technical, the demonstration here is very simple:



III.6(a): A unique gradient path joining two critical cells



III.6(b): Revering the gradient path

Figure III.6: Canceling critical cells

Proof. Let $\alpha_0, \beta_0, \ldots, \alpha_r, \beta_r, \alpha_{r+1}$ be the unique V–path of the theorem, with $\beta_0 = \beta$ and $\alpha_{r+1} = \alpha$. W is obtained from V by reversing the gradient vector field along the V–path (cf figure III.6):

$$\begin{aligned} W = V \quad \setminus \quad & \{\{\alpha_1 \prec \beta_1\}, \ldots, \{\alpha_r \prec \beta_r\}\} \cup \\ \cup \quad & \{\{\alpha_1 \prec \beta\}, \{\alpha_2 \prec \beta_1\}, \ldots, \{\alpha_r \prec \beta_{r-1}\}, \{\alpha \prec \beta_r\}\}. \end{aligned}$$

□

III.2 Discrete Morse functions

Actually, the topological information lies in the discrete gradient vector field. However, some topological properties are more easily expressed in terms of discrete Morse functions.

III.2.1 Discrete Morse functions and its critical cells

A discrete Morse function on a given cell complex is a real function almost increasing with the dimension. There can be at most one exception of the increasing per cell, and this exception relates to the pairing of the discrete gradient vector field.

Definition III.6 (Discrete Morse function) *A function $f : K \to \mathbb{R}$ mapping each cell of a cell complex K to a real value is a discrete Morse function if it satisfies, for every cell $\sigma^{(p)} \in K$:*

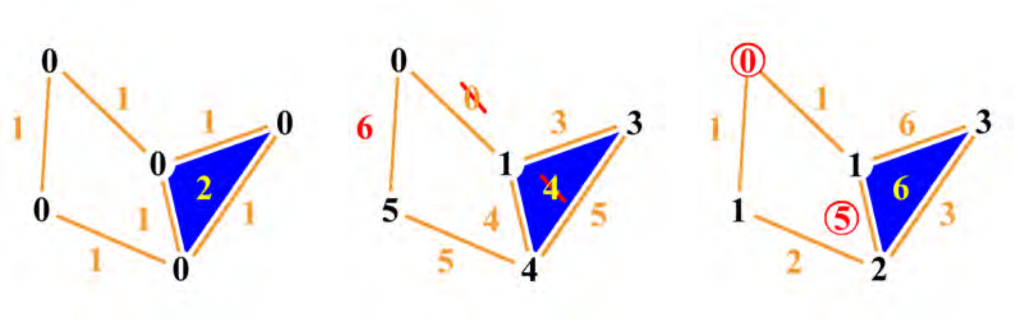
$$\begin{aligned} \# \left\{\tau^{(p+1)} \succ \sigma^{(p)} : f(\tau) \leq f(\sigma)\right\} \leq 1 \\ \textit{and } \# \left\{\upsilon^{(p-1)} \prec \sigma^{(p)} : f(\upsilon) \geq f(\sigma)\right\} \leq 1 \end{aligned}$$

In other words, for each cell σ, f assigns at most one face of σ to a value greater than $f(\sigma)$, and at most one bounding cell of σ to a value less than $f(\sigma)$. There is at most one "counterbalancing" face and one "counterbalancing" bounding cell for every cell. It is easy to show that a cell cannot have both of them. A cell that has none of them will be called *critical*:

Definition III.7 (Critical cell) *A cell $\sigma^{(p)}$ is a* critical cell *of f if:*

$$\begin{aligned} \# \left\{\tau^{(p+1)} \succ \sigma^{(p)} : f(\tau) \leq f(\sigma)\right\} = 0 \\ \textit{and } \# \left\{\upsilon^{(p-1)} \prec \sigma^{(p)} : f(\upsilon) \geq f(\sigma)\right\} = 0 \end{aligned}$$

Figure III.7 gives some example of discrete Morse functions. Of course, not every function is valid: on figure III.7(b) for example, the face (with value



III.7(a): trivial (valid) III.7(b): invalid III.7(c): valid (optimal)

Figure III.7: Examples of discrete Morse functions

4) and the edge with value 0 are assigned values invalid for definition III.6. The critical cells of figure III.7(c) are assigned 0 and 5.

III.2.2 Integrating a discrete gradient vector field

For a given discrete Morse function, every cell has at most one "counterbalancing" face or bounding cell. If there is one, we can define a pair of the discrete gradient vector field with the cell and its "counterbalancing" one. For example, the discrete Morse function of figure III.7(c) corresponds to the discrete gradient vector field of figure III.3(b).

Theorem III.8 *For every discrete Morse function f, there exists a discrete gradient vector field V with the same critical cells as f.*

Proof. We can define V for every cell $\sigma^{(p)}$ by:

$$V\left(\sigma^{(p)}\right) = \begin{cases} \tau^{(p+1)} \text{ such that } \tau \succ \sigma \text{ and } f(\tau) \leq f(\sigma) & \text{if such } \tau \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

□

Trivial discrete Morse function We saw in section III.1.2 that the empty set is a discrete gradient vector field, for which every cell is critical. This corresponds to a trivial discrete Morse function f which assign to every cell its dimension: $f\left(\sigma^{(p)}\right) = p$ (cf figure III.7(a)). This process of deducing a discrete Morse function from a discrete gradient vector field can be generalized for a general discrete gradient vector field.

Theorem III.9 *For every discrete gradient vector field V, there exists a discrete Morse function f with the same critical cells as V.*

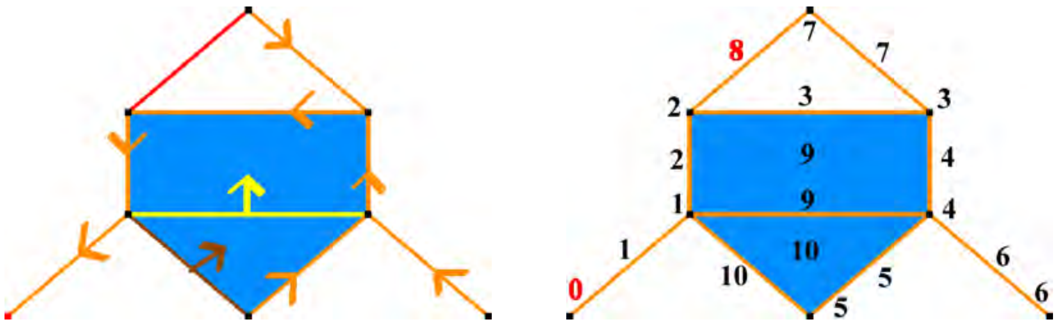


Figure III.8: A discrete gradient vector field and its corresponding discrete Morse function.

The proof of this theorem can be found in [For98, theorem 9.3]. Another way of constructing f out of V can be found in the algorithm of section VI.2.4. Figure III.8 shows a result of this algorithm.

III.2.3 Operations on discrete Morse elements

Let f and g be discrete Morse functions defined respectively on two cell complexes K and L, and V and W their corresponding discrete gradient vector fields.

Restriction. If L is a subcomplex of K, then $g = f_{|L}$ is a valid discrete Morse function.

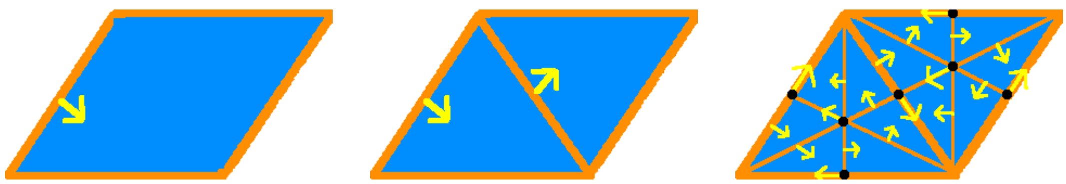
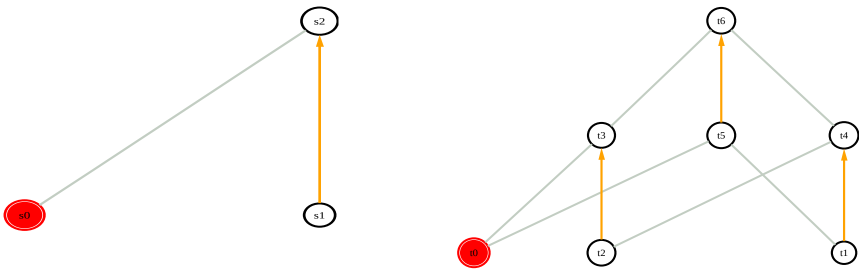


Figure III.9: Refining a discrete gradient vector field: all the new vertices, edges and faces are paired easily.

Refinement. If L is a subdivision of K, one can construct g out of f in order to get the same number of critical cells. This can be done by refining locally the discrete gradient vector field on each subdivided cell, as on figure III.9. A complete demonstration can be found in [For98, section 12].

Cartesian product. The cartesian product $K \times L$ is a cell complex with the following incidence relation: $(\alpha_K, \alpha_L) \prec (\beta_K, \beta_L)$ if either $\alpha_K = \beta_K$ and $\alpha_L \prec \beta_L$, or $\alpha_K \prec \beta_K$ and $\alpha_L = \beta_L$. A discrete gradient vector field $V \times W$ can be defined on $K \times L$ in order to have $\sum_q m_q(f) \cdot m_{p-q}(g)$ critical cells of index p:

$$\begin{cases} if\ \alpha_K\ is\ not\ critical,\ V(\alpha_K) = 0, & (V \times W)(\{\alpha_K, \alpha_L\}) = 0 \\ if\ \alpha_K\ is\ not\ critical,\ V(\alpha_K) = \beta_K, & (V \times W)(\{\alpha_K, \alpha_L\}) = \{\beta_K, \alpha_L\} \\ if\ \alpha_K\ is\ critical,\ W(\alpha_L) = 0, & (V \times W)(\{\alpha_K, \alpha_L\}) = 0 \\ if\ \alpha_K\ is\ critical, W(\alpha_L) = \beta_L, & (V \times W)(\{\alpha_K, \alpha_L\}) = \{\alpha_K, \beta_L\} \end{cases}$$



III.10(a): Segment III.10(b): Triangle

Figure III.10: The Hasse diagram of an optimal discrete gradient vector field on a Segment and on a Triangle.

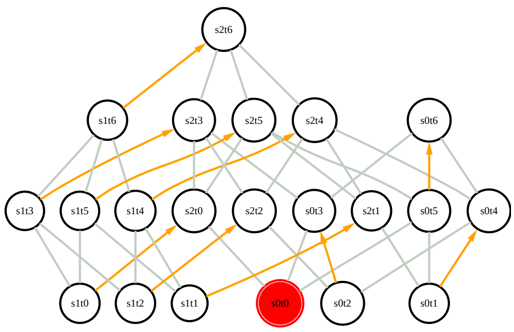


Figure III.11: The Hasse diagram of the cartesian product of the discrete gradient vector fields of figure III.10.

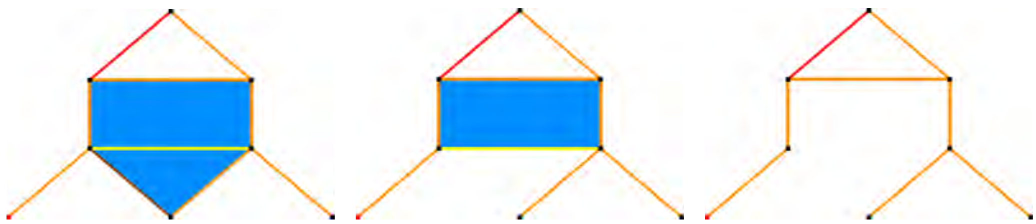
This process is a kind of lexicographic priority. Figures III.10 and III.11 give an example for the simple case of a segment *cartesian* a triangle.

III.3 Topological properties

With the definition of a discrete Morse function, we would like to get the same intuition as with smooth Morse functions, in particular the notion of height. We will thus define the cut $K(c)$ of a cell complex K at a certain height c as follow:

$$K(c) = \bigcup_{\sigma \in K,\ f(\sigma) \leq c} \ \bigcup_{\tau \prec \sigma} \tau.$$

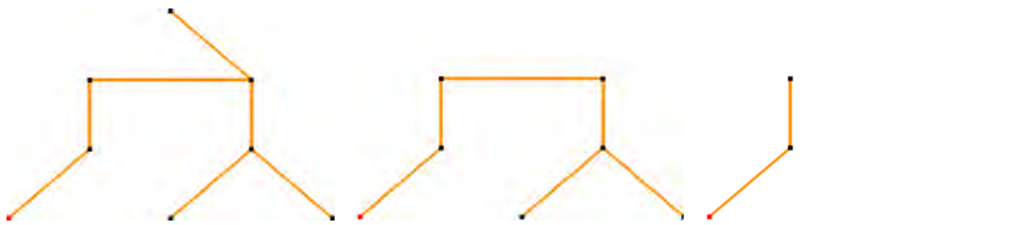
III.3.1 Homotopy properties



III.12(a): $K(10)$: Initial complex

III.12(b): $K(9)$: A collapse triangle / boundary edge

III.12(c): $K(8)$: A collapse triangle / edge



III.12(d): $K(7)$: Removing a critical cell

III.12(e): $K(6)$: A collapse edge / vertex

III.12(f): $K(2)$: Last steps until the critical vertex

Figure III.12: Steps of the collapse of a simple cell complex.

Morse proved that moving the height at which the manifold is cut does not change topology if we do not go across a critical height. The same theorems state for the discrete case:

Theorem III.10 *If $a < b$ are real numbers such that $[a, b]$ contains no critical values of f, then $K(b) \searrow K(a)$.*

Theorem III.11 *If $a < b$ are real numbers such that $f^{-1}([a, b])$ contains a unique critical cell $\sigma^{(p)}$, then $K(b)$ is homotopy equivalent to $K(a) \bigcup_{\partial e^p} e^p$, where e^p denotes a p–cell with boundary ∂e^p.*

The proofs of those theorems can be found in [For98, theorems 3.3 and 3.4]. As a direct corollary of the above theorems, we can enounce:

Corollary III.12 *K is simple–homotopy equivalent to a cell complex with exactly $m_p(f)$ cells of dimension p.*

Those results are of great significance for the field of computation topology: once a discrete Morse function has been defined on a cell complex, one can calculate its homotopy and homology groups from a very reduced number of cells. This would allow in some cases to use exponential algorithm in an admissible time.

III.3.2 Discrete Morse inequalities

Other topological properties originally proved by Morse are grouped in the *Morse inequalities*. They follow directly from the sub–additivity of the Betti numbers and the retraction mentioned above. Those inequalities are valid whatever the field is chosen to calculate the Betti numbers [For98, corollary 8.3]. A proof of those inequalities can be found in [Mil63].

Theorem III.13 (Strong Morse inequalities) *For a given finite cell complex K, any discrete Morse function f defined on it satisfies:*

$$\forall p, \quad m_p\left(f\right) - m_{p-1}\left(f\right) + \cdots \pm m_0\left(f\right) \geq \beta_p\left(K\right) - \beta_{p-1}\left(K\right) + \cdots \pm \beta_0\left(K\right)$$

Theorem III.14 (Weak Morse inequalities) *For a given finite cell complex K of dimension n, any discrete Morse function f defined on it satisfies:*

$$\begin{array}{rcl} \forall p, & & m_p\left(f\right) \geq \beta_p\left(K\right) \\ & & \\ \chi\left(K\right) & = & \#_n\left(K\right) - \#_{n-1}\left(K\right) + \cdots \pm \#_0\left(K\right) \\ & = & m_n\left(f\right) - m_{n-1}\left(f\right) + \cdots \pm m_0\left(f\right) \\ & = & \beta_n\left(K\right) - \beta_{n-1}\left(K\right) + \cdots \pm \beta_0\left(K\right) \end{array}$$

where $\chi\left(K\right)$ is the Euler characteristic and $\#_p\left(K\right)$ the number of p–cells of K.

