UNIVERSITÉ PARIS 7 — DENIS DIDEROT
UFR D'INFORMATIQUE

THÈSE

présentée et soutenue publiquement
le 8 décembre 2003
pour l'obtention du Diplôme de

Docteur de l'Université Paris 7

Spécialité : Algorithmique

par

Éric Colin de Verdière

Shortening of curves and decomposition of surfaces

Examining board :

M. Christian Choffrut (Univ. Paris 7)	president
Mme Christiane Frougny (Univ. Paris 8)	examiner
M. Michel Pocchiola (ENS Paris)	thesis advisor
M. Günter Rote (FU Berlin)	reviewer
Mme Marie-Françoise Roy (Univ. Rennes 1)	reviewer
Mme Brigitte Vallée (CNRS & Univ. Caen)	examiner

Other rewiever :
M. Herbert Edelsbrunner (Duke Univ.)

Research done in the
Laboratoire d'informatique
de l'École normale supérieure
UMR 8 548 du CNRS

UNIVERSITÉ PARIS 7 — DENIS DIDEROT
UFR D'INFORMATIQUE

THÈSE

présentée et soutenue publiquement
le 8 décembre 2003
pour l'obtention du Diplôme de

Docteur de l'Université Paris 7

Spécialité : Algorithmique

par

Éric Colin de Verdière

Shortening of curves and decomposition of surfaces

Examining board :

M. Christian Choffrut (Univ. Paris 7)	president
Mme Christiane Frougny (Univ. Paris 8)	examiner
M. Michel Pocchiola (ENS Paris)	thesis advisor
M. Günter Rote (FU Berlin)	reviewer
Mme Marie-Françoise Roy (Univ. Rennes 1)	reviewer
Mme Brigitte Vallée (CNRS & Univ. Caen)	examiner

Other rewiever :
M. Herbert Edelsbrunner (Duke Univ.)

Research done in the
Laboratoire d'informatique
de l'École normale supérieure

UMR 8 548 du CNRS

The thesis was originally written in French, except Chapters 4 and 5, which are also in English in the original version. This document is a translation by the author.
Errors and typos might have been introduced during the translation (or even exist in the French version). If you find such errors, I would really appreciate the feedback.

Acknowledgements

I would like to thank first Michel Pocchiola, who guided and encouraged me during these three years of thesis. He was able to propose interesting research topics, to improve my ability in writing, and to give me adequate advices when I needed help. I particularly appreciated his help when writing this document.

I thank Jacques Stern, the headmaster of the Computer Science Department at ENS. Many thanks also to Jacques Beigbeder, Kilian Hart, and Catherine Le Bihan, maintaining a high-quality computer installation, to Joëlle Isnard, Michelle Angely, Jean-Claude Denise, and Valérie Mongiat for their availability and efficiency in the necessary administrative tasks, and to the team of the library.

I would like to say my gratitude to Herbert Edelsbrunner, Günter Rote, and Marie-Françoise Roy, reviewers of this thesis; I thank them for the interest they expressed in reading my work. Thanks also to Christian Choffrut, Christiane Frougny, and Brigitte Vallée, who accepted to be in my examining board.

Many thanks also to Jean-Daniel Boissonnat and to all members of the Prisme team for their kindness during the few visits I did at Inria Sophia-Antipolis. I appreciated to work with Mariette Yvinec on conforming Delaunay triangulations. Thanks to David Cohen-Steiner, with whom every discussion can generate new ideas, and to Pierre Alliez, who always knows how to relate theory and applications.

I would like to thank also the other researchers with whom I had the privilege and the pleasure to work, in particular Gert Vegter, who initiated my learning in mathematical writing, and Francis Lazarus, who also read my work on Tutte's theorem.

Let me also thank, for many reasons, several persons for their help: my uncle Yves Colin de Verdière, Robert Connelly, Michael Floater, Valérie Pham-Trong, Michael Starbird, and Anne Verroust.

I wish to thank the persons I met everyday during this thesis, and whose help was precious. Thanks to those who shared with me the office of the students of the team: Pierre Angelier, Xavier Goaoc, and Laurent Rineau, who helped me in programming, and Luc Habert, newcomer in the team.

I am really grateful to my family, wife's family, and friends who encouraged me and helped to make these years enjoyable. Finally and over all, thanks to Mathilde for her priceless help and her support, particularly when I was writing this thesis.

Contents

Introduction

Decomposing, deforming, shortening while preserving topology

Decomposing, deforming and shortening are elementary operations on geometric objects. This thesis is concerned with these three elementary operations, with a leitmotiv: we wish that they preserve topological properties of the object.

Topology is a quite recent field of geometry: although isolated observations were made quite early (Euler's formula was already known by Descartes, in the 17th century), this field really emerged in the middle of the 19th century (see [45, p. 235], [98]). Topology leaves out the metric and projective characteristics of the objects, such as distance, area, angle, alignment; the topological properties are those which can be derived merely from the notions of neighborhood and continuous map. An important notion is the notion of homeomorphism (bijective bicontinuous map): in topology, two homeomorphic objects (etymologically, which have the same shape) will be regarded as identical.

We will thus consider decompositions, deformations and shortenings which maintain and/or give prominence to topological properties.

To decompose an object means to subdivide it into simpler objects. We will for instance decompose a surface into surfaces which are topologically simpler, and decompose the tridimensional space into tetrahedra in such a way that they fit a given shape. Such a decomposition enables to understand the topology of the object.

To deform an object into another is to find a continuous geometric transformation between these objects. We will compute deformations which preserve the topology between two embeddings of a given graph in the plane.

To shorten an object means to transform it into an object which has a "size" as small as possible. We will shorten, by deformations, curves drawn on surfaces and graphs embedded in the plane.

Our works are of algorithmic nature: the goal is to effectively compute such operations on geometric objects. They are quite theoretic, but are motivated by the manipulation of geometric objects, of increasing complexity, by computers. One immediately thinks about tridimensional models which proliferate on the Internet and in video games, but we should also mention the domain of computer-aided design (CAD), whose industrial impacts are very important, biology, because the knowledge of the shape of a molecule is a crucial step towards the understanding of its function [131], or robotics, where the movements of the objects must be planned carefully [87]. Decomposing a mesh, deforming a polyhedral model into another one, computing shortest paths, are problems which occur in the real world.

Shortening of curves and decomposition of surfaces

Our main contribution is concerned with the shortening of curves on surfaces while preserving topological properties of these curves. It belongs to the domain of *computational topology*. This expression appeared for the first time, to our knowledge, in a paper by Vegter and Yap in 1990 [154]. It is a field of *computational geometry* whose purpose is "to identify and formalize topological questions in computer applications and to study algorithms for topological problems" [51] (see also [11, 153]). In other words, it is sometimes justified and fruitful to study topological questions independantly from any geometry, with a specific vocabulary and adequate techniques (borrowed especially to algebraic topology and combinatorial topology), in order to put them back afterwards in their geometric framework.

This constatation might be not new, but it is at least formalized and justified with the apparition of this domain. Topology has always had an important place in the field of computational geometry. The study of line arrangements [80] leads naturally to the theory of oriented matroids [17], which are a natural topological axiomatization. The *topological methods* [159] (homology, fixed-points and equipartition theorems) enable to solve questions arising from discrete and computational geometry. The problem of embedding a space into another also has a deep topological part: graph embeddings, knot theory and decidability of the existence of homeomorphisms. The importance of topology is strengthened nowadays, notably because of the applications (geometric modeling, meshes, biogeometry) [51]. Topology can thus be viewed as a characteristic being more or less present in any problem of computational geometry.

The area of computational topology concerned with this work is the algorithmic study of surfaces and curves drawn on them. Fundamental questions (deformation, homeomorphy, decomposition) arise in applications in computer science and have close relations with topology. The underlying topological problems had never been studied from the algorithmic viewpoint. Important theorems have been found since the beginning of the 20th century, which classify surfaces up to homeomorphism or which give necessary and sufficient conditions so that two curves are homotopic (admit a continuous transformation between them). In these two examples, results dating back to one century are constructive and also enable to effectively compute a homeomorphism between these surfaces or a homotopy between these curves. There remained to give efficient algorithms for these problems and to study precisely their complexity.

Our study is about the *shortening* (we will also use the term *optimization*) of families of curves on surfaces while preserving some of their topological properties. It is thus a specialization of the problem of shortest paths computation, a fundamental problem in combinatorial optimization [137, Chapters 6–9]. More precisely, a graph embedding on a surface being fixed, we are looking for the shortest embedding which is isotopic to that embedding, with fixed vertices; a family of simple, pairwise disjoint cycles being fixed, we are looking for the shortest family of simple, pairwise disjoint cycles whose cycles are homotopic to the cycles of the former family. Our goal is thus to fix the topology and to "improve" as much as possible the geometry of these curves (paths of the graph embedding, or cycles), in the sense that the computation of a shortest curve among the curves which have the same topological properties yields a more canonical representation.

To solve this problem, we will use *decompositions* of the surface into surfaces which are topologically simple. We will give greedy algorithms which allow to *shorten* the curves of these decompositions, while maintaining topological properties of the curves. These algorithms work in a framework where the curves are drawn on the vertex-edge graph of a polyhedral surface. We will show that the resulting curves are, individually (*i.e.*, independantly from the other curves of the decomposition), shortest curves among the curves on the vertex-edge graph which are topologically equivalent. Hence, to *optimize* a curve or a family of curves, it will be sufficient to extend this family into a topological decomposition of the surface, to *shorten* this decomposition, and to remove the curves which were previously added. We will show that this leads to algorithms which are polynomial in their input and in the longest-to-shortest edge ratio of the vertex-edge graph.

The interest of this work in practice is to be able to shorten, not only a single curve on a surface, but also decompositions of the surface into topologically simple surfaces: cutting a surface into a topological disk (a region homeomorphic to a disk) is often the first stage of the *parameterization* process, which aims at creating a correspondence between a surface and a planar region, and which is a recurrent problem in computer graphics.

Other contributions

This thesis contains two other contributions.

The first one, which is an extension of our D.E.A. work, focuses on the *deformation* of graphs embeddings in the plane and, to a smaller extent, on their *shortening*. In contrast to the previous work, we consider here graphs rectilinearly embedded in the plane, and the vertices will move: we wish to create a continuous family of embeddings (isotopy) between two given embeddings. Let us consider a planar graph for which some vertices are fixed in the plane; Tutte's barycentric embedding theorem [152] yields an effective and simple way to compute the embedding which minimizes the sum of the squares of the lengths of the edges among all embeddings satisfying this constraint (*shortening*). By changing certain coefficients, one can in fact create a *deformation* between two graph embeddings. Apart from the study of this deformation process, our work consisted in giving another proof of Tutte's theorem and in refuting its natural generalization in higher dimension. Tutte's theorem is often used in the second stage of the parameterization process: it enables to create a homeomorphism between a topological disk and a planar region. The creation of deformations with the help

of this theorem is motivated by the generation of *morphings* between two objects.

The last contribution of this thesis is concerned with conforming Delaunay triangulations in three dimensions (one should say "tetrahedralizations"). A polyhedral object in the tridimensional space being fixed, we wish to *decompose* the space into simplexes (tetrahedra, triangles, edges, and vertices) which fit the shape of the object, *i.e.*, each face of the object must be a union of simplexes: we shall say that the triangulation *conforms* to the polyhedral object. We additionally impose that this triangulation is the Delaunay triangulation of a finite number of points (containing in particular the vertices of the polyhedral object). We give an algorithm to compute such a conforming Delaunay triangulation where the number of inserted points is quite small in practice. The motivation of this work comes from mesh generation and geometric modeling, where it is often desirable to decompose the space into cells conforming to a geometric object. The interest of Delaunay triangulations is that the shapes of the cells are often reasonable and that they have numerous well-studied and exploited features.

Organization of the document

This dissertation is made of five chapters.

The first three chapters are concerned with the optimization of curves on surfaces and the decomposition of surfaces. The first chapter introduces the main notions of topology and combinatorial topology that we will use. The second chapter presents an overview of the previous works which have a link with our work. The third chapter contains our results on this topic.

The last two chapters are about the two other works included in this thesis: in Chapter 4, we revisit Tutte's barycentric theorem, from the point of view of the deformation of graph embeddings. Finally, in Chapter 5, we describe an algorithm for computing conforming Delaunay triangulations in 3D.

Chapter 1 introduces the notions used in Chapters 2 and 3, which are independant. Each of the chapters 4 and 5 can be read independantly from the remaining part of the thesis.

Chapter 3 extends the results and simplifies the proofs of [40]; it contains, with minor modifications, the paper [41]; it introduces new techniques which are easier to use from the algorithmic point of view. The content of Chapter 4 has been published in [42] and in a preliminary version of this paper. Chapter 5 contains nearly literally the paper [38].

Chapter 1

Topology of surfaces

The purpose of this chapter is to present the main notions and results of topology used in Chapters 2 and 3 of this dissertation. Proofs will thus be omitted in most cases. The reader can choose to skip this chapter and to come back whenever needed, to skim through it, or to read it entirely. The index indicates in boldface the pages of this chapter in which the terms are defined.

We refer the interested reader to [92, 6], which are quite elementary books devoted to this topic, and which we used for the redaction of this chapter; in the absence of any other indication, the proofs of the results presented in this chapter can be found in these books. Related books are also [145] for a description of more diversified topics, [114, 91] about algebraic topology, and [117] for graphs embedded on surfaces.

1.1 Surfaces, curves and graphs embeddings

In this section, we recall the usual notions in topology of curves and surfaces, and also the notion of embedded graph.

1.1.1 Surfaces

A *topological space* is a set X with a collection $\mathcal{O}$ of subsets of X, called *open sets*, satisfying the three following axioms:

- the empty set and X are open;
- any union of open sets is an open set;
- any finite intersection of open sets is open.

There is, in particular, no notion of metric (distance, angle, area) in a topological space. On the other hand, one can, with the notion of *neighborhood* (a neighborhood of $x \in X$ is a set containing an open set containing x), define the notion of continuity, of limit, of continuous map, and so on. The topological spaces considered in this thesis are assumed to be *Hausdorff*, which means that two distinct points have disjoint neighborhoods. If X and Y are two topological

spaces, a map $f : X \to Y$ is a *homeomorphism* if it is continuous, bijective, and if its inverse f^{-1} is also continuous.

A *surface* is a topological space in which each point has a neighborhood homeomorphic to the unit disk $\{(x, y) \in \mathbf{R}^2 \mid x^2 + y^2 < 1\}$ or to the unit half-disk $\{(x, y) \in \mathbf{R}^2 \mid x^2 + y^2 < 1, \quad x \geq 0\}$. The *boundary* of a surface $\mathcal{M}$, denoted by $\partial\mathcal{M}$, is the set of the points of this surface which have no neighborhood homeomorphic to the unit disk. The *interior* of $\mathcal{M}$ is the complementary part of its boundary.

A surface is thus "abstract": the only knowledge we have of it is the neighborhoods of each point. A surface is not necessarily embedded in $\mathbf{R}^3$.

A topological space X is *compact* if any set of open sets whose union is X admits a finite subset whose union is still X. Unless otherwise specified, the surfaces considered here are compact.

1.1.2 Curves, connectivity, and the Jordan–Schönflies theorem

Let X be a topological space. A *path* in X is a continuous map $p : [0, 1] \to X$; its *endpoints* are $p(0)$ and $p(1)$. A path is *simple* if it is one-to-one. A *closed path* in X, or *loop*, is a path whose endpoints coincide; they are called the *basepoint* of the loop. A loop ℓ is *simple* if $\ell|_{[0,1)}$ is one-to-one. The *relative interior* of a path $p : [0, 1] \to X$ (possibly closed) is the map $p|_{(0,1)}$.

An *infinite path* in X is a continuous map $p : \mathbf{R} \to X$; it is simple if it is one-to-one; its *relative interior* is the infinite path itself.

A *cycle* is a continuous map $\gamma : S^1 \to X$ (where S^1 denotes the unit circle). A cycle differs from a loop because a loop has (equal) endpoints; on a cycle, no point is distinguished. A cycle γ is *simple* if γ is one-to-one. The relative interior of a cycle is the cycle itself. It is sometimes convenient to view a cycle as an infinite path $\gamma : \mathbf{R} \to X$ such that $\gamma(.) = \gamma(1 + .)$. If γ is a cycle (considered as a map from $\mathbf{R}$ into X), the *loops associated to* γ are the loops which have the form $\gamma(x + .)|_{[0,1]}$, where $x \in \mathbf{R}$.

A *curve* denotes a path, possibly closed, an infinite path, or a cycle. The *image set*, or *image*, of a curve $c : I \to X$ is the set $c(I) \subseteq X$. We will sometimes, by abuse of notation, identify a curve with its image set.

Let X be a topological space. X is *connected*[1] if, for any points a and b in X, there exists a path in X whose endpoints are a and b. If X is connected, we shall say that a curve c *separates* X if $X \setminus c$ is not connected. The *connected components* of a topological space X are the classes of the equivalence relation on X defined by: a is equivalent to b if there exists a path between a and b.

Theorem 1.1 (Jordan–Schönflies — see [148]) *Let γ be a simple cycle in $\mathbf{R}^2$. Then there exists a homeomorphism of $\mathbf{R}^2$ to $\mathbf{R}^2$ such that the image of γ by this homeomorphism is the unit circle $t \mapsto (\cos(2\pi t), \sin(2\pi t))$.*

Figure 1.1 illustrates this theorem. In particular, any simple cycle in $\mathbf{R}^2$ separates $\mathbf{R}^2$ into two connected components (*Jordan curve theorem*).

[1] In this thesis, the only type of connectivity considered is connectivity by arcs.

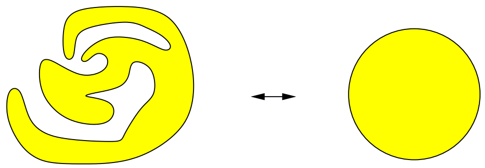


Figure 1.1: Illustration of the Jordan–Schönflies theorem.

1.1.3 Graphs and embeddings of graphs

A *graph* $G = (V, E)$ is the data of a set of *vertices* V and of a set of *edges* E, each edge having two (possibly equal) vertices which are its *endpoints*. There are numerous variations on the notion of graph:

- the graph can be *oriented* (each edge e has a source vertex and a target vertex) or not (the endpoints of each edge are not distinguished);
- *multiple edges* can be allowed or not (that is, the fact that several edges can have the same endpoints);
- *loops* (that is, edges whose two endpoints are equal) can be allowed or not;
- the graph can be *finite* (that is, V and E are finite sets) or not.

From this notion of graph, one defines the notions of an edge *incident* to a vertex, of two *adjacent* vertices, of *degree* of a vertex, of *connectivity* of the graph, of *path*, and of *closed path*. We will assume that the reader is familiar with all these notions, and refer for example to [44, Section 5.4] or [20]. Let us mention that, in this dissertation, multiple edges and loops will be allowed unless otherwise specified, and that the graphs considered will be finite except in very few cases.

We now introduce the notion of *embedding* of a graph $G = (V, E)$ in a topological space X, which is a way to represent G with points for its vertices and paths for its edges. More precisely, it is the data of two maps:

- Γ_V which associates to each vertex of G a point of X;
- Γ_E which associates to each edge e of G a path in X between the images by Γ_V of the endpoints of e,

in such a way that:

- the map Γ_V is one-to-one (two distinct vertices are sent to different points of X);
- for each edge e, the relative interior of $\Gamma_E(e)$ is one-to-one (the image of an edge is a simple path, except possibly at its endpoints);
- for all distinct edges e and e', the relative interiors of $\Gamma_E(e)$ and $\Gamma_E(e')$ are disjoint (two edges cannot cross);

- for each edge e and for each vertex v, the relative interior of $\Gamma_E(e)$ does not meet $\Gamma_V(v)$ (no edge passes through a vertex).

The *faces* of a graph embedding are the connected components of the complementary part of the image of the vertices and edges of this graph. A graph is said to be *planar* if it admits an embedding in $\mathbf{R}^2$.

1.2 Polyhedral surfaces

1.2.1 Definition and properties

We will give a way to build surfaces by gluing together polygons along their edges.

Let us consider a topological space made of a disjoint union P of simple polygons (*i.e.*, without holes, *i.e.*, homeomorphic to the closed unit disk). Let us choose an orientation of the edges of the polygons of P, and let $A = (A_1, \ldots, A_n)$ be a partition of these edges. One can create the quotient space P' obtained from P by *identification*, or *gluing*, of the edges of A_i for all i (following the prescribed orientations of the edges). This set can be provided with a topology [92, p. 116]. We shall say that the n *edges* of P' are the sets $A_1, \ldots, A_n$. This identification of edges naturally induces an identification on the vertices of the polygons of P: two vertices are identified on the surface if they are the sources, or the targets, of two edges of P belonging to a same set A_i.

The topological space P' is not necessarily a surface, since the identifications can create points which have no neighborhood homeomorphic to the unit disk or half-disk. The following definition aims at excluding such cases.

A *polyhedral surface* is a topological space obtained by identifications of edges and vertices of a finite number of simple polygons (called *faces* of the polyhedral surface), in such a way that:

- the resulting space is a surface;
- two distinct edges or vertices of a given polygon are not identified on the surface;
- the intersection of two distinct polygons is either empty, a common vertex, or a common edge.

See Figure 1.2.

This in particular implies that:

- an edge is incident to two polygons (or one single polygon if it is on the boundary of the surface);
- for each vertex v, the polygons having v as a vertex can be ordered in a cyclic or linear sequence $P_1, P_2, \ldots, P_k, P_1$ or $P_1, P_2, \ldots, P_k$, so that two consecutive polygons in this sequence are identified along an edge incident to v (Figure 1.3).

A polyhedral surface is a surface. The converse is non-obvious:

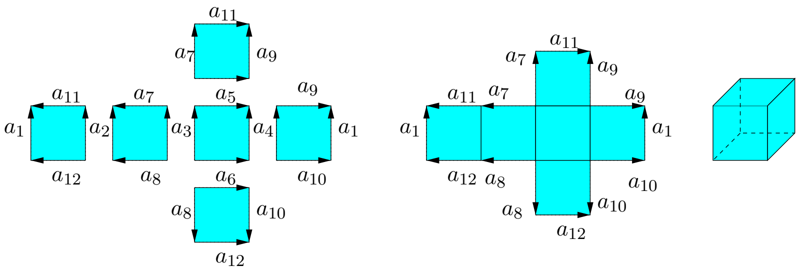


Figure 1.2: Construction of a polyhedral surface (a cube) with six squares.

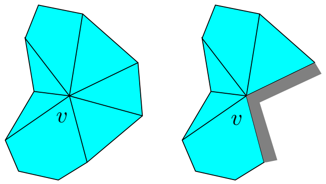


Figure 1.3: The polygons incident to some vertex v can be ordered cyclically (if v is a point interior to the surface) or linearly (if v is a point on the boundary).

Theorem 1.2 (see [148] or [57]) *Any (compact) surface is homeomorphic to a polyhedral surface.*

This amounts to saying that, on any compact surface, one can embed a finite graph $G = (V, E)$ such that:

- each face of G is homeomorphic to the open unit disk, and incident to at least three edges;
- the boundary of each face is made of a cycle which alternates with distinct vertices and edges;
- the closure of two distinct faces are disjoint, or share exactly one vertex or one edge.

This theorem implies that, in order to study the topology of surfaces, it is sufficient to restrict ourselves to polyhedral surfaces.

The *vertex-edge graph* of a polyhedral surface is the graph induced by the vertices and edges of the polyhedral surface. A *combinatorial surface*, or *triangulated surface*, is a polyhedral surface whose faces are triangles. A homeomorphism between a triangulated surface and a surface $\mathcal{M}$ yields a *triangulation* of $\mathcal{M}$. Since any polygon can be triangulated, any surface admits a triangulation.

1.2.2 Cutting surfaces

The Jordan–Schönflies theorem enables to define rigorously the *cutting* of a surface $\mathcal{M}$ along a simple curve c. The key is the following lemma:

Lemma 1.3 *Let $\mathcal{M}$ be a surface, and let c be a simple, possibly closed path, whose relative interior is in the interior of $\mathcal{M}$. Then there exists a triangulation T of $\mathcal{M}$ such that the image of c is a (simple) path in the vertex-edge graph of T.*

To prove the existence of T is quite easy if we can find a triangulation of $\mathcal{M}$ whose vertex-edge graph intersects c at a finite number of points. Indeed, it is then sufficient to subdivide each triangle t along each piece of c which crosses t (such a piece separates t by the Jordan curve theorem). But the difficulty is that one must get rid of the possibly infinite number of intersections. This lemma can be shown by adapting the proofs of the fact that any compact surface is triangulable (for example the proof in [57]).

Epstein [67, Appendix] states a theorem which is a bit different, but his method enables in particular to prove this result. Let us fix a triangulation of $\mathcal{M}$. The idea is to take real numbers $0 = a_0 < a_1 < \ldots < a_{n-1} < a_n = 1$ such that, for each i, $c|_{[a_i,a_{i+1}]}$ is included in a disk, and to prove by induction on i, using the Jordan–Schönflies theorem, that there exists a homeomorphism of $\mathcal{M}$ which sends $c|_{[0,a_i]}$ into a piecewise linear curve (with respect to a given initial triangulation).

This lemma enables in particular to define the topological space obtained by *cutting* $\mathcal{M}$ along a simple path c: let T be a triangulation given by Lemma 1.3; one defines this space as the combinatorial surface obtained from T by removing the identifications of the edges contained in c. Iterating the process, it is possible to cut the surface along a graph embedding whose edges have their relative interiors in the interior of $\mathcal{M}$.

In Chapter 3, we will repeatedly use the cutting of a surface along a simple curve or along several simple, pairwise disjoint curves.

1.3 Classification and decomposition of surfaces

1.3.1 Topological invariants

We introduce here quantities which can be computed for a polyhedral surface S, but which are *topological invariants* in the sense that, if S and S' are homeomorphic surfaces, then these quantities are equal for S and S'. Two trivial examples of topological invariants of a polyhedral surface S are the number of its connected components and the number of its boundaries (more precisely, the number of connected components of its boundary): it is obvious that these quantities depend in fact solely on the underlying surface $\mathcal{M}$.

Let S be a polyhedral surface. The boundary of a polygon of S is a cycle of edges, which can be oriented in two different ways. Choosing an orientation of this polygon yields an orientation of the edges of its boundaries. S is *orientable* if it is possible to orient each of its polygons in such a way that each edge incident to two polygons is oriented in opposite ways by these two polygons (Figure 1.4); if this is possible, such a choice of orientation is called an *orientation* of the surface, which is said to be *oriented*.