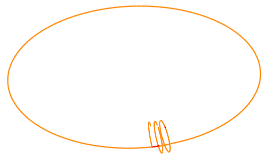
III.3.3 Collapse and discrete Morse functions

Collapsing a complex does not change its (simple) homotopy type. So we should be able to extend a discrete Morse function with a cell complex without adding any critical cell:

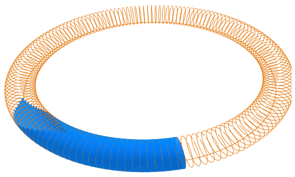
Theorem III.15 *Let L be a subcomplex of K such that $K \searrow L$. Let f be a discrete Morse function on L and let $c = max_{\sigma \in L} f(\sigma)$. Then f can be extended*

III.13(a): $K(0)$: critical vertex

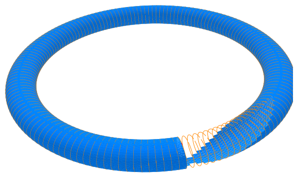
III.13(b): First critical edge



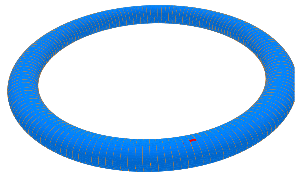
III.13(c): Second critical edge



III.13(d): Extension



III.13(e): More extension



III.13(f): Critical face

Figure III.13: The critical steps of the extension of a torus.

to a discrete Morse function g on K with $L = K(c)$, and such that there are no critical cell in $K \setminus L$.

Proof. By induction on the number of elementary collapses required, it is sufficient to prove it when $K = L \cup \sigma \cup \tau$, where σ is a cell of K and $\tau \prec \sigma$ one of its free faces. We can define g on K by $\forall\ \alpha \in L,\ g(\alpha) = f(\alpha)$ and $g(\sigma) = c + 1,\ g(\tau) = c + 1$. $\square$

Figure III.13 shows different steps of the extensions of a Morse function as described in proposition III.15, except for the critical values (in red).

III.3.4 Sphere theorems

We just mention here another very interesting theorem of Morse theories. A proof of those discrete versions can be found in [For98, section 5].

Theorem III.16 *If K is a cell complex with a discrete Morse function f with exactly two critical cells, then K is homotopy equivalent to a sphere.*

Proposition III.17 *If S is a PL n–sphere, then, by performing a finite sequence of bisections, S can be subdivided to a polyhedron K which has a Morse function with exactly 2 critical cells.*

III.4 Optimality in discrete Morse functions

As we mentioned above, the critical cells of a discrete Morse function defined on a cell complex describe its simple homotopy type. Having a small number of critical cells would accelerate the computation of topological properties.

Definition III.18 (Optimal discrete Morse function) *We will say a discrete Morse function is* optimal *when it has the minimum possible number of critical cells.*

Unfortunately, the general problem of finding an optimal discrete Morse function is *MAX–SNP hard*, i.e. an NP–hard problem for which any polynomial approximation algorithm can lead to a result arbitrary far from the optimum.

III.4.1 Existence of an optimum

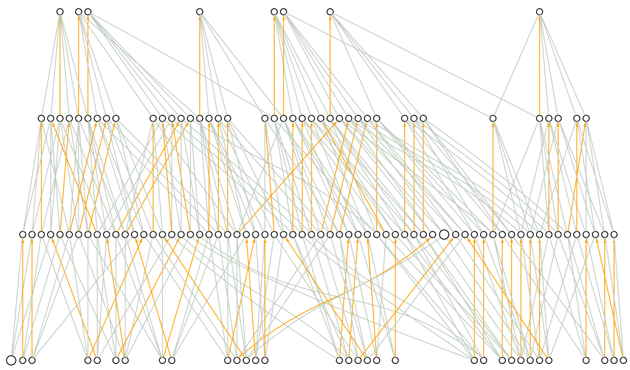


Figure III.14: The number of matchings in the Hasse diagram is finite.

Although there is an infinity of discrete Morse function for a given cell complex K, there is only a finite number of discrete gradient vector field (cf figure III.14). In fact, a discrete gradient vector field can be seen as a matching in the Hasse diagram (cf section III.1.3). There is less than $2^{(\#K)^2}$ such matchings. Thus, there exists a minimum to the possible numbers of critical cells of K.

III.4.2 A related problem

We know from theorems III.10 and III.11 that the critical cells have a tight relation with collapsibility. Eğecioğlu and Gonzalez studied the Erasability and the Collapsibility problem for simplicial[1] complexes of dimension 2 in [Ege95]:

Collapsibility problem	
Instance:	A pair (K, n), where K is a finite simplicial complex of dimension 2 and n is a non–negative integer
Question:	Does K contain a subset F of 2–simplicies of cardinality at most n such that $K \setminus F$ collapses to a point?

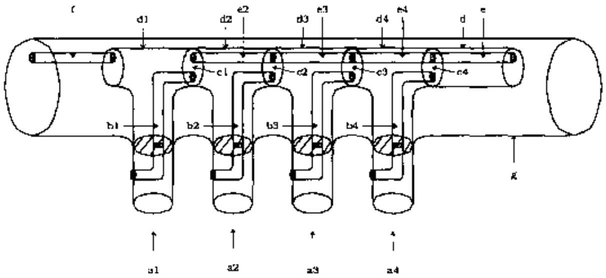


Figure III.15: A "gadget" used in Eğecioğlu and Gonzalez proof.

In particular, they proved that this problem is MAX–SNP hard, reducing the problem to the Vertex Cover problem by 2–cell complexes as the one of figure III.15.

III.4.3 Complexity of the optimum

We will cf that the problem of determining the minimum possible number of critical cells of any discrete Morse function defined on a cell complex is also MAX–SNP hard, by reduction to the Collapsibility problem.

Morse optimality problem	
Instance:	A pair (K, n), where K is a finite cell complex of dimension at least 2 and n is a non–negative integer
Question:	Does there exist a discrete Morse function on K with at most n critical cells?

Let's consider a simplicial complex K of dimension at most 2. Suppose we find a discrete Morse function on K with the minimum number of critical cells. The number of critical vertices is the number of connected components.

[1]a simplex is a particular type of cell

Thus, the discrete Morse function has the minimum number of critical faces (cf hteorem III.14). Then, by theorem III.10, the subset F of the critical faces (i.e. cells of dimension 2) would answer the question of the Collapsibility problem. We deduce the following theorem:

Theorem III.19 *The problem of finding optimal discrete Morse functions is a reduction of the Collapsibility problem, and is thus a MAX–SNP hard problem.*

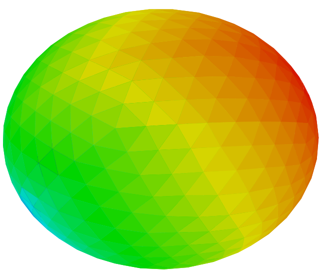
IV
Optimal discrete Morse functions on surfaces

IV.1 Optimality conditions

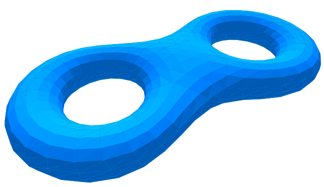
We saw in section III.4.3 that the general problem of finding an optimal discrete Morse function is NP–hard. However, for the case of 2–manifolds, this problem can be solved in linear time by the algorithm presented in section IV.2. The proof of the optimality relies on the classification theorem for surfaces introduced in the next section. A proof of those theorems can be found [Arm79].

IV.1.1 The surface classification theorem

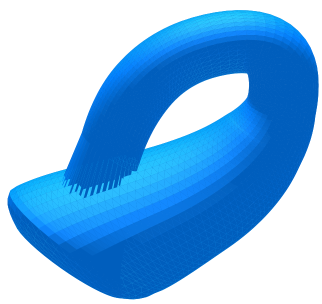
Theorem IV.1 (Classification theorem for surfaces) *Any compact connected surface without boundary is homeomorphic to exactly one of the following surfaces: a sphere S^2, a connected sum of $g > 0$ tori $T(g)$, or connected sum of $g > 0$ projective planes $M(g)$. No two of these surfaces are homeomorphic.*



IV.1(a): 2–sphere.



IV.1(b): Sum of 2 tori.



IV.1(c): Klein bottle: connected sum of 2 projective planes.

Figure IV.1: Examples of surfaces without boundary.

The sphere (figure IV.1(a)) and connected sums of tori (figure IV.1(b)) are orientable surfaces, whereas surfaces with Möbius strips are not orientable (figure IV.1(c)). g is called the genus of the surface. Actually, the Betti numbers (with coefficient in $\mathbb{Z}_2$) and the orientability completely characterize the topology of a surface, as we can explicitly calculate the homology groups of the standard surfaces [Arm79]:

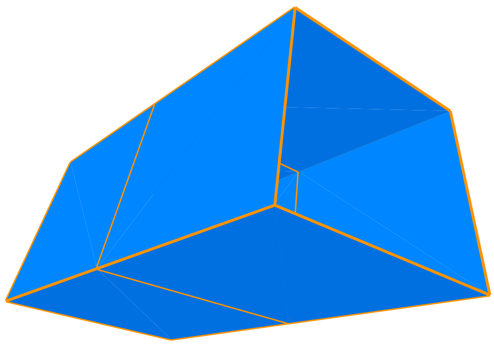
Proposition IV.2 (Homology groups of the standard surfaces)

$$
\begin{array}{rclcrclcrcl}
H_0\left(S^2\right) & = & \mathbb{Z}_2 & , & H_1\left(S^2\right) & = & 0 & , & H_2\left(S^2\right) & = & \mathbb{Z}_2 \quad . \\
H_0\left(T(g)\right) & = & \mathbb{Z}_2 & , & H_1\left(T(g)\right) & = & 2g\cdot\mathbb{Z}_2 & , & H_2\left(T(g)\right) & = & \mathbb{Z}_2 \quad . \\
H_0\left(M(g)\right) & = & \mathbb{Z}_2 & , & H_1\left(M(g)\right) & = & g\cdot\mathbb{Z}_2 & , & H_2\left(M(g)\right) & = & \mathbb{Z}_2 \quad .
\end{array}
$$

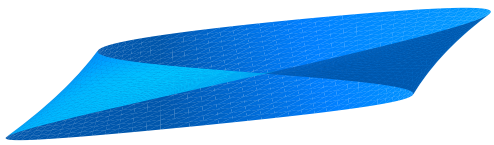
IV.1.2 Surfaces with boundary

Proposition IV.3 *Any compact connected surface with a non–empty boundary is homeomorphic to exactly one of the following surfaces: a sphere, a finite connected sum of tori, or finite connected sum of projective planes, in each case with some finite number of open disks removed.*

A proof of this extension of the theorem for closed surfaces can also be found in [Arm79]. The homology group H_0 remains unchanged, and H_2 vanishes for surfaces with boundary.



IV.2(a): Cylinder.



IV.2(b): Möbius strip.

Figure IV.2: Examples of surfaces with a non–empty boundary.

For example, identifying the boundaries of two disks with the boundary of a cylinder (figure IV.2(a)) creates a space homeomorphic to the sphere. Identifying the boundary of a disk with the boundary of a Möbius strip (figure IV.2(b)) creates a space homeomorphic to the projective plane.

IV.1.3 Sufficient conditions for reaching optimality

We have now all the elements to enounce sufficient conditions to guarantee that a discrete Morse function is optimal (definition III.18):

Proposition IV.4 (Surfaces without boundary) *Let f be a discrete Morse function defined on a compact connected surface without boundary. If f has exactly one critical vertex and one critical face, and possibly many critical edges, then it is optimal.*

Proposition IV.5 (Surfaces with boundary) *Let f denote a discrete Morse function defined on a compact connected surface with a non–empty boundary. If f has one critical vertex, no critical face, and possibly many critical edges, then it is optimal.*

Proof. In the two cases of the theorem, we suppose that $m_0 (f) = \beta_0 (K) = 1$ and $m_2 (f) = \beta_2 (K)$. From the second weak Morse inequality (theorem III.14), we know that $\chi (K) = m_2 (f) - m_1 (f) + m_0 (f) = \beta_2 (K) - \beta_1 (K) + \beta_0 (K)$. So we deduce that $m_1 (f) = \beta_1 (K)$. From the first weak Morse inequality (theorem III.14), which states that $m_p (f) \geq \beta_p (K)$, we reached the lower bound to the number of critical cells. Therefore, f is optimal. □

IV.2 Algorithm

Given a finite cell complex K that has the topology of a 2–manifold, the algorithm proceeds on each connected component in 4 steps:

1. Construct a spanning tree T on the dual pseudograph of K.
2. If K has a boundary, add one loop to T.
3. Define a discrete Morse function on T.
4. Define a discrete Morse function on the complement of T.

First step: Construction of a face–spanning tree. The face–spanning tree T can be constructed out of the dual pseudograph (cf section II.2.4) by any of the standard algorithms [Hop73]. In particular, we can use some mesh compression's strategies. For example, the EdgeBreaker's compression algorithm [Lop02] (figure IV.3) constructs a spiraling face–spanning tree (figure IV.4).

Second step: Addition of one loop. We test whether the manifold has a boundary during the first step. If we found a loop, we add it to T, so T

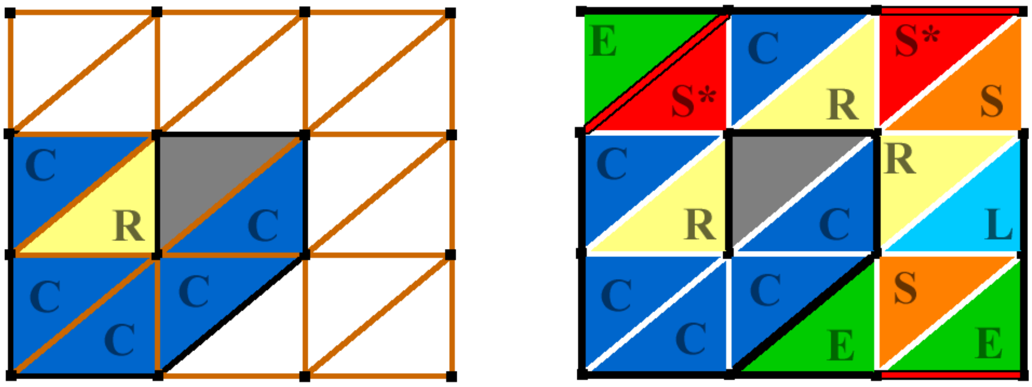
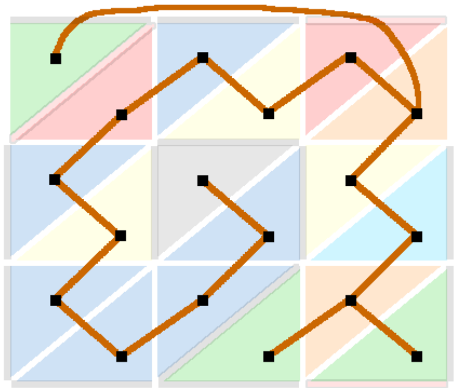
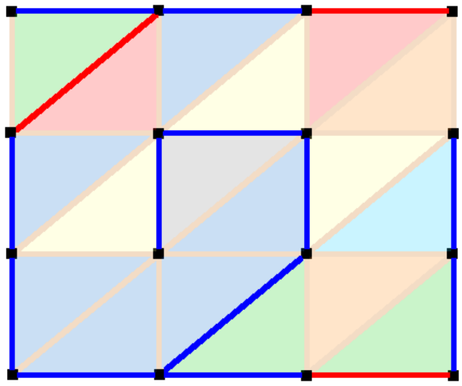


Figure IV.3: EdgeBreaker mesh compression codes on a triangulated torus.