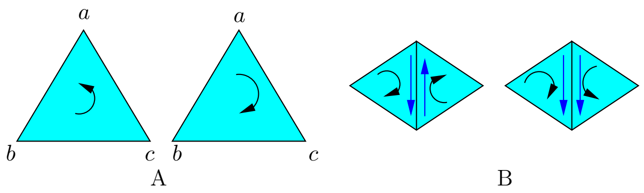


Figure 1.4: Definition of the orientability of a surface. A: the two possible orientations for a triangle abc. B: On the left, the edge incident to the triangles is oriented in two opposite ways. On the right, the orientations of the edge induced by the two triangles are the same.

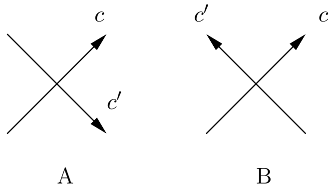


Figure 1.5: Two orientations of crossings between two curves c and c'. A: c' pierces c from left to right. B: c' pierces c from right to left.

Let c and c' be two curves on a surface, having for intersection a unique point p. We shall say that c and c' *cross* if there exists a neighborhood U of p and a homeomorphism of this neighborhood into the unit disk which sends $c \cap U$ and $c' \cap U$ into two secant line segments. On an oriented surface, one can define without ambiguity the orientation of a crossing between c and c' (Figure 1.5).

Let S be a polyhedral surface with V vertices, E edges and F faces. The *Euler characteristic* of S, denoted by $\chi(S)$, is the signed integer $V - E + F$.

Theorem 1.4 *Let S and S' be two polyhedral surfaces which are homeomorphic. Then:*

- *S is orientable if and only if S' is orientable;*
- *$\chi(S) = \chi(S')$.*

In other words, the Euler characteristic and the orientability are topological invariants. Let us give an idea of the proof. A polyhedral surface being fixed, one can *subdivide* one or several faces, by decomposing them into smaller faces. It is easy to see that such a subdivision changes neither the orientability of the surface nor its Euler characteristic (Figure 1.6). To prove the previous result, it is thus sufficient to prove that S and S' can be subdivided in such a way that these refinements are the same. It is possible to represent S and S' as embedded graphs G and G' on a same surface $\mathcal{M}$ (see the previous section). If G and G' intersect at a finite number of points, it seems plausible that it is possible to find a common subdivision for G and G'. What makes the proof difficult is that, *a priori*, G and

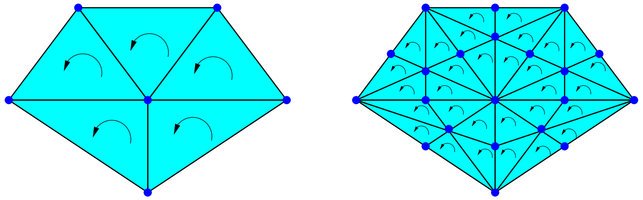


Figure 1.6: A subdivision of the surface changes neither its orientability nor its Euler characteristic.

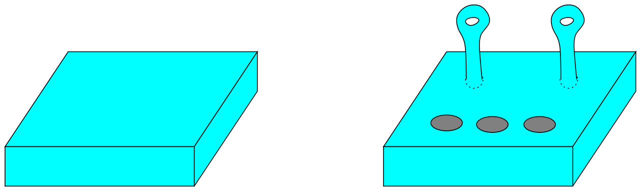


Figure 1.7: Any compact surface is homeomorphic to a sphere (here, a parallelepiped) to which are glued g handles and removed b open disks. Here, $g = 2$ and $b = 3$.

G' can have an infinite number of intersection points. It is nevertheless possible to show that G and G' have combinatorially equivalent subdivisions.

All surfaces considered in this thesis are orientable, and we often omit this restriction.

1.3.2 Classification theorem and polygonal schemata

1.3.2.1 Classification theorem

In the previous section, we have given topological invariants: two polyhedral surfaces which have different invariants cannot be homeomorphic. Conversely, are two surfaces with the same invariants homeomorphic? The following fundamental theorem classifies the surfaces up to homeomorphism:

Theorem 1.5 (Classification theorem for surfaces) *Let $\mathcal{M}$ be a compact, connected, orientable surface. There exist two unique non-negative integers g and b such that $\mathcal{M}$ is homeomorphic to a sphere to which are glued g handles and removed b pairwise disjoint disks. In addition, $\chi(\mathcal{M}) = 2 - 2g - b$.*

Figure 1.7 illustrates this theorem.

In particular, the number of boundaries b and the Euler characteristic characterize the homeomorphy class of a compact, connected, orientable surface. The integer g is thus a topological invariant, called the *genus* of the surface. Another consequence of this theorem is that the compact, orientable surfaces are all embeddable in $\mathbf{R}^3$.

Here are the values of g and b for a few surfaces (Figure 1.8):

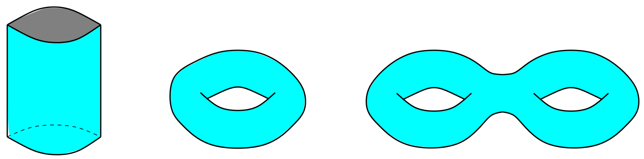


Figure 1.8: From left to right: a cylinder, a torus, and a double-torus.

- for a sphere, $g = 0$ and $b = 0$;
- for a disk, $g = 0$ and $b = 1$;
- for a cylinder, $g = 0$ and $b = 2$;
- for a torus, $g = 1$ and $b = 0$;
- for a double-torus, $g = 2$ and $b = 0$.

We will say that a surface is a *topological sphere* (resp. *disk*) if it is homeomorphic to a sphere (resp. to a disk).

1.3.2.2 Sketch of proof and polygonal schemata

We will give the method of the proof, which is instructive. In the course of this description, we will introduce the notions of *polygonal schema*, *reduced polygonal schema*, and *canonical polygonal schema*, which are families of curves decomposing the surface into a disk. The origin of the proof can be found in the paper by Brahana [21]; most textbooks in topology give this proof (see for example [74, Chapter 4]). By Theorem 1.2, we can assume that $\mathcal{M}$ is a polyhedral surface. The first stage of the proof consists in showing that $\mathcal{M}$ admits a polygonal schema, *i.e.*, a *pattern*.

A *polygonal schema* of a connected surface $\mathcal{M}$ is the data of a simple polygon with an identification of its edges (choice of an orientation and partition of the edges), such that we obtain the surface $\mathcal{M}$ by performing these identifications. The interest of a polygonal schema is that it defines a surface in a particularly simple way: it suffices to give the list of the edges on the boundary of the polygon by indicating which identifications have to be done (see Figures 1.9 and 1.10, and also Figure 1.2, middle part). In the case of a surface with boundary, the edges of the polygonal schema corresponding to the edges of the boundary of the surface are identified with no other edge of the schema.

Any graph embedded on $\mathcal{M}$, which has exactly one face and such that this face is a topological disk, gives rise to a polygonal schema of $\mathcal{M}$ by cutting this graph along its edges (see Figures 1.9 and 1.10). By abuse of language, we will also say that such a graph is a polygonal schema of $\mathcal{M}$.

Intuitively, any connected polyhedral surface $\mathcal{M}$ admits a polygonal schema, built by removing as many edge identifications as possible while maintaining the connectivity of the surface. Formally, let us consider the *dual graph* $G^* = (V^*, E^*)$ of the vertex-edge graph $G = (V, E)$ of $\mathcal{M}$: it is the graph whose vertices are the

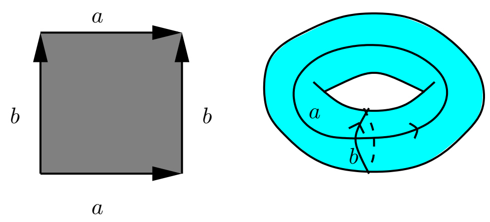


Figure 1.9: On the left, a polygonal schema of the torus. On the right, the surface obtained after identification of the edges on the boundary of the schema. The polygonal schema can be described by the word $ab\bar{a}\bar{b}$ (in counterclockwise order).

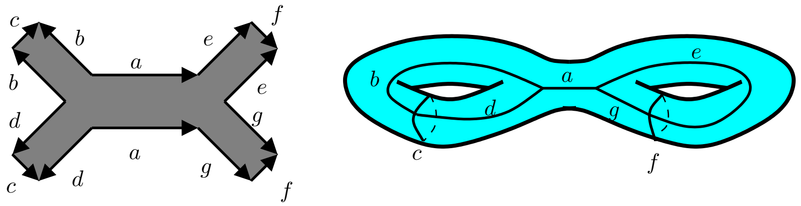


Figure 1.10: On the left, a polygonal schema of the double-torus, of the form $\bar{a}b\bar{c}\bar{b}dc\bar{d}agf\bar{g}e\bar{f}\bar{e}$. On the right, the surface obtained after identification of the edges on the boundary of the schema.

faces of $\mathcal{M}$ and such that two vertices are linked with an edge if the corresponding faces of the polyhedral surface are adjacent (Figure 1.11). The dual graph of $\mathcal{M}$ is connected; if one computes a *spanning tree* T^* of this graph and removes all identifications of edges on the polyhedral surface except the ones corresponding to edges of T^*, one gets a polygonal schema (Figure 1.11D).

The remaining part of the proof consists in transforming the polygonal schema to prove that the surface admits a polygonal schema of a very particular form. These elementary operations are *cut-and-paste* operations: one cuts the polygonal schema into two pieces, which are then glued along edges which must be identified on the surface, so that this operation does not change the surface itself. See Figure 1.12. With such operations, one proves that the surface admits a *reduced polygonal schema*, then a *canonical polygonal schema*.

Let us assume that $\mathcal{M}$ is boundaryless; a *reduced polygonal schema* of $\mathcal{M}$ is a polygonal schema of $\mathcal{M}$ in which all vertices of the boundary of the schema are identified into a single point on the surface[2]. After identification of the edges of the polygonal schema, one obtains a set of closed paths which are simple and pairwise disjoint, except at some vertex v_0, where they all meet; the complementary part in $\mathcal{M}$ of these closed paths yields a topological disk. Such a set of loops is called

[2]Let us emphasize that this notion is independent from any triangulation of $\mathcal{M}$. In particular, the paths of a polygonal schema are not necessarily included in the vertex-edge graph of $\mathcal{M}$, if $\mathcal{M}$ is polyhedral.

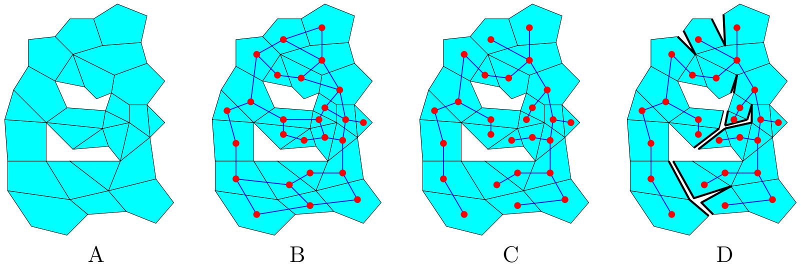


Figure 1.11: The creation of a polygonal schema of a polyhedral surface (here, with three boundaries and with genus zero). A: The surface. B: The dual graph G^* of the vertex-edge graph. C: A spanning tree T^* of this graph. D: Cutting of the surface along the dual edges of $G^* \setminus T^*$. The edges in bold lines must be paiwise identified to get again the surface.

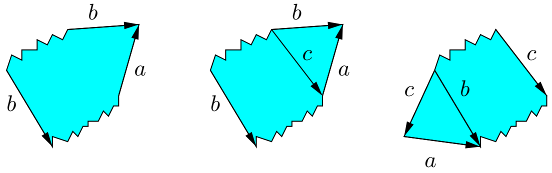


Figure 1.12: Surgery of a polygonal schema.

a *fundamental system of loops*, and v_0 is the *basepoint* of this system.

There is no standard definition of a reduced polygonal schema on a surface with boundary. We shall take the following definition: if $\mathcal{M}$ has at least one boundary, a *reduced schema* of $\mathcal{M}$ is a polygonal schema such that, after identification of the edges and vertices, each boundary of $\mathcal{M}$ contains exactly one vertex of the schema, and such that there is at most one vertex outside the boundary of $\mathcal{M}$. Any polygonal schema can be transformed into a reduced polygonal schema.

A *canonical polygonal schema* of a surface $\mathcal{M}$ is a polygonal schema in which the boundary of the polygon has the form

$$a_1 b_1 \bar{a}_1 \bar{b}_1 \ \ldots \ a_g b_g \bar{a}_g \bar{b}_g \ c_1 d_1 \bar{c}_1 \ \ldots \ c_b d_b \bar{c}_b \quad \text{or} \quad a\bar{a}, \tag{1.1}$$

where $(g, b) \in \mathbf{N}^2 \setminus \{(0, 0)\}$. (The edges d_i are identified with no other edge and thus correspond to the boundary of the surface.) The particular case $a\bar{a}$ corresponds to the case where the surface is the sphere. See Figure 1.13 for an example of a canonical schema of a sphere with 4 boundaries, and Figure 1.14 for an example on the double-torus. It is possible to prove that any canonical schema is reduced, and that any reduced schema can be transformed into a canonical schema.

Hence, any compact, connected, orientable surface is homeomorphic to the surface obtained by identifying the edges of a polygonal schema defined by Formula (1.1), for some $(g, b) \in \mathbf{N}^2$ (by convention, $g = b = 0$ if the schema has the form $a\bar{a}$). One proves that the Euler characteristic of such a surface is $2 - 2g - b$,

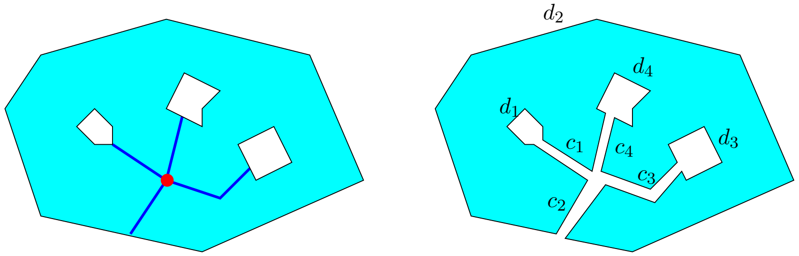


Figure 1.13: A polygonal schema on a sphere with 4 boundaries.

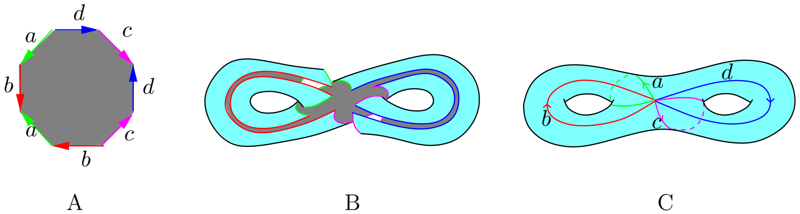


Figure 1.14: A: A canonical polygonal schema of the double-torus. B: The identification of the edges of the schema. C: We get a double-torus containing a set of loops which are simple and pairwise disjoint except at a common point, such that cutting the surface yields a topological disk: a *fundamental system of loops* of the double-torus.

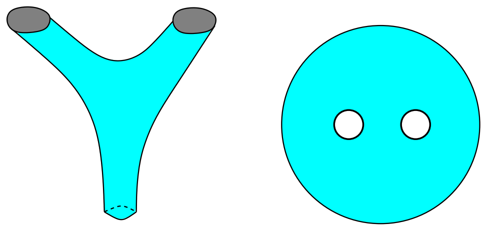


Figure 1.15: Two representations of a pair of pants.

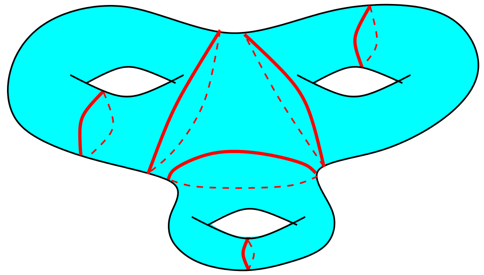


Figure 1.16: A pants decomposition of a triple-torus.

and that a sphere with g handles glued and b disjoint open disks removed admits a canonical polygonal schema defined by Formula (1.1).

Given a compact, orientable surface having k connected components, its *genus* g is the sum of the generi of its connected components, and its *number of boundaries* b is the sum of the numbers of boundaries of its connected components. Using the previous theorem, we see that the Euler characteristic of such a surface is $2k - 2g - b$.

1.3.3 Pants decompositions

We now introduce another type of surface decomposition, which will be used in Chapter 3. A *pair of pants* is a surface with genus 0 and with 3 boundaries (Figure 1.15). A *pants decomposition* (see [90] or [149, p. 269]) of $\mathcal{M}$ is a set of simple, pairwise disjoint cycles such that cutting $\mathcal{M}$ along these cycles yields pairs of pants (Figure 1.16). Any compact, connected, orientable surface whose Euler characteristic is negative (which excludes the sphere, disk, cylinder and torus) admits a pants decomposition. One can create a pants decomposition of a surface by cutting iteratively this surface along an *essential cycle*, *i.e.*, a simple cycle such that no connected component of the surface cut along this cycle is a disk or a cylinder. A pants decomposition is made of $3g + b - 3$ cycles.

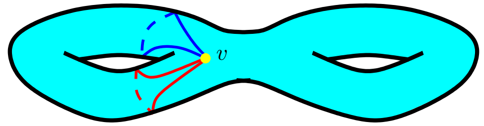


Figure 1.17: The two cycles on this double-torus are freely homotopic, but not homotopic when considered as loops with basepoint v.

1.4 Homotopy, isotopy, and universal covering space

The notions of homotopy and isotopy enable to determine if it is possible to *deform* one curve into another one. The universal covering space is a space which enables to make homotopy computations.

1.4.1 Homotopy

Let p and q be two paths on a surface $\mathcal{M}$. The *concatenation* of p and q, denoted by $p.q$, is the path defined by:

- $(p.q)(t) = p(2t)$, if $0 \leq t \leq 1/2$;
- $(p.q)(t) = q(2t - 1)$, if $1/2 \leq t \leq 1$.

A *reparameterization* of a path p is a path of the form $p \circ \varphi$, where $\varphi : [0, 1] \to [0, 1]$ is bijective and increasing. If the paths are considered up to reparameterization, the concatenation is associative. The *inverse* of a path p, denoted by p^{-1}, is the map $t \mapsto p(1 - t)$.

The notion of homotopy corresponds to the intuitive idea of deformation. Two paths p and q on $\mathcal{M}$, having both a and b as endpoints, are *homotopic* if there exists a continuous family of paths whose endpoints are a and b between p and q. More formally, a *homotopy* between p and q is a continuous map $h : [0, 1] \times [0, 1] \to \mathcal{M}$ such that $h(0, .) = p$, $h(1, .) = q$, $h(., 0) = a$, and $h(., 1) = b$. This definition applies in particular to the case of loops.

Two cycles γ and γ' are *homotopic* if there is a continuous family of cycles joining γ to γ', *i.e.*, a continuous map $h : [0, 1] \times S^1 \to \mathcal{M}$ such that $h(0, .) = \gamma$ and $h(1, .) = \gamma'$.

Homotopy of cycles (also called *free homotopy*) and homotopy of loops (also called *homotopy with fixed basepoint*) are two equivalence relations which are quite different, see Figure 1.17 for an example. A loop (resp. a cycle) is *contractible* if it is homotopic to a constant loop (resp. to a constant cycle).

Let v_0 be a point of $\mathcal{M}$. The relation "is homotopic to" partitions the set of loops with basepoint v_0 into *homotopy classes*. Let us denote by $[\ell]$ the homotopy class of a loop ℓ. The set of homotopy classes can be equipped with the law "." defined by $[\ell].[\ell'] = [\ell.\ell']$, and, with this law, the set of homotopy classes of loops with basepoint v_0 is a group, called the *fundamental group* of $(\mathcal{M}, v_0)$ and denoted by $\pi_1(\mathcal{M}, v_0)$ or more concisely $\pi_1(\mathcal{M})$, whose unit element is the class of contractible loops.

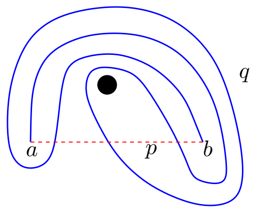


Figure 1.18: Two homotopic simple paths are not necessarily isotopic. Here, the surface considered is the plane minus one disk; paths p (in dashed lines) and q (in solid lines), which have a and b as endpoints, are simple, homotopic, but not isotopic.

In particular, the fundamental group of the disk or the sphere is trivial: two paths having the same endpoints are homotopic. The fundamental group of the cylinder is $\mathbf{Z}$ (the homotopy class of a loop is the same as the signed number of times it winds around the hole), and the fundamental group of the torus is $\mathbf{Z}^2$. (The fundamental group of the pair of pants is the free group with two generators.)

Free homotopy can be interpreted within the fundamental group, as follows. It can be proved easily that two cycles γ and γ' are homotopic if and only if, for any two loops ℓ and ℓ' associated repectively to γ and γ', there exists a path β joining $\ell(0)$ to $\ell'(0)$ such that the loop $\beta^{-1}.\ell.\beta.\ell'^{-1}$ is contractible. Two loops ℓ and ℓ' with basepoint v_0 are thus homotopic *as cycles* if $[\ell]$ and $[\ell']$ are *conjugates* in the fundamental group.

1.4.2 Isotopy

An *isotopy* h between two simple curves (paths, loops, or cycles) is a homotopy such that the curves remain simple during the whole deformation: for each $t \in [0, 1]$, $h(t, .)$ must be a simple curve (path, loop, or cycle).

For simple curves, the notions of homotopy and isotopy are close but not identical. Two homotopic simple paths are not necessarily isotopic, as proved by a counterexample due to Feustel [71], drawn on Figure 1.18. Nevertheless, there are cases where two homotopic simple paths must be isotopic:

Lemma 1.6 *Let D be a closed disk, and c_1 and c_2 be two simple paths in D with the same endpoints, and which intersect the boundary of D precisely at their endpoints. Then c_1 and c_2 are isotopic in the interior of D augmented with tbe endpoints of c_1.*

This result is a simple consequence of the Jordan–Schönflies theorem (and is even trivial if the relative interiors of c_1 and c_2 are disjoint).

For two simple loops, we have the following theorem by Epstein [67, Theorem 4.1]:

Theorem 1.7 *Let ℓ_1 and ℓ_2 be two simple, homotopic, non-contractible loops on an orientable surface. Then ℓ_1 and ℓ_2 are isotopic.*

(This theorem is false if the loops are contractible, because two simple cycles, bounding a disk and oriented in opposite directions, are not isotopic. The orientability of the surface is also necessary.)

1.4.3 A few simple results

The same paper [67] by Epstein contains several results related to homotopy of curves on surfaces, which will be used in Chapter 3 and deserve to be mentioned here. They are quite intuitive. Two of these results describe the surfaces obtained by cutting along some loops:

Theorem 1.8 (Epstein [67, Theorem 1.7]) *Let ℓ be a simple, contractible loop on a surface. Then ℓ is the boundary of a topological disk.*

Lemma 1.9 (Epstein [67, Lemma 2.4]) *Let γ and γ' be two simple, pairwise disjoint, homotopic but non-contractible cycles. Then γ and γ' bound a cylinder.*

A loop ℓ and a signed integer k being fixed, the kth power of ℓ is the concatenation of $|k|$ times the loop ℓ (if $k \geq 0$) or ℓ^{-1} (if $k < 0$).

Lemma 1.10 (Epstein [67, Lemma 4.3]) *Let ℓ be a non-contractible loop on an orientable surface, and $k \geq 1$. Then the kth power of ℓ is non-contractible.*

More precisely:

Theorem 1.11 (Epstein [67, Theorem 4.2 and Lemma 4.3]) *Let ℓ be a non-contractible loop on an orientable surface. Let $k \geq 2$. Then there exists no simple loop homotopic to the kth power of ℓ.*

1.4.4 Universal covering space

Informally, the *universal covering space* of a surface $\mathcal{M}$ is a surface $\widetilde{\mathcal{M}}$ such that it is possible to *lift* curves from $\mathcal{M}$ into $\widetilde{\mathcal{M}}$ and such that two paths are homotopic in $\mathcal{M}$ if and only if these paths can be lifted to paths which have the same endpoints in $\widetilde{\mathcal{M}}$. The universal covering space is thus a tool to compute homotopy.

1.4.4.1 Examples

Let $\mathcal{M}$ be the annulus (which is homeomorphic to a cylinder) depicted on Figure 1.19A. If this annulus is cut along the dashed line segment, we obtain a rectangle; if we glue together infinitely many copies of this rectangle, we obtain an "infinite strip", depicted on Figure 1.19B, which will be denoted by $\widetilde{\mathcal{M}}$. There is a natural "projection" π from $\widetilde{\mathcal{M}}$ onto $\mathcal{M}$, such that a path in $\mathcal{M}$ can be *lifted* to a path (in fact, infinitely many paths) in $\widetilde{\mathcal{M}}$. We see that two paths c and c' are homotopic in $\mathcal{M}$ if two lifts of c and c' starting at the same vertex of $\widetilde{\mathcal{M}}$ have the same targets. The two loops represented on the figure are not homotopic, because one of them is contractible (its lifts in $\widetilde{\mathcal{M}}$ are closed curves), and the other one is non-contractible (its lifts are not closed).

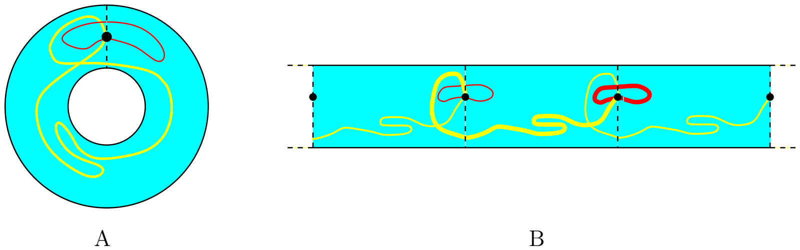


Figure 1.19: A: An annulus $\mathcal{M}$ and two loops with the same basepoint (in black). B: Its universal covering space $\widetilde{\mathcal{M}}$, with lifts of these loops. The vertices of $\widetilde{\mathcal{M}}$ in black are the lifts of the basepoint.

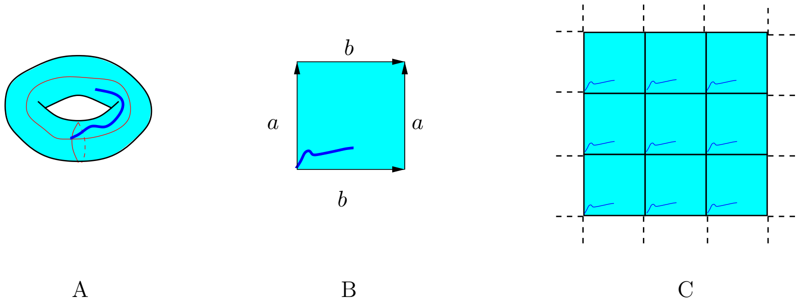


Figure 1.20: A: A torus. B: A polygonal schema of the torus. C: The universal covering space of the torus.

The same figure can be drawn for the torus (Figure 1.20A). If this torus $\mathcal{M}$ is viewed as a canonical polygonal schema (Figure 1.20B), a square whose opposite sides will be identified to obtain $\mathcal{M}$, its universal covering space consists of infinitely many copies of this polygonal schema glued along the sides of the schema: hence, it is the plane (Figure 1.20C).

1.4.4.2 Definition and properties

Precisely, a *universal covering space* of a connected surface $\mathcal{M}$ is the data of a pair $(\widetilde{\mathcal{M}}, \pi)$, where:

- $\widetilde{\mathcal{M}}$ is a (possibly non-compact) surface which is *simply connected*, *i.e.*, every loop in $\widetilde{\mathcal{M}}$ is contractible;
- π is a continuous map from $\widetilde{\mathcal{M}}$ onto $\mathcal{M}$, called *projection*, which is a *local homeomorphism*: any point x of $\mathcal{M}$ has an open, connected neighborhood U such that $\pi^{-1}(U)$ is a disjoint union of open sets $(U_i)_{i\in I}$ and $\pi|_{U_i} : U_i \to U$ is a homeomorphism.

It is known that each connected surface has a universal covering space (see [114, Chapter 5]). On the other hand, two universal covering spaces are isomorphic

(that is, there is a homeomorphism between them which "projects" to the identity map). This enables to speak without ambiguity of *the* universal covering space of a surface $\mathcal{M}$.

A *lift* of a path p is a path $\widetilde{p}$ in $\widetilde{\mathcal{M}}$ such that $\pi \circ \widetilde{p} = p$. An *automorphism* of $(\widetilde{\mathcal{M}}, \pi)$ is a homeomorphism $h : \widetilde{\mathcal{M}} \to \widetilde{\mathcal{M}}$ which preserves the structure of the covering: $\pi \circ h = \pi$.

The main properties of $(\widetilde{\mathcal{M}}, \pi)$ that we will use are:

- the *lift property*: let p be a path in $\mathcal{M}$ whose source is y; let $x \in \pi^{-1}(y)$. Then there exists a unique path $\widetilde{p}$ in $\widetilde{\mathcal{M}}$, whose source is x, such that $\pi \circ \widetilde{p} = p$;
- the *homotopy property*: two paths p_1 and p_2 with the same endpoints are homotopic in $\mathcal{M}$ if and only if they have two lifts $\widetilde{p}_1$ and $\widetilde{p}_2$ sharing the same endpoints in $\widetilde{\mathcal{M}}$;
- the *intersection property*: a path p in $\mathcal{M}$ self-intersects if and only if either a lift of p self-intersects, or two lifts of p intersect.

The notion of lift is defined in a similar way for the other types of curves: a *lift* of an infinite path $p : \mathbf{R} \to \mathcal{M}$ is an infinite path $\widetilde{p}$ such that $\pi \circ \widetilde{p} = p$. A *lift* of a cycle γ (considered as a map from $\mathbf{R}$ into $\mathcal{M}$ such that $\gamma(.) = \gamma(1 + .)$) is an infinite path $\widetilde{\gamma} : \mathbf{R} \to \widetilde{\mathcal{M}}$ such that $\pi \circ \widetilde{\gamma} = \gamma$.

1.4.4.3 A construction of the universal covering space

Let $\mathcal{M}$ be a connected polyhedral surface. It is possible [92, pp. 209–212] to describe in full generality a construction of the universal covering space of $\mathcal{M}$. We will only give an idea of the construction in the case where the polyhedral surface $\mathcal{M}$ has no vertex in its interior (following the description by Hershberger and Snoeyink [94]). Schipper [135] and Dey and Schipper [54] have given algorithms to build the universal covering space in the case of boundaryless surfaces.

The algorithm maintains a portion S of the universal covering space of $\mathcal{M}$ which has been built, which is a topological disk. S is made of copies of polygons of $\mathcal{M}$ (in other words, the image, by π, of each point of S is known). The edges of S can be of two types: there are *active* edges, beyond which the construction of the universal covering space needs to be proceeded, and *inactive* edges, which are still incident to two polygons in S (or to one polygon, if they project to the boundary of $\mathcal{M}$). Initially, S consists of a copy of one single polygon of $\mathcal{M}$, and all the edges of this copy are active, except the edges which project to the boundary of $\mathcal{M}$. The following process is iterated:

- let p be a polygon of S with an active edge a;
- let p' be a copy of the polygon of $\mathcal{M}$ adjacent to $\pi(p)$ through $\pi(a)$;
- we glue p' to p via the edge a;
- the edges of p' which do not project to the boundary of the surface are made active, except a, which is made inactive.

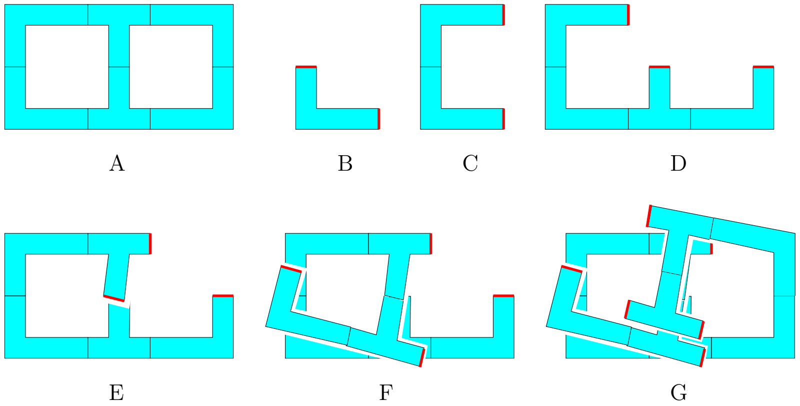


Figure 1.21: The construction of the universal covering space of a pair of pants (a polygon with two holes of the plane). The active edges are depicted in bold lines. A: The polyhedral surface itself. B: Initialization of the construction with one single polygon. C: After one elementary step, S consists of two polygons. D: A few stages later. E, F, G: Continuation of the process. Several polygons of S project to a same polygon of $\mathcal{M}$.

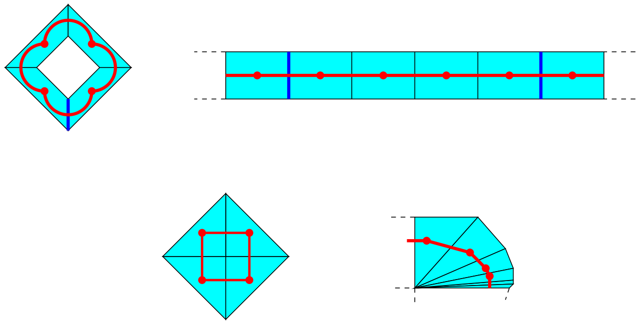


Figure 1.22: On top, the construction of the universal covering space of an annulus represented by a triangulation whose vertices are all on the boundary. The dual graph of the triangulation is represented. At the bottom, an example showing that the same construction will fail if there are vertices of the triangulation in the interior of $\mathcal{M}$: in this example, the algorithm does not construct the universal covering space, but an object containing a vertex incident to infinitely many triangles.

Figure 1.21 presents an example of this construction. This process is, of course, infinite (except in the case of the sphere or the disk). To show that the map π is a local homeomorphism, the acyclicity of the dual graph G^* of the vertex-edge graph G of $\mathcal{M}$ is crucial (this comes from the fact that the vertices of G are on the boundary of the surface), see Figure 1.22. On the other hand, it is clear that the space built by the algorithm is simply connected, because the dual graph of its vertex-edge graph is a tree.

1.4.4.4 Properties of curves in the universal covering space

The Jordan curve theorem gives a separation property for cycles in the plane. The following lemma, which will be used in Chapter 3, extends this property to the case where the space is the universal covering space of a surface.

Lemma 1.12 *Let $\mathcal{M}$ be a surface and $(\widetilde{\mathcal{M}}, \pi)$ be its universal covering space. Let c be a simple curve on $\mathcal{M}$ which is:*

- *either a path which intersects $\partial\mathcal{M}$ exactly at its endpoints,*
- *or a cycle in the interior of $\mathcal{M}$.*

Then each lift $\widetilde{c}$ of c in $\widetilde{\mathcal{M}}$ separates $\widetilde{\mathcal{M}}$ into two connected components.

Proof. A proof can be found in [24, p. 417]; we give an intuition of a possible proof. We will exhibit a construction of the universal covering space of $\mathcal{M}$ for which it will clearly appear that the lifts of c are separating (we hence implicitely use the uniqueness of the universal covering space up to isomorphism).

Let us cut $\mathcal{M}$ along c, thus obtaining a surface $\mathcal{M}'$. The boundary of $\mathcal{M}'$ is made of pieces coming from the boundary of $\mathcal{M}$ and of other pieces coming from the cutting of $\mathcal{M}$ along c. Let $(\widetilde{\mathcal{M}'}, \pi)$ be the universal covering space of $\mathcal{M}'$. The boundary of $\widetilde{\mathcal{M}'}$ is made of pieces which are lifts of c, and of pieces which are lifts of pieces of boundaries of $\mathcal{M}'$.

For each piece $\widetilde{c}_1$ of boundary of $\widetilde{\mathcal{M}'}$ which is a lift of c, we create a copy $\widetilde{\mathcal{M}'}_{\widetilde{c}_1}$ of $\widetilde{\mathcal{M}'}$. We then glue $\widetilde{c}_1$ to a portion of boundary of $\widetilde{\mathcal{M}'}_{\widetilde{c}_1}$ which has the opposite orientation. We thus obtain a surface whose boundary is also made of pieces which are lifts of c or lifts of pieces of boundaries of $\mathcal{M}'$. We continue the construction, by gluing to this surface new copies of $\widetilde{\mathcal{M}'}$.

The topological space obtained is clearly simply connected (the dual graph of the copies of $\widetilde{\mathcal{M}'}$ glued together is a tree, and each copy is simply connected). It remains to prove that it is a surface which is a covering space of $\mathcal{M}$. Moreover, each lift of c separates this surface into two connected components. □

Chapter 2

Previous works

The main contribution of this thesis, in Chapter 3, is based on the study and the computation of shortest paths on surfaces, within a given homotopy class. In this chapter, we present an overview of the existing related results. We will survey successively:

- the works considering the computation of shortest paths, in a graph and on a surface;
- the results of computational topology related to the problems of homotopy and decomposition of surfaces into polygonal schemata;
- the algorithms concerned with the computation of shortest paths within a given homotopy class;
- the application domains of all these works.

We will situate our work among these results when they appear. Reading this chapter is not necessary to understand the next one.

We have chosen not to describe here a state of the art of the works related to Chapters 4 and 5: it seemed more interesting to describe in detail the related works of one single contribution. However, these two chapters both contain a presentation, much more concise than the present chapter, of their related works.

2.1 Computation of shortest paths

A part of our work is directed towards the *shortening* of curves. In this section, we focus on the problem of computing shortest paths. Our goal here is not to give an exhaustive view of the domain: several chapters of textbooks and papers are devoted to this topic (we refer the interested reader to [137, Chapters 6–9] and [115]). We simply indicate a few significant works which will be cited later in the thesis. We first examine the problem of computing shortest paths in a graph, then in a planar region, and finally on a polyhedral surface (each polygon being Euclidean).

2.1.1 Shortest paths in graphs

By their generality, graphs appear in many problems, being either geometric (embedded on a surface or in the space) or abstract (modeling connections between persons, entities, networks, ...). The computation of shortest paths in graphs is a fundamental problem in combinatorial optimization.

Schrijver [137, Chapters 6–9] has recently described in an extensive way the various algorithms to compute shortest paths in graphs; we refer the interested reader to this book (see also [44, pp. 514–578]). There are several variations of the problem: is it an oriented graph or not? is the graph weighted, or do all edges have unit weights? if it is weighted, can the weights be negative? are we looking for a shortest path between two points, or for the set of shortest paths from a given point, or for the set of all shortest paths in the graph? We restrict ourselves to the case of a weighted non-oriented graph $G = (V, E)$, with non-negative weights, for which we want to compute a shortest path between two vertices s and t. Quite surprisingly, most algorithms solving this problem compute, with the same complexity, a shortest path between s and *all* vertices of the graph. In fact, they compute a spanning tree of G, rooted at s, such that all simple paths in this tree starting at s are shortest paths.

Dijkstra's well-known method [56], dating back to 1959, builds such a tree. Let us first explain how to compute the distance between s and all vertices of the graph. The algorithm maintains:

1. for each vertex v, a value $d(v) \in \mathbf{R}_+ \cup \{+\infty\}$ which equals the (possibly infinite) length of a path from s to v; initially, the map d equals 0 at s and $+\infty$ for all other vertices. The value of $d(v)$ will only decrease in the course of the algorithm and will reach, at the end, the distance between s and v;
2. a set $U \subseteq V$, initially empty, of vertices for which the value of d is exactly the distance to vertex s;
3. the following property: at each stage of the algorithm, for any edge (u, v), $u \in U$ and $v \in V \setminus U$, $d(v)$ is smaller or equal than $d(u) + |uv|$. ($|uv|$ denotes the length of edge (u, v).)

The algorithm consists in finding iteratively $v \in V \setminus U$ such that $d(v)$ is minimal. In this situation, the distance between s and v must be equal to $d(v)$; v is thus appended to U. To maintain the third property, the algorithm proceeds to a *relaxation* step: for each vertex v' incident to v, if $d(v') > d(v) + |vv'|$, we let $d(v')$ be $d(v) + |vv'|$. The process is continued until $U = V$.

The algorithm can be easily adapted to compute a tree of shortest paths rooted at s: each vertex different from s maintains the value of its father, updated at each relaxation step. From the complexity viewpoint, the difficulty is to access quickly to the vertex of $V \setminus U$ whose value of d is miminal. Using this method with Fibonacci heaps, one can compute a tree of shortest paths in time $O(|E| + |V| \log |V|)$.

Many variations of the problem have been studied; let us mention a result which is useful in our context, due to Henzinger, Klein, Rao, and Subramanian [93]

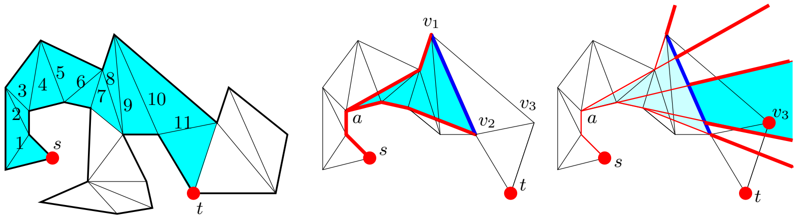


Figure 2.1: The funnel algorithm. On the left, the triangles strip joining vertices s and t, and edges $e_1, \ldots, e_{11}$. In the middle, a funnel defined by s and edge v_1v_2. On the right, the search of the shortest path from s to v_3 amounts to finding in which "sector" is v_3.

in 1997. Let G be a planar, non-oriented graph. There exists an algorithm with linear complexity $O(|E|)$ to compute a shortest path between s and each vertex of G. The cornerstone of their method is the existence, for a planar graph, of good *separators*, which are "small" sets of vertices separating the graph into "not too large" connected components; they also use relaxation steps as in Dijkstra's method.

2.1.2 Shortest paths in a planar region

Now, let us consider the more geometric case of the computation of shortest paths on a surface. We will treat of the case of a planar region and of a polyhedral surface. We refer the reader interested by geometric shortest paths in general to [115].

We start with the case of a simple polygon (*i.e.*, without holes) in the plane. The *funnel* algorithm, by Lee and Preparata [107], enables to compute a shortest path between two vertices s and t in a simple polygon P. We start by triangulating the interior of P without adding new vertices (see for example [19, p. 278]); let T be such a triangulation. The dual graph of T is a tree, and the set of triangles joining s to t is a strip of triangles B. The shortest path between s and t is necessarily in B. Let us call $e_1, \ldots, e_n$ the list of interior edges of B, in the order from s to t (Figure 2.1, left). To compute the shortest path between s and t, it is sufficient, knowing the shortest path between s and the endpoints of e_i, to be able to compute the shortest path between s and the endpoints of e_{i+1}.

Let v_1 and v_2 be the endpoints of e_i. The crucial fact used in the algorithm is the structure of the shortest paths c_1 (resp. c_2) between s and v_1 (resp. s and v_2). Starting from s, c_1 and c_2 may overlap at the beginning, until some vertex a, where they diverge and follow concave chains to vertices v_1 and v_2; hence the suggestive term of funnel, which denotes the area comprised between a, v_1, and v_2 (Figure 2.1, middle part). This simple structure enables, given the shortest paths c_1 and c_2, to deduce the shortest path between s and the vertex v_3 of edge e_{i+1} distinct from v_1 and v_2: it suffices to determine in which "sector" of the plane, delimited by the supporting lines of the line segments of c_1 and c_2 touching the funnel, is v_3 (Figure 2.1, on the right). If the triangulation is given,

the computation of the shortest path between s and t can be done in time linear in the number of triangles of the strip B (in fact, it is theoretically possible to triangulate P in linear time [30]).

We have just considered the problem of computing a shortest path in a simple polygon. But what about a polygonal planar region (the plane minus some obstacles for example)? Two main approaches are known. The first one lies in the computation of the visibility graph of the scene (see for example [46, Chapter 15] or [5]), which contains all shortest paths between vertices. In the second approach, the propagation of a "wave" from a vertex s is simulated, and the time at which this wave reaches the target vertex t is the distance between s and t. In other words, we maintain the set of points at a certain distance of s. The paper by Hershberger and Suri [95] gives an optimal algorithm, in time $O(n \log n)$, where n is the number of vertices of the scene, to compute a shortest path.

To conclude, let us mention a paper by Papadopoulou [124] which is related to our work. The problem is the following: let P be a simple polygon, and let (s_i, t_i) be k pairs of points on the boundary of P; if it exists, compute the family of k shortest paths which are simple and pairwise disjoint between s_i and t_i, $i = 1, \ldots, k$. Of course, it may well happen that such a family does not exist: for example, if $k = 2$ and if the order of the vertices on the boundary of the polygon is s_1, s_2, t_1, t_2, any path between s_1 and t_1 must cross any path between s_2 and t_2. Paths are supposed to be disjoint in a weak sense: they are allowed to go along together, but not to cross (in other words, there exists an arbitrarily small perturbation which makes the paths simple and pairwise disjoint). The algorithm by Papadopoulou has complexity $O(k + n \log k)$, where n is the complexity of the polygon. In Chapter 3, we will also have to find a family of shortest paths which are simple and pairwise disjoint, but in the case of a surface which is not necessarily planar and while maintaining the homotopy class.

2.1.3 Shortest paths on polyhedral surfaces

Let us now start the computation of shortest paths on a polyhedral surface. The *topological* definition of a polyhedral surface has been given in Chapter 1; we add to this definition a *metric* property: each polygon must be isometric to a polygon of the Euclidean plane (in particular, the lengths of the identified edges must match together).

We have to distinguish between the notion of shortest path and the notion of *geodesic*, which is a path *locally minimal*, that is, a local perturbation of this path increases its length. A shortest path is a geodesic, the converse being false. Starting with a path between two points, it is possible to deform it into a geodesic by local optimizations; on the contrary, to compute a shortest path, such methods are not sufficient. We will only focus on shortest paths computations, not on geodesics.

An intermediate problem is to compute the sequence of edges crossed by the shortest path. A crucial property is the *unfolding* property: if a shortest path crosses an edge e, and if we flatten the two polygons incident to e, then the shortest path is a line segment in this representation. If the sequence of edges crossed by the shortest path is known, this property shows that the funnel algorithm can be used in the corresponding strip of triangles to compute the shortest path. From a theoretical viewpoint, it is possible to study [118, 1] the sequences of edges crossed by all shortest paths on a convex polyhedron, which yields an indication on the structure of these shortest paths.

Mitchell, Mount, and Papadimitriou [116] have given an algorithm to compute exact shortest paths on a (not necessarily convex, and of arbitrary topology) polyhedral surface, in time $O(n^2 \log n)$, where n is the complexity of the surface. Their method relies on the "continuous Dijkstra" technique: a point s being fixed, the algorithm computes a subdivision of the polyhedron such that, in each region, the shortest paths to s cross the same edges of the surface. Chen and Han [31] have shown that the problem is solvable in time $O(n^2)$. Their method is to build a tree of sequences of edges crossed by the shortest paths (without subdividing the surface); they manage to bound the size of the tree.

Approximation algorithms have also been developed. Generally, the vertex-edge graph of the surface is refined in a particular way, and the shortest path is computed within this graph. In practice, these algorithms have a good behaviour and are quite simple to implement [99]; it is even possible to bound the quality of the approximation [104]. "Fast marching methods" can also be used: they simulate the propagation of a wave from a starting point, see [138, pp. 289–298].

On the other hand, computing a shortest path in $\mathbf{R}^3$ with obstacles is much more difficult: Canny and Reif [27] have shown that this problem is NP-complete. Approximation algorithms are developed.

2.2 Curves on surfaces: homotopy and decomposition

The topology of surfaces, as briefly described in the last chapter, dates back to the first half of the 20th century. We will here focus on the algorithmic aspects of this domain, by pointing out the problems of homotopy of curves on surfaces. This way, we restrict ourselves to a small part of computational topology; we will not try to mention problems related to homology, and our discussion will conduct us only very concisely to Morse theory. For general references on computational topology, we refer the reader to publications [153, 11, 51]. Moreover, we defer to the next section the topics concerning the shortening of curves while preserving their homotopy class, which are more directly related to our works.

In a first part, we will describe the algorithms for computing polygonal schemata, which are often the preliminary step to solve homotopy problems on surfaces. Then, methods enabling to decide if two curves are homotopic will be described, since it is quite a central problem. Finally, we will more concisely mention the existing techniques to decide if a curve is homotopic to a simple curve, or to "uncross" a set of curves.

Unless otherwise specified, the surface $\mathcal{M}$ is a triangulated surface (the algorithms often extend to the case where $\mathcal{M}$ is polyhedral), and the curves are cycles or possibly closed paths in the vertex-edge graph of $\mathcal{M}$.

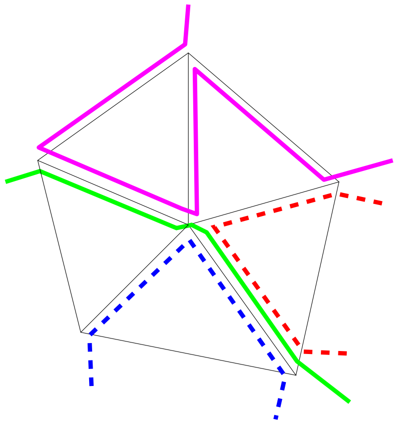


Figure 2.2: Paths going along the vertex-edge graph of a polyhedral surface $\mathcal{M}$. For each edge e of this graph, the order, from left to right, of the edges of the paths going along e is known.

2.2.1 Surface decomposition

2.2.1.1 Reduced or canonical schemata

For a given surface $\mathcal{M}$, the existence of a polygonal schema has been known since the end of the 19th century: the classical proof of the classification theorem for surfaces exactly amounts to showing that each surface has a canonical polygonal schema. This proof is effective: starting with a polygonal schema, it yields a way to compute a canonical schema, as noted in 1921 by Brahana [21].

Let us first emphasize that there may exist no reduced polygonal schema whose loops are on the vertex-edge graph G of a surface $\mathcal{M}$. If the surface is boundaryless, of genus g, the basepoint of the corresponding system of loops must have degree at least $4g$, which is of course not always the case. Hence, it is necessary to draw loops outside G. In fact, from an algorithmic point of view, this is not necessary: it is sufficient to assume that several loops can go along a same edge of G, so that it is possible to "spread them apart" to make them simple and pairwise disjoint. In Chapter 3, we will also have to deal with paths which "go along" each other on the vertex-edge graph; see Figure 2.2, and, for more details, Section 3.1, page 60.

Lazarus, Pocchiola, Vegter, and Verroust [105] have worked on the computation of a canonical schema in the case of a combinatorial surface. They have given two algorithms with optimal complexity $O(gn)$, for a boundaryless orientable surface $\mathcal{M}$. Their first algorithm, sketched in [154], is incremental: the principle is to maintain, at each step of the algorithm, a surface S whose boundary is a simple cycle on the vertex-edge graph of $\mathcal{M}$. At the beginning, S is the surface $\mathcal{M}$ with one triangle removed. The surface is shrinked by iteratively removing triangles incident to its boundary. When the removal of such a triangle modifies the topology of the boundary (which is not a simple cycle any more), there are two cases according to whether this removal disconnects S or not. If it does, the algorithm is recursively run on each of its connected components (if the genus of the component is non-zero). Otherwise, this topological modification of the boundary enables to compute a pair of loops, which will be part of the final canonical polygonal schema, and the algorithm is run on the surface cut along these loops.

The previous method directly builds a canonical schema. On the contrary, the second algorithm proposed in [105] is based on the use of Brahana's method [21]: a polygonal schema is computed, which is then put under reduced form, and then under canonical form. The computation of an initial schema is made simply by computing a subgraph G' of G such that its complementary part is an open disk (see Section 1.3.2). In fact, G' has $2g$ independant cycles, and these cycles will play a part in the following, because they generate the fundamental group of $\mathcal{M}$. A reduced schema is obtained by extending the cycles of G' into loops having a common basepoint, then it is transformed into a canonical schema by proceeding combinatorially to so-called "Brahana transformations", which are cut-and-paste operations; the difficulty resides in the obtention of an overall $O(gn)$ complexity.