IV.4(a): T



IV.4(b): G

Figure IV.4: The resulting face–spanning tree T and its complement graph G.

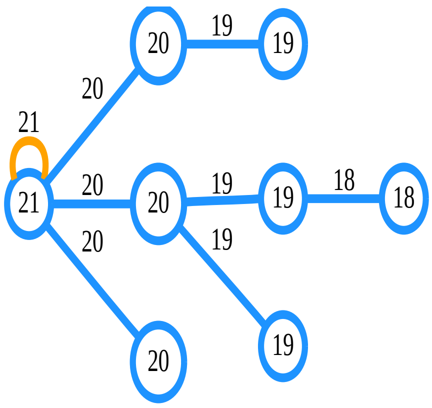


Figure IV.5: Loop added at step 2 to a face–spanning tree T of figure II.1 (no critical cell).

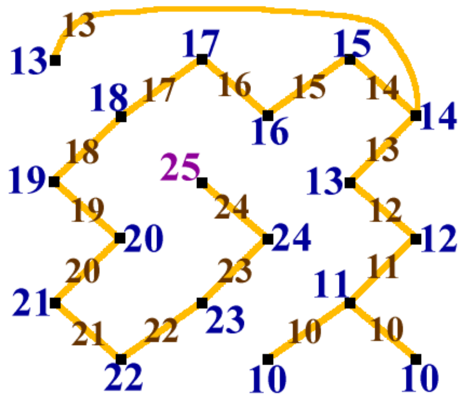


Figure IV.6: The discrete Morse function on the spanning tree T of figure IV.3 (1 critical face).

becomes a pseudograph. For example on figure II.7, the loop (with value 21) has been added to T.

Third step: Definition of the function on T. We select a root of T, and we assign to every node of T (i.e. 2–cells of K) its height in the tree plus a constant c. We assign to every link of T (i.e. 1–cells of K) the minimum value of its two end nodes (cf figure II.7). The result of this process on the example of figure IV.3 is shown on figure IV.6.

In our construction, the initial value of c must be at least $\#_0 K + 1$.

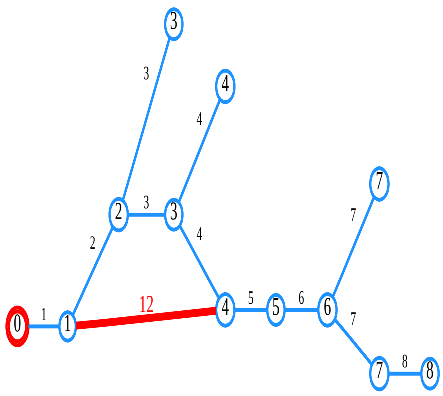


Figure IV.7: The complement graph G of the cylinder model and its discrete Morse function (1 critical vertex and 1 critical edge).

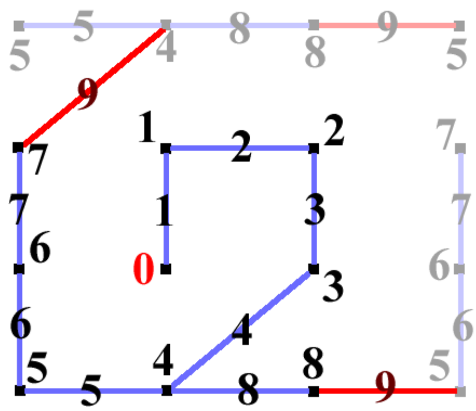


Figure IV.8: The discrete Morse function on the complement graph G of figure IV.3 (1 critical vertex and 2 critical edges).

Fourth step: Definition of the function on the complement of T. We will now consider G, the complement of T: G is a graph with no loop whose nodes are the vertices of K, and whose links are the edges of K that are not represented in T.

We build another spanning tree U on G. We assign to every node of G its edge distance to a selected root of U; and to every link of U the maximum value of its two end nodes.

We finally assign the value (c-1) to each link of $G \setminus U$ (cf the critical edge of figure II.3 with value 12). The result of this process on the example of figure IV.3 is shown on figure IV.8.

IV.3 First extension to non–manifolds

If our complex does not have the topology of a 2–manifold, some links can be incident to 3 faces and the above demonstration does not work anymore. However, the algorithm still produces a valid discrete Morse function, which is still optimal in several cases.

In fact, a cell complex of dimension 2 is not a manifold if it combines some of the following 3 reasons:

1. Dangling edge (cf figure IV.9).
2. Singular vertex (cf figure IV.10).
3. Non–regular edge (cf figure IV.11).

The neighborhood of a point of a face is always $\mathbb{R}^2$.

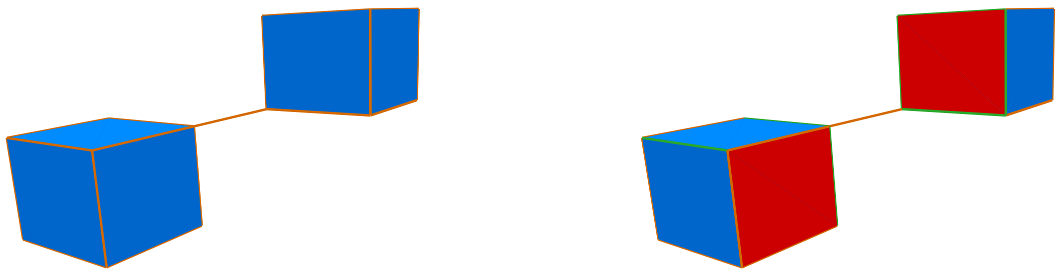


Figure IV.9: A non–manifold complex with a dangling edge (on the left) and its critical cells (on the right): 1 critical vertex and 2 critical faces.

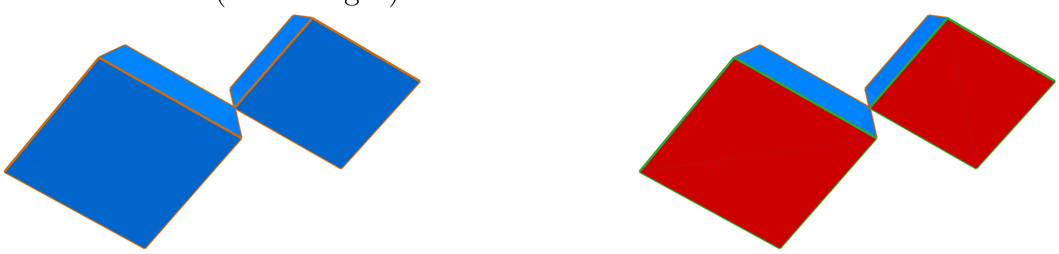


Figure IV.10: A non–manifold complex with a corner vertex (on the left) and its critical cells (on the right): 1 critical vertex and 2 critical faces.

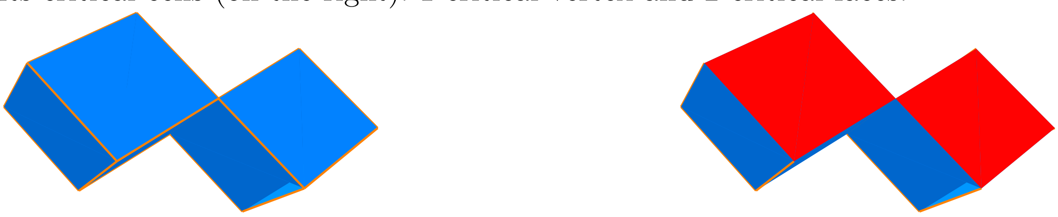


Figure IV.11: A non–manifold complex with a non–regular edge (on the left) and its critical cells (on the right): 1 critical vertex and 2 critical faces.

1. dangling edge. That case resumes to a graph glued to a complex. The graph will not appear in steps 1, 2 and 3 of the algorithm, and will be processed in step 4. The homology of a graph is its connectivity, therefore the algorithm still reaches the optimality in that case.

2. singular vertex. This corresponds to several cell complexes glued at a vertex or a pinched manifold. At step 1 of the algorithm, there would not be one spanning tree T, but several. The optimality of the function resumes to the optimality on each cell complex. Step 4 will generate a connected graph, and there would be only one critical vertex. So, this case doesn't affect the optimality of our result.

3. non–regular edge. That case is the most difficult one. We will only give a heuristic that always build valid discrete Morse function, but we know from the theory (cf section III.4.3) that for some special cases the resulting functions can be arbitrary far from the optimum. Some other heuristics are discussed further on in section VI.4. In that case, we will define the dual pseudograph with the same nodes, and with the links that are incident to 1 or 2 nodes only (i.e. only the regular edges). The algorithm then runs normally, and the non–regular links that cannot be part of the spanning tree of step 4 will remain critical.

IV.4 Proof and analysis

Valid discrete Morse function. From the construction on the trees T (step 3) and U (step 4), the function f, resulting of the algorithm run on one connected component K, respects the inequalities of definition III.6. Moreover, there is exactly one critical vertex: the root of U. If K has no boundary, the root of T will be the unique critical face. In the other case, there is no critical face in K.

We now just need to check that the edges of K are assigned valid Morse values. From the initial value of the constant c, the critical edges are the links of $G \setminus U$, which are assigned a value greater than the value of any vertex, and inferior to the value of any face (there is $\#_0 K < c$ nodes in G, so at most $\#_0 K$ different regular values). The inequalities of definition III.6 are obvious for each cell represented in the trees T and U. From the choice of the initial value of the constant c, every cell of T has a value greater than any cell of G. Thus, those inequalities are strictly respected between the edges of T and the vertices of G, and between the edges of U and the faces in T.

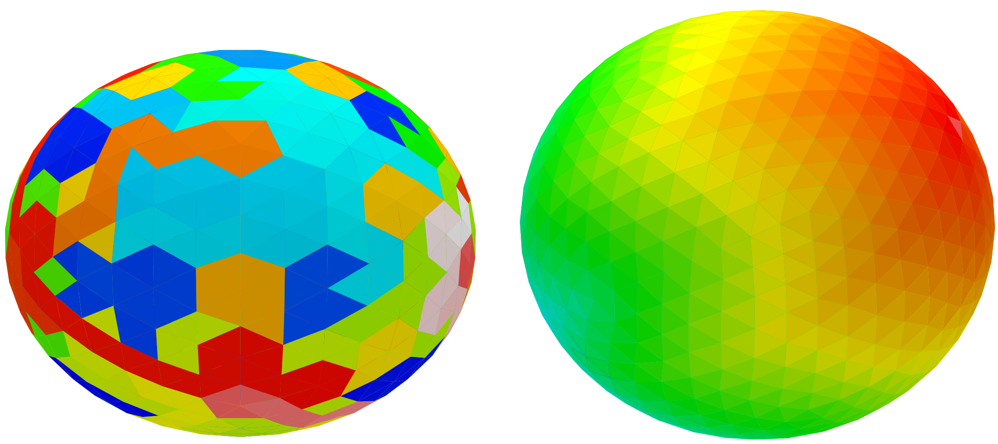
Optimal discrete Morse function. Thus, our construction yields a valid discrete Morse function f with exactly 1 critical vertex ($m_0(f) = 1$), possibly many critical edges, and 1 critical face ($m_2(f) = 1$) if K is a manifold without boundary, and no critical face ($m_2(f) = 0$) if K is a manifold with boundary. From propositions IV.4 and IV.5, we know that f is optimal.

Complexity. Once the spanning trees are built, the algorithm visits each node and link at most once. Therefore, the steps 2, 3 and the second part of step 4 are of linear complexity. Building a spanning tree can be linear with a simple greedy algorithm [Hop73]. So, the whole algorithm is linear in time with respect to $\#K$.

IV.5 Mixing with geometry

The above construction is completely independent of the geometry. One of the powers of Forman's discrete Morse theory reflects here on two points. First, the whole algorithm is done without any floating–point operation. Second, it is possible to add some external constraints, for example geometrical ones. There are different constraints we can add on our discrete Morse function:

- The face–spanning tree T can be chosen to be a minimal spanning tree. This leads to a complexity in $O(\#K \cdot \log \#K)$.
- The loop added at step 2 can minimize the same function, in order to have the root of the face–spanning tree at a minimal position.
- The roots of the face–spanning trees T of step 1 and U of step 4, can also be at a minimal position.
- The value assigned at steps 3 and 4 can be augmented for some branches of the tree to separate a part when reconstructing the complex by successive heights, as in theorems III.10 and III.11.



IV.12(a): Without geometrical constraint IV.12(b): With geometrical constraint

Figure IV.12: Two discrete gradient vector fields on a 2–sphere, both with 2 critical cells.

The way we include geometric constraint does not change the optimality of the resulting function. For example, figure IV.12 shows two discrete gradient vector field on a 2–sphere, both with 2 critical cells. This can be used to imagine or justify geometrical algorithms.

IV.6 Results

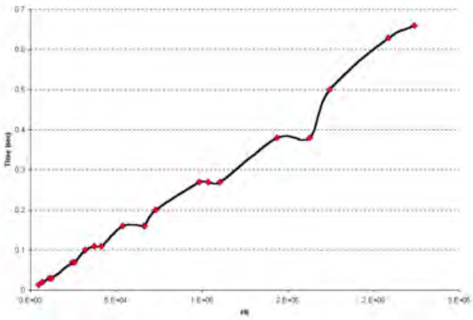


Figure IV.13: Execution time vs. the number of cells of the cell complex K: the complexity is linear.

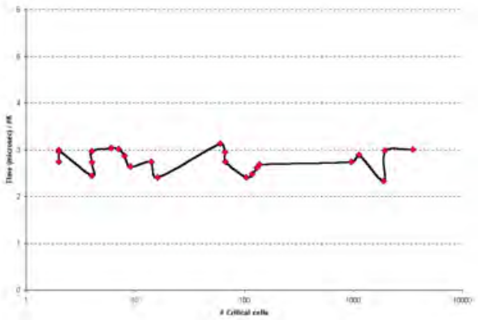


Figure IV.14: Ratio execution time / size of K vs. the number of critical cells of K: independence.

We tested our algorithm on more than 150 models from various types: triangulations, quadrangulations and general polygons; manifolds and non-manifolds; models with a consistent topology and raw scans or VRML importation with deficient topology (figures IV.15, IV.16 and IV.17). The algorithm always built a valid discrete Morse function. For all the manifolds cases, the resulting function was optimal. For the non-manifolds complexes (in particular for the examples of Moriyama and Takeuchi [Mor00]), the function had at most 4 redundant critical cells. The experimental results on a Pentium III, 550 MHz, confirm the linear complexity with respect to $\#K$ (figure IV.13) and the independence of the execution time towards the topology (figure IV.14).

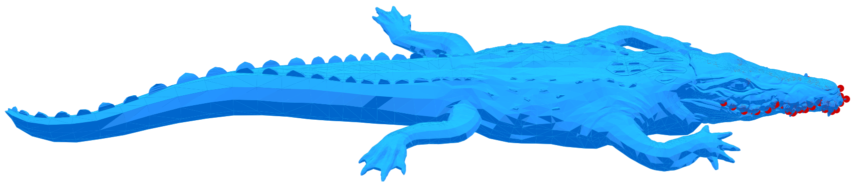


Figure IV.15: An optimal discrete Morse function on a VRML–imported model. Each tooth is a different connected component (here a topological sphere), and has two critical cells. So there are 65 critical vertices and 65 critical faces (64 teeth + 1 for the body).

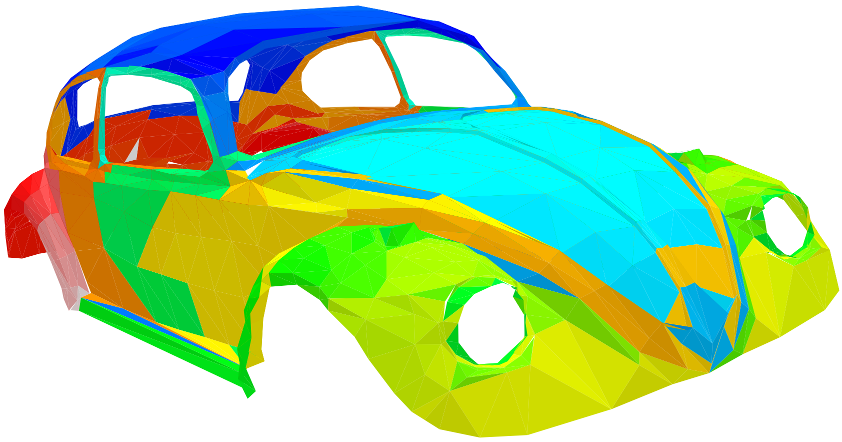


Figure IV.16: An optimal discrete gradient vector field on a beetle model with 2 connected components.

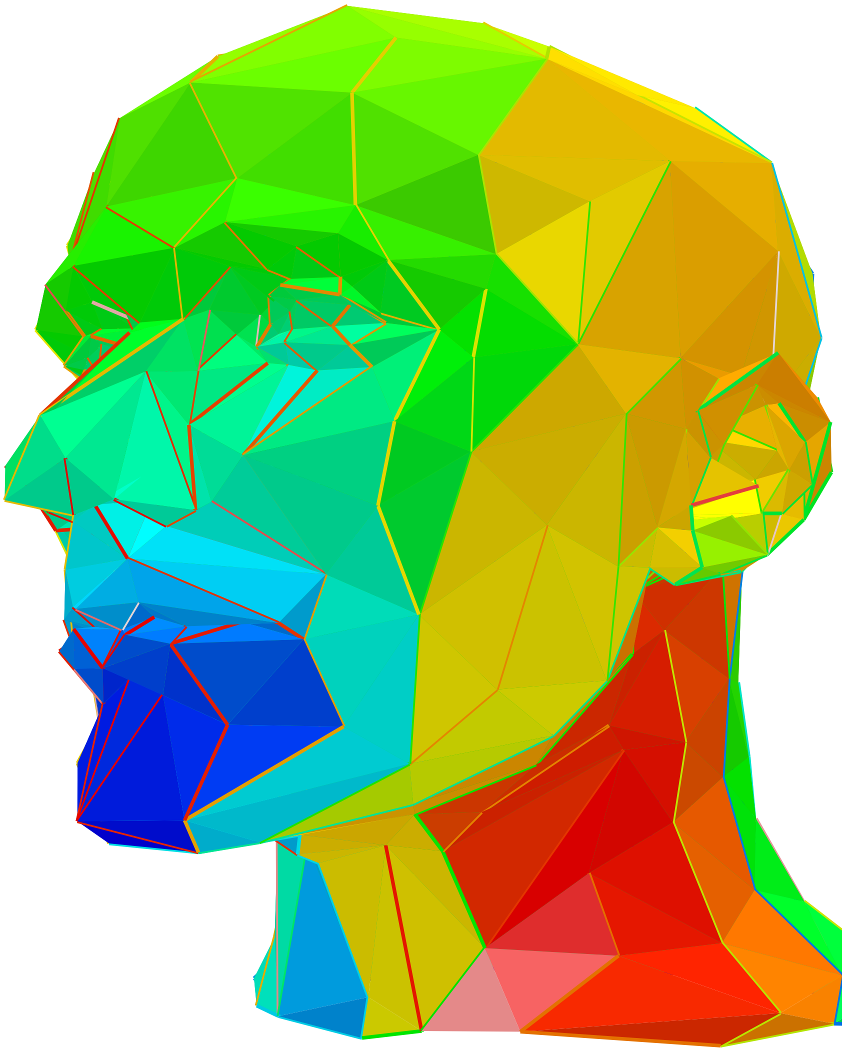


Figure IV.17: An optimal discrete gradient vector field with the complement graph U.

V
Structure of a discrete Morse function

V.1 Layers of a combinatorial vector field

The graph structure is usually used to represent complex objects and their relations. We already did so in section II.2.3 where we represented a cell complex by its Hasse diagram, and in section II.1.6 where we represented a hypergraph by bipartite graphs. This allowed us to give a simple definition of a discrete gradient vector field as an acyclic matching on this diagram (cf section III.1.3).

However, the Hasse diagram is still a complex object to visualize, and we will cf in this section how to represent it in a more simple way. Seeing the Hasse diagram as a collection of bipartite graphs, we will be able to visualize a cell complex as a collection on hypergraphs and a discrete Morse function as hyperforests on those hypergraphs.

V.1.1 Layers of the Hasse diagram

We saw in section II.2.4 that, in an n–combinatorial manifold, a (n-1)–cell is incident to either 1 or 2 n–cells. So the dual layer $n/(n\text{-}1)$ of the Hasse diagram will be represented by a pseudograph, called the dual pseudograph. This pseudograph can be seen as the hypergraph representation of the *dual layer* $n/(n\text{-}1)$ of the Hasse diagram.

Definition V.1 (Layer of the Hasse diagram) *The* layer p/q *of the Hasse diagram,* $|p-q| = 1$*, of a cell complex* K *is an oriented simple bipartite graph. Its* $\boldsymbol{N}$ *and* $\boldsymbol{L}$ *class of nodes are the* p*– and* q*–cells of* K *respectively. Its links joins nodes representing incident* p*– and* q*–cells of* K*.*

This definition puts a difference between layers of type p/q and q/p by differentiating the **N** and **L** classes of nodes. The orientation of those layers is the same as the one of the original Hasse diagram. For example, figure V.2 shows the hypergraph of the layer 2/1 of the Hasse diagram of figure V.1.

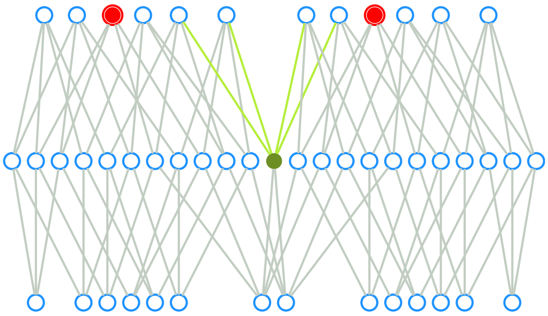


Figure V.1: The Hasse diagram of the double cube of figure IV.11.

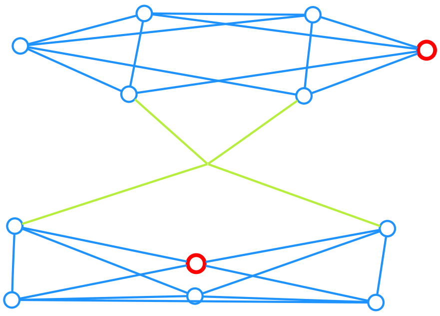


Figure V.2: The hypergraph of the layer 2/1 of figure V.1.

V.1.2 Interpretation of the algorithm for surfaces

One of the most efficient compression strategies relies on topological surgery [Tau98]: cutting the surfaces along a dual spanning–forest and encoding the remaining disc. We used a similar method for constructing optimal discrete Morse function on 2–manifolds (cf chapter IV).

At steps 1 and 2 of the algorithm of section IV.2 we built a discrete gradient vector field as a face–spanning tree extracted from the dual pseudograph. Therefore, a discrete gradient vector field restricted to the top layer of the Hasse diagram of a manifold can be represented by a tree. At step 4, we built a vertex–spanning tree of K, and defined our discrete gradient vector field on it. Step 3 and last part of step 4 are the integration of this vector field, in the sense of theorem III.9.

Thus, the construction of optimal discrete Morse function on surfaces can be seen as processing the top dual layer of the Hasse diagram, and then its first layer. The process on each layer consists of removing the circuits, i.e. constructing a spanning–tree. This is easy for simple graphs and pseudographs. Unfortunately, for the general case of hypergraphs this is a much harder problem. This is why the optimality problem is NP–hard (cf theorem III.19).

V.1.3 Reduced layer of a combinatorial vector field

Considering successive layers of the Hasse diagram is redundant: each p-cell of K appears in 4 layers ($p/(p$+1), $p/(p$-1), $(p$+1$)/p$, $(p$-1$)/p$). When the Hasse diagram is oriented by a discrete gradient vector field (cf section III.1.3), the matching splits on the different layers, with no redundancy. The following reduction allows such partition:

Definition V.2 (Reduced layers of a combinatorial vector field) *Let K be a cell complex, $\mathcal{V}$ a combinatorial vector field defined on it and B the* *layer p/q of the Hasse diagram oriented by $\mathcal{V}$, with $|p-q|=1$. The* reduced layer *B' is an oriented bipartite graph defined as follow:*

> *The **N** class of nodes of B' is the p–cells of K unpaired or paired with a q–cell of K in $\mathcal{V}$*
>
> *The **L** class of nodes of B' is the q–cells of K paired with a p–cells in $\mathcal{V}$*
>
> *The orientation of B' is the same as the one of the original Hasse diagram.*

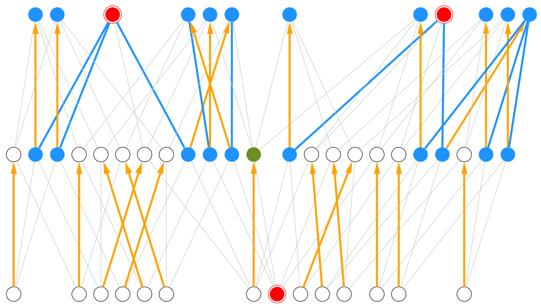


Figure V.3: The reduced layer 2/1 of the double cube (blue nodes).

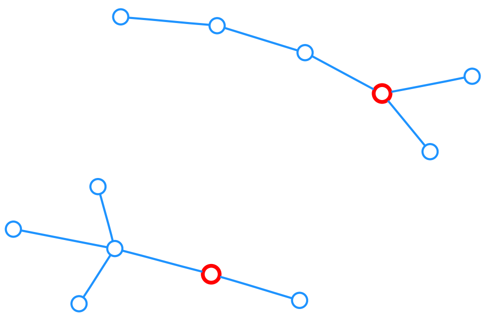


Figure V.4: The hypergraph of the reduced layer 2/1 of figure V.3.

For example, figure V.3 shows in blue the edges in the Hasse diagram of figure V.1 that belongs to the reduced layer 2/1. The corresponding hypergraph (cf section V.2.2) is a forest (cf figure V.4). We can deduce from the definition the following proposition:

Proposition V.3 *With the notations of the definition*

*(i) If $\mathcal{V}$ is a discrete gradient vector field, the critical p–cells of $\mathcal{V}$ appears in reduced layers in the **N** class.*

*(ii) A p–cell of K is an **N** node of exactly one reduced layer.*

(iii) Any $\mathcal{V}$–path is entirely represented in two reduced layer.

Proof. **(i)**. The critical cells of a discrete gradient vector field are unpaired. Therefore, item **(i)** follows directly from definition V.2.

(ii). If a p–cell is unpaired in $\mathcal{V}$, it appears in the **N** class of the reduced layers $p/(p+1)$ and $p/(p-1)$. If it is paired with a q–cell in $\mathcal{V}$, it appears in the **N** class of the reduced layer p/q and in the **L** class of the reduced layer q/p.

(iii). A $\mathcal{V}$–path is an alternate sequence of paired p– and $(p+1)$–cells. From definition V.2, all the paired p– and $(p+1)$–cells appear twice: once in the layer $p/(p+1)$, and once in the layer $(p+1)/p$. □

V.2 Discrete Morse functions, acyclic matchings and hyperforests

We saw in section III.2.2 that the notions of discrete Morse function and discrete gradient vector field are equivalent. A discrete gradient vector field has been defined as an acyclic matching in the Hasse diagram (cf section III.1.3). This involves two problems: creating a matching, and removing cycles. Those two problems are separately well understood (cf [Lov86] for matching theory, and [Hop73] for graph algorithms). However, when combined, they create NP hard problems (cf section III.4.3). In this section, we will give another point of view on discrete Morse theory in terms of the simplest (linear instead of quadratic) of those two problems: creating forests. We will prove our combined problem can be seen as a *hyperforest* creation problem.

V.2.1 Hyperforests

We defined a forest as a graph with no circuit in section II.1.1. Here is a natural extension of forests for hypergraphs [Ber70]:

Definition V.4 (Oriented hypercircuit) *An* oriented hypercircuit *in a hypergraph is a sequence of distinct nodes $n_0, n_1, \ldots, n_{r+1}$ such that $n_{r+1} = n_0$ and for all $0 \leq i \leq r$, n_i is the source of a hyperlink leading to n_{i+1}.*

Definition V.5 (Hyperforest) *We will say that a simply oriented hypergraph is a* hyperforest *if each node is the source of at most one hyperlink, and if it does not contain any hypercircuit.*

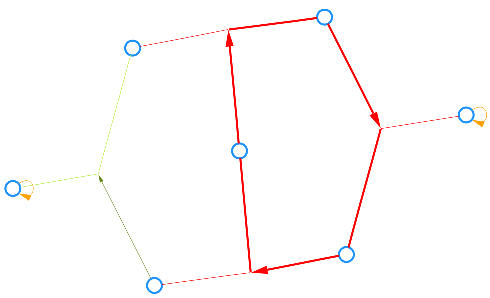


Figure V.5: A hypercircuit (in red).

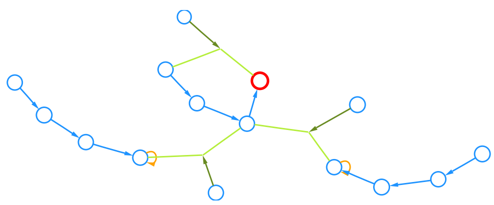


Figure V.6: A part of the hyperforest 2/1 resulting while processing a solid 2-sphere.

For example, figure V.5 shows a hypercircuit, and figure V.6 shows a part of a hyperforest. We can deduce from this definition the following properties:

Proposition V.6 *Let HF be a hyperforest, and R one of its regular component.*

(i) The regular components of a HF are simple trees.

(ii) There is at most one node in R which is the source of either a loop or non–regular hyperlink.

(iii) The dual $\mathcal{D}(HF)$ of HF is also a hyperforest.