2.2.1.2 Short schemata

In 2002, Erickson and Har-Peled [68] considered the problem of finding the shortest polygonal schema of a surface. Let $\mathcal{M}$ be a polyhedral surface, possibly non-oriented, of genus g and with k boundaries; assume that each edge has a non-negative *weight*. The goal is to cut $\mathcal{M}$ along some of its edges to obtain a topological disk, while minimizing the sum of the weights of the edges of the cutting. (Here, each edge of the surface contains at most one path of the polygonal schema.) The paper contains several results:

- this problem is NP-hard, even in the case where all edges have unit weights. It is indeed possible to reduce this problem to the problem of the *rectilinear Steiner tree*: if n points on a square grid of size $m \times m$ are fixed, finding the shortest tree in the grid which contains all points is NP-hard;

- there is an algorithm to solve this problem in time $n^{O(g+k)}$. Let G be the shortest polygonal schema on $\mathcal{M}$, considered as a subgraph of the vertex-edge graph of $\mathcal{M}$. Let $\hat{G}$ be the graph obtained from G by removing the dangling edges which are not incident to the boundary of $\mathcal{M}$, and by removing the vertices of degree 2; $\hat{G}$ is proved to have $O(g+k)$ vertices and $O(g+k)$ edges; then an exhaustive search is performed;

- the authors also gave a polynomial algorithm to compute an $O(\log^2 g)$-approximation of the shortest polygonal schema. Their approach consists in cutting iteratively the surface along short essential cycles, until surfaces of zero genus are obtained, and then in cutting these surfaces with a spanning tree joining the boundaries of these surfaces. They also gave a method to compute exactly the shortest essential cycle in the vertex-edge graph of $\mathcal{M}$.

2.2.1.3 Other types of decompositions

We wished to limit our description to the case of the decompositions by polygonal schemata, because they are the closest to our work. However, there exist other types of topological decompositions of surfaces.

In the conclusion of [68], the existence of pants decompositions of surfaces is mentioned, but, to our knowledge, no algorithm on this subject exists in the literature.

Morse–Smale complexes constitute an important type of surface decomposition. A property is that they depend on a map $f : \mathcal{M} \to \mathbf{R}$ (if $\mathcal{M}$ is embedded in $\mathbf{R}^3$, f can be the "height function" of the surface). The idea is to decompose $\mathcal{M}$ along the flow lines of f passing through critical points of f. This is mathematically well-studied; difficulties occur in the polyhedral case. Edelsbrunner, Harer, and Zomorodian [63] have given an algorithm to compute Morse–Smale complexes decomposing a surface $\mathcal{M}$. This approach can be generalized to higher dimensions [62].

Another problem [28] is the decomposition of a two-dimensional simplicial complex, which is, a priori, not a surface, into surfaces: the complex is cut along its edges so that, after cutting, the complex is a union of surfaces.

2.2.2 Contractibility and homotopy tests

The two following questions have received much attention, in the development of algebraic topology and since the apparition of computational topology:

- the *contractibility problem* is as follows: on a combinatorial surface $\mathcal{M}$, decide if a closed path c is contractible. Another problem is to decide if two given paths c_1 and c_2 are homotopic; it is trivially equivalent to the contractibility problem since c_1 and c_2 are homotopic if and only if the concatenation of c_1 and of the inverse of c_2 is a contractible closed path;

- the problem of deciding whether two given cycles γ_1 and γ_2 are homotopic. Solving this problem clearly enables to solve the first one, a closed path being contractible if and only if the corresponding cycle is contractible. This problem is thus more difficult than the previous one.

We here indicate the two main approaches to solve these problems. Unless otherwise specified, the surface $\mathcal{M}$ is without boundary.

2.2.2.1 Approach using the universal covering space

A first approach is based on the universal covering space. Let us assume that we are able to build the universal covering space $(\widetilde{\mathcal{M}}, \pi)$ of our surface $\mathcal{M}$ (or, at least, a sufficiently large part of this covering space), and to build lifts of paths of $\mathcal{M}$ in $\widetilde{\mathcal{M}}$. Then, deciding whether a path c is contractible reduces to lift c to a path $\widetilde{c}$, which is closed if and only if c is contractible.

Schipper [135] used this property. His algorithm incrementally builds the part of the universal covering space containing the lift of a curve. This construction is based on the preliminary computation of a canonical polygonal schema of $\mathcal{M}$; then, copies of this polygonal schema are glued in the course of the algorithm. With this construction, he can solve the contractibility problem in $O(gn + g^2k)$ time, where k is the number of edges of the path, g is the genus of the surface and n is its complexity.

Dey and Schipper [54] have improved the previous result: the contractibility test can be achieved in $O(n + k \log g)$ time. The method is similar, but the polygonal schema is not necessarily canonical any more, and its storage is done in a more efficient way, which reduces the overall complexity.

The two previous papers also work in the non-orientable case ($\mathcal{M}$ is still boundaryless). This method using the universal covering space is interesting by itself. But this does not yield any algorithm to decide whether two cycles are homotopic.

2.2.2.2 Algebraic approach

It is natural to reformulate these problems in algebraic terms. A path c is contractible if and only if c represents the unit element of the fundamental group; two cycles γ_1 and γ_2 are homotopic if and only if the homotopy classes of (any) two loops ℓ_1 and ℓ_2 associated to γ_1 and γ_2 are conjugates.

Let us consider a polygonal schema P of $\mathcal{M}$, on an orientable surface of genus g. We assume for simplification that P is canonical: it has the form

$$a_1 b_1 \bar{a}_1 \bar{b}_1 \dots a_g b_g \bar{a}_g \bar{b}_g.$$

Any path in $\mathcal{M}$ retracts on the edges of P. On the other hand, the fundamental group of $\mathcal{M}$, written $\pi_1(\mathcal{M})$, is generated by the paths of P. It is indeed the free group with $2g$ generators

$$[a_1], [b_1], \dots, [a_g], [b_g],$$

quotiented by the relation

$$[a_1][b_1][\bar{a}_1][\bar{b}_1] \dots [a_g][b_g][\bar{a}_g][\bar{b}_g] = 1 \quad (2.1)$$

(1 denoting the unit element of $\pi_1(\mathcal{M})$). It is thus possible to *code* the homotopy class of a path by a *reduced* word (containing no factor of the form $x\bar{x}$ or $\bar{x}x$) on the alphabet

$$A = \{a_1, \bar{a}_1, b_1, \bar{b}_1, \dots, a_g, \bar{a}_g, b_g, \bar{b}_g\}. \quad (2.2)$$

This coding is ambiguous because of Relation 2.1: several words correspond to a same element of the universal covering space. The contractibility and homotopy problems can thus be split into two subproblems:

- the *translation* of the curves into algebraic terms, by *words* representing their classes in the fundamental group. This stage requires the computation of a polygonal schema;
- the *computation* in the fundamental group, that is, determining whether an element of this group, coded by a word, is the unit element, or if two elements of this group are conjugates.

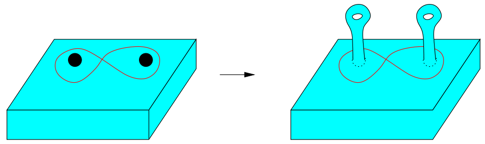


Figure 2.3: Transformation of a boundaryless surface to a surface with boundary by addition of handles. The surface $\mathcal{M}$ (on the left) is a sphere (a parallelepiped) with two boundaries, and the surface $\bar{\mathcal{M}}$ (on the right) is a double-torus, that is, a sphere with two handles. Any path in $\mathcal{M}$ is contractible in $\mathcal{M}$ if and only if it is contractible in $\bar{\mathcal{M}}$.

Dehn [49], as far back as 1912, proved that the contractibility problem and the homotopy problem for cycles were decidable, on boundaryless orientable surfaces. He gave an algorithm to compute in the fundamental group, by coding, in a non-ambiguous way, a homotopy class with a reduced word on the alphabet A. However, he did not consider the problem of computing a polygonal schema, and the complexity of his algorithm is not optimal. See [145, p. 186].

Nearly a century afterwards, in 1999, Dey and Guha [53] gave an optimal algorithm to solve both problems. Their approach follows the two above steps. They however avoid the computation of a canonical polygonal schema, as in [54]. The second step, more difficult, is algebraic; it relies on results of combinatorial group theory [84]. Their result is that testing homotopy between two cycles with complexity k_1 and k_2, on a possibly non-orientable surface of complexity n, can be done in $O(n + k_1 + k_2)$ time. Their study however excludes three surfaces of low genus for which the result of group theory does not apply.

2.2.2.3 The surfaces with boundary

The papers cited above do not mention the homotopy tests on surfaces with boundary. Nevertheless, any surface with boundary $\mathcal{M}$ can be extended to a boundaryless surface $\bar{\mathcal{M}}$ containing $\mathcal{M}$, by gluing a "handle" to each boundary, see Figure 2.3. It is intuitively clear that a path included in $\mathcal{M}$ is contractible in $\mathcal{M}$ if and only if it is contractible in $\bar{\mathcal{M}}$; a rigorous proof of this fact is immediate by Seifert–Van-Kampen theorem, a classical theorem of algebraic topology (see for example [145, pp. 124–132]). This, with the algorithm by Dey and Guha [53], implies that the problem of determining whether two cycles are homotopic on an orientable surface with boundary is possible in $O(n + k_1 + k_2)$ time.

2.2.2.4 The case of the plane

To our knowledge, only one paper by Cabello, Liu, Mantler, and Snoeyink [25] tackles the problem of testing the contractibility in the plane with obstacles. (Later, in Section 2.3.3, page 49, we will see that other papers aim at computing the shortest path homotopic to a given path in the plane, which in particular solves this problem; but the complexity is not necessarily optimal.)

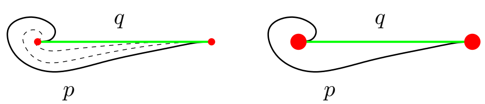


Figure 2.4: The two notions of homotopy where the endpoints of the paths are obstacles. First case, on the left: the endpoints of the paths are "pins"; both paths p and q are homotopic, p having the possibility to rotate around its left endpoint (a few stages of the homotopy are represented in dashed lines). Second case, on the right: the obstacles are disks, and the paths p and q are not homotopic any more.

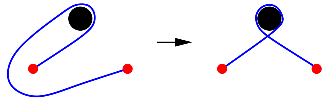


Figure 2.5: The shortest path homotopic to a given simple path is not necessarily simple. Ths surface here is the plane minus one disk.

Let P be a set of points in the plane, called obstacles; we wish to decide whether two paths are homotopic in $\mathbf{R}^2 \setminus P$. Actually, it is possible to define, in addition to the usual notion of homotopy where the curves are disjoint from the obstacles, two variations of homotopy where the endpoints of the paths are themselves obstacles (Figure 2.4):

- the case where the endpoints of the paths are "pinned" on a point of P: a path c is a continous map from the open set $(0, 1)$ into the plane minus P, with $\lim_{0^+} c$ and $\lim_{1^-} c$ belonging to P;
- the case where the endpoints of the paths are "pushpins": each point of P is considered as a larger obstacle (a closed disk with radius ε); a path is defined as above, but the endpoints of the paths are on the boundary of the disk. Thus, two paths with the same source, one of them winding around the obstacle and the other one going straight ahead, are considered as being non-homotopic under this definition, although they were homotopic under the previous definition.

We will have to run into these subtleties later; in fact, the proof of Theorem 3.2, in Chapter 3, consists exactly in playing with these two definitions. The paper by Cabello *et al.* [25] applies to the standard homotopy notion as well as the two other ones. The case of the usual notion is more intricate in the manipulation of simple paths, because the shortest path homotopic to a given simple path can self-intersect, as shown in Figure 2.5. The same phenomenon occurs for loops, see Figure 2.6.

Assume that p and q are two paths with complexity n, the set P containing also n points. In the case where p and q are both simple, but can intersect together, Cabello *et al.* have given an algorithm of complexity $O(n \log n)$ to test whether p and q are homotopic in $\mathbf{R}^2 \setminus P$. The method can be decomposed into three stages:

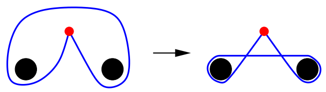


Figure 2.6: The shortest loop homotopic to a given simple loop is not necessarily simple. The surface is the plane minus two disks.

- a *compression* of the paths p and q, to store them in a form which does not contain their geometric properties but which keeps the information of their homotopy classes. This is done by defining an *aboveness relation* containing the points of P and the x-monotone subpaths of p and q;
- a *canonization* of these representations, using shortcuts of the unused parts;
- a *comparison* of the representations (this step is non-obvious, because, after the canonization, two homotopic paths may have different representations).

If p and q are not necessarily simple, the authors give an algorithm with complexity $O(n^{3/2})$. We will not give the details of this algorithm which uses many results and data structures of computational geometry (red-blue segments intersections, orthogonal range queries, crossing number of a graph, and so on). Testing the homotopy between non-simple paths p and q is indeed at least as difficult as *Hopcroft's problem*, which seems to be not solvable in less than $O(n^{4/3})$ time.

2.2.2.5 Extensions

All the questions raised in the case of the surfaces can be asked in the case of more general objects. But undecidable problems appear. There are simplicial complexes of dimension 2, or 4-manifolds, for which the contractibility problem is not decidable: indeed, any group with a finite presentation is the fundamental group of a simplicial complex of dimension 2 (and also of a manifold of dimension 4), and deciding, in such a group, if a given element is the unit element is in general unsolvable. See Stillwell [145, p. 247], who mentions other related results.

The complexity of these problems for manifolds of dimension 3 is unknown, but these problems are supposed to be very difficult: independantly from the algorithmic questions, the classification of 3-manifolds is not known. A key ingredient in this direction would be to prove *Poincaré's conjecture*, which claims that a boundaryless, compact, connected, simply connected 3-manifold is homeomorphic to S^3. Recent works due to Perelman seem to be an important step towards a proof of this conjecture [125, 126].

2.2.3 Uncrossing curves

Another question which received attention is the following: a curve (path or cycle) on $\mathcal{M}$ being fixed, is there a simple curve homotopic to this curve? More generally, a family of curves $C = (C_1, \ldots, C_n)$ being fixed, how to "uncross" these curves,

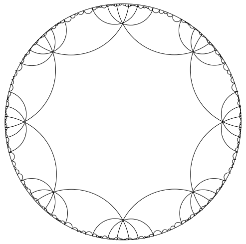


Figure 2.7: A tessellation of the hyperbolic disk with regular octogons (with the hyperbolic metric).

in the following sense: find $C' = (C'_1, \ldots, C'_n)$ such that C_i and C'_i are homotopic for all i, the curves of C' having as few intersections as possible?

2.2.3.1 A not really constructive method

At which condition is a loop homotopic to a given simple loop? Poincaré gave, in 1904, a necessary and sufficient condition, on a boundaryless orientable surface $\mathcal{M}$. Remember that the universal covering space of the torus consists of a regular tessellation of the plane with squares. Such a construction is impossible with the Euclidean metric for surfaces of higher genus: for the double-torus, for instance, it would be necessary to build a tessellation of the plane with regular octogons, each vertex being incident to eight octogons. This is however possible using a non-Euclidean metric: one can build a tessellation of the hyperbolic plane (also called *Poincaré's disk*) with such polygons (see Figure 2.7, and [149] for a detailed description of this metric). Let ℓ be a loop, and let $\widetilde{\ell}$ be a lift of ℓ in the universal covering space built as indicated; let a and b be the endpoints of ℓ, and let ℓ' be the projection onto $\mathcal{M}$ of the unique geodesic path between a and b; ℓ' is homotopic to ℓ. Let us consider the set of all lifts of ℓ'; obviously, if they are all pairwise disjoint except at their endpoints, then ℓ' is simple, and thus ℓ is homotopic to a simple loop. Poincaré proved the converse statement: ℓ is homotopic to a simple loop if and only if the lifts of ℓ' are pairwise disjoint, except at their endpoints. Hence, ℓ is simple if and only if the shortest loop homotopic to ℓ is simple. This noteworthy fact is specific to the metric (Euclidean or hyperbolic, depending on the genus of the surface), as we explained in Figure 2.6. Several decades where needed to turn this constatation into a fully effective process (see [145, pp. 190–194]).

2.2.3.2 The algebraic viewpoint

Chillingworth also expressed interest in this problem: in [33], he gave a method to determine whether a given cycle is homotopic to a simple cycle. His method is algebraic and uses the *winding number* of a curve with a vector field. In [34], his result is extended and enables to determine whether a family of cycles is "uncrossable" (*i.e.*, if this family is representable by a family of simple, pairwise disjoint cycles, while keeping their homotopy classes). The method is quite similar,

but it uses, apart from the winding numbers, a way to code the homotopy class of a curve which consists in writing down the list of the paths of a polygonal schema crossed by this curve. We will introduce in Chapter 3, page 68, a variation of this code. However, this does not yield a way to compute a family of simple, disjoint cycles, if such a family exists.

The works by Cohen and Lustig [37] and Lustig [112] are in the same vein: the goal is to compute the minimal number of intersections between two cycles within a given homotopy class. The cycles are given in an algebraic form (on a homotopy basis obtained with a canonical schema), but the ideas use the hyperbolic disk. The algorithm has been implemented.

2.2.3.3 The use of elementary uncrossing operations

de Graaf and Schrijver [48] have studied the following more general problem: a family of cycles $\Gamma = (\gamma_1, \ldots, \gamma_n)$ on a surface (possibly non-orientable and/or with boundary) being fixed, how is it possible to transform Γ into a family $\Gamma' = (\gamma'_1, \ldots, \gamma'_n)$ such that, for all i, γ_i and γ'_i are homotopic, while minimizing the number of crossings of Γ'? They proved that this is always possible by using a few elementary uncrossing operations, called *Reidemeister moves*, which do never increase the number of intersections. Additionally, each cycle γ'_i in Γ' has a minimal number of self-intersections in its homotopy class, and, for all $i \neq j$, γ'_i and γ'_j cross a minimal number of times among all curves γ_i and γ_j in their homotopy classes. This yields an algorithm (not so effective, though) to compute such a Γ'. The proof of this result is quite difficult and relies on the study of the properties of a (hypothetical) family of cycles which would not reduce to a minimally crossing family by Reidemeister moves, the family having a minimal number of cycles; it consists in showing the result for simple surfaces (sphere, open disk, projective plane), then in proving the same result for more complex surfaces by using a Euclidean or hyperbolic metric on them.

Let us note that Hass and Scott [88] had studied the same problem before, in the particular case of a single curve; the type of result they obtain is a bit different, since they look for disks bounded by one or two pieces of curves on the surface, containing possibly other pieces of curves in their interior.

One of the elementary Reidemeister moves is the "uncrossing" of two cycles which cross twice, thus bounding a topological disk. If one of the cycles is the shortest cycle in its homotopy class, and if a Reidemeister move is possible, this means that the other cycle can be shortened by proceeding to this Reidemeister move, without changing the homotopy class (Figure 2.8). This constatation will be generalized and used several times in the proofs of Chapter 3. A key ingredient of these proofs is to show that some algorithm uncrosses curves, the uncrossing being done with such moves.

2.3 Homotopic shortest paths

In this section, we continue our review of computational topology of curves on surfaces, with an emphasis on the following problem: a curve c on a surface being

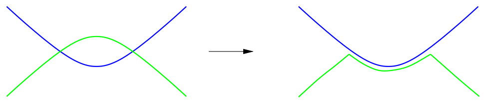


Figure 2.8: One of the Reidemeister moves.

given, how to find a curve homotopic to c which is *optimal*, that is, as short as possible within its homotopy class?[1] In other words, how to shorten c as much as possible while keeping its topological properties? More generally, we can consider the simultaneous optimization of several simple, pairwise disjoint curves.

This question is not purely topological, contrary to the subjects described in the previous section: metric questions are superimposed to the topological problem. We will briefly discuss this problem on smooth surfaces in general, with results asserting the existence of shortest homotopic cycles. We will then have a more algorithmic viewpoint, in the case of the plane with obstacles, then of locally Euclidean surfaces. These questions are in the heart of our thesis: we will present in Chapter 3 our contribution in this domain.

2.3.1 Smooth surfaces

2.3.1.1 Hyperbolic surfaces

We have seen earlier that the case of paths in the hyperbolic disk is very special: a loop is homotopic to a simple loop if and only if the shortest loop which is homotopic to this loop is itself simple. This property remains true for any *hyperbolic surface*, *i.e.*, a surface locally isometric to the hyperbolic plane. In fact, in this context, there is a unique closed geodesic cycle in each homotopy class, which is itself the shortest cycle in its homotopy class [24, Theorem 1.6.6]. Any iterative process which locally shortens a cycle will thus converge to an optimal cycle. This yields an elementary algorithm to approximate optimal cycles on such surfaces, see [24, Appendix].

2.3.1.2 Surfaces with a Riemannian metric

A *Riemannian surface* is a surface equipped with a metric, *i.e.*, a positive definite quadratic form on the tangent space at each point, form depending continuously on the point of the surface. A surface smoothly embedded in $\mathbf{R}^3$, equipped with the induced metric (the distance between two points of the surface being the length of the shortest path on the surface between these two points) is a simple example of a Riemannian surface. A quantity which is intrinsic to each point of a Riemannian surface is its (Gauß) *curvature*. Intuitively, the curvature of a surface $\mathcal{M}$ at some point gives the following information. In the plane, the ratio between the circumference and the radius of a circle equals 2π. This property no longer holds in the general case: on a sphere, if r is small enough, the set of points at

[1]In Chapter 3, we will consider simple curves. We shall thus say that a curve is *optimal* if it is simple and if there exists no shorter simple curve in its homotopy or isotopy class.

(geodesic) distance r from a point p is a circle whose circumference is smaller than $2\pi r$. On a horse saddle, or a crisps, the circumference is greater than $2\pi r$. The curvature of $\mathcal{M}$ at p is zero (resp. positive, resp. negative) if the circumference of a circle with radius r centered at p, divided by r, tends to a limit equal to (resp. greater than, resp. less than) 2π when r tends to 0. A hyperbolic surface is, in fact, a Riemannian surface whose curvature is constant, equal to -1.

A result dating back to 1929, due to Lusternik and Schnirelmann [111], states that each Riemannian surface which is homeomorphic to the sphere contains at least three simple closed geodesics. In the general case, each simple, non-contractible cycle admits at least one shortest homotopic cycle, and, furthermore, each such cycle is necessarily simple if the surface is orientable (this is due to Freedman, Hass, and Scott [75]). More generally, let us consider a finite number of cycles on a Riemannian surface, each of them being as short as possible in its homotopy class. What can we say on the crossings between these cycles? It seems plausible that they have a minimal number of intersections allowed by their homotopy classes; this has also been showed in [75]. Conversely, Neumann-Coto [121] proved that any minimally crossing finite set of cycles is a set of geodesics, as short as possible in their homotopy classes, for some metric. For a general textbook on closed geodesics, see [102].

2.3.1.3 Continuous transformations of curves into geodesics

Let us now mention several processes which enable to transform a curve drawn on a Riemannian surface into a geodesic. These processes do not (in general) compute shortest homotopic paths, but have been extensively studied.

Birkhoff's process [16], found in 1917, yields a very simple way to this purpose. The idea is, starting with a curve parameterized with its length, to sample regularly this curve with a finite number of points, and to replace each piece of curve between two consecutive points by a geodesic segment. We iterate by taking as sample points the set of middle points of the geodesic segments obtained.

In 1993, Hass and Scott [89] gave an elementary algorithm to deform one or several curves into geodesics. Unlike Birkhoff's process, their process satisfies in an obvious way the fact that the number of intersections or self-intersections between curves can only decrease in the course of the algorithm. The idea is to take a family of disks covering the surface, and to shorten each maximal piece of curve by a shortest path within this disk.

Another type of curve evolution, using the curvature, is the following. Let us consider a cycle drawn in the plane or on a Riemannian surface. Let the curve evolve in the following way: each point of the surface moves in the direction of its normal vector at each point, with some speed. This yields a family of partial differential equations which arise frequently in physics (fluid mechanics, material science), when the curve represents an *interface* between two regions (see [138] for a textbook devoted to this topic). In particular, if each point of the curve evolves with a speed proportional to the curve curvature at this point towards the curvature center, it is a *curve shortening* problem. The behaviour of this process has been well-studied. In the case of the plane, if the initial curve is simple, it remains simple during the whole evolution process; at some time, it becomes and remains convex, and then retracts to one single point [82]. In the case of Riemannian surfaces, the curve can converge to a point or to a closed geodesic [83, 78]. It is thus a shortening process of cycles on a surface which maintains the homotopy class and the simplicity during the whole deformation.

2.3.2 "Flat surfaces"

Hershberger and Snoeyink [94], in 1994, found an algorithm which computes the shortest path homotopic to a given path, in the following realm. The surfaces have a boundary, and are triangulated in such a way that the vertices of the triangulation are on the boundary of the surface. In addition, each edge has a length, and each triangle is provided with the Euclidean metric induced by the length of its edges. An example of such a surface is a polygon with holes in the plane: it is always possible to triangulate such a polygon without introducing new vertices. Let $\mathcal{M}$ be such a surface.

The method lies in computing a portion S of the universal covering space of $\mathcal{M}$ which contains a lift of the input path p. As the surface contains no vertex in its interior, we know, according to Section 1.4.4.3, how to build S. Let us call a and b the endpoints of p. We will prove that the shortest path q in $\widetilde{\mathcal{M}}$ between a and b is inside S. If it were not the case, since the dual graph of $\widetilde{\mathcal{M}}$ is a tree, q would cross twice, in opposite directions, an edge of S; this contradicts the definition of a shortest path. It is thus sufficient to compute a shortest path in S between a and b. The funnel algorithm (Section 2.1.2) applies as is to solve this problem, although S is not necessarily isometrically embeddable in the plane.

The algorithm can be extended, with a simple modification, to compute the shortest cycle homotopic to a given cycle.

2.3.3 The case of the plane

Two very recent papers treat of the optimization of a finite number of paths in the particular case of the plane with obstacles: the goal is to use techniques from computational topology to find efficient algorithms.

In 2002, Efrat, Kobourov, and Lubiw [65] worked on the following problem. Let $C = (C_1, \dots, C_p)$ be a family of simple, pairwise disjoint paths in the plane; let P be a set of n paths in the plane, containing the endpoints of the paths in C, but no other point of C. The goal is to *optimize* the paths in C, that is, to find a family $C' = (C'_1, \dots, C'_p)$ of pairwise disjoint, simple paths such that C_i and C'_i are homotopic for all i, with the C'_i as short as possible. (The authors consider the first of the two homotopy models introduced for point obstacles, indicated in Section 2.2.2.4, page 42, and not the usual homotopy notion.)

Actually, the resulting paths are allowed to intersect or to self-intersect, by going along each other, provided they do not cross (we have described a similar situation in the end of Section 2.1.2, page 36, when describing [124]). The reason of this choice is that, without this convention, an optimal family C' would not necessarily exist, the configuration space being an open set.

The first step consists in the creation of "shortcuts" in the input paths, while maintaining their simplicity. The crucial remark is then that many paths are in fact made of $O(n)$ polygonal x-monotone arcs, each of these arcs corresponding to several overlapping pieces of paths. The second step is thus to shorten these pieces, using the funnel algorithm, and with the help of the shortest paths which have still been computed. Finally, the pieces of the paths are glued together, and are the output of the algorithm.

An important restriction of this paper is that the usual homotopy notion is not considered: the endpoints of the paths must belong to the obstacles. In this particular case, two simple homotopic paths are isotopic (see [67, 71]). Additionally, each resulting path is a shortest path in a given homotopy class, and is simple. We will find similar situations in Chapter 3.

Bespamyatnikh [13] improves the previous algorithm. By speeding up the second step, he proves that only shortest paths computations in monotone polygons need to be performed, thus obtaining an $O(k + n \log n)$ complexity, k being the input and output size, and n being the number of obstacles. He also gives an algorithm in the case where the input paths are not simple.

2.4 Applications

In this section, we review the applications of the fields we have introduced. We will first focus on geodesics and shortest paths in general. Then we will explain why surface decompositions are useful in several domains. Then, we will give applications of the computations of shortest homotopic paths. The work that will be presented in the next chapter falls into this category.

2.4.1 Applications of shortest paths

We exclude here the case of shortest paths in graphs, whose applications are numerous, to focus on shortest paths (or geodesics) in metric spaces.

Shortest paths appear naturally in physics. The Snell–Descartes law indicates how light refracts at the boundary between two homogeneous regions with different indices; it can be exactly formulated by saying that the travel of the light between two points is the shortest path between these two points. Waves propagate in the same way; radars and sonars exploit this property: they emit waves and measure their return time. Applications of shortest paths can be found in seismic analysis [138, pp. 298–304].

Shortest paths problems can also be found in robotics (see for example [101] or [87]). In path planning problems for robots in a scene with obstacles, it is natural to optimize the moves. The general problem is, of course, much more complicated: the size of the robot must be taken into account, because it cannot be assimilated to a point; generally, the optimized criterion is the execution time of a given task, and the dynamics of the robot also plays a part.

A set of points P in a metric space being fixed, the *Voronoi diagram* of P is the decomposition of the space induced by the sets of points of the space which have the same nearest neighbor among all points of P. Voronoi diagrams in the plane have been extensively studied in computational geometry (see for example [19]) and are directly related to shortest paths problems. In particular, the computation of a Voronoi diagram on a surface must use the notion of (geodesic) distance between points of the surface.

In computer-aided design and geometric modeling, geodesics or shortest paths are also used. A user who intends to draw a path between two points can specify the two extremal points of the curve to be drawn, and then let a program find the shortest path; if the result is unsatisfying, he/she can specify new control points. The shortest path between two points is, in this situation, simply used as a "canonical" choice. Of course, this notion is limited: a small perturbation of the surface can change the shortest path between two points from all to all. But the natural tool of the ruler is thus generalized to the case of a surface.

2.4.2 Applications of short cuttings of surfaces

We now indicate a few applications of the computations of surface decompositions (by polygonal schemata or pants decompositions). Short decompositions are sought in most applications, which often concern computer graphics: in this field, unusefully complicated drawings must be avoided.

2.4.2.1 Parameterization

Surface parameterization [72, 50] is an essential tool in computer graphics. Parameterizing a surface $\mathcal{M}$ means to put it into correspondence with a planar domain (often a topological disk, but possibly several disks or several planar surfaces with holes). Of course, a bijective correspondence between $\mathcal{M}$ and a planar region does exist only if $\mathcal{M}$ is an orientable surface with zero genus and at least one boundary. In the general case, the parameterization problem is subdivided into two subproblems:

1. first, the surface is split into planar surfaces (with the help of a polygonal schema, or by cutting the surface into several topological disks; pants decompositions could also be useful). This step has an influence on the quality of the parameterization: to avoid visible artefacts (in texture mapping or in the creation of a mesh), the cut must be as short as possible. Chapter 3 will provide, among other things, a way to create such a decomposition with short paths;

2. each of the resulting surfaces is thus put into correspondence (with a piecewise linear homeomorphism, the surface being usually polyhedral) to a planar region. We will see in Chapter 4 a way to create such a homeomorphism, using Tutte's barycentric embedding theorem. In practice, the distortion of the correspondence generally has to be miminized; this is an important subproblem of the parameterization problem, and approaches such as [72] are based on Tutte's theorem.

We now mention a list of applications using parameterization.

Visualization. Parameterizing a surface yields a representation of this surface in the plane. This enables to view the whole surface (there are no hidden faces), and to easily represent informations on it (if each point of the surface has a color, for example); it is exactly what is done in cartography, by representing the earth on a planar domain. Parameterization is the natural algorithmic tool corresponding to the notions of atlases and charts in differential geometry. When the surface is drawn as a polygonal schema, one has a *pattern* of the surface, which also enables to visualize its topology.

Texture mapping. In computer graphics, a common operation is to give to a given surface the aspect of some material (wood, cloth, ...). This is made by *texture mapping*: the surface is put into correspondence with a rectangle containing some colored pattern representing the material, called *texture* [113, 109, 127]. The decomposition of the initial surface into planar surfaces is thus necessary. Of course, when putting into correspondence the planar surface(s) with the texture, care must be taken about the gluing conditions of the polygonal schema: otherwise, discontinuities would appear in the texture where the surface has been cut.

During the texture mapping on complex surfaces, a common technique consists in cutting the surface along points whose curvature is maximal: this enables to minimize the distortion induced by the parameterization. This reinforces the interest of the cutting. In fact, it is possible, while cutting the surface into a polygonal schema, to favour the cutting along regions of high curvature, by giving a smaller weight to regions of high curvature. If this is not sufficient, it is still possible to continue the cutting of the surface.

Mesh creation and numerical computations. Ordinary or partial differential equations are very common in physics, and it is crucial to be able to simulate them. Analysis methods by *finite elements* are used, *i.e.*, the space is discretized by a mesh [76]. The shape of the elements of the resulting mesh has a direct influence on the efficiency of the numerical simulation: usually, elements as regular as possible (no triangle with small angles) are sought, and a more refined mesh at certain places is sometimes desirable.

For a surface of $\mathbf{R}^3$, the computation of such a mesh can use a surface parameterization stage. Parameterization often enables to reduce the problem to computations in the plane, which makes them simpler and faster (see for example [150]).

Remeshing and compression. The apparition of three-dimensional scanners and the always increasing size of meshes (several hundreds of millions of polygons for the modeling of an airliner) lead to a new problem: how to deal with geometric models?

Remeshing [4] a mesh of a surface is to find a new mesh which is geometrically close to the initial mesh but which has a different number of elements (vertices, edges, faces). This in particular enables to *simplify* the geometry: the mesh coming from a 3D scanner is extremely regular, and many points are geometrically of poor significance (points located on flat regions, for example); they can be removed. It is also possible to *augment* the geometry: starting with a rough mesh, points can be added to smooth the object.

Remeshing a mesh also enables to *compress* it efficiently, for storage or transmission purposes: in [86], a remeshing method is presented, in which the resulting mesh is stored in a very compact way.

In the papers which have just been cited, a parameterization of the input surface is used: this enables to work in the plane. This is also done in the paper [2] that we wrote with P. Alliez, O. Devillers, and M. Isenburg, whose goal is to remesh a tridimensional object: it is necessary to use a parameterization stage, and thus to compute a polygonal schema. The parameterization is chosen so that it is as conformal as possible (angles should be preserved as much as possible), and it is proved that the distortion introduced by this parameterization is not harmful for the result (more precisely, the effect of this distortion can be compensated). In this precise case, decomposing the surface into a fundamental system of loops instead of some polygonal schema is advisable: the vertices of the polygonal schema must be treated separately, and it is better to minimize their number.

2.4.2.2 Other applications in computer graphics

We now describe other problems in computer graphics for which a topological decomposition of surfaces can be useful.

Multiresolution analysis and topology filtering. Multiresolution analysis consists in the creation of a *hierarchy* of meshes, more or less refined, representing a given surface. In order to do that, it is necessary to decompose the initial mesh into topologically elementary surfaces; this enables to avoid the removal of handles or holes during the mesh simplification [59, 79]. On the contrary, it one wishes to *filter* the topology of a mesh, that is, to remove the "topological noise" constituted by the small holes and the small handles, a cutting of the surface can also help. Let us note however that, in this latter case, the use of the smallest essential cycle may be more adequate (see the description of [68] in Section 2.2.1.2, page 39).

Geometric compression. We have explained above that surface parameterization can be useful for the compression of geometric models. Independantly from this, the decomposition of a surface (without explicitely computing a parameterization) might be useful for compression purposes.

The compression of a mesh is usually subdivided into two stages [3]. The first stage is the storage of the combinatorial part of the mesh, *i.e.*, the underlying combinatorial surface: incidence relations between vertices, edges, and faces. The second stage is the storage of the geometry, *i.e.*, the coordinates of the vertices. During the first stage, it can help to have a planar simplicial complex, because efficient algorithms to code planar triangulations are known [128]. Cutting the

surface to get a topological disk can thus be useful for compressing the mesh connectivity.

Metamorphoses. A discipline of computer graphics aims at creating *metamorphoses*, also known as *morphings* (see [106] for a general description and references). These are continuous deformations between two (three-dimensional) objects. These metamorphoses are frequently used nowadays for visual effects (advertisings, movies, ...). One of the goals is the development of tools enabling the automatic creation of metamorphoses between two surfaces of $\mathbf{R}^3$. In order for such a transformation to exist, it is of course necessary that both surfaces have the same topology. But this is not sufficient: there must exist a continuous family of homeomorphisms of $\mathbf{R}^3$ (called *ambiant isotopy*), which takes one surface into the other. For example, the surface of a torus (embedded in the standard way in $\mathbf{R}^3$) and the surface of a knotted string both have the same topology, but we cannot hope to create a continuous deformation without creating self-intersections in the intermediate shapes. An intermediate stage to reach this ultimate goal (which seems quite far!) can be to decompose the surfaces into polygonal schemata, which at least allows to compute a homeomorphism between both surfaces. To do this, canonical schemata must be used: if the lists of edges of the two polygonal schemata are not the same (for example $ab\bar{a}\bar{b}cd\bar{c}\bar{d}$ for one schema and $abcd\bar{a}\bar{b}\bar{c}\bar{d}$ for the other one), the correspondence between the schemata do not induce a homeomorphism between the surfaces.

2.4.3 Applications of shortest homotopic paths

The results of Chapter 3 allow not only to shorten decompositions of surfaces, but also to find a shortest simple path among all paths homotopic to a given simple path. We here focus on applications of these results.

We have evoked the domain of robotics to justify the interest of shortest paths computations. This remains true for shortest paths computations within a given homotopy class: it can be necessary to force a robot to travel through certain places. Optimizing a slalom (described with markers) reduces to computing a shortest homotopic path with respect to the obstacle markers!

Shortest homotopic paths are also used in VLSI systems (Very Large-Scale Integration): the problem is to be able to connect pins from several electronic components of a chip, while minimizing the length of the wires. If all connections are in the plane, and if the topology of the net is prescribed, the problem amounts to computing shortest homotopic paths with known homotopy classes. In addition, it is necessary to detect whether these connections are possible without crossings, that is, to force the wires to be simple and disjoint [108].

Another application domain is cartography and GIS (Geographic Information Systems). Quite schematically, a geographic map is stored with polygonal lines, separating the plane into regions. In many cases, a very high resolution is not necessary (details would anyway be visible only from some zoom resolution), and harmful (computation and display times are increased). It is thus necessary to be able to *simplify* the geometry of a polygonal line, while maintaining, among others:

the geometry of the curve (the new curve cannot be too far apart from the initial curve), the position of the curve with respect to some points (after simplification, the costal cities must appear on the earth), and the simplicity of the simplified curve; see [47] for more details. The two last conditions are topological conditions of homotopy and simplicity, considered in the next chapter.

A very concrete application of shortest simple homotopic path computations has been presented to us[2]. Machines such as televisions emit an electromagnetic field with an electron gun; this electron gun is made of a coil of brass winded on a quite complicated surface by a robot. Several types of surfaces can be adequate, but it is wished to minimize its size (particularly its depth) while maintaining some properties of the field. It is thus necessary to be able to compute the electromagnetic field created by such a coil, depending on the chosen surface. A first step consists thus in computing the position of the wire; this exactly amounts to computing a shortest path within a given homotopy class; this path is simple because the wire does not overlap.

For all these applications, however, the surface is quite simple, and, in particular, it is difficult to imagine applications in the case of surfaces with handles. But another application domain is computer-aided design, where surfaces considered in practice can have non-zero genus. For example, an operator must draw a path on a surface, along which the surface has to be cut. To get a nice drawing, it is natural to wish to improve the *geometry*, that is, to replace the initial path by a shortest path. But the *topology* prescribed by the user is not to be changed: the new path must be in the same homotopy class as the old path, and must remain simple. Our optimization algorithms achieve in particular this goal.

Discussion

The state of the art that we have just presented shows that the topological problems concerning curves on surfaces, from a fundamental and from the applications viewpoints, are recognized to be interesting. In particular, the shortening of curves within a given homotopy class and the decomposition of surfaces have been the subject of active works. However, the results are improvements in some specific cases, and the problems, which are difficult, are far from being solved in all cases.

Regarding the topological decompositions of surfaces, former works contain algorithms to compute polygonal schemata. Often relying on classical proofs of theorems of topology, exploited under a new viewpoint, they give asymptotical bounds on the complexity of these problems and on the size of the resulting schemata (see [154, 105]). These decompositions are, in particular, useful to compute portions of the universal covering space [135, 54]. But they are geometrically unsatisfying: for the applications, notably in computer graphics, it is necessary to have decompositions of surfaces whose curves are shorter and regular. A possibility is to treat the curves of a decomposition by a geodesic smoothing, as suggested at the end of [105], but this is not sufficient for complex surfaces (the notion of geodesic is only local, contrary to the notion of shortest path). The problem is

[2]Personal communication by Dominique Michelucci, for which I thank him.

thus not only *topological*, but also *geometric*: it is an *optimization* problem. The paper by Erickson and Har-Peled [68] (published nearly at the same time as our first work in this topic) is a first answer: a quite general problem is NP-difficult, but there is an approximation algorithm in polynomial time. To our knowledge, this algorithm has not been implemented; without any experimental results, it is difficult to know if the approximation is reasonable in practice. Our work yields a polynomial algorithm which solves exactly a slightly different problem.

Concerning the computations of homotopy of curves on surfaces, the contractibility and homotopy tests have been well-studied [135, 54, 53], and there are also algorithms to determine the minimal number of intersections between curves in given homotopy classes [34, 37, 112]. When shortest homotopic paths are considered (adding, this way, an optimization problem to the purely topological questions, like in the previous paragraph), it appears that the problem has been often raised and that solutions have been found in some particular cases (in the plane [25, 65, 13], or on surfaces locally isometric to regions of the plane [94]), using advanced methods and data structures from computational geometry, and, sometimes, by mixing them in a very complex and clever way. But no general approach has been given for this problem.

The contribution of the next chapter aims at progressing towards these two directions. We consider a surface provided with a metric, and we try to optimize a family of curves (to compute the shortest family which has the same topological properties). This uses a decomposition of the surface into topologically elementary surfaces; in particular, such a decomposition, or one single curve, can be optimized with our algorithms.

Chapter 3

Optimization of curves on surfaces

Abstract. In this chapter, we wish to *optimize* families of curves, that is, to shorten them while maintaining some of their topological properties. Let $\mathcal{M}$ be an orientable surface; let G be a weighted graph embedded on $\mathcal{M}$. Let us agree that the length of a curve on $\mathcal{M}$ is the sum of the weights of the edges of G crossed by the curve. This includes in particular the case where the surface $\mathcal{M}$ is polyhedral, and where the curves are paths in the vertex-edge graph of the surface.

We consider two types of families of curves to be optimized: a graph embedding being fixed on $\mathcal{M}$, we wish to find the shortest graph embedding isotopic to the first embedding, with fixed vertices; a family of simple, pairwise disjoint cycles being fixed, we wish to find the shortest family of simple, pairwise disjoint cycles whose cycles are homotopic to the cycles of the initial family. This in particular contains the case of one single simple path or cycle, of a fundamental system of loops, or of a pants decomposition.

The method we propose consists in extending these curves to a topological decomposition of the surface, which is optimized by greedy processes. The analysis of these optimization processes yields results of *individual optimality* of each of the curves of the resulting family, and of *simplicity* of the shortest curve homotopic to a given simple curve. We obtain algorithms to optimize families of curves in the vertex-edge graph of a polyhedral surface which are polynomial in the input of the algorithm and in the longest-to-shortest edge ratio of the surface.

Introduction

Let $\mathcal{M}$ be a compact, connected, orientable surface, possibly with boundary. Let G be a weighted graph embedded on $\mathcal{M}$. The *length* of any curve c is defined to

be the sum of the weights of the edges of G crossed by c (counting multiplicities).

In this chapter, we shall consider two types of embeddings of curves:

- embeddings of graphs on $\mathcal{M}$, the graph having no isolated vertex, and having possibly loops and multiple edges; in other words, a family of simple, pairwise disjoint paths except possibly at common endpoints;
- *embeddings of cycles* on $\mathcal{M}$, *i.e.*, families of simple, pairwise disjoint cycles on $\mathcal{M}$.

The *curves* we will consider are the paths of an embedding of graph, or the cycles of an embedding of cycles. An embedding of graph or of cycles will be denoted by $s = (s_1, \ldots, s_n)$, where the s_i are the curves of the embedding.

We wish to *optimize* such embeddings, *i.e.*, to shorten as much as possible their curves (with fixed endpoints, in the case of an embedding of graph), while maintaining some of their topological properties: we impose that the resulting curves are homotopic or isotopic to the initial curves.

Let $s = (s_1, \ldots, s_n)$ be an embedding of graph or of cycles. To optimize s, we will proceed as follows:

1. we *extend* $s = (s_1, \ldots, s_n)$ by adding curves, to obtain an embedding of graph or of cycles $(s_1, \ldots, s_N)$ (with $N \geq n$) which decomposes the surface $\mathcal{M}$ into topologically elementary surfaces (disks, cylinders, pairs of pants). Such an embedding of graph or of cycles will be called *cut system*;
2. we *optimize* the cut system $(s_1, \ldots, s_N)$ with a quite simple iterative step, thus obtaining a cut system $(s'_1, \ldots, s'_N)$. We prove that, for all i, s'_i is a curve which has the same topological properties as s_i and is as short as possible among the curves having these properties;
3. we *extract* from $(s'_1, \ldots, s'_N)$ the first n curves $(s'_1, \ldots, s'_n)$, which constitute the desired optimal embedding s'.

More specifically, after Step 2:

- for cut systems by cycles, each cycle s'_i is as short as possible among the cycles homotopic to s_i;
- for cut systems by graph, each path s'_i is as short as possible among the simple paths isotopic to s_i in $\mathcal{M}$ minus the vertices of s which are not endpoints of s_i.

The most difficult step is Step 2: although the optimization process is quite simple to understand, the proof of optimality of the result is tricky, notably in the case of cycles. The first step is not very difficult, and the third one is trivial. Cut systems are close to polygonal schemata and pants decompositions, in the sense that these also decompose a surface into topologically elementary surfaces. Besides, these processes can be applied to polygonal schemata and pants decompositions.

The curves of s are thus, *each, individually*, as short as possible among the curves which have the same topological properties. In particular, this enables to prove the following theorems:

- let c be a simple path whose endpoints are on the boundary of $\mathcal{M}$, and let C be the set of all (not necessarily simple) paths with minimal length among the paths which are homotopic to c. There exists an element of C which is simple (this result is false in general if the endpoints are not on the boundary of $\mathcal{M}$);

- let γ be a simple cycle, and let Γ be the set of all (not necessarily simple) cycles with minimal length among the cycles which are homotopic to γ. There exists an element of Γ which is simple;

- a fundamental system of loops $s = (s_1, \ldots, s_n)$ being fixed on $\mathcal{M}$ (assumed to be boundaryless in this case), any shortest fundamental system of loops $t = (t_1, \ldots, t_n)$ such that s_i is homotopic to t_i for all i is made of loops t_i which are individually as short as possible among the simple loops homotopic to s_i;

- a pants decomposition $s = (s_1, \ldots, s_n)$ being fixed on $\mathcal{M}$, any shortest pants decomposition $t = (t_1, \ldots, t_n)$ such that s_i is homotopic to t_i for all i is made of cycles t_i which are individually as short as possible among all cycles homotopic to s_i.

An important particular case is the case where the surface $\mathcal{M}$ is a polyhedral surface and where the graph G is the dual graph of the vertex-edge graph of $\mathcal{M}$. In this case, the curves can be regarded as being drawn “on” the vertex-edge graph (in fact, in an arbitrarily small tubular neighborhood, several curves having the possibility to go along a same edge while remaining disjoint): the length of a curve, in the sense previously defined, coincides with the length of the corresponding path in the vertex-edge graph. In this framework, the optimization processes yield an effective way to compute the previously described optimal curves (or families of curves). The complexity of the algorithms is polynomial in their input (complexity of the surface and of the initial curves), and in the ratio between the largest and the smallest weights of the edges of G.[1] The implementation of these algorithms does not yield particular difficulties: no complex data structure is required. The optimization algorithms are very simple, the implementation of the completion of a family of curves into a cut system would need more time to be implemented. The only arithmetic operations required are the addition and the comparison on real numbers (in fact, in the ring generated by the weights of the edges of G).

The complexity of these problems was previously unknown. Erickson and Har-Peled [68] studied the related problem of computing the shortest polygonal schema of a polyhedral surface, and proved that it is NP-hard, even in the case where the weights equal one. Our work shows in particular that a simpler problem,

[1] The algorithm is thus polynomial, in the strictest sense, only if the ratio between the greatest and the smallest weight is bounded from above by a strictly positive fixed constant.

the problem of computing a shortest fundamental system of loops homotopic to a given system, is polynomial (with unit weights).

Applications of optimization of curves, polygonal schemata or pants decompositions have been given in Section 2.4, pages 50–55, we will not further describe these applications.

To the best of our knowledge, no algorithm had been previously developed, even to optimize a single curve within a given homotopy class on a surface. To optimize a path or a loop, there however exists a naïve method: lift this curve into the universal covering space, compute a shortest path between the endpoints of this lift, and project the resulting curve onto the surface. However, the final curve may be non-simple (by projection onto the surface, self-intersections can occur, see Figures 2.5 and 2.6, page 43). In some cases, it may be desirable to allow self-intersections; but, if the input path is simple, it is natural to wish to obtain a simple path as output. Furthermore, using this method, the number of vertices of the universal covering space to explore can be exponential, hence also the cost of the method, even with unit weights (see the Appendix to this chapter, page 111).

This chapter has been the subject of publications [40, 41], written with F. Lazarus. The statement and the proof of the optimality theorem for embeddings of cycles are similar to those in [41]. The optimality theorem for embeddings of graphs is stated under a really more general form than in [40] (the embeddings of graphs to be optimized can have several vertices, contrary to fundamental systems of loops, and on a surface which may have boundaries), and proved in a far simpler way (the universal covering space is less used, and the reductions on crossing words are only parenthesized). Sections 3.4 and 3.5 also have important differences with these papers.

This chapter is organized as follows. We first present in more details the framework and the notion of length we consider here. Then, the iterative optimization processes and the statements of the two theorems concerning the optimality of their results are presented; these theorems are then proved. Afterwards, we describe how to extend an embedding of graph or of cycles to a cut system. Finally, we explain how all of this can be implemented.

3.1 Framework of the study

3.1.1 Definition of length

This section introduces notions which will be used in the whole chapter. $\mathcal{M}$ denotes a compact, connected, orientable surface, possibly with boundary, and G is a non-oriented graph embedded on $\mathcal{M}$ such that the relative interior of each edge is in the interior of $\mathcal{M}$. G is assumed to be *weighted*: each edge e of G has a non-negative *weight*.

We shall consider families of curves (paths or cycles) drawn on $\mathcal{M}$. We will assume that these families are *regular* with respect to G, in the following sense:

- the set of intersection points between the curves and G is finite, and, at such points, exactly one piece of curve and one edge of G intersect and actually cross;

- if a point is an endpoint of a path, it is in the relative interior of no curve;
- the set of (self-)intersection points between relative interiors of curves is finite, and, at such points, exactly two pieces of curves intersect and actually cross at this point;
- the relative interior of each curve is in the interior of $\mathcal{M}$.

A curve is said to be *regular* if it constitutes itself a regular family of curves. If a regular curve c crosses edges $e_{k_1}, \ldots, e_{k_n}$ of G, its *length*, denoted by $|c|$, is defined as the sum of the weights of $e_{k_1}, \ldots, e_{k_n}$, counting multiplicities. The *length* of a regular family of curves s, denoted by $|s|$, is the sum of the lengths of its curves.

Length additivity holds with this notion of length: if c_1 is a path from a_1 to a_2, and c_2 is a path from a_2 to a_3, in such a way that the concatenation of c_1 and c_2 is a *regular* path c, and such that a_2 is not on G, then $|c| = |c_1| + |c_2|$. We will frequently use this length additivity.

To gain intuition, the reader can assume that all curves are piecewise linear with respect to a fixed triangulation of $\mathcal{M}$ — such a surface being triangulable [57]. This does not cause harm to the interest of this chapter, and avoids topological complications.

3.1.2 Particular case of a polyhedral surface

To illustrate the introduction of this notion of length, let us here give an important particular case, to which we will restrict ourselves in Section 3.5. Assume that $\mathcal{M}$ is a polyhedral surface, whose vertex-edge graph H is weighted.[2]

Let us choose for G the dual graph of H embedded on $\mathcal{M}$, defined as follows. (Here, the graph G can have vertices on the boundary of $\mathcal{M}$, hence the definition is slightly different from the definition given in Chapter 1.) There is one vertex of G in each face of H and one edge of G crossing each edge of H which is in the interior of $\mathcal{M}$. Moreover, for each edge e of H on the boundary of $\mathcal{M}$, a vertex of G is put on e, and linked by an edge to the vertex of G which is in the face of H incident to e. Thus, to each edge e of H corresponds exactly one edge e^* of G, whose weight is defined to be the weight of e.