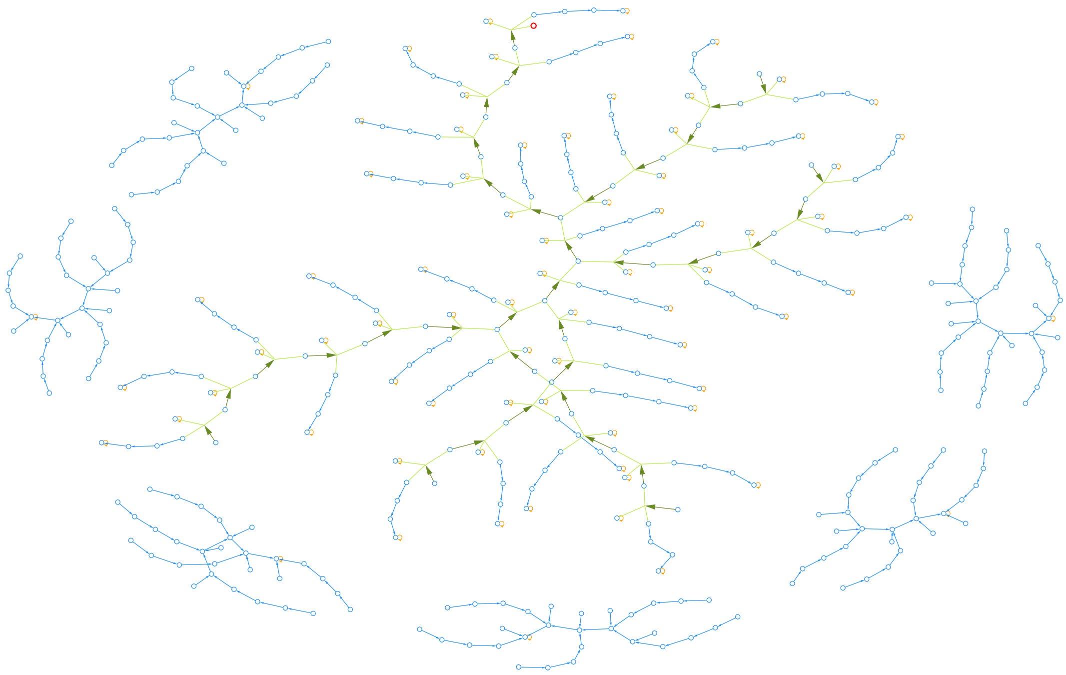


Figure V.7: The hyperforest 2/1 resulting while processing a model of $S^2 \times S^1$.

On figure V.7 for example, we can see how the non-regular hyperlink (in green) form a kind of forest.

Proof. **(i)**. Suppose R had a (simple) circuit $n_0, n_1, \ldots, n_{r+1} = n_0$. So there is $(r+1)$ nodes and $(r+1)$ regular links in this circuit. As a node cannot be the source of two links, each node is the source of exactly one link.

Suppose, without loss of generality, that n_0 is the source of the link $\{n_0, n_1\}$. n_1 is incident to two links of the circuit: $\{n_0, n_1\}$ and $\{n_1, n_2\}$. As it is not the source of the first one, so it is the source of $\{n_1, n_2\}$. Continuing those deductions, we prove that all the links of the circuit are oriented in such a way to form an oriented hypercircuit.

As HF is a hyperforest, this leads to a contradiction. Therefore, R is a simple tree.

(ii). Let k be the number of nodes of R. As R is a tree, it has (k-1) (regular) links. The end nodes of those links are nodes of R, as those links are regular (cf definition II.7). Therefore, among those k nodes, there are (k-1) nodes that are the source of regular hyperlinks. So there is at most $k-(k-1) = 1$ node in R which is the source of either a loop or a non–regular hyperlink.

(iii). Let $l_0, l_1, \ldots, l_{r+1} = l_0$ be a hypercircuit of $\mathcal{D}(HF)$. Let n_i be a common node of l_i and l_{i+1} in HF, different from n_{i-1} (l_{i-1} is different from l_i). Let n_{t+1} be the first node of the sequence to be equal to a precedent node n_s. Then $n_s, n_{s+1}, \ldots, n_{t+1} = n_s$ is a hypercircuit of HF. But HF is a hyperforest. Therefore, $\mathcal{D}(HF)$ has no hypercircuit. From the orientation defined at section II.1.5, $\mathcal{D}(HF)$ is a hyperforest. $\square$

V.2.2 Discrete gradient vector field and hyperforests

We defined a discrete gradient vector field as an acyclic matching in the Hasse diagram (cf section III.1.3), and a hyperforest as a hypergraph without hypercircuits. It seems natural to cf a discrete gradient vector field as a collection of hyperforests, extracted from the hypergraphs of the different layers of the Hasse diagram.

There are two hypergraphs representations of ranks p and q ($|p-q| = 1$) of the Hasse diagram: the direct layer p/q and the dual layer q/p. But the dual of a hyperforest is also a hyperforest. Therefore, considering a discrete gradient vector field as a collection of hyperforests is consistent.

Definition V.7 (Hypergraphs of a combinatorial vector field) *Let K be a cell complex, $\mathcal{V}$ a combinatorial vector field defined on it and B' the reduced layer p/q of $\mathcal{V}$ ($|p-q| = 1$). The p/q–hypergraph of $\mathcal{V}$, H, is the hypergraph representation of B': $H = \mathcal{B}^{-1}(B')$. H is oriented as follow: the source node of a hyperlink of H is the node representing its paired cell in $\mathcal{V}$.*

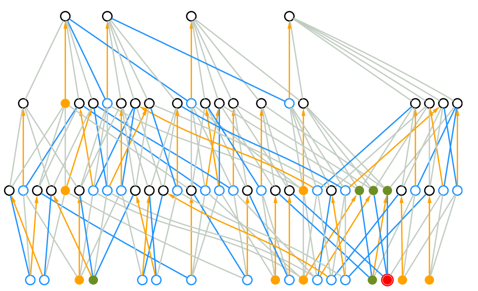


Figure V.8: The Hasse diagram of a discrete gradient vector field on a 4 cubes solid model.

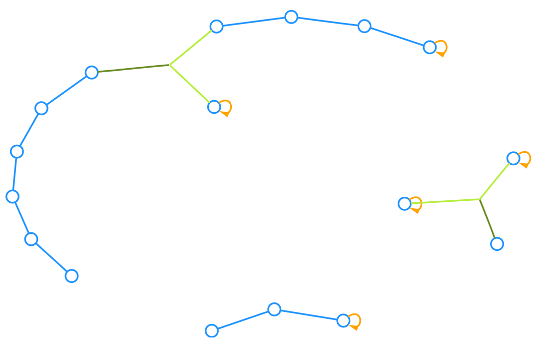


Figure V.9: The 1/0–hyperforest of the discrete gradient vector field of figure V.8.

For example, figure V.9 shows the hyperforest of the Hasse diagram of figure V.8.

Theorem V.8 *Let $\mathcal{V}$ be a combinatorial vector field. $\mathcal{V}$ is a discrete gradient vector field on an n–cell complex K if and only if the 0/1, 1/2, . . . (n-1)/n hypergraphs of $\mathcal{V}$ are hyperforests.*

As the dual of a hyperforest is a hyperforest (proposition V.6), the theorem is valid for any sequence obtained by replacing the p/q–hypergraph by the q/p–hypergraph of $\mathcal{V}$.

Proof. The orientation of HF ensures the first condition of definition V.5. By proposition V.3, every cell of K is represented in one of the reduced layers and any $\mathcal{V}$–path is represented in one of the hypergraphs. Therefore, we just need to prove that a closed $\mathcal{V}$–path is a hypercircuit in one of the hypergraphs.

Let $n_0, n_1, \ldots, n_{r+1} = n_0$ be an oriented hypercircuit in the p/q–hypergraph HF. From definition V.4, n_i is the source of a hyperlink l_i incident to n_{i+1}. This hyperlink l_i represents a q–cell β_i of K, and n_i represents a p–cell α_i. As n_i is the source of l_i, we know form the orientation of definition V.7 that α_i and β_i are incident and form a pair in $\mathcal{V}$. So $\alpha_0, \beta_0, \ldots, \alpha_r, \beta_r, \alpha_{r+1}$ is a $\mathcal{V}$–path. As $n_{r+1} = n_0$ and $r \geq 1$, this is a closed $\mathcal{V}$–path.

This argument can be reversed to prove that a closed $\mathcal{V}$–path is hypercircuit in one of the p/q–hypergraphs. □

We will now define the analogue of critical cells for hyperforests. This will be the foundation of the algorithm of chapter VI. A critical element of a discrete gradient vector field will be represented by a regular component of one of its hyperforest. For example, on figure V.7, the critical node in red corresponds to a critical component of the hyperforest (connecting 3 nodes).

Definition V.9 (Critical component) *A regular component of a hyperforest will be called* critical *if none of its node is the source of either a loop or a non–regular hyperlink.*

Proposition V.10 *Let HF be the p/q–hyperforest of K. The number of critical components of HF is exactly the number $m_p(K)$ of critical p–cells of K.*

Proof. From proposition V.3 every possible critical p–cell is represented HF and its corresponding reduced layer B'.

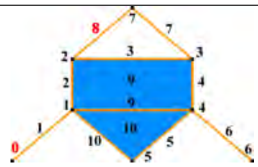
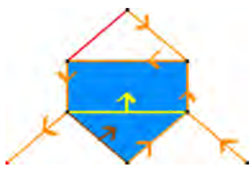
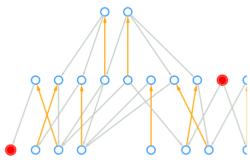
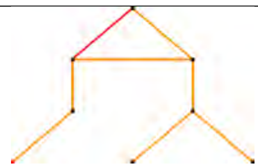
The isolated nodes of B' are not matched with any cell of K, and remain isolated nodes in HF. Those nodes are critical components, according to definition V.9.

We know from proposition V.6 that each regular component R is a simple tree. In such a tree with k nodes, there is (k-1) (regular) links. All links are oriented, so among those k nodes, (k-1) are the sources of links of R, and is therefore not critical. If R is not a critical component, there is exactly one node of R which is the source of either a loop or a non–regular hyperlink, i.e. it is not critical.

If R is a critical component, this node is neither the source of a loop nor of a non–regular hyperlink. From the definition II.7 of a regular component, this node is not incident to any regular hyperlink not in R. All those links of R are already paired with other nodes. So this node is unpaired in B'. From definition V.2, it cannot be paired with a cell outside B'. Therefore, it is an unpaired node, i.e. a critical cell. $\square$

V.2.3 3 points of view for optimality in discrete Morse theory

Theorems III.8, III.9 and V.8 prove the equivalence of the building blocs of discrete Morse theory:

Discrete Morse function	A function $f : K \longrightarrow \mathbb{R}$, almost increasing with the dimension.	 .
Discrete gradient vector field	An acyclic matching in the Hasse diagram.	 
Collection of hyperforests	Hypergraphs with no hypercircuits.	 

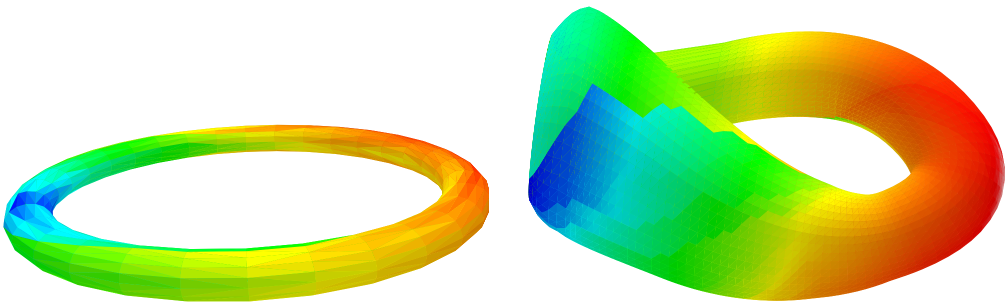
In chapter IV, we were mainly interested in optimality: Morse theory is a powerful tool to describe topological properties, and a small number of critical cells give a more concise information. This characterization can be done for the hyperforests: an optimal discrete Morse function will have the minimal possible number of critical components in each hyperforest extracted from the p/q layer. There are as many non–critical elements in a hyperforest as its number of hyperlinks (non–critical elements are paired with an incident hyperlink). Therefore, an optimal discrete Morse function has the maximum number of hyperlinks in each of its hyperforests. Therefore, the problem of finding a maximal hyperforest in a hypergraph is MAX–SNP hard also.

V.3 Discrete Morse numbers as topological invariants for 3–manifolds

We know that the critical elements of a Morse function on K are related to the topology of K. But are those elements a complete characterization of the topology of K? We already know part of the answer. We will show in this section that the discrete Morse numbers are topological invariants for 3–manifolds.

V.3.1 Accuracy of Morse numbers

Definition V.11 (Morse numbers) *The p–th Morse number $\mathcal{M}_p(K)$ of a cell complex K is the minimum possible number of critical p–cells, considering all possible discrete Morse functions defined on K.*



V.10(a): Torus V.10(b): Klein bottle

Figure V.10: A torus and Klein bottle with both a minimal discrete gradient vector field: 1 critical vertex, 2 critical edges, 1 critical face.

Morse theory is related to the simple homotopy [Coh73] type of a topological space. For example, an optimal Morse function defined on a knot and on the unknot will give rise to the same decomposition. Moreover, if we only consider the number of critical cells, and not their incidences, we are not able to distinguish between a torus and a Klein bottle (figures V.10). The homotopy type is able to make such a distinction. However, we know from the sphere theorem III.16 that they give a complete characterization of the sphere. Finally, Morse numbers are more precise than the Betti numbers: consider the homological sphere of Poincaré. The Morse number for this space cannot be 1–0–0–1, as the Betti numbers, because the homological sphere is not homotopic to a sphere. In fact, our algorithm of chapter VI gives the optimal answer 1–2–2–1 (there is a need of 2 generators for the fundamental group).

Those Morse numbers could be seen as an information between homology and simple homotopy.

V.3.2 Invariance proof for 3–manifolds

Discrete Morse numbers are linked to simple homotopy. To prove their invariance, we could prove that topologically equivalent cell complexes are simple homotopic, and that simple homotopic spaces have the same discrete Morse numbers. Unfortunately, the first affirmation is not true in the general case. We will use the following theorems, which demonstration can be found respectively in [Moi52] and [Coh73, 25.1].

Theorem V.12 (3–Manifold Hauptvermutung) *Any two triangulations of a topological 3–manifold have a common subdivision.*

Theorem V.13 *If K_* is a subdivision of K, then K and K_* are simple–homotopy equivalent.*

The proof of the invariance now follows:

Theorem V.14 (Invariance of discrete Morse numbers) *Let K and L be homeomorphic 3–manifolds. Then for all p, $\mathcal{M}_p(K) = \mathcal{M}_p(L)$.*

Proof. Let f be an optimal discrete Morse function defined on K. We will prove the theorem by the absurd. Suppose L would have its Morse number of index n higher than the one of K: $\mathcal{M}_n(L) > m_n(f)$. We will construct a discrete Morse function g on L with the same number of critical elements as f.

From theorem V.12, there exists a common subdivision to K and L. We deduce from theorem V.13 that L can be obtained from K by a finite number of collapses and extensions.

If M_* is an extension of M, and f is a discrete Morse function defined on M, we know from section III.2.3 that we can define a discrete Morse function f_* on M_* with the same number of critical elements as f. If M collapses on M_*, we know by proposition III.15 that we can extend f_* on M without adding any critical element.

Therefore, we can build a discrete Morse function g on L with the same number of critical elements as f. This contradicts $\mathcal{M}_n(L) > m_n(f)$. $\square$

VI
Constructing discrete Morse functions

VI.1 Data structure

In this chapter, we aim to build a discrete Morse function on a cell complex K. Such a function assigns to every cell of K a real value, so we need at least a cell structure and one entry per cell. The cell structure will be composed of the following fields:

- an identifier (unsigned long)
- the dimension (unsigned char)
- the value of the Morse function (unsigned long)
- the value of the discrete gradient vector field (0 or cell identifier)
- a list with the star cells' identifiers
- a list with the boundary cells' identifiers
- a flag to indicate being an edge of one of the hyperforests
- a "non–regular" flag
- a "loop" flag
- a "critical" flag
- a "visited flag" for graph traversals
- a component identifier for the forest creation

The component identifier will be used when mixing with geometry to create a minimal spanning forest, and will be used as a union–find structure [Tar75]. A cell complex structure will just be a matrix of every cell in each dimension. We also added a coordinate matrix for rendering those models.

VI.2 Algorithms for constructing discrete Morse functions out of hyperforests

The algorithms introduced here are extensions of the algorithm of section IV.2. They are also a natural deduction of the theory developed in chapter V.