Any path in H has a length, in the graph-theoretical sense (sum of the weights of the edges of this path in H) equal to the length, in the previous sense (sum of the weights of the edges of G crossed by the path). Each regular embedding of graph or of cycles on $\mathcal{M}$ can be retracted onto a family of paths on H without changing the homotopy classes of the curves, see Figure 3.1: each crossing of a curve with an edge of G corresponds to an edge of a path in H; this transformation is length-preserving. The resulting paths can fail to be simple or disjoint, because

[2]Note that we do not assume each polygon to be Euclidean. In particular, the triangle inequality needs not be satisfied.

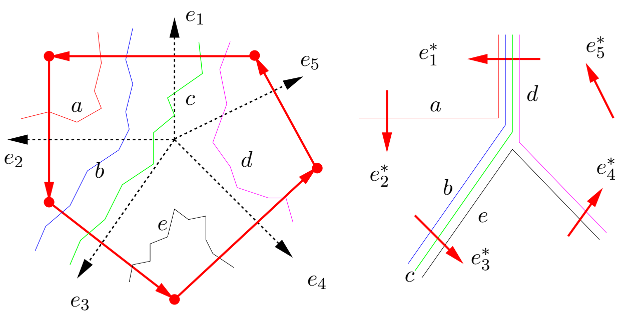


Figure 3.1: A retraction of a regular embedding of graph or of cycles (a, b, c, d, e) on H, in a neighborhood of a vertex of H whose incident edges are $e_1, \ldots, e_5$. The corresponding edges of G are denoted by $e_1^*, \ldots, e_5^*$.

they can pass several times along a given edge or a given vertex; we will however say that they are simple and pairwise disjoint in the sense that it is always possible to perturb them (the perturbation being arbitrarily small) to get simple, disjoint paths; this by analogy with the continuous case, where two paths can go along arbitrarily close together while being disjoint.

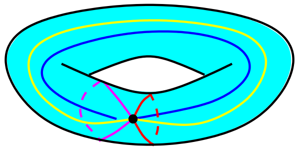
From an algorithmic point of view, we are thus interested in the optimization of curves on H which are simple and pairwise disjoint (in the previous sense). The curves are paths on the graph H; as the curves are allowed to go together along a same edge of H, we will assume that, for each edge e of H, the order of the edges of the paths going along e from left to right is known. The fact that the paths are simple and pairwise disjoint is expressed by a condition on the edges of the paths arriving at each vertex of H; we will come back to this aspect in Section 3.5. We could express all our results in this *combinatorial* framework; however, for the proofs, we need to work with *topological* tools on these curves. We thus choose to describe and prove our results in a framework where all curves are « spread apart » on $\mathcal{M}$.

3.2 Optimization theorems for cut systems

An *embedding of graph* on $\mathcal{M}$ is (in this chapter) a family of simple, disjoint paths, except possibly at common endpoints. An *embedding of cycles* on $\mathcal{M}$ is a family of simple, pairwise disjoint cycles on $\mathcal{M}$.

We will define the notion of cut system, which is an embedding (of graph or of cycles) decomposing the surface into topologically elementary surfaces. We will present two theorems concerning the optimization of cut systems, one for graphs, the other one for cycles.

$\mathcal{M}$ is a surface, and G is a graph embedded on $\mathcal{M}$, such that the length of a curve c on $\mathcal{M}$ is computed as being the sum of the weights of the edges of G crossed by c, see Section 3.1.



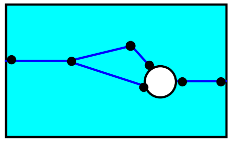


Figure 3.2: Two examples of cut systems by graph: on the left, on a torus, a cut system by graph with one single vertex; on the right, on a sphere with two holes, a cut system by graph with seven vertices.

3.2.1 Cut systems by graph

3.2.1.1 Main result

Definition 3.1 *A* cut system by graph *of $\mathcal{M}$ is a regular embedding of graph $s = (s_1, \ldots, s_N)$ on $\mathcal{M}$ such that:*

- *no path of s is a contractible loop;*
- *for each $i \in [1, N]$, the surface obtained by cutting $\mathcal{M}$ along all paths of s except s_i is a disjoint union of closed disks. Furthermore, the endpoints of s_i are on the boundary of the disk containing s_i.*

See Figure 3.2 for two examples. The set of *vertices* of s is the set of all endpoints of paths in s. The closed disks *defined* by s (or $s \setminus s_i$) are the closed disks obtained by cutting $\mathcal{M}$ along the corresponding paths.

Let $s = (s_1, \ldots, s_N)$ be a cut system by graph, and let $i \in [1, N]$. We define the shortening operation, denoted by shrt_i, as follows. System s is transformed into a family of paths $s' = (s'_1, \ldots, s'_N)$ such that:

- if $k \neq i$, $s'_k = s_k$;
- s'_i has the same endpoints as s_i, is a simple path or loop[3], and its relative interior is in the same open face of $s \setminus s_i$ as the relative interior of s_i;
- s' is a regular embedding of graph;
- s'_i is as short as possible among the paths satisfying the previous conditions.

There are infinitely many s' satisfying these conditions; the set of such s' is denoted by $\mathrm{shrt}_i(s)$. In other words, computing an element of $\mathrm{shrt}_i(s)$ amounts to shortening s_i in the face of $s \setminus s_i$ it is contained in. It is quite easy to notice (see Lemma 3.3 below) that each element $s' \in \mathrm{shrt}_i(s)$ is a cut system by graph, homotopic to s in $\mathcal{M}$ (in the following sense: s_k and s'_k are homotopic in $\mathcal{M}$, for each k), and not longer than s.

Each shortening operation can be viewed as a local optimization (a path is shortened, taking into account the fact that it should not cross the other ones).

[3]Thus, s'_i is a simple path, except that its endpoints can be the same point if s_i is a closed path

In this sense, it is a *greedy* process. The following theorem asserts that iterating shortening operations shrt_i on a system yields, after stabilization of the lengths of the paths, a *global* optimum for *each* of the paths taken individually. Let

$$\mathrm{shrt} = \mathrm{shrt}_N \circ \mathrm{shrt}_{N-1} \circ \cdots \circ \mathrm{shrt}_1,$$

i.e., shrt associates to any cut system the set of systems that can be obtained by applying the N shortening operations successively.

Theorem 3.2 *Let $s^0 = (s^0_1, \ldots, s^0_N)$ be a cut system by graph; for each $k \in \mathbf{N}$, let $s^{k+1} \in \mathrm{shrt}(s^k)$. Then there exists $m \in \mathbf{N}$ such that s^m and s^{m+1} have the same length. Moreover, under these conditions, for each $i \in [1, N]$:*

1. *s^m_i is a shortest path among all simple paths (or loops) isotopic to s^0_i in $\mathcal{M}$ minus the vertices of s^0 which are not endpoints of s^0_i;*
2. *if the vertices of system s^0 are all on the boundary of $\mathcal{M}$, then s^m_i is a shortest path among all paths homotopic to s^0_i in $\mathcal{M}$.*

We shall say that such a system s^m is *optimal*.

Item 1 of Theorem 3.2 looks quite complicated. We now try to explain with a few examples that all the ingredients of this conclusion are necessary. Then, we will deduce a few important corollaries of this theorem.

3.2.1.2 Comments

Let us start with a simple lemma:

Lemma 3.3 *Let s be a cut system by graph, and let $i \in [1, N]$; let $s' \in \mathrm{shrt}_i(s)$. Then s_i and s'_i are paths or loops which are isotopic in the open face of $s \setminus s_i$ containing s_i, to which the endpoints of s_i are appended.*

Proof. Let $\mathcal{M}'$ be the closed disk defined by $s \setminus s_i$ containing s_i. Let us first assume that the endpoints of s_i (and thus also of s'_i) are disjoint on $\mathcal{M}$. Then s_i and s'_i, viewed on $\mathcal{M}'$, are two simple paths in $\mathcal{M}'$, which intersect the boundary of $\mathcal{M}'$ precisely at their (common) endpoints. Hence, s_i and s'_i are isotopic in the interior of $\mathcal{M}'$ to which their endpoints are appended, by Lemma 1.6, page 27.

If s_i (and hence also s'_i) is a loop, the same reasoning applies: after cutting, the loop s_i is transformed into a simple path (with distinct endpoints), for otherwise it would be a loop in a disk, hence contractible, contradicting the definition of a cut system by graph. The same is true for s'_i. □

In particular, s_i and s'_i are isotopic in $\mathcal{M}$ minus all vertices of s which are not endpoints of s_i. We will see later (Section 3.4) that each regular embedding of graph in $\mathcal{M}$ can be extended to a cut system by graph whose vertices are the same.

One can ask whether s^m_i is not, in fact, a shortest path among all simple paths homotopic to s^0_i in $\mathcal{M}$ minus the vertices of s^0 which are not endpoints of s^0_i. This is incorrect. Indeed, let us recall (Figure 1.18, page 27) that two simple

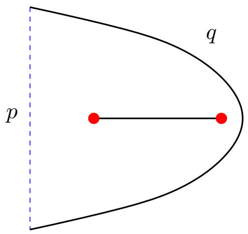


Figure 3.3: The shortest path isotopic to a given path is not necessarily isotopic to this path in $\mathcal{M}$ minus the vertices which are not its endpoints. Here, the small disks represent the vertices of the cut system which are not the endpoints of q.

homotopic paths are not necessarily isotopic. If we consider a compact surface containing Figure 1.18, and if we include q in a cut system by graph, which is then optimized, the path corresponding to q will necessarily be strictly longer than p by Lemma 3.3, and because p and q are not isotopic.[4]

Can we say that s_i^m is a shortest path among all simple paths which are homotopic or isotopic to s_i^0 in $\mathcal{M}$? The answer is, again, negative: Figure 3.3 presents an example where the shortest path p homotopic or isotopic to q in $\mathcal{M}$ is not homotopic to q in $\mathcal{M}$ minus the endpoints of q which are not its endpoints. It is possible to build a cut system by graph containing q, and to optimize this cut system: the path corresponding to q will necessarily be longer than p.

3.2.1.3 Two corollaries

Two embeddings of graph s and t are *isotopic with fixed vertices* if there exists a continuous family of embeddings of graph, whose vertices are the vertices of s, joining s to t. A corollary of Theorem 3.2 is the following:

> **Corollary 3.4** *s^m is a shortest cut system by graph among all cut systems by graph isotopic, with fixed vertices, to s^0.*

Proof. System s^m is isotopic, with fixed vertices, to s^0 by Lemma 3.3. Let $t = (t_1, \dots, t_N)$ be a cut system by graph isotopic, with fixed vertices, to s^0. The existence of such an isotopy immediately yields an isotopy between s_i^0 and t_i, in $\mathcal{M}$ minus the vertices of s^0 which are not endpoints of s_i^0. By Theorem 3.2, s_i^m cannot be longer than t_i. □

Let us consider the set E of all cut systems by graph belonging to a given class of isotopy (with fixed vertices). E can be provided with a reflexive and transitive relation defined by $s \preceq t$ if and only if, for all $i \in [1, N]$, $|s_i| \le |t_i|$. The previous corollary asserts the existence of a smaller element for this relation.

[4] In these examples, we do not give the underlying graph G, hoping that the intuition given by the case of the Euclidean length will be sufficient to convince the reader; but everything really works in our framework, with a suitable graph G.

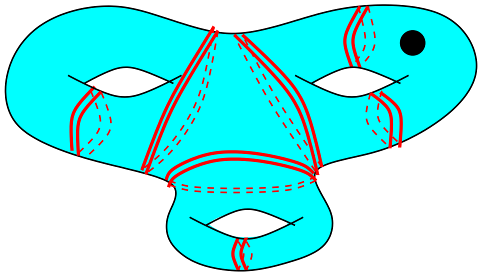


Figure 3.4: A cut system by cycles on a genus three surface with one boundary.

If the graph has only one vertex, we have:

> **Corollary 3.5** *Let s be a cut system by graph having one single vertex. Then, for each i, s_i^m is a shortest loop among all simple loops homotopic to s_i^0.*

Proof. Loop s_i^m is homotopic to s_i^0. Let t_i be a shortest loop among all simple loops homotopic to s_i^0. As s_i^0 and t_i are simple, homotopic, and non-contractible, t_i is homotopic to s_i^0 by Epstein's theorem (Theorem 1.7, page 27). But, by Theorem 3.2, s_i^m is a shortest loop among all simple loops isotopic to s_i^0 in $\mathcal{M}$, hence it cannot be longer than t_i. □

We will apply this corollary later to the case of fundamental systems of loops.

In the conclusion of Corollary 3.5, the word "simple" cannot be removed. Indeed, the shortest loop homotopic to a given loop is not necessarily simple, as proved on Figure 2.6, page 44.

3.2.2 Cut systems by cycles

Definition 3.6 *A* cut system by cycles *of $\mathcal{M}$ is a regular embedding of cycles $s = (s_1, \dots, s_N)$ on $\mathcal{M}$ such that:*

- *no cycle is contractible;*
- *the surface obtained by cutting $\mathcal{M}$ along the cycles of s is a union of (closed) cylinders and pairs of pants, as well as the surface obtained by cutting $\mathcal{M}$ along the cycles of $s \setminus s_i$, for each $i \in [1, N]$.*

See Figure 3.4 for an example. We will see later that each compact, connected, orientable surface, except the sphere and the disk, admit cut systems by cycles. The surfaces *defined* by s (or $s \setminus s_i$) are the connected components of the surface obtained by cutting $\mathcal{M}$ along the corresponding cycles; these are cylinders or pairs of pants.

Let $s = (s_1, \dots, s_N)$ be a cut system by cycles, and let $i \in [1, N]$. We define a shortening operation, again denoted by shrt_i, as follows. System s is transformed into a family of cycles $s' = (s'_1, \dots, s'_N)$ such that:

- if $k \neq i$, $s'_k = s_k$;
- s'_i is simple, and is homotopic to s_i in the cylinder or the pair of pants defined by $s \setminus s_i$ in which s_i is;
- s' is a regular embedding of cycles;
- s'_i is as short as possible among all cycles satisfying the previous conditions.

Again, $\mathrm{shrt}_i(s)$ denotes the set of all such s'. This shortening operation thus transforms a cut system by cycles into another cut system by cycles which is homotopic (in the sense that s_k and s'_k are homotopic cycles for each k), and which is not longer. Again, let

$$\mathrm{shrt} = \mathrm{shrt}_N \circ \mathrm{shrt}_{N-1} \circ \cdots \circ \mathrm{shrt}_1.$$

We now have the following optimality theorem:

Theorem 3.7 *Let $s = (s_1, \ldots, s_N)$ be a cut system by cycles. For each $k \in \mathbf{N}$, let $s^{k+1} \in \mathrm{shrt}(s^k)$. For some $m \in \mathbf{N}$, s^m and s^{m+1} have the same length. Under these conditions, for each $i \in [1, N]$, s^m_i is a shortest cycle among all (not necessarily simple) cycles which are homotopic to s^0_i.*

We shall again say that such a system is *optimal*. Consider now the set E of cut systems by cycles which belong to a given homotopy class. It is possible to provide E with a reflexive and transitive relation defined by $s \preceq t$ if and only if, for each $i \in [1, N]$, $|s_i| \leq |t_i|$. The previous theorem asserts the existence of a smaller element for this relation, and, furthermore, each smaller element is made of cycles which are, individually, shortest cycles in their homotopy class.

3.2.3 Variations

Before proving these theorems, let us simply mention that there exist slight variations for the optimization processes. These variations relax the definition of a cut system: the second condition of this definition is replaced by the weaker condition that cutting $\mathcal{M}$ along s should yield a union of elementary surfaces (disks, cylinders, or pairs of pants).

For the first variation, the operation shrt_i consists in shortening s_i in $\mathcal{M}$ cut along the curves of $s \setminus s_i$ (which is not necessarily a topological disk in the case of cut systems by graph), while maintaining the homotopy class of s_i. We used this approach in [40] for fundamental systems of loops. Actually, in that paper, the complexity of the algorithm is not as good as in the present chapter (each shortening step can require to look for a simple shortest homotopic path in some cylinder), and the proof is longer. In the case of cycles, this also makes things more tricky, because cutting $\mathcal{M}$ along $s \setminus s_i$ can create spheres with four boundaries or tori with one boundary.

A second variation is to define, for each i, two operations shrt_i and $\overline{\mathrm{shrt}_i}$, consisting in shortening s_i, in the surface obtained by cutting $\mathcal{M}$ along s, on the left (resp. right) of s_i. The algorithm has the same asymptotic complexity and is as natural as the one presented here. But proofs seem slightly easier in the present framework.

3.3 Proofs of the optimization theorems for cut systems

This section contains the proofs of Theorems 3.2 and 3.7 of the last section. We will start by proving the first theorem in a particular case where the proof is quite short. We will then deduce the general case with a few topological considerations. The proof of the second theorem uses arguments which are analogous to the first theorem, but is really longer and more complex. The reason is that the whole proof uses the universal covering space, and that it is necessary to first state simple properties on curves on cylinders and pairs of pants which are as short as possible in their homotopy class.

A key ingredient for the proofs of these theorems is the notion of crossing word, which enables to code the way curves cross. We introduce it now.

3.3.1 Crossing words

3.3.1.1 Preliminaries on words

We first introduce some classical notions on words (see for example [97, Chapter 1] or [7, Chapter 0]). Let X be a non-empty set of symbols; let $\bar{X}$ be the set of symbols $\{\bar{x},\ x \in X\}$. A *letter* is an element of $X \cup \bar{X}$; we call *alphabet*, and denote by A, the set of these letters. A *word* on A is a finite sequence of elements of A. The empty word is denoted by ε. The *length* of a word is the number of its letters. Two words w_1 and w_2 being given, their *concatenation* is denoted by $w_1 w_2$. A word v is a *factor* of some word w if there exist two words u_1 and u_2 such that w can be written $u_1 v u_2$.

Let w be a word. If w contains a factor of the form $a\bar{a}$ or $\bar{a}a$, where a is some symbol, let w' be the word obtained by removing this factor from w; we shall say that w' is deduced from w by an *elementary a-reduction.* An *elementary reduction* is an elementary a-reduction for some a. An *(a-)reduction* is a succession of zero, one or several elementary (a-)reductions. A word is said *(a-)irreducible* if it can be applied no elementary (a-)reduction.

Lemma 3.8 *Let w be a word; there exists exactly one irreducible (resp. a-irreducible) word w' such that w reduces (resp. a-reduces) to w'.*

Proof. The proof is elementary, but we mention it because the same techniques will be used later. We prove the result in the case of reductions, the arguments being the same in the case of a-reductions.

The proof relies on the *confluence property*: if w, w_1, and w_2 are words such that w elementarily reduces to w_1 and w_2, then there exists w_3 such that w_1 and w_2 both reduce to w_3. It is a simple case analysis: the factors of w removed to obtain w_1 and w_2 are either the same in w, or disjoint in w, or they partly overlap in w. If they are the same, the property is obviously true. If they are disjoint in w, then it is also true, because we can take for w_3 the word w where the two factors have been removed. Otherwise, both factors partly overlap, and there is a factor of w_3 of the form $a\bar{a}a$ or $\bar{a}a\bar{a}$; then (up to swapping w_1 and w_2), the first reduction consists in the removal of the first two letters of this factor, and the second one consists in the removal of the last two letters. It implies that, actually, $w_1 = w_2$, and the result follows.

The lemma is deduced from this property by induction on the number of letters of w. It is equals 0 or 1, the result is obvious. If w is not irreducible, let w_1 and w_2 be any two words such that w elementarily reduces to w_1 and w_2; it is sufficient to prove that each of w_1 and w_2 reduces to one single irreducible word, which is the same for both words. We know that there exists w_3 such that w_1 and w_2 reduce to w_3. By the induction hypothesis, each of w_1, w_2, and w_3 reduces to one single irreducible word, which must thus be the same for w_1, w_2, and w_3; this concludes the proof. □

We will denote by $\mathrm{red}(w)$ (resp. $\mathrm{red}_a(w)$) the unique irreducible (resp. a-irreducible) word w' such that w reduces (resp. a-reduces) to w'. We will say that a word w is *parenthesized* if $\mathrm{red}(w) = \varepsilon$.[5]

Lemma 3.9 *Let w_1 and w_2 be two words. Then w_1w_2 is parenthesized if and only if w_2w_1 is parenthesized.*[6]

Proof. By symmetry, it is sufficient to prove one implication. This is done by induction on the length of w_1w_2, the result being trivial if the length equals 0 or 1. Let w be a factor in w_1w_2 of the form $a\bar{a}$ (the case $\bar{a}a$ being analogous).

If w is entirely contained in w_1, let w'_1 be the word obtained from w_1 by this elementary reduction; by the induction hypothesis, and since w'_1w_2 is parenthesized, $w_2w'_1$ is also parenthesized; then so is w_2w_1. The case where w is entirely included in w_2 can be treated in a similar way.

Hence, there remains the case where $w_1 = w'_1a$ and $w_2 = \bar{a}w'_2$. In this case, since w_1w_2 is parenthesized, so is $w'_1w'_2$, hence by the induction hypothesis, so is $w'_2w'_1$, hence so is $\bar{a}w'_2w'_1a = w_2w_1$. □

3.3.1.2 Crossing word

Let C be a family of curves (paths, possibly closed or infinite, or cycles) which are simple and pairwise disjoint, on an oriented surface. Let A be the set of letters of the form c or $\bar{c}$, where $c \in C$. Let p be a path intersecting C in a generic way (the number of intersection points is finite, and, at such points, p crosses one curve of C). Let us walk along p and, each time we encounter a crossing with a curve c, let us write the letter c or $\bar{c}$, according to the orientation of the crossing. The resulting word is called the *crossing word* of p with C, denoted by C/p.

All surfaces considered will be assumed to be oriented. In particular, $\mathcal{M}$ is provided with an orientation, which naturally induces an orientation of its universal covering space $(\widetilde{\mathcal{M}}, \pi)$.

The following lemmas show that the parenthesized words appear naturally in the crossing words.

[5] Parenthesized words are also called *Dyck words*, see [12, p. 35].

[6] This lemma is obvious for the reader aware of the fact that the set of reduced words is a group (the free group), since in a group $u.v$ is the unit element if and only if $v.u$ is the unit element.

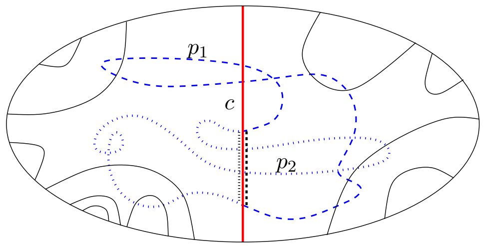


Figure 3.5: A step of the proof of Lemma 3.10.

Lemma 3.10 *Let C be a family of simple, pairwise disjoint curves on an oriented surface, such that each of these curves separate the surface into two connected components. Let p be a closed path intersecting generically C. Then the crossing word C/p is parenthesized.*

Proof. We prove the result by induction on the number of crossings between p and the curves of C. The result is trivial if p does not cross any curve of C. Hence, let us assume that there is at least one crossing between p and a curve $c \in C$.

As p is a closed path, it must cross c at least once with the opposite orientation. Consider now p as a cycle γ (*i.e.*, let us forget the basepoint of p). The two crossings split the cycle γ into two paths p_1 and p_2 (Figure 3.5). For $k = 1, 2$, it is possible to extend p_i to a closed path p'_i, so that $C/p'_i = C/p_i$, by adding a piece of a path which goes along a part of c. By the induction hypothesis, C/p_1 and C/p_2 are parenthesized. In addition, C/p equals, up to cyclic permutation,

$$c\ \ (C/p_1)\ \ \bar{c}\ \ (C/p_2) \quad \text{or} \quad \bar{c}\ \ (C/p_1)\ \ c\ \ (C/p_2).$$

By Lemma 3.9, we obtain that C/p is parenthesized. □

The following lemma is a corollary of Lemma 3.10.

Lemma 3.11 *Let C be a family of simple, pairwise disjoint paths in $\mathcal{M}$, which intersect $\partial\mathcal{M}$ exactly at their endpoints; also, let p be a contractible closed path in $\mathcal{M}$, intersecting C generically. Then, C/p is parenthesized.*

Proof. In the universal covering space $(\widetilde{\mathcal{M}}, \pi)$ of $\mathcal{M}$, let us consider the set of all lifts $\widetilde{C}$ of the paths in C. These lifts are simple and pairwise disjoint, and each of them separates $\widetilde{\mathcal{M}}$ into two connected components, by Lemma 1.12, page 32.

Let $\widetilde{p}$ be a lift of p; as p is contractible, $\widetilde{p}$ is closed. By Lemma 3.10, $\widetilde{C}/\widetilde{p}$ is parenthesized. But C/p is obtained from this word by projection, that is, each letter of $\widetilde{C}/\widetilde{p}$, which denotes a lift of some curve in C, is replaced by the projection of this curve in $\mathcal{M}$; thus C/p is parenthesized. □

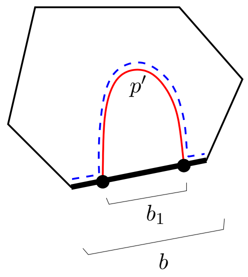


Figure 3.6: The proof of Proposition 3.14.

3.3.2 Proof of Theorem 3.2

3.3.2.1 Proof of Theorem 3.2 in a particular case

Definition 3.12 *A* cut system by paths $s = (s_1, \dots, s_N)$ *is a cut system by graph such that the* $2N$ *endpoints of the paths* s_i *are pairwise disjoint and on the boundary of* $\mathcal{M}$.

We will prove the following particular case of Theorem 3.2:

Theorem 3.13 *Let* $s^0 = (s_1^0, \dots, s_N^0)$ *be a cut system by paths; for each* $k \in \mathbf{N}$, *let* $s^{k+1} \in \text{shrt}(s^k)$. *Then there exists* $m \in \mathbf{N}$ *such that* s^m *and* s^{m+1} *have the same length. Moreover, for each* $i \in [1, N]$, s_i^m *is a shortest path among all paths homotopic to* s_i^0 *in* $\mathcal{M}$.

Clearly, if s^0 is a cut system by paths, then s^k is also a cut system by paths, for each $k \in \mathbf{N}$.

Each surface considered here is assumed to be equipped with an embedding of a weighted graph G, such that the length of a regular path is computed as being the sum of the weights of the edges of G crossed by the path. The following proposition will be used several times:

Proposition 3.14 *Let* D *be a closed disk, and let* b *be a simple path on the boundary of* D. *Assume that there is no simple path, in* D, *having the same endpoints as* b *and which is shorter than* b.

Let p *be a regular path in* D, *whose endpoints are on* b. *Let* b_1 *be the subpath of* b *whose endpoints are the endpoints of* p. *Then* $|b_1| \leq |p|$.

Proof. Let p' be the path obtained from p after the removal of its loops; p' is simple and not longer than p. Now, if $|b_1| > |p|$ (Figure 3.6), then $|b_1| > |p'|$; but then, by replacing, in b, the subpath b_1 by p', we would obtain a simple path having the same endpoints as b_1, and whose length is smaller (p being regular, its endpoints are not on G), which contradicts the hypothesis. □

For $i \in [1, N]$, let c_i be a path which "goes along" s_i^0, as follows:

- c_i is disjoint from all paths in s^0;

- there exist two paths p_i and q_i on $\partial\mathcal{M}$, with length zero, whose relative interiors do not meet any path of s^0, and such that s_i^0 is homotopic to $p_i.c_i.q_i$.

In particular, c_i has its endpoints on the boundary of $\mathcal{M}$, and these endpoints do not meet s^0.

Let $i \in [1, N]$ be fixed. Let t_i be a shortest path among all paths homotopic to c_i. In particular, t_i is not necessarily simple. Note that if s^0 and t_i constitute a regular family of paths then, by Lemma 3.11, s^0/t_i is parenthesized (since $t_i.c_i^{-1}$ is contractible and c_i does not cross s^0). The idea of the proof is to show that the shortening operations "uncross" s^0 and t_i.

Let $s = (s_1, \dots, s_N)$ be a cut system by paths at some stage of the process. Let $j \in [1, N]$; let $r \in \text{shrt}_j(s)$. Even if it means slightly perturbing t_i, we may assume that s (resp. r) and t_i constitute a regular family. In the remaining part of the proof, we will write the crossing words of r or s with t_i in a slightly different way, by omitting the "s" and the "r" (for example, we shall write $1\ \overline{5}\ \overline{2}$ instead of $s_1\ \overline{s_5}\ \overline{s_2}$). This allows for example to say that $s/t_i = r/t_i$ if t_i crosses neither s_j nor r_j. In a similar way, we will write red_j instead of red_{s_j} or red_{r_j}.

Lemma 3.15 $\text{red}_j(s/t_i) = \text{red}_j(r/t_i)$.

Proof. Let us consider a maximal subpath $\hat{t}_i$ of t_i which crosses no path of $s \setminus s_j = r \setminus r_j$: $\hat{t}_i$ belongs to a closed disk D defined by $s \setminus s_j$, hence $s/\hat{t}_i$ and $r/\hat{t}_i$ are only made of letters of the form j or $\bar{j}$.[7] These words are empty if s_j is not inside D; we now assume that s_j belongs to D.

Paths s_j and r_j have the same endpoints, which are on the boundary of D, and no other point of s_j and r_j is on the boundary of D. Moreover, $\hat{t}_i$ has its endpoints on the boundary of D, and these endpoints are different from the endpoints of s_j and r_j. From this, it comes that s_j separates the endpoints of $\hat{t}_i$ if and only if r_j separates the endpoints of $\hat{t}_i$ (Jordan curve theorem). If it is the case, $\text{red}_j(s/\hat{t}_i)$ and $\text{red}_j(r/\hat{t}_i)$ both equal either j or $\bar{j}$; otherwise, we have $\text{red}_j(s/\hat{t}_i) = \varepsilon = \text{red}_j(r/\hat{t}_i)$.

It is thus possible to write t_i as a concatenation of paths $\hat{t}_{i,1}, \dots, \hat{t}_{i,n}$ such that each path $\hat{t}_{i,k}$ intersects generically s, and such that, for each k, $\text{red}_j(s/\hat{t}_{i,k}) = \text{red}_j(r/\hat{t}_{i,k})$. The result follows. □

The following proposition proves, roughly but not exactly, that the application of shrt_j to s has for effect to proceed to all possible j-reductions on the crossing word s/t_i. This is in fact true if we accept to replace t_i by a path t_i' having the same properties as t_i:

Proposition 3.16 *There exists a path* t_i' *on* $\mathcal{M}$ *such that:*

- t_i' *has the same properties as* t_i *(the lengths and homotopy classes of* t_i *and* t_i' *are equal;* t_i' *and* s *(resp.* r*) constitute a regular family);*

[7] In this proof, $\hat{t}_i$ does not cross generically s because some endpoints of $\hat{t}_i$ can be on s; it is nevertheless possible to define in a natural way the crossing word, with the convention that the intersections between s and the endpoints of $\hat{t}_i$ are not taken into account.

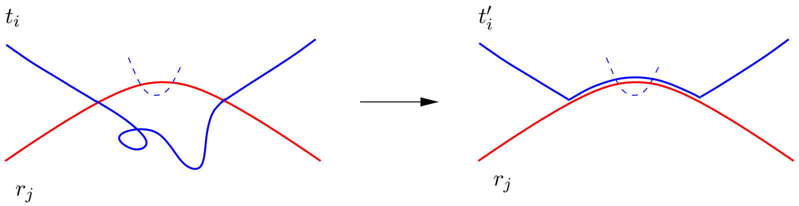


Figure 3.7: The uncrossing operation between curves r_j and t_i in the proof of Proposition 3.16. Path t_i is not necessarily simple and r_j can cross other pieces of t_i.

- $r/t'_i = \mathrm{red}_j(s/t_i)$.

Proof. If r/t_i is j-irreducible, there is nothing to prove by Lemma 3.15. Assume that some elementary j-reduction is possible on r/t_i. Let $\hat{r}_j$ and $\hat{t}_i$ be the two subpaths of r_j and t_i associated with this reduction. These two paths do not cross r, hence are in a closed disk defined by r. It is obvious that $|\hat{t}_i| \leq |\hat{r}_j|$, and there is in fact equality by Proposition 3.14. We can thus "uncross" the intersections, by replacing, in t_i, the part $\hat{t}_i$ by a path which goes along $\hat{r}_j$ (Figure 3.7); this changes neither the length of the path nor its homotopy class, and removes two crossings. The proof is finished by induction. □

Since s^0/t_i is parenthesized, the previous proposition also proves that, at some stage of the process, s/t_i is the empty word (up to changing t_i by a path having the same properties), because the length of s/t_i strictly decreases at each application of shrt. The following proposition aims at studying what happens once s and t_i are disjoint.

Proposition 3.17 *Let us assume that s and t_i are disjoint and that $j = i$. Then $|r_j| = |t_i|$.*

Proof. Path t_i is in a topological disk defined by s. Up to removing the loops of t_i, which are contractible, we can assume that t_i is simple. Thus, path $p_i.t_i.q_i$ is also simple, and of the same length as t_i; it has the same endpoints as s_j and is in a closed disk defined by s. Slightly perturb $p_i.t_i.q_i$ into a simple path u_i which meets $\partial\mathcal{M}$ exactly at its endpoints and which has the same length as t_i. Then, by definition of $r \in \mathrm{shrt}_j(s)$, we have $|r_j| \leq |u_i| = |t_i|$. As t_i is a shortest path within its homotopy class, and because p_i and q_i have zero length, there is in fact equality. □

Let us conclude the proof of Theorem 3.13, which is a particular case of Theorem 3.2.

Proof of Theorem 3.13. Let s and $s' \in \mathrm{shrt}_j(s)$ be two cut systems considered in the course of the optimization; by abuse of notation, we will write $s' = \mathrm{shrt}_j(s)$ (which infers that, in the course of the optimization process, s' is chosen as element of $\mathrm{shrt}_j(s)$).

Let $i \in [1, N]$, and let t_i^0 be a shortest path among all paths homotopic to c_i. Assume that t_i^0 and s^0 constitute a regular family of paths. By Lemma 3.11, we know that s^0/t_i^0 is parenthesized. By Proposition 3.16, it is possible to build a sequence $(t_i^k)_{k\in\mathbf{N}}$ of shortest paths homotopic to t_i^0 such that the length of the crossing word s^k/t_i^k strictly decreases, until it becomes the empty word ε at some stage p. By applying again $i-1$ times Proposition 3.16, we get that there exists t_i, having the same properties as t_i^0, which does not cross $\mathrm{shrt}_{i-1} \circ \cdots \circ \mathrm{shrt}_1(s^p)$. By Proposition 3.17, $|s_i^{p+1}|$ and $|t_i|$ have the same length.

The preceding reasoning still shows that, for each i, the sequence $(|s_i^k|)_{k\in\mathbf{N}}$ becomes stationary at some point $p+1$, and that, once it is the case, s_i^{p+1} is a shortest homotopic path. There remains to show that this sequence is stationary as soon as two consecutive terms are equal, *i.e.*, $|s^m| = |s^{m+1}|$. (This fact provides a quite simple criterion to know when the optimization process can be stopped.) Thus, let us assume that two cut systems s and $s' = \mathrm{shrt}(s)$ have the same length, and let $i \in [1, N]$; it suffices to prove that s_i has the same length as t_i (a shortest path homotopic to c_i).

The word s/t_i is parenthesized; let us assume that an elementary j-reduction is possible on this word, and let $\hat{s}_j$ and $\hat{t}_i$ be the two subpaths associated to this reduction. We will prove that both endpoints have the same length. This will imply that it is possible, like in Proposition 3.16, to modify t_i without changing its length nor its homotopy class to proceed to this j-reduction. Hence, by induction, it will be possible to assume $s/t_i = \varepsilon$ (up to changing t_i).

Assume $j \neq 1$; only $(\mathrm{shrt}_1(s))_1$ appears in the word $(\mathrm{shrt}_1(s))/\hat{t}_i$. By Lemma 3.11, this word is parenthesized and, by Proposition 3.14, it is possible to iteratively proceed to the reductions, by replacing $\hat{t}_i$ by a path $\hat{t}'_i$ which has the same length and the same homotopy class as $\hat{t}_i$, and which does not cross $\mathrm{shrt}_1(s)$. By iterating this process, we obtain the existence of some path $\hat{t}''_i$, of the same length and the same homotopy class as $\hat{t}_i$, which crosses no path of $s'' := \mathrm{shrt}_{j-1} \circ \cdots \circ \mathrm{shrt}_1(s)$. Moreover, the endpoints of $\hat{t}''_i$ are on s''_j; as the elements of $\mathrm{shrt}_j(s'')$ have the same length as s'', then $\hat{t}''_i$ cannot be shorter than $\hat{s}_j$. Thus $|\hat{s}_j| = |\hat{t}''_i| = |\hat{t}_i|$.

We can thus assume, up to a change of t_i, that $s/t_i = \varepsilon$. By applying $i-1$ times Proposition 3.16, we can assume, up to changing t_i, that $(\mathrm{shrt}_{i-1} \circ \cdots \circ \mathrm{shrt}_1(s))/t_i = \varepsilon$. Proposition 3.17 then proves that $|(\mathrm{shrt}(s))_i| = |t_i|$; because $|(\mathrm{shrt}(s))_i| = |s_i|$, we deduce that $|s_i| = |t_i|$, which was to be proved. □

3.3.2.2 Proof of Theorem 3.2 in the general case

We will use Theorem 3.13 to prove the more general Theorem 3.2. The idea is to remove small disks to $\mathcal{M}$ around the vertices of the cut system by graph s^0 to obtain a surface with boundary $\hat{\mathcal{M}}$, to define from s^0 a cut system by paths $\hat{s}^0$ on $\hat{\mathcal{M}}$, and to prove that the optimizations of s^0 and $\hat{s}^0$ are processed in the same way.

We have seen that the map shrt_i is not defined in a unique way: there are several s'_i satisfying the required conditions so that $s' \in \mathrm{shrt}_i(s)$. In particular, intuitively, the exact position of s'_i does not matter: if we deform a subpath of s'_i in the interior of a face of G, while keeping the paths simple and pairwise disjoint, it should be clear for the reader that the course of the optimization will remain unchanged. We will now formalize this fact.

Let s be a cut system by graph; for each vertex v of s, let D_v be a closed disk containing v in its interior[8] and disjoint from G. We assume that D_v is chosen in such a way that, for each j, s_j crosses the boundary of D_v a minimal number of times (as many times as s_j has endpoints at v). We also assume that the disks D_v are disjoint. The existence of such disks is easy to prove if we assume that all the curves considered are piecewise linear with respect to a fixed triangulation of $\mathcal{M}$, and comes, in the general case, from Lemma 1.3, page 18.

Lemma 3.18 *Let $i \in [1, N]$, and let $s' \in \mathrm{shrt}_i(s)$. There exists $s'' \in \mathrm{shrt}_i(s)$ such that:*

- *s_i and s''_i coincide inside the disks D_v;*
- *the course of the optimization process is the same, starting with s' or with s''. More precisely, let f be the composition, in an arbitrary order, of a finite number of operations rac_j; then, for each element in $f(s')$, there exists an element in $f(s'')$ whose curves have the same lengths.*

Proof. We begin by building disks d_v, D'_v, and D''_v around each vertex v, in such a way that:

$$d_v \sqsubset D_v \sqsubset D'_v \sqsubset D''_v,$$

the notation $A \sqsubset B$ meaning that the closure of A is included in the interior of B. We may assume that the disks d_v, D'_v, and D''_v are disjoint from G and that the disks D''_v are pairwise disjoint. We may also assume that s'_i meets the boundary of d_v a minimal number of times.

Let v be an endpoint of s_i. First assume that v has degree 1 (considered as vertex of s), which implies that it is on the boundary of $\mathcal{M}$, by the definition of a cut system by graph. We will show that there exists a homeomorphism h_1 from $\mathcal{M}$ to $\mathcal{M}$, fixed outside D''_v and at v, such that s_i and $h_1(s'_i)$ are the same inside D_v. We build a first homeomorphism which sends d_v to D'_v, while fixing v and $\mathcal{M} \setminus D''_v$ (Figure 3.8, on the left and in the middle). By composition with a second homeomorphism, fixing v and $\mathcal{M} \setminus D'_v$ (Figure 3.8, on the right), we can ensure that the image of s'_i inside D_v is any simple path having v as endpoint (Jordan–Schönflies theorem), hence in particular the piece of s_i which is inside D_v.

If v has degree greater than 1, then the paths different from s_i decompose the disk D''_v into sectors (see Figure 3.9). If s_i is a closed path with basepoint v, then the endpoints of s_i are in two different sectors of D''_v: otherwise, s_i would be the boundary of a face of s, which is a disk, and s_i would be contractible, thus contradicting the definition of a cut system by graph. The same reasoning can be applied as before in each sector of D''_v containing an endpoint of s_i.

[8]Here, the notions of closure, interior, and boundary are taken relatively to the ambiant space $\mathcal{M}$. In particular, if v is on the boundary of $\mathcal{M}$, the interior of D_v contains a piece of boundary of $\mathcal{M}$. In this case, D_v shall be chosen so that its boundary is homeomorphic to a closed disk.

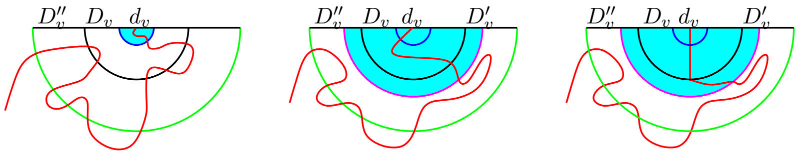


Figure 3.8: The creation of the homeomorphism in the proof of Lemma 3.18. The path on the figure is a piece of s'_i.

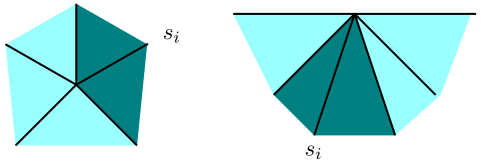


Figure 3.9: The sectors of D''_v defined by the paths of $s \setminus s_i$, in the proof of Lemma 3.18. On the left, the case where v is in the interior of $\mathcal{M}$; on the right, the case where v is on the boundary of $\mathcal{M}$.

Using the same technique, we can, for each sector of a disk D''_w containing no endpoint of s_i, create a homeomorphism fixed outside this sector such that the image of s'_i by this homeomorphism does not enter D_w. Finally, we have built a homeomorphism h, fixed on the vertices of s, on the paths of $s \setminus s_i$, and outside the disks D''_v, such that $h(s'_i)$ and s_i coincide in all disks D_v.

Let $s'' = h(s')$; clearly, $s'' \in \mathrm{shrt}_i(s)$. Furthermore, for any j and for any systems t and t', we have $t' \in \mathrm{shrt}_j(t)$ if and only if $h(t') \in \mathrm{shrt}_j(h(t))$. This last assertion shows that the course of the optimization is the same, continuing with s' or s'': the lengths of the considered systems will be the same during the whole optimization. □

Proof of Theorem 3.2. Let V be the set of vertices of s. Let s and $s' \in \mathrm{shrt}_i(s)$ be two cut systems by graph considered in the course of the optimization. First of all, by Lemma 3.3, we know that s_i and s'_i are isotopic in $\mathcal{M}$ minus the elements of V which are not endpoints of s_i. In particular, s_i and s'_i are homotopic in $\mathcal{M}$.

We build disks D_v, as indicated earlier, such that the paths s^0 cross the boundary of the disks D_v a minimal number of times. By Lemma 3.18, we may assume that, at each shortening step, the subpaths of the paths of the system in the interior of the disks D_v do not change.

Let U be the union of the interiors of the disks D_v; the space $\hat{\mathcal{M}} := \mathcal{M} \setminus U$ is a surface with boundary. For any cut system by graph s considered in the course of the optimization process, one sees that the intersection of s with $\hat{\mathcal{M}}$ is a cut system by paths, denoted by $\hat{s}$. If s and $s' \in \mathrm{shrt}_i(s)$ are two cut systems by graphs considered, then $\hat{s}'$ belongs to $\mathrm{shrt}_i(\hat{s})$. Indeed, to each simple path in $\hat{\mathcal{M}}$, disjoint from $\hat{s}_j$ for all $j \neq i$, having the same endpoints as $\hat{s}_i$, corresponds a simple path (or loop) in $\mathcal{M}$, disjoint from s_j for $j \neq i$, having the same endpoints as s_i, its relative interior being in the same face of $s \setminus s_i$ as the relative interior of s_i; moreover, these paths have the same length.

The sequence $(\hat{s}^k)_{k\in\mathbf{N}}$ thus satisfies $\hat{s}^{k+1} \in \mathrm{shrt}(\hat{s}^k)$. We can apply Theorem 3.13: there exists $m \in \mathbf{N}$ such that $|\hat{s}^{m+1}| = |\hat{s}^m|$, and, for any i, $\hat{s}_i^m$ is a shortest path among all homotopic paths in $\hat{\mathcal{M}}$.

In a first step, let us assume that all vertices of s^0 are on the boundary of $\mathcal{M}$; let $i \in [1, N]$. Let t_i be a path homotopic to s_i^0 in $\mathcal{M}$, as short as possible (but not necessarily simple). Path t_i is thus homotopic, in $\mathcal{M}$, to some path t_i', of the same length as t_i, coinciding with s_i^0 inside each of the disks D_v. In particular, t_i' does not enter the disk D_v, if v is not an endpoint of s_i^0. We obtain a path $\hat{t}_i'$, included in $\hat{\mathcal{M}}$, having the same length as t_i, homotopic in $\mathcal{M}$ to $\hat{s}_i^0$. Since all vertices of s_i^0 are on the boundary of $\mathcal{M}$, this implies that $\hat{t}_i'$ is homotopic *in* $\hat{\mathcal{M}}$ to $\hat{s}_i^0$, and thus that $\hat{s}_i^m$ cannot be longer than t_i. This proves Item 2 of Theorem 3.2. We now prove the first item.

Let $i \in [1, N]$. Call v_0 and v_1 the endpoints of s_i^0; let $V' = V \setminus \{v_0, v_1\}$. Let t_i be a shortest simple path (or loop) isotopic to s_i^0 in $\mathcal{M} \setminus V'$. To prove the result, it suffices to prove that $\hat{s}_i^m$ and t_i have the same length.

Let $h : [0, 1] \times [0, 1] \to \mathcal{M} \setminus V'$ be an isotopy between s_i^0 and t_i: for each $t \in [0, 1]$, $h(t, .)|_{[0,1)}$ is one-to-one, $h(t, 0) = v_0$ and $h(t, 1) = v_1$. For $k = 0, 1$, $h^{-1}(D_{v_k})$ is a neighborhood of the compact set $[0, 1] \times \{k\}$; thus there must exist $\varepsilon > 0$ such that

$$h([0, 1] \times [0, \varepsilon]) \subseteq D_{v_0} \quad \text{and} \quad h([0, 1] \times [1 - \varepsilon, 1]) \subseteq D_{v_1}.$$

Let h' be the restriction of h to $[0, 1] \times [\varepsilon, 1 - \varepsilon]$. Let $r : \mathcal{M} \setminus V \to \mathcal{M} \setminus U$ be a continuous map which is the identity on $\mathcal{M} \setminus U$ and which sends, for each $v \in V$, $D_v \setminus \{v\}$ to the boundary of D_v. Since h is an isotopy in $\mathcal{M} \setminus V'$, and since $h(., k) = v_k$ for $k = 0, 1$, the map $h'' := r \circ h'$ is well-defined, continuous, and its image set is inside $\hat{\mathcal{M}}$. Furthermore:

- $h''(., \varepsilon)$ is on the boundary of D_{v_0}, and $h''(., 1 - \varepsilon)$ is on the boundary of D_{v_1};
- $h''(0, .)$ is the concatenation of some path on the boundary of D_{v_0}, of $\hat{s}_i^0$, and of some path on the boundary of D_{v_1};
- $h''(1, .)$ is the concatenation of some path on the boundary of D_{v_0}, of some path $\hat{t}_i$ which has the same length as t_i, and of some path on the boundary of D_{v_1}.

From this, we deduce that $\hat{s}_i^0$ is homotopic, in $\hat{\mathcal{M}}$, to a path which has the same length as t_i; hence $|s_i^m| = |t_i|$, which was to be proved. □

3.3.3 Proof of Theorem 3.7

We now prove Theorem 3.7. The ideas are similar to the proof of Theorem 3.13, but the proof is more sophisticated. This comes from the fact that we need to shorten cycles within a given homotopy class in a cylinder or a pair of pants, in contrast to the situation with embeddings of graphs. In this section, regularity of families of curves is always assumed, although omitted in most statements.

3.3.3.1 Crossing words sets

Let C be a family of curves (paths or cycles) on $\mathcal{M}$, which are simple and pairwise disjoint. We want to define the analogue of the crossing word between a given lift of a cycle in $\mathcal{M}$ and the lifts of the curves in C.

We first introduce the notion of *geometric lift*. Recall from Chapter 1 that a lift of a cycle $\gamma : \mathbf{R} \to \mathcal{M}$ (such that $\gamma(1+.) = \gamma(.)$) is an infinite path $\widetilde{\gamma} : \mathbf{R} \to \widetilde{\mathcal{M}}$ such that $\pi \circ \widetilde{\gamma} = \gamma$. Let us introduce an equivalence relation $\sim$ on the lifts of γ by: $\widetilde{\gamma} \sim \widetilde{\gamma}'$ if and only if there exists $k \in \mathbf{Z}$ such that $\widetilde{\gamma}(.) = \widetilde{\gamma}'(k + .)$. From a geometric point of view, two lifts which are in the same equivalence class are the same (in particular, their image sets on $\mathcal{M}$ are the same). A *geometric lift* is an equivalence class of this relation. Obviously, for a contractible cycle γ, the notions of lift and of geometric lift coincide; by definition, they also coincide for paths. This notion enables to define the crossing word between a path in $\widetilde{\mathcal{M}}$ and the *geometric* lifts of the curves in C: indeed, since C contains simple and pairwise disjoint curves, the geometric lifts of C correspond precisely to the connected components of $\pi^{-1}(\bar{C})$, where $\bar{C} \subseteq \mathcal{M}$ means the union of the image sets of the curves in C.

Let γ be a non-contractible cycle in $\mathcal{M}$ which intersects the curves in C generically, and let $\widetilde{\gamma}$ a lift of γ. We wish to avoid the definition of the crossing word of $\widetilde{\gamma}$ with a set of geometric lifts, because this crossing word is potentially infinite. Instead, we restrict ourselves to portions of $\widetilde{\gamma}$, called *lifted periods*. A *lifted period* of $\widetilde{\gamma}$ is a path of the form $\widetilde{\gamma}(a + .)|_{[0,1]}$ for some $a \in \mathbf{R}$.

Let $\widetilde{C}$ be the set of the geometric lifts of C. We will use the following convention: if c is a curve in C, then the geometric lifts of c will be denoted by c^α, $\alpha \in I_c$. (It is clear that the geometric lifts of a curve are enumerable, and we can thus choose $I_c \subseteq \mathbf{N}$, but the indexation set I_c does not matter at all.) We consider the words on the alphabet made of the letters c^α and $\bar{c}^\alpha$, where $c^\alpha \in \widetilde{C}$. The *crossing words set* of $\widetilde{\gamma}$ with $\widetilde{C}$, denoted by $[\widetilde{C}/\widetilde{\gamma}]$, is the set of crossing words $\widetilde{C}/\widetilde{p}$, over all lifted periods $\widetilde{p}$ of $\widetilde{\gamma}$.[9] Our first task will be to show that the crossing words set $[\widetilde{C}/\widetilde{\gamma}]$ is entirely determined once we know one of its elements.

We note that $\widetilde{\gamma}$ induces an automorphism $\tau_{\widetilde{\gamma}}$ in $\widetilde{\mathcal{M}}$, as follows. Let $v \in \widetilde{\mathcal{M}}$. Let $\widetilde{p}$ be a lifted period of $\widetilde{\gamma}$; consider a path β^0 joining $\widetilde{p}(0)$ to v and call β^1 the lift of $\pi(\beta^0)$ starting at $\widetilde{p}(1)$. The target v' of β^1 satisfies $\pi(v) = \pi(v')$; intuitively, $\widetilde{\gamma}$ translates v to v'. It is readily seen that v' does not depend on the choice of β^0 and $\widetilde{p}$. We therefore define $\tau_{\widetilde{\gamma}}(v) := v'$. In particular, $\tau_{\widetilde{\gamma}}$ sends a geometric lift of a curve $c \in C$ to another geometric lift of c.