The algorithms process each layer or dual layer of the Hasse diagram. For each of those, they define a hyperforest extracted from the layer's hypergraph. Then they give an orientation to those hyperforests, i.e. they define a discrete gradient vector field. At the same time, they can define the discrete Morse function on each cell.

VI.2.1 Hypergraph of the regular components

Constructing a hyperforest on a regular component can be done the same way as constructing a spanning tree [Hop73]. The problem resides in the hyperlinks that join those regular components. We can represent them, again, with a hypergraph. This reduction is the principle of one of the heuristics of section VI.4.2.

Definition VI.1 (Hypergraph of the regular components) *Let H be a hypergraph. The* hypergraph $\mathcal{C}(H)$ of the regular components *of H is the hypergraph with one node for each regular component of H, and whose hyperlinks are the loops and the non–regular hyperlinks of H.*

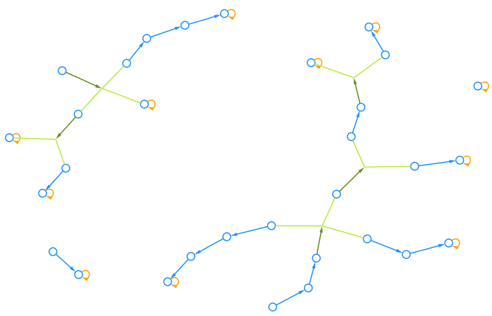


Figure VI.1: A 1/0–hyperforest of a pile of $3 \times 3 \times 1$ cubes.

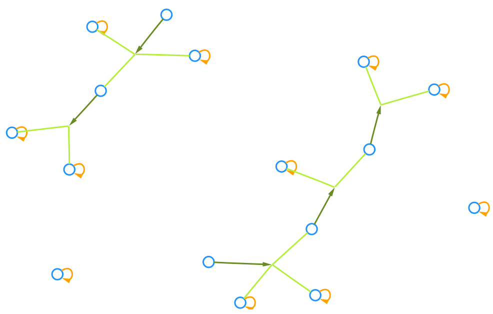


Figure VI.2: The hypergraph of the regular components of figure VI.1.

Notice that the hypergraph of the regular components of a hyperforest HF is also a hyperforest (cf figures VI.1 and VI.2). In that case, we know from section V.2.3 that there are exactly $r - h$ critical p–cells, where r is the number of regular components of a p/q–hyperforest, and l its number of loops and non–regular hyperlinks.

VI.2.2 Choosing the roots

For a given hyperforest HF, we will process its regular components from the root to the leaves. Therefore, we need to choose a root regular component R in the hyperforest $\mathcal{C}(HF)$ of the regular components of HF.

In each connected component of $\mathcal{C}(HF)$, there is either a critical component R or at least one loop incident to a regular component R. If not, there would be the same number of nodes and non–regular hyperlinks. As a node can be the source of at most one hyperlink, there would be a hypercircuit in $\mathcal{C}(HF)$. And as $\mathcal{C}(HF)$ is a hyperforest, this is impossible.

In each regular component R, we will choose a root. If the component is a critical component, the root can be any node of R. If the component is not critical, we know from definition V.9 that exactly one node is the source of a loop or a non–regular hyperlink. The root node of R will be that one.

The root regular component R and the roots of the regular components can be chosen among the different nodes incident to a loop to comply with geometric conditions (cf section VI.4.4).

VI.2.3 Construction of a discrete gradient vector field

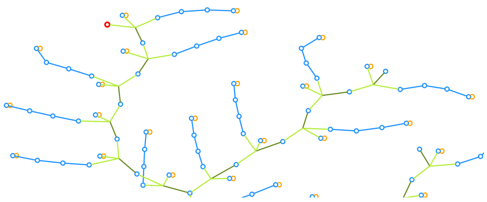


Figure VI.3: A part of the hyperforest 2/1 of $S^2 \times S^1$.

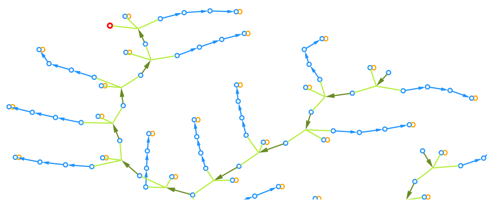


Figure VI.4: Orienting the hypergraph of figure VI.3.

For each regular component R, we need to pair each of its nodes, except one if R is critical. For example, figure VI.4 shows the result of this process on the model of figure VI.3. We first choose a root node in the component as described in section VI.2.2. We pair the leaves of R with their unique incident link. Considering $R^{(1)}$ composed of the unpaired nodes and links of R, we pair the leaf nodes of $R^{(1)}$ with their unique incident link. And we repeat the process on $R^{(2)}$ composed of the unpaired elements of $R^{(1)}$, and so forth.

After this process, the last node is the root node. If the component is critical, it remains unpaired and will be critical. If the component is not critical, we pair the root node with its incident loop or non–regular hyperlink. So every hyperlink of the hyperforest and every node except one per critical component have been paired.

VI.2.4 Construction of a discrete Morse function

We will discuss the different heuristics used for hyperforest creation in section VI.4. Depending on the heuristic used, two kinds of hyperforests can appear: direct hyperforest (of type $p/(p+1)$) and dual hyperforest (of type $(p+1)/p$).

Direct hyperforest. In a direct hyperforest HF, the nodes represent cells of lower dimension than the hyperlinks. Therefore, the nodes of HF should be assigned a smaller value than their incident links. This will be our primary construction.

Dual hyperforest. In a dual hyperforest HF, the nodes represent cells of higher dimension than the hyperlinks. Therefore, the nodes of HF should be assigned a greater value than their incident links. We can transform the primary construction to satisfy with this observation by considering the discrete Morse function $g : x \mapsto C - f(x)$, and we can choose C in such a way that $\mathrm{Im} g = \mathrm{Im} f$.

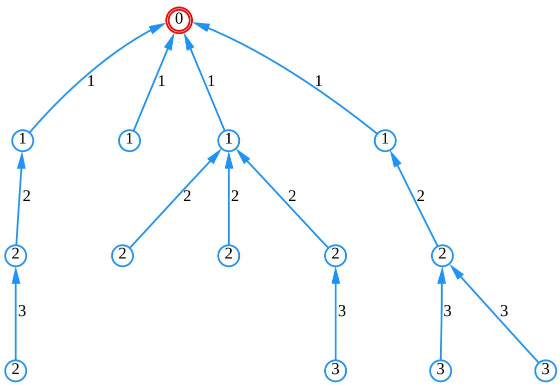


Figure VI.5: The edge distances on a small tree example.

For the root regular component R of HF, we assign the value c to its root node and its incident loop or non–regular hyperlink if any (cf section VI.2.2). Then, we assign to each node of R its edge distance (cf figure VI.5) to the root of R plus c, and to every link of R the maximum value of its two end nodes, as in step 4 of algorithm of section IV.2.

If the root regular component was isolated in $\mathcal{C}(HF)$, we already assigned all the elements of this connected component of $\mathcal{C}(HF)$. If this root component was not isolated, we repeat the process above on a regular component incident to this root regular component, with the initial value of c equal to the greatest value already assigned. Processing this way each connected component of $\mathcal{C}(HF)$ we assign all the elements of HF.

However, to ensure the inequalities of definition III.6, we must avoid that a cell of another layer of the hypergraph interferes with the cells of the hypergraph we considered. A simple way of doing so is to set the initial value c to the number of cells of K of dimension less or equal to p.

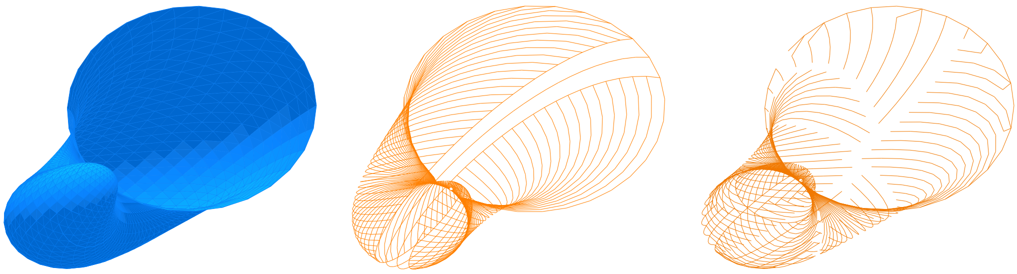
Notice that this discrete Morse function will have the same critical elements as the discrete gradient vector field of section VI.2.3. This gives another proof of theorem III.9: considering a discrete gradient vector field, we can build its Hasse diagram and a collection of the direct hyperforests of the layers $0/1$, $1/2$,..., $(n\text{-}1)/n$. Then we can construct the discrete Morse function as above, and thus prove the theorem.

VI.3 Optimality considerations

The main part of the algorithm is to create the hyperforest. As we have seen in sections VI.2.3 and VI.2.4, we do not need to care about orientation. Our goal is to reach optimality. Unfortunately, this is not possible in polynomial time (unless P=NP). Moreover, we saw in section III.4.3 that any polynomial approximation can be arbitrary far from the optimum. Therefore, we chose to extend our algorithm of chapter IV, which is proven to be optimal for 2–manifolds.

VI.3.1 Validity of local optimization

As we have seen in section V.3, the minimal number of critical cells is an invariant at least for 3–manifolds. Therefore, maximizing the number of hyperlinks in each layer of the Hasse diagram gives a global maximum:



VI.6(a): Original model VI.6(b): C_1 VI.6(c): C_2

Figure VI.6: A contractible space and the complement cell complexes C_1 and C_2 of two different hyperforests HF_1 and HF_2 defined on it.

Consider two different $n/(n\text{-}1)$–hyperforests HF_1 and HF_2 giving the same number of critical cells (or critical components). Now call C_1 and C_2 the two cell complexes represented by the cells of dimension $\leq n$ and whose $(n$-$1)$–cells are not in HF_1 and HF_2 respectively (cf figure VI.6). From theorems III.10 and III.11, C_1 and C_2 are simple homotopic. Therefore, they have the same discrete Morse number. We conclude by induction that, in the case of 3–manifolds, maximizing the number of hyperlinks in each hyperforest generates an optimal discrete Morse function.

VI.3.2 Regular components

Each regular component R of H is determined before any construction of HF. For any hyperforest HF, consider RT the simple graph which nodes are the n nodes of R and which links are the regular hyperlinks of HF incident to those nodes. As R is a regular component of H, there is no regular hyperlink incident to a node of R and a node out of R, so RT is well defined. As HF is a hyperforest, there is no circuit in RT: RT is a collection of k trees. So RT has $(n-k)$ links. The maximum number of links will thus be for k minimal, i.e. RT being a unique (connected) tree. This optimum can be reached by constructing a spanning tree on each regular component of H [Hop73].

Figure VI.7: Replacing a non–regular hyperlink by a loop.

VI.3.3 Loops

Each connected component of $\mathcal{C}(HF)$ is either critical or incident to either a loop or a non–regular hyperlink. The problem of regular hyperlinks has been resolved optimally, and we want now to maximize the number of loops and non–regular hyperlinks of HF. If a critical component is incident to a loop in H, then adding this loop to HF generates another hyperforest with one more hyperlink and one less critical component. If a regular component is incident to a loop l in H and to a non–regular hyperlink nl in H and HF, then replacing nl by l in HF generates another hyperforest with the same number of hyperlinks (and less risk to create a hypercircuit). This process is illustrated on figure VI.7. Therefore, we can always generate a hyperforest HF with the maximum possible number of hyperlinks such that every regular component incident to a loop in H is incident to a loop in HF.

VI.4 Different heuristics

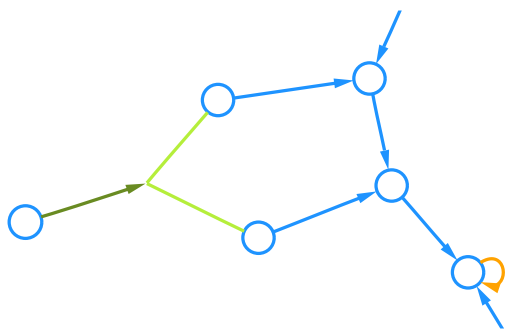


Figure VI.8: Detail of a hyperlink insertion in the dual hyperforest appearing with a solid torus model.

In section VI.3, we proved that reaching an optimal discrete gradient vector field can be obtained, at least for 3–manifolds, by maximizing the number of hyperlinks of the hyperforests HF extracted from the layers H of the Hasse diagram (definition V.1). This maximum can always be reached by, for each regular component R of H, generating a spanning tree, adding the links of this tree to HF, and if R is incident to a loop, adding it to HF. Then, we can process the non–regular hyperlinks of H. We aim that HF has the maximum number of hyperlinks, as it will minimize the number of critical cells. For example, adding the hyperlink on the left side of figure VI.8 allows us to pair it with the node on the left. Thus, there will be less critical (unpaired) nodes.

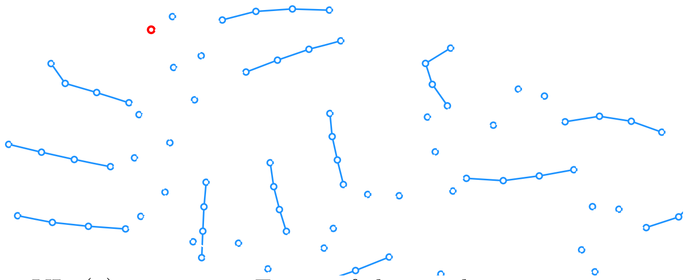
VI.4.1 Algorithm outline

We must first choose which layers of the Hasse diagram we process. In fact, we can process all of them, indifferently from their direct or their dual hypergraph representation. We know that the dual pseudograph of a manifold has no non–regular hyperlink, and that the direct hypergraph of the first layer is a simple graph. Those two simple cases could be useful as the construction of hyperforest is linear on pseudograph and quadratic on general hypergraphs. For example, a solid model could be processed by the following sequence of layers: 0/1,1/2,3/2; or 3/2,2/1,0/1.

In this work, all the algorithms to extract a hyperforest HF out of a hypergraph H process by the following steps (cf figures VI.9(a), VI.9(b) and fig:s2xs1 2):

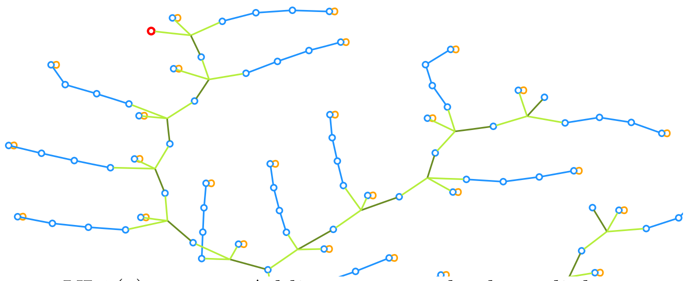
1. Initiate HF with the nodes of H.
2. Generate a spanning tree on every regular component of H.
3. Add all the links of those spanning trees to HF.
4. If a regular component is incident to some loops, add one of them to HF.
5. Process the non–regular hyperlinks of H.

The 4 first steps of the algorithm are linear, and guaranteed to be optimal in any case. The last step requires some heuristics as detailed below.



VI.9(a): steps 1–3: Forest of the regular components.

VI.9(b): step 4: Adding loops.



VI.9(c): step 5: Adding non–regular hyperlinks.

Figure VI.9: Steps of the algorithm on a part of the hyperforest 2/1 of $S^2 \times S^1$.

VI.4.2 Hypergraph simplification

Let HF be the hyperforest being created out of the hypergraph H. There is one critical cell of HF in each of its critical component (proposition V.10).

There is one obvious case for non–regular hyperlinks. A non–regular hyperlink nl can create a hypercircuit in a regular component R when it is incident more than one time to R[1], and when the source of nl is a node of R. A hyperlink can be added to HF only if it is incident to at least one critical component. If a hyperlink loops in all of its incident regular critical components of HF, we can remove it from H as it will never be part of HF. This condition is satisfied if the hyperlink is not incident to any critical component.

There are two obvious configurations for a regular component R of H: when the regular component is incident to a loop and when it is incident to

[1]hyperlinks are families of nodes, and not set of nodes

only one hyperlink. In the first case, we can add the loop to HF, and remove R and its loop from H (step 4). In the second case, if the hyperlink does not create a hypercircuit in R, we add it to HF, and remove R and this hyperlink from H. The two graphs of figure VI.7 are obvious in that sense.

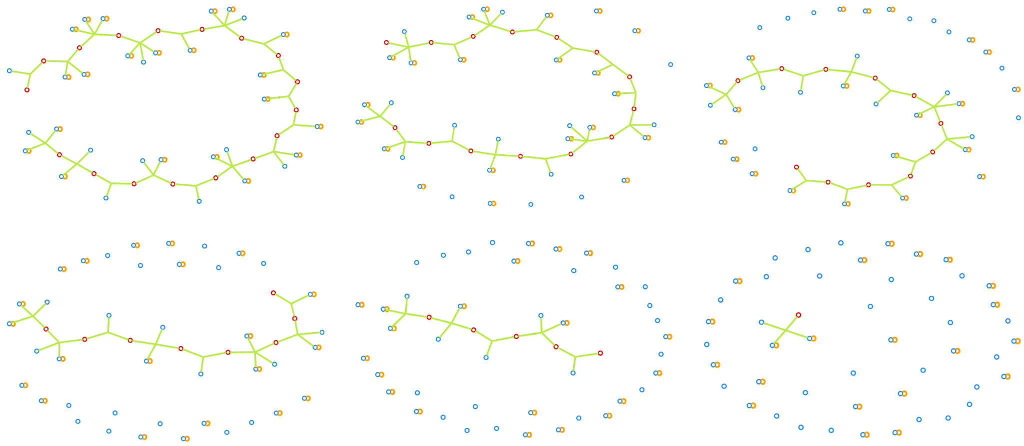


Figure VI.10: Successive simplifications of a hypergraph (3 per step).

Completely simplifying the hypergraph $\mathcal{C}(H)$ of the regular components (cf section VI.2.1) require a quadratic time of execution (cf figure VI.10. However, if some hyperlinks of $\mathcal{C}(HF)$ have not been removed, we need some further process. We can either apply an exponential algorithm to reach optimality if the size of the hypergraph would allow it, or apply one of the other heuristics to complete the hyperforest.

VI.4.3 Greedy methods

Let HF be the hyperforest being created out of the hypergraph H. We can try to add the hyperlinks of H to HF in order they appear after a sort. The criterion for a hyperlink not to be added to HF (and to be removed from H) is the obvious case of section VI.4.2: when it is multiply incident to every critical component it is incident to.

The priority on links (which appears in the sort) can be quite arbitrary, as there is no polynomial approximation. We tested 3 of them:

- minimal number of incident regular components.
- minimal number of incident critical components in HF.
- maximal number of incident non–critical components in HF.

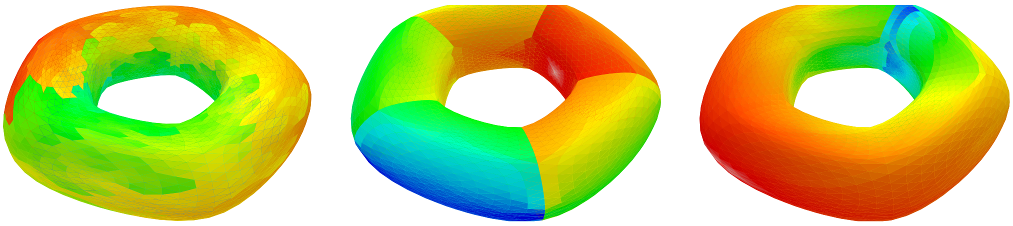
The problem that appears with those criterions is that the priority must be calculated again each time a hyperlink is added to HF (as some components change status from critical to non–critical). So the complexity of such a heuristic is quadratic, as the one of section VI.4.2.

VI.4.4 Mixing with geometry

As for the algorithm of chapter IV (cf section VI.4.4), we can impose some more conditions on our discrete Morse functions. However, there is a difference with that case: the geometry can influence the result, as the hyperforest of a layer will be different if the hyperforest of the precedent layer processed is a geometrical minimum (cf tables 6.21 a 6.26).

There are different constraints we can add on our hyperforest HF:

- The spanning tree of the regular components of HF can be chosen to be a minimal spanning tree.
- The loops added to the regular components of HF can minimize the same function, in order to have the root of the spanning trees at a minimal position.
- The roots of the spanning trees of the critical components of HF can also be at a minimal position.
- The priority used in the greedy heuristics (cf section VI.4.3) can be derived from the same geometrical function.



VI.11(a): No geometrical constraint VI.11(b): Minimal distance to origin VI.11(c): Minimal Z coordinate

Figure VI.11: Discrete gradient vector fields with geometrical constraints.

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	1208	530	14	402
Dual	7258	728	8	934
Sym Direct	3580	658	50	702
Sym Dual	3842	566	6	722

Table VI.1: Number of redundant cells per method on the panel of models of tables VI.2 and VI.3. Robust Morse: 56.

VI.5 Results

We compared the different heuristics of section VI.4 on two kinds of models: Hachimori's examples [HacModels] (mainly non–constructible, cf Table VI.2) and other solid models at the Mat&Mídia Laboratory (Table VI.3). The different heuristics we implemented are:

– Direct: processing the layers 0/1, 1/2, 2/3.

– Dual: processing the layers 3/2, 2/1, 1/0.

– Sym Direct: processing the layers 0/1, 1/2, 3/2.

– Sym Dual: processing the layers 3/2, 2/1, 0/1.

– HG Simpl: only simplifying the hypergraph, with no further proceeding.

– Min Def: priority to the hyperlinks incident to the minimum of critical components.

– Min Deg: priority to the hyperlinks of minimum degree.

– Max Cpl: priority to the hyperlinks incident to the maximum of non–critical components.

We added a simple comparison called "Robust Morse", which is a reinforced version of Direct/Min Def. The algorithm is optimal for surfaces in the case of

Sym Dual (chapter IV. The results of those processes are given on Table VI.1. The detailed results appear on Tables 6.4 to 6.26.

To illustrate the effect of geometry constraints on hyperforest creation, we applied different geometrical constraints on the solids models of Table VI.3:

– No geometrical constraint: the hyperlinks appear in the order of the model importation (cf figure VI.11(a)).

– Minimal distance to origin: the hyperlinks are first ordered by their distance to a reference point, and the spanning–trees are geometrically minimal, with the closest root to the origin (cf figure VI.11(b)).

– Minimal Z coordinate: the hyperlinks are first ordered by their height, and the spanning–trees are geometrically minimal, with the lowest height root (cf figure VI.11(c)).

Model	Topology	Number of cells	Euler	Robust	Best Morse
bing	3–ball	(480,2511,3586,1554)	1	(1,2,2,0)	(1,1,1,0)
bjorner	projective plane + one facet	(6,15,11)	2	(1,0,1)	
c-ns	contractible	(12,37,26)	1	(1,2,2)	(1,1,1)
c-ns2	contractible	(13,39,27)	1	(1,2,2)	(1,0,0)
c-ns3	contractible	(10,31,22)	1	(1,1,1)	
dunce hat	Dunce hat	(8,24,17)	1	(1,1,1)	
gruenbaum	3–ball	(14,54,70,29)	1	(1,0,0,0)	
knot	3–ball	(380,1929,2722,1172)	1	(1,1,1,0)	
lockeberg	3–sphere	(12,60,96,48)	0	(1,0,0,1)	
mani-walkup-C	3–sphere	(20,126,212,106)	0	(1,0,0,1)	
mani-walkup-D	3–sphere	(16,106,180,90)	0	(1,0,0,1)	
nonextend	contractible	(7,19,13)	1	(1,1,1)	(1,0,0)
poincare	homology sphere	(16,106,180,90)	0	(1,2,2,1)	
projective	projective plane	(6,15,10)	1	(1,1,1)	
rudin	3–ball	(14,66,94,41)	1	(1,0,0,0)	
simon	contractible	(7,20,14)	1	(1,1,1)	(1,0,0)
ziegler	3–sphere	(10,38,50,21)	1	(1,0,0,0)	

Table VI.2: Results on Hachimori's models [HacModels].

Model	Topology	Number of cells	Euler	Robust	Best
Pile of Cubes	contractible	(572,1477,1266,360)	1	(1,0,0,0)	
s2xs1	$S^2 \times S^1$	(192,588,612,216)	0	(1,3,4,2)	(1,1,1,1)
s3	3–sphere	(162,522,576,216)	0	(1,1,1,1)	(1,0,0,1)
solid 2sphere	2–sphere	(64,144,108,26)	2	(1,0,1,0)	
Furch knotted ball	3–ball	(600,1580,1350,369)	1	(1,1,1,0)	

Table VI.3: Results on solid models.

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,526,572,46)	(1,119,119,0)	(1,1,1,0)	(1,84,88,4)
Dual	(410,1084,675,0)	(32,186,155,0)	(3,5,3,0)	(40,179,140,0)
Sym Direct	(1,526,526,0)	(1,119,119,0)	(1,1,1,0)	(1,84,84,0)
Sym Dual	(1,675,675,0)	(1,155,155,0)	(1,3,3,0)	(1,140,140,0)

Table VI.4: bing. Cells: (480,2511,3586,1554). Robust Morse: (1,2,2,0).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,1,2)	(1,0,1)	(1,0,1)	(1,0,1)
Dual	(2,2,2)	(1,0,1)	(1,0,1)	(1,0,1)
Sym Direct	(1,0,1)	(1,0,1)	(1,0,1)	(1,0,1)
Sym Dual	(1,1,2)	(1,0,1)	(1,0,1)	(1,0,1)

Table VI.5: bjorner. Cells: (6,15,11). Robust Morse: (1,0,1).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,8,8)	(1,2,2)	(1,2,2)	(1,1,1)
Dual	(2,2,1)	(1,1,1)	(1,1,1)	(1,1,1)
Sym Direct	(1,3,3)	(1,2,2)	(1,2,2)	(1,2,2)
Sym Dual	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)

Table VI.6: c-ns. Cells: (12,37,26). Robust Morse: (1,2,2).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,3,3)	(1,2,2)	(1,2,2)	(1,2,2)
Dual	(3,2,0)	(1,0,0)	(1,0,0)	(1,0,0)
Sym Direct	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)
Sym Dual	(1,0,0)	(1,0,0)	(1,0,0)	(1,0,0)

Table VI.7: c-ns2. Cells: (13,39,27). Robust Morse: (1,2,2).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,5,5)	(1,2,2)	(1,1,1)	(1,1,1)
Dual	(2,2,1)	(1,1,1)	(1,1,1)	(1,1,1)
Sym Direct	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)
Sym Dual	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)

Table VI.8: c-ns3. Cells: (10,31,22). Robust Morse: (1,1,1).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,7,7)	(1,2,2)	(1,1,1)	(1,2,2)
Dual	(2,2,1)	(1,1,1)	(1,1,1)	(1,1,1)
Sym Direct	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)
Sym Dual	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)

Table VI.9: dunce hat. Cells: (8,24,17). Robust Morse: (1,1,1).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,5,5,0)	(1,0,0,0)	(1,0,0,0)	(1,1,1,0)
Dual	(10,18,9,0)	(1,2,2,0)	(1,0,0,0)	(1,1,1,0)
Sym Direct	(1,5,5,0)	(1,0,0,0)	(1,0,0,0)	(1,1,1,0)
Sym Dual	(1,9,9,0)	(1,2,2,0)	(1,0,0,0)	(1,1,1,0)

Table VI.10: gruenbaum. Cells: (14,54,70,29). Robust Morse: (1,0,0,0).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,408,439,31)	(1,83,83,0)	(1,2,2,0)	(1,75,77,2)
Dual	(270,734,465,0)	(25,134,110,0)	(2,3,2,0)	(21,120,100,0)
Sym Direct	(1,408,408,0)	(1,83,83,0)	(1,2,2,0)	(1,75,75,0)
Sym Dual	(1,465,465,0)	(1,110,110,0)	(1,2,2,0)	(1,100,100,0)

Table VI.11: knot. Cells: (380,1929,2722,1172). Robust Morse: (1,1,1,0).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,7,20,14)	(1,2,3,2)	(1,0,0,1)	(1,0,0,1)
Dual	(11,23,13,1)	(1,2,2,1)	(1,0,0,1)	(1,2,2,1)
Sym Direct	(1,7,7,1)	(1,2,2,1)	(1,0,0,1)	(1,0,0,1)
Sym Dual	(1,13,13,1)	(1,2,2,1)	(1,0,0,1)	(1,2,2,1)

Table VI.12: lockeberg. Cells: (12,60,96,48). Robust Morse: (1,0,0,1).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,31,68,38)	(1,10,16,7)	(1,0,0,1)	(1,5,7,3)
Dual	(19,56,38,1)	(2,11,10,1)	(1,0,0,1)	(2,9,8,1)
Sym Direct	(1,31,31,1)	(1,10,10,1)	(1,0,0,1)	(1,5,5,1)
Sym Dual	(1,38,38,1)	(1,10,10,1)	(1,0,0,1)	(1,8,8,1)

Table VI.13: mani-walkup-C. Cells: (20,126,212,106). Robust Morse: (1,0,0,1).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,26,58,33)	(1,7,10,4)	(1,0,0,1)	(1,2,4,3)
Dual	(16,54,39,1)	(2,8,7,1)	(1,0,0,1)	(1,9,9,1)
Sym Direct	(1,26,26,1)	(1,7,7,1)	(1,0,0,1)	(1,2,2,1)
Sym Dual	(1,39,39,1)	(1,7,7,1)	(1,0,0,1)	(1,9,9,1)

Table VI.14: mani-walkup-D. Cells: (16,106,180,90). Robust Morse: (1,0,0,1).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)
Dual	(2,1,0)	(1,0,0)	(1,0,0)	(1,0,0)
Sym Direct	(1,0,0)	(1,0,0)	(1,0,0)	(1,0,0)
Sym Dual	(1,0,0)	(1,0,0)	(1,0,0)	(1,0,0)

Table VI.15: nonextend. Cells: (7,19,13). Robust Morse: (1,1,1).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,28,63,36)	(1,9,12,4)	(1,2,3,2)	(1,7,10,4)
Dual	(14,47,34,1)	(2,12,11,1)	(1,2,2,1)	(3,12,10,1)
Sym Direct	(1,28,28,1)	(1,9,9,1)	(1,2,2,1)	(1,7,7,1)
Sym Dual	(1,34,34,1)	(1,11,11,1)	(1,2,2,1)	(1,10,10,1)

Table VI.16: poincare. Cells: (16,106,180,90). Robust Morse: (1,2,2,1).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,2,2)	(1,1,1)	(1,1,1)	(1,1,1)
Dual	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)
Sym Direct	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)
Sym Dual	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)

Table VI.17: projective. Cells: (6,15,10). Robust Morse: (1,1,1).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,11,12,1)	(1,3,3,0)	(1,0,0,0)	(1,1,1,0)
Dual	(12,29,18,0)	(1,4,4,0)	(1,0,0,0)	(1,4,4,0)
Sym Direct	(1,11,11,0)	(1,3,3,0)	(1,0,0,0)	(1,1,1,0)
Sym Dual	(1,18,18,0)	(1,4,4,0)	(1,0,0,0)	(1,4,4,0)

Table VI.18: rudin. Cells: (14,66,94,41). Robust Morse: (1,0,0,0).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,4,4)	(1,1,1)	(1,1,1)	(1,1,1)
Dual	(1,0,0)	(1,0,0)	(1,0,0)	(1,0,0)
Sym Direct	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)
Sym Dual	(1,0,0)	(1,0,0)	(1,0,0)	(1,0,0)

Table VI.19: simon. Cells: (7,20,14). Robust Morse: (1,1,1).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,3,3,0)	(1,0,0,0)	(1,0,0,0)	(1,0,0,0)
Dual	(7,12,6,0)	(1,1,1,0)	(1,0,0,0)	(1,0,0,0)
Sym Direct	(1,3,3,0)	(1,0,0,0)	(1,0,0,0)	(1,0,0,0)
Sym Dual	(1,6,6,0)	(1,1,1,0)	(1,0,0,0)	(1,0,0,0)

Table VI.20: ziegler. Cells: (10,38,50,21). Robust Morse: (1,0,0,0).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,258,303,45)	(1,56,66,10)	(1,0,0,0)	(1,64,74,10)
Dual	(199,327,129,0)	(12,35,24,0)	(1,0,0,0)	(14,28,15,0)
Sym Direct	(1,258,258,0)	(1,56,56,0)	(1,0,0,0)	(1,64,64,0)
Sym Dual	(1,129,129,0)	(1,24,24,0)	(1,0,0,0)	(1,15,15,0)

(a) No geometrical constraint

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,20,20,0)	(1,0,0,0)	(1,0,0,0)	(1,3,3,0)
Dual	(48,73,26,0)	(1,0,0,0)	(1,0,0,0)	(5,6,2,0)
Sym Direct	(1,20,20,0)	(1,0,0,0)	(1,0,0,0)	(1,3,3,0)
Sym Dual	(1,26,26,0)	(1,0,0,0)	(1,0,0,0)	(1,2,2,0)

(b) Minimal distance to origin

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,32,32,0)	(1,3,3,0)	(1,0,0,0)	(1,2,2,0)
Dual	(126,228,103,0)	(8,17,10,0)	(1,0,0,0)	(9,14,6,0)
Sym Direct	(1,32,32,0)	(1,3,3,0)	(1,0,0,0)	(1,2,2,0)
Sym Dual	(1,103,103,0)	(1,10,10,0)	(1,0,0,0)	(1,6,6,0)

(c) Minimal Z coordinate

Table VI.21: Pile of Cubes. Cells: (572,1477,1266,360). Robust Morse: (1,0,0,0).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,104,216,113)	(1,20,27,8)	(1,1,1,1)	(1,24,31,8)
Dual	(99,208,110,1)	(8,28,21,1)	(1,1,1,1)	(8,33,26,1)
Sym Direct	(1,104,104,1)	(1,20,20,1)	(1,1,1,1)	(1,24,24,1)
Sym Dual	(1,110,110,1)	(1,21,21,1)	(1,1,1,1)	(1,26,26,1)

(a) No geometrical constraint

Direct	(1,53,103,51)	(1,6,8,3)	(1,1,1,1)	(1,10,15,6)
Dual	(6,8,3,1)	(1,1,1,1)	(1,1,1,1)	(1,1,1,1)
Sym Direct	(1,53,53,1)	(1,6,6,1)	(1,1,1,1)	(1,10,10,1)
Sym Dual	(1,3,3,1)	(1,1,1,1)	(1,1,1,1)	(1,1,1,1)

(b) Minimal distance to origin

Direct	(1,69,124,56)	(1,12,15,4)	(1,1,1,1)	(1,16,24,9)
Dual	(32,79,48,1)	(1,1,1,1)	(1,1,1,1)	(2,11,10,1)
Sym Direct	(1,69,69,1)	(1,12,12,1)	(1,1,1,1)	(1,16,16,1)
Sym Dual	(1,48,48,1)	(1,1,1,1)	(1,1,1,1)	(1,10,10,1)

(c) Minimal Z coordinate

Table VI.22: s2xs1. Cells: (192,588,612,216). Robust Morse: (1,3,4,2).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,73,154,82)	(1,11,16,6)	(1,0,0,1)	(1,12,18,7)
Dual	(74,166,93,1)	(6,18,13,1)	(1,1,1,1)	(7,20,14,1)
Sym Direct	(1,73,73,1)	(1,11,11,1)	(1,0,0,1)	(1,12,12,1)
Sym Dual	(1,93,93,1)	(1,13,13,1)	(1,1,1,1)	(1,14,14,1)

(a) No geometrical constraint

Direct	(1,99,208,110)	(1,20,26,7)	(1,1,1,1)	(1,11,17,7)
Dual				
Sym Direct	(1,99,99,1)	(1,20,20,1)	(1,1,1,1)	(1,11,11,1)
Sym Dual				

(b) Minimal distance to origin

Direct	(1,44,90,47)	(1,0,0,1)	(1,0,0,1)	(1,5,7,3)
Dual	(34,66,33,1)	(1,0,0,1)	(1,0,0,1)	(4,6,3,1)
Sym Direct	(1,44,44,1)	(1,0,0,1)	(1,0,0,1)	(1,5,5,1)
Sym Dual	(1,33,33,1)	(1,0,0,1)	(1,0,0,1)	(1,3,3,1)

(c) Minimal Z coordinate

Table VI.23: s3. Cells: (162,522,576,216). Robust Morse: (1,1,1,1).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,16,17,0)	(1,2,3,0)	(1,0,1,0)	(1,2,3,0)
Dual	(19,25,8,0)	(2,1,1,0)	(1,0,1,0)	(1,0,1,0)
Sym Direct	(1,16,17,0)	(1,2,3,0)	(1,0,1,0)	(1,2,3,0)
Sym Dual	(1,7,8,0)	(1,0,1,0)	(1,0,1,0)	(1,0,1,0)

(a) No geometrical constraint

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,5,6,0)	(1,0,1,0)	(1,0,1,0)	(1,0,1,0)
Dual	(21,27,8,0)	(1,0,1,0)	(1,0,1,0)	(3,2,1,0)
Sym Direct	(1,5,6,0)	(1,0,1,0)	(1,0,1,0)	(1,0,1,0)
Sym Dual	(1,7,8,0)	(1,0,1,0)	(1,0,1,0)	(1,0,1,0)

(b) Minimal distance to origin

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,1,2,0)	(1,0,1,0)	(1,0,1,0)	(1,0,1,0)
Dual	(13,16,5,0)	(1,0,1,0)	(1,0,1,0)	(2,1,1,0)
Sym Direct	(1,1,2,0)	(1,0,1,0)	(1,0,1,0)	(1,0,1,0)
Sym Dual	(1,4,5,0)	(1,0,1,0)	(1,0,1,0)	(1,0,1,0)

(c) Minimal Z coordinate

Table VI.24: solid 2sphere. Cells: (64,144,108,26). Robust Morse: (1,0,1,0).

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,221,235,14)	(1,46,47,1)	(1,1,1,0)	(1,47,48,1)
Dual	(241,446,206,0)	(18,62,45,0)	(1,1,1,0)	(14,53,40,0)
Sym Direct	(1,221,221,0)	(1,46,46,0)	(1,1,1,0)	(1,47,47,0)
Sym Dual	(1,206,206,0)	(1,45,45,0)	(1,1,1,0)	(1,40,40,0)

(a) No geometrical constraint

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,52,52,0)	(1,6,6,0)	(1,1,1,0)	(1,7,7,0)
Dual	(165,302,138,0)	(3,5,3,0)	(1,2,2,0)	(11,36,26,0)
Sym Direct	(1,52,52,0)	(1,6,6,0)	(1,1,1,0)	(1,7,7,0)
Sym Dual	(1,138,138,0)	(1,3,3,0)	(1,2,2,0)	(1,26,26,0)

(b) Minimal distance to origin

	HG Simpl	Min Deg	Min Def	Max Cpl
Direct	(1,50,50,0)	(1,9,9,0)	(1,1,1,0)	(1,9,9,0)
Dual	(159,302,144,0)	(2,11,10,0)	(1,1,1,0)	(10,34,25,0)
Sym Direct	(1,50,50,0)	(1,9,9,0)	(1,1,1,0)	(1,9,9,0)
Sym Dual	(1,144,144,0)	(1,10,10,0)	(1,1,1,0)	(1,25,25,0)

(c) Minimal Z coordinate

Table VI.25: Furch. Cells: (600,1580,1350,369). Robust Morse: (1,1,1,0).

VII
Future Works

This work was focused on Forman's discrete Morse theory. We analyzed the building blocs of this theory, and proved the layered structure of discrete Morse functions. We represented this layer structure by a collection of hyperforests and gave a complete characterization of the critical cells in terms of regular components of hyperforests. We used this analysis to introduce a scheme for constructing discrete Morse function on finite cell complexes of arbitrary dimension. This construction is quadratic in time in the worst cases, and is proven to be linear and optimal in the case of 2-manifolds. The experimental results showed our algorithm gave an optimal result in most of the cases. This opens the question of which conditions on the cell complex would ensure the optimality of the resulting function.

An important application of this work to computer graphics would be in the field of geometric compression. The algorithm Grow&Fold of A. Szymczak and J. Rossignac [Szy00] could be justified and enhanced by our algorithm to minimize the number of so-called "glue faces" in order to achieve a better encoding. This work has been done in an optimal way for the case of surfaces with handles in [Lop02].

We plan to continue this work in three directions. First, as mentioned above, apply Forman's theory and the analysis of our algorithm for solid mesh compression. Second, we will try to apply discrete Morse theory to resolve singularities that arise from shape reconstruction. Finally, we look forward to produce a topologically consistent morphing based on mapping directly the discrete gradient field between two objects of the same homotopy type.

Bibliography

[Arm79] M. Armstrong. *Basic topology.* McGraw-Hill, London, 1979.

[Ber70] C. Berge. *Graphes et hypergraphes.* Dunod, Paris, 1970.

[Ber99] M. W. Bern, D. Eppstein, et al. Emerging challenges in computational topology. ACM Computing Research Repository, 1999.

[Boi98] J.-D. Boissonnat and M. Yvinec. *Algorithmic Geometry.* Cambridge University Press, 1998.

[Coh73] M. M. Cohen. *A course in simple homotopy theory.* Springer, New York, 1973.

[Del93] C. J. A. Delfinado and H. Edelsbrunner. An Incremental Algorithm for Betti Numbers of Simplicial Complexes. In *Proceedings of 9th Annual Symposium on Computer Geometry*, pages 232–239, 1993.

[Dey99] T. K. Dey, H. Edelsbrunner, and S. Guha. Computational topology. In B. Chazelle, J. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, volume 223 of *Contemporary mathematics*, pages 109–143. American Mathematical Society, Providence, 1999.

[Dey01] T. K. Dey and S. Guha. Algorithms for manifolds and simplicial complexes in euclidean 3-Space. preprint, 2001.

[Ede01] H. Edelsbrunner, J. L. Harer, and A. Zomorodian. Hierarchical Morse Complexes for Piecewise Linear 2-Manifolds. In *Proceedings of the 17th Symposium of Computational Geometry*, pages 70–79, 2001.

[Ege95] Ö. Eğecioğlu and T. F. Gonzalez. A computationally intractable problem on simplicial complexes. *Computational Geometry: Theory and Applications*, 6:85–98, 1996.

[For95] R. Forman. A discrete Morse theory for cell complexes. In S. T. Yau, editor, *Geometry, Topology and Physics for Raoul Bott*. International Press, 1995.

[For98] R. Forman. Morse theory for cell complexes. *Advances in Mathematics*, 134:90–145, 1998.

[For00] R. Forman. Applications of combinatorial differential topology. In *Proceedings of the SullivanFest*, 2000.

[For01] R. Forman. A user guide to discrete Morse theory. In *Séminaire Lotharingien de Combinatoire*, volume 48, 2001.

[GFZneato] S. C. North. Neato User's Guide. Technical report, AT&T Bell Laboratories, Murray Hill, NJ, 1992.

[GFZdot] E. Koutsofios and S. C. North. Drawing graphs with dot. Technical report, AT&T Bell Laboratories, Murray Hill, NJ, 1993.

[HacModels] M. Hachimori. Simplicial complex library. www.qci.jst.go.jp/-˜hachi.

[Har98] J. C. Hart. Morse theory for implicit surface modeling. In H.-C. Hege and K. Polthier, editors, *Mathematical Visualization*, pages 257–268, Berlin, 1998. Springer.

[Hop73] J. Hopcroft and R. E. Tarjan. Efficient algorithms for graph manipulation. *Communications of the ACM*, 16:372–378, 1973.

[Lew01] T. Lewiner, H. Lopes, and G. Tavares. Optimal discrete Morse functions for 2-manifolds. *Computational Geometry: Theory and Applications*, 26(3):221–233, 2003.

[Lew02] T. Lewiner, H. Lopes, and G. Tavares. Visualizing Forman's discrete vector field. In H.-C. Hege and K. Polthier, editors, *Visualization and Mathematics III*, pages 95–112. Springer-Verlag, Heidelberg, 2003.

[Lop96] H. Lopes. *Algorithms to build and unbuild 2 and 3 dimensional manifolds*. PhD thesis, Department of Mathematics, PUC-Rio, 1996.

[Lop02] H. Lopes, J. Rossignac, A. Safonova, A. Szymczak, and G. Tavares. Edgebreaker: a simple compression for surfaces with handles.

In *7th ACM Siggraph Symposium on Solid Modeling and Application*, pages 289–296, 2002.

[Lov86] L. Lovász and M. D. Plummer. *Matching Theory.* Van Nostrand Reinhold, Amsterdam, 1986.

[Lun69] A. Lundell and S. Weingram. *The topology of CW complexes.* Van Nostrand Reinhold, New York, 1969.

[Mar58] A. Markov. Insolvability of the problem of homeomorphy. In *Proceedings of the International Congress of Mathematics*, pages 300–306, 1958.

[Mey02] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete Differential–Geometry Operators for Triangulated 2–Manifolds. In H.-C. Hege and K. Polthier, editors, *Mathematical Visualization III.* Springer, Berlin, 2002.

[Mil63] J. W. Milnor. *Morse theory.* Princeton University Press, Princeton, NJ, 1963.

[Moi52] E. E. Moïse. Affine structures in 3–manifolds. *Annals of Math*, 56(2):96–114, 1952.

[Mor00] S. Moriyama and F. Takeuchi. Incremental construction properties in dimension two— shellability, extendable shellability and vertex decomposability. In *Proceedings of the 12th Canadian conference on computational geometry*, pages 65–72, Fredericton, 2000.

[Shi91] Y. Shinagawa, T. Kunii, and Y. Kergosien. Surface coding based on Morse theory. *IEEE Computer Graphics and Applications*, 11:66–78, 1991.

[Szy00] A. Szymczak and J. Rossignac. Grow & Fold: Compressing the connectivity of tetrahedral meshes. *Computer–Aided Design*, 32(8/9):527–538, 2000.

[Tar75] R. E. Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the ACM*, 22(2):215–225, 1975.

[Tau98] G. Taubin and J. Rossignac. Geometric compression through topological surgery. *ACM Transactions on Graphics*, 17(2):84–115, 1998.

[Veg97] G. Vegter. Computational topology. In J. E. Goodman and J. O'Rourke, editors, *Handlebook of Discrete Computational Geometry*, pages 517–536. CRC Press, 1997.