Define a permutation $\phi_{\widetilde{\gamma}}$ over the set of words by the rule, if w is any word and $c^\alpha \in \widetilde{C}$: $\phi_{\widetilde{\gamma}}(c^\alpha w) = w\ \tau_{\widetilde{\gamma}}(c^\alpha)$, $\phi_{\widetilde{\gamma}}(\bar{c}^\alpha w) = w\ \overline{\tau_{\widetilde{\gamma}}(c^\alpha)}$, and $\phi_{\widetilde{\gamma}}(\varepsilon) = \varepsilon$.

Proposition 3.19 *For any word w in $[\widetilde{C}/\widetilde{\gamma}]$, we have: $[\widetilde{C}/\widetilde{\gamma}] = \{\phi_{\widetilde{\gamma}}^n(w), n \in \mathbf{Z}\}$.*

Proof. First, let $a^1 < a^2$ be two real numbers such that exactly one crossing occurs between the curves in $\widetilde{C}$ and $\widetilde{\gamma}|_{[a^1,a^2]}$. For $k = 1, 2$, let $w^k = \widetilde{C}/\widetilde{\gamma}|_{[a^k,a^k+1]}$.

[9] In fact, we consider only the lifted periods whose endpoints are disjoint from lifts of $\widetilde{C}$; this to avoid a tedious definition of the crossings when an intersection occurs at an endpoint of $\widetilde{p}$.

We have $w^2 = \phi_{\widetilde{\gamma}}(w^1)$; indeed,

$$\begin{aligned}\tau_{\widetilde{\gamma}}(c^\alpha \cap \widetilde{\gamma}|_{[a^1,a^2]}) &= \tau_{\widetilde{\gamma}}(c^\alpha) \cap \tau_{\widetilde{\gamma}}(\widetilde{\gamma}|_{[a^1,a^2]}) \\ &= \tau_{\widetilde{\gamma}}(c^\alpha) \cap \widetilde{\gamma}|_{[a^1+1,a^2+1]}.\end{aligned}$$

From this fact, it is easy to conclude. □

If w is a word, we define $[w]_{\widetilde{\gamma}}$ to be the set $\{\phi^n_{\widetilde{\gamma}}(w), n \in \mathbf{Z}\}$. The sets of words having this form are called the *$\widetilde{\gamma}$-words sets*; in particular, the crossing words set $[\widetilde{C}/\widetilde{\gamma}]$ is a $\widetilde{\gamma}$-words set. Note that $\phi_{\widetilde{\gamma}}$ does not affect the length of a word, so that the *length* of a $\widetilde{\gamma}$-words set is well-defined. Let W be a $\widetilde{\gamma}$-words set. If there exists $w \in W$ containing a factor $c^\alpha \bar{c}^\alpha$ or $\bar{c}^\alpha c^\alpha$, where $c^\alpha \in \widetilde{C}$, we denote by w' the word resulting from removing this factor from w; we say that W (which equals $[w]_{\widetilde{\gamma}}$) *elementarily c-reduces* to $[w']_{\widetilde{\gamma}}$. We define the *elementary reductions*, *(c)-reductions*, and *(c)-irreducible words set* in the obvious way.

Lemma and Definition 3.20 *Let c be a curve in C. Any $\widetilde{\gamma}$-words set W c-reduces (resp. reduces) to exactly one c-irreducible (resp. irreducible) $\widetilde{\gamma}$-words set. We define* $\mathrm{red}_c(W)$ *(resp.* $\mathrm{red}(W)$*) to be this $\widetilde{\gamma}$-words set.*

Proof. We prove the result for reductions, the same argument holds for c-reductions. Let w be a word; a *simplification* on w consists in either an elementary reduction on w (removal of $c^\alpha\bar{c}^\alpha$ or $\bar{c}^\alpha c^\alpha$), or in the removal of the first and last letter of w, if the first is of the form c^α (resp. $\bar{c}^\alpha$) and the last of the form $\overline{\tau_{\widetilde{\gamma}}(c^\alpha)}$ (resp. $\tau_{\widetilde{\gamma}}(c^\alpha)$). It is easily proved that W elementarily reduces to W' if and only if, for *any* $w \in W$, there exists $w' \in W'$ such that w simplifies to w'.

We say that w and w' are *equivalent* if $[w]_{\widetilde{\gamma}} = [w']_{\widetilde{\gamma}}$. Let w be a word; suppose that w_1 and w_2 are derived from w by a simplification. It can be shown by an easy case analysis that there exist equivalent words w'_1 and w'_2 such that, for $i = 1, 2$, w'_i is derived from w_i by zero or one simplification.

Let W be any $\widetilde{\gamma}$-words set. Assume that W elementarily reduces to W_1 and W_2. Let $w \in W$; by the first paragraph, there exist w_1 and w_2 in W_1 and W_2 such that w simplifies to w_1 and w_2. It follows, by the second paragraph, that there exists W_3 deduced from W_1 and W_2 by zero or one elementary reduction.

We can now prove the result by induction on the length of $\widetilde{\gamma}$-words sets; the lemma is trivial if the length is 0 or 1. Assume that the lemma is true for all $\widetilde{\gamma}$-words sets of length at most n; let W be a $\widetilde{\gamma}$-words set of length $n+1$. If W is reducible, consider any two $\widetilde{\gamma}$-words sets, W_1 and W_2, derived from W by an elementary reduction. By the preceding paragraph, W_1 and W_2 reduce (with zero or one elementary reduction) to some W_3. From our induction hypothesis, each of W_1, W_2, and W_3 yields only one irreducible $\widetilde{\gamma}$-words set, which must hence be the same for all these words sets. This concludes the proof. □

Suppose that an elementary c-reduction is possible on $[\widetilde{C}/\widetilde{\gamma}]$. Fix a lifted period $\widetilde{p}$ of $\widetilde{\gamma}$, such that this elementary reduction corresponds to a factor $c^\alpha\bar{c}^\alpha$ or $\bar{c}^\alpha c^\alpha$ on $\widetilde{C}/\widetilde{p}$. Let a and a' be the two intersection points corresponding to the two letters of this factor, and c^α_1 and $\widetilde{p}_1$ be the subpaths of c^α and $\widetilde{p}$ which are

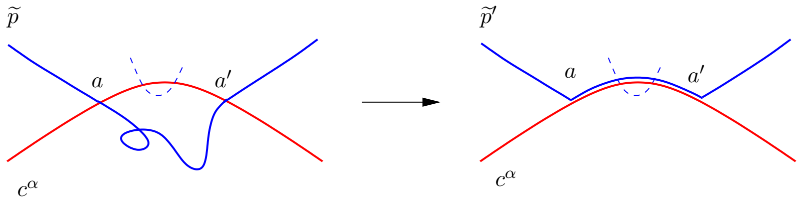


Figure 3.10: The fundamental operation of uncrossing two curves c^α and $\widetilde{p}$, crossing at a and a', corresponding to an elementary reduction on $[\widetilde{C}/\widetilde{\gamma}]$. Paths $\widetilde{p}_1$ and c^α_1 are the subpaths of $\widetilde{p}$ and c^α between a and a'. Path $\widetilde{p}_1$ is not necessarily simple, and c^α_1 can cross other pieces of $\widetilde{p}$.

between a and a'. The projections of these two subpaths on $\mathcal{M}$ are homotopic. We will often use the following elementary operation of *uncrossing* (Figure 3.10): we replace $\widetilde{p}_1$ by a path having the same endpoints which goes along c^α_1. The effect of this operation is the removal of the two crossings a and a'. Let $\widetilde{p}'$ be the path $\widetilde{p}$ after this operation, and let $\widetilde{\gamma}'$ be the corresponding geometric lift of the cycle γ' for which $\widetilde{p}'$ is a lifted period; $[\widetilde{C}/\widetilde{\gamma}]$ elementarily reduces to $[\widetilde{C}/\widetilde{\gamma}']$, and the cycles γ and γ' are homotopic.

Proposition 3.21 *Let C be a family of simple, pairwise disjoint curves on $\mathcal{M}$; let $\widetilde{C}$ be the set of geometric lifts of the curves in C. We assume that each element of $\widetilde{C}$ separates $\widetilde{\mathcal{M}}$ into two connected components. Let γ be a cycle in $\mathcal{M}$, homotopic to some cycle γ' disjoint from C, and let $\widetilde{\gamma}$ be a lift of γ. Then* $\mathrm{red}([\widetilde{C}/\widetilde{\gamma}]) = [\varepsilon]_{\widetilde{\gamma}}$.

Proof. Let p and p' be the restrictions of γ and γ' to $[0,1]$. There exists a path β joining $p(0)$ to $p'(0)$ such that the path $q := \beta^{-1}.p.\beta.p'^{-1}$ is contractible in $\mathcal{M}$. Let $\widetilde{q}$ be a lift of q, concatenation of the inverse of β^0, $\widetilde{p}$, β^1, and the inverse of $\widetilde{p}'$ (respectively lifts of β, p, β, and p'). We choose $\widetilde{q}$ so that $\widetilde{p}$ is a lifted period of $\widetilde{\gamma}$.

Since p' is disjoint from $\widetilde{C}$, $w := \widetilde{C}/\widetilde{q}$ is the concatenation of $\widetilde{C}/(\beta^0)^{-1}$, $\widetilde{C}/\widetilde{p}$, and $\widetilde{C}/\beta^1$. Furthermore, $\tau_{\widetilde{\gamma}}(\beta^0)$ is equal to β^1; hence, if the kth letter of $\widetilde{C}/\beta^0$ is equal to c^α (resp. $\bar{c}^\alpha$), then the kth letter of $\widetilde{C}/\beta^1$ is equal to $\tau_{\widetilde{\gamma}}(c^\alpha)$ (resp. $\overline{\tau_{\widetilde{\gamma}}(c^\alpha)}$). It follows that $[w]_{\widetilde{\gamma}}$ reduces to $[\widetilde{C}/\widetilde{p}]_{\widetilde{\gamma}} = [\widetilde{C}/\widetilde{\gamma}]$. Now, by Lemma 3.10, w is parenthesized, so $[w]_{\widetilde{\gamma}}$ also reduces to $[\varepsilon]_{\widetilde{\gamma}}$. Lemma 3.20 concludes. □

3.3.3.2 Curves on cylinders and pairs of pants

In this section, we use crossing words to prove some basic facts regarding curves on cylinders and pairs of pants. All surfaces considered here have corresponding embedded graphs G used to compute the length of the curves.

For cycles. Let us study the existence, in pairs of pants or cylinders, of shortest homotopic cycles which are simple.

Proposition 3.22 *Let K be a cylinder or a pair of pants, and γ be a cycle homotopic to a boundary of K. There exists a* simple *cycle homotopic to and not longer than γ.*

We will only give a proof when K is a pair of pants; the proof of the case where K is a cylinder is simpler. The proof relies on the two following lemmas. Let p be a shortest path between the two boundaries of K which are not homotopic to γ; p can be chosen to be simple. Let K' be the cylinder obtained when cutting K along p, and let p' be a (simple) shortest path in K' joining one point of each boundary of K'.

Lemma 3.23 *There exists a cycle γ', homotopic to and not longer than γ, which does not cross p.*

Proof. Let $\widetilde{p}$ be the set of geometric lifts of p, and $\widetilde{\gamma}$ be a lift of γ. $[\widetilde{p}/\widetilde{\gamma}]$ reduces to $[\varepsilon]_{\widetilde{\gamma}}$ by Proposition 3.21; if it is not empty, let $\widetilde{\gamma}_1$ and $\widetilde{p}_1$ be the subpaths of $\widetilde{\gamma}$ and of the element of $\widetilde{p}$ corresponding to an elementary reduction. Since p is a shortest path, $|\widetilde{p}_1| \leq |\widetilde{\gamma}_1|$, and we can, like in Figure 3.10, proceed to the elementary reduction by changing γ to another cycle, which is homotopic to and not longer than γ, and has two crossings less than γ with p. The proof is finished by induction on the number of crossings between γ and p. □

Lemma 3.24 *There exists a cycle γ'' in the cylinder K', not longer than γ', homotopic to γ' in K', which crosses p' exactly once.*

Proof. Let $\widetilde{\gamma}'$ be a lift of γ', and $\widetilde{p}'$ be the set of the geometric lifts of p' in the universal covering space of K'. γ' is homotopic, in K', to its boundaries. The crossing words set $[\widetilde{p}'/\widetilde{\gamma}']$ reduces to some irreducible words set $[w]_{\widetilde{\gamma}'}$, where w contains exactly one letter. Indeed, the number of letters with and without bar in w differ by exactly one; if w contains more than one letter, then two letters with and without a bar must be consecutive in w, which implies a possible elementary reduction. Hence, if γ' crosses p' at least twice, an elementary reduction is possible. Let $\widetilde{\gamma}'_1$ and $\widetilde{p}'_1$ be the subpaths of $\widetilde{\gamma}$ and of the element of $\widetilde{p}'$ in the universal covering space of K' corresponding to this reduction. Like in the proof of Lemma 3.23, we can modify γ' to get a not longer, homotopic cycle which crosses p' twice less, the homotopy being taken in K'. We finish by induction on the length of $[\widetilde{p}'/\widetilde{\gamma}']$. □

Proof of Proposition 3.22. By Lemmas 3.23 and 3.24, we may assume that γ does not cross p, and crosses p' exactly once, say at some point a. Cut the cylinder K' along p'; we obtain two copies a' and a'' of a, and γ is transformed into a path between a' and a''. Hence, a shortest path between a' and a'' leads to a cycle in K' which can be chosen to be simple, not longer than γ, and homotopic to γ in K. □

In fact, the techniques used in the proof of this proposition yield an algorithm to compute a shortest cycle homotopic to a given boundary of a cylinder or pair of pants; this will be discussed in more details in Section 3.5.

For paths. Here are some results on shortest homotopic paths on cylinders or pairs of pants.

Proposition 3.25 *Let K be a cylinder or a pair of pants, and γ be one boundary of K. Assume γ is a shortest cycle among the simple cycles homotopic to γ. Let q be a path in K whose endpoints are on γ and which is homotopic to a path whose image set on K is inside the image set of γ. Then q^*, the shortest path on γ homotopic to q, is not longer than q.*

Again, we only give the proof for a pair of pants. Let p be a shortest path between the two boundaries of K which are not homotopic to γ; p can be chosen to be simple. Let K' be the cylinder obtained when cutting K along p.

Lemma 3.26 *There exists a path q', homotopic to and not longer than q, which does not cross p.*

Proof. Analogous to Lemma 3.23. □

Proof of Proposition 3.25. By Lemma 3.26, and since $q^* = q'^*$, we may assume that q belongs to K'. The proof is by induction on the number $c(q)$ of self-crossings of q. If $c(q) = 0$, then q is homotopic, in K', to a simple subpath of γ, and the result follows from the minimality of γ. Assume that $c(q) > 0$, and that the result is true for all smaller values of $c(q)$. Let c_1 be a simple closed subpath of q, and let q_1 be equal to the path q where c_1 is removed; $c(q_1) = c(q) - 1$. The loop c_1 is either contractible or freely homotopic to the boundaries of K', by Theorem 1.11, page 28.

If c_1 is contractible, then $q^* = q_1^*$. By the induction hypothesis, $|q_1^*| \leq |q_1|$; we also have $|q_1| \leq |q|$, which concludes this case. If c_1 is homotopic to γ or its reverse, q is homotopic to $q_1.\ell$ or $q_1.\ell^{-1}$, where ℓ is a loop associated with γ, and thus $|q^*| \leq |q_1^*| + |\gamma|$. Using the induction hypothesis, this cannot be greater than $|q_1| + |\gamma|$, which, in turn, cannot be greater than $|q_1| + |c_1| = |q|$. □

For cut systems by cycles. Let us start with a simple proposition.

Proposition 3.27 *Let s be a cut system by cycles of $\mathcal{M}$. Let s_{k_1} and s_{k_2} be cycles of s which are $\pm$* homotopic *(*i.e., *s_{k_1} and s_{k_2} are homotopic, or s_{k_1} and the reverse of s_{k_2} are homotopic). Then there is a sequence $s_{k_1} = s_{i_1}, s_{i_2}, \ldots, s_{i_p} = s_{k_2}$ of $\pm$ homotopic cycles of s such that two consecutive cycles in this sequence bound a cylinder whose interior is disjoint from the cycles of s.*

Proof. By Lemma 1.9, s_{k_1} and s_{k_2} bound a cylinder K. Any cycle of s inside K must be $\pm$ homotopic to s_{k_1} and s_{k_2}, for otherwise such a cycle would be contractible or non-simple. It follows that each component of the surface K cut along all cycles of s is a cylinder. Hence the cycles of s in K can be ordered $s_{k_1} = s_{i_1}, s_{i_2}, \ldots, s_{i_p} = s_{k_2}$ so that two consecutive cycles in this sequence bound a cylinder containing no other cycle of s. □

Proposition 3.28 *Let s be a cut system by cycles of $\mathcal{M}$. Assume that γ is inside one component K of the surface obtained by cutting $\mathcal{M}$ along s (K is a cylinder or a pair of pants), and homotopic in $\mathcal{M}$ to a cycle s_k. Then γ is homotopic,* in *K, to one boundary of K.*

The difficulty of this proposition lies in the fact that γ can be non-simple; otherwise, the same technique as the previous proposition could be applied. We start with a simple lemma.

Lemma 3.29 *Let K be a compact surface included in $\mathcal{M}$ whose boundaries are non-contractible in $\mathcal{M}$. Then any cycle inside K which is contractible (in $\mathcal{M}$) is also contractible in K.*

Proof. Let $(\widetilde{\mathcal{M}}, \pi)$ be the universal covering space of $\mathcal{M}$; it is sufficient to prove that $\pi^{-1}(K)$ is simply connected. If it were not the case, there would exist a simple non-contractible cycle γ in $\pi^{-1}(K)$. Such a cycle bounds a disk D in $\widetilde{\mathcal{M}}$, by Theorem 1.8, page 28, and D is not entirely contained in $\pi^{-1}(K)$. Therefore a lift $\widetilde{\gamma}'$ of a boundary γ' of K is inside D. This is impossible since $\widetilde{\gamma}'$ contains infinitely many lifts of some point of $\mathcal{M}$ (by Lemma 1.10, page 28). $\square$

Proof of Proposition 3.28. We again assume that K is a pair of pants. Let $\widetilde{s}$ denote the geometric lifts of the cycles in s. Let γ' be a cycle "running along" s_k: γ' is simple, disjoint from all cycles of s, and homotopic, in the surface obtained after cutting $\mathcal{M}$ along $s \setminus s_k$, to s_k. Let p and p' be the restrictions of γ and γ' to $[0, 1]$. There exists a path β joining $p(0)$ to $p'(0)$ such that the path $q := \beta^{-1}.p.\beta.p'^{-1}$ is contractible in $\mathcal{M}$. Let $\widetilde{q}$ be a lift of q, concatenation of the inverse of a lift β^0 of β, a lift $\widetilde{p}$ of p, a lift β^1 of β, and the inverse of a lift $\widetilde{p}'$ of p'.

Without loss of generality, assume that $\widetilde{s}/\beta^0$ is irreducible. If this crossing word is empty, then q is contractible in K by Lemma 3.29, hence γ and γ' are homotopic in K; so are γ and s_k, and the proof is complete. Assume this crossing word is non-empty.

Since p and p' do not cross s, $\widetilde{s}/\widetilde{q}$ is the concatenation of $\widetilde{s}/(\beta^0)^{-1}$ and $\widetilde{s}/\beta^1$. Because $\widetilde{s}/\beta^0$ is irreducible, so are $\widetilde{s}/(\beta^0)^{-1}$ and $\widetilde{s}/\beta^1$; since $\widetilde{s}/\widetilde{q}$ can be elementarily reduced (because q is a closed path), there is exactly one possible elementary reduction on this word, which concerns the last letter of $\widetilde{s}/(\beta^0)^{-1}$ and the first letter of $\widetilde{s}/\beta^1$. Hence, the first elements of $\widetilde{s}$ crossed by β^0 and β^1 must be the same, say s_j^α. Let β' be the beginning of β before its first crossing with s; we get that $\beta'^{-1}.p.\beta'$ is homotopic to the nth power of a loop ℓ_j associated to s_j in $\mathcal{M}$, for some n. This homotopy can in fact be taken in K by Lemma 3.29. We now have to prove that $n \in \{-1, 1\}$.

There exists a loop ℓ_k associated to s_k, and a path δ, such that $\delta.\ell_k.\delta^{-1}$ is homotopic (in $\mathcal{M}$) to the nth power of ℓ_j. Hence ℓ_k is homotopic to the nth power of $\delta^{-1}.\ell_j.\delta$. Since ℓ_k is simple, it follows by Theorem 1.11 that $|n| \leq 1$. Hence γ is homotopic, in K, to s_j or its reverse. $\square$

3.3.3.3 Proof of Theorem 3.7

Let us consider a cut system by cycles $s = (s_1, \ldots, s_N)$ of $\mathcal{M}$; let $\widetilde{s}$ be the set of all geometric lifts of s in $\widetilde{\mathcal{M}}$. For $i \in [1, N]$, let us choose an enumeration $(s_i^\alpha)_{\alpha \in I_i}$ of the geometric lifts of s_i.

Fix $j \in [1, N]$; let $r \in \mathrm{shrt}_j(s)$. We consider the set $\widetilde{r}$ of all the geometric lifts of the cycles in r, and we give an enumeration of the geometric lifts of r_k, for each

k, as follows. We choose the enumeration of the geometric lifts of r_k, for $k \neq j$, as the geometric lifts of s_k (*i.e.*, $r_k^\alpha = s_k^\alpha$). It remains to define the enumeration of the geometric lifts of r_j. Let P_j be the surface defined by $s \setminus s_j$ which contains s_j. We first choose a cycle γ_j in P_j, so that γ_j is disjoint from s_j and r_j. (This is always possible since s_j is homotopic to a boundary of P_j.) Thus s_j and γ_j bound a cylinder in P_j by Lemma 1.9, page 28; the lifts of this cylinder in $\widetilde{\mathcal{M}}$ are disjoint infinite strips which contain no lift of s or γ_j in their interior. We can thus choose an enumeration of the geometric lifts of γ_j such that, for each α, s_j^α and γ_j^α bound a strip which contains no lift of s or γ_j in its interior. Doing the same with γ_j and r_j (which together bound a cylinder in P_j), we get an enumeration of the geometric lifts of r_j.

Finally, fix $i \in [1, N]$; let t_i be a shortest cycle among all cycles homotopic to s_i (t_i is not necessarily simple), and $\widetilde{t}_i$ be a lift of t_i. Henceforth, the crossing words regarding the geometric lifts of s and r will be written by using the enumeration described above, by omitting the "r" and the "s". For example, we shall write ${}^3_1 \, \overline{{}^7_5} \, \overline{{}^4_2}$ instead of $s_1^3 \, \overline{s_5^7} \, \overline{s_2^4}$. This allows to write, for example, that $[\widetilde{r}/\widetilde{t}_i] = [\widetilde{s}/\widetilde{t}_i]$ if t_i does not cross r_j nor s_j.

Proposition 3.30 $\mathrm{red}_j([\widetilde{r}/\widetilde{t}_i]) = \mathrm{red}_j([\widetilde{s}/\widetilde{t}_i])$.

We denote by $\widetilde{s}_j$ the set of geometric lifts of s_j, with the enumeration induced by $\widetilde{s}$. The same holds for $\widetilde{r}_j$.

Lemma 3.31 *Let p be a path in P_j whose endpoints are on the boundary of P_j, and $\widetilde{p}$ be a lift of p. Then $\widetilde{s}_j/\widetilde{p}$ and $\widetilde{r}_j/\widetilde{p}$ reduce to the same irreducible word.*

Proof. We first assume s_j and r_j are disjoint; they bound a cylinder K inside P_j. The lifts of K in the universal covering space of $\mathcal{M}$ are pairwise disjoint infinite strips bounded by r_j^α and s_j^α, by the choice of the enumeration of the geometric lifts of r_j. Furthermore, $\widetilde{p}$ has its endpoints outside these strips. Let us split $\widetilde{p}$ into subpaths $\widetilde{p}_i$, $i = 1, \ldots, n$, each entering in exactly one strip, and exactly once in this strip, and so that their endpoints are outside the strips. Clearly, $\widetilde{s}_j/\widetilde{p}_i$ and $\widetilde{r}_j/\widetilde{p}_i$ reduce to the same irreducible word (there are two cases according to whether $\widetilde{p}_i$ enters and exits the strip through the same boundary or not); the result follows.

For the general case, let γ_j be a simple cycle homotopic in P_j to r_j and s_j, and which does not cross r_j nor s_j. We again enumerate the geometric lifts of γ_j so that γ_j^α and r_j^α (or s_j^α) bound an infinite strip. Applying the reasoning of the above paragraph to s_j and γ_j, and then to γ_j and r_j, we get the result. $\square$

Proof of Proposition 3.30. Assume first that t_i is contained in P_j. By Proposition 3.21, we have $\mathrm{red}([\widetilde{r}/\widetilde{t}_i]) = [\varepsilon]_{\widetilde{t}_i} = \mathrm{red}([\widetilde{s}/\widetilde{t}_i])$. But this also equals $\mathrm{red}_j([\widetilde{r}/\widetilde{t}_i])$ and $\mathrm{red}_j([\widetilde{s}/\widetilde{t}_i])$, and this concludes the proof. If t_i is not entirely contained in P_j, then let t'_i be a maximal subpath of t_i which is inside P_j, and $\widetilde{t}'_i$ be a lift of t'_i; it is sufficient to prove that $\widetilde{r}_j/\widetilde{t}'_i$ and $\widetilde{s}_j/\widetilde{t}'_i$ reduce to the same irreducible word; but this follows from Lemma 3.31. $\square$

Proposition 3.32 *There exists a cycle t'_i, homotopic to and not longer than t_i, and a lift $\widetilde{t}'_i$ of t'_i, such that $[\widetilde{r}/\widetilde{t}'_i]$ is a $\widetilde{t}_i$-words set equal to* $\mathrm{red}_j([\widetilde{s}/\widetilde{t}_i])$.

PROOF. By Proposition 3.30, $[\widetilde{r}/\widetilde{t}_i]$ j-reduces to $\mathrm{red}_j([\widetilde{s}/\widetilde{t}_i])$. If $[\widetilde{r}/\widetilde{t}_i]$ is j-irreducible, there is nothing to show. Otherwise, an elementary j-reduction is possible on $[\widetilde{r}/\widetilde{t}_i]$. We can apply Proposition 3.25 to the subpath of t_i corresponding to this j-reduction, and apply the uncrossing operation to t_i. We obtain a lift $\widetilde{t}'_i$ of a cycle t'_i which is homotopic to and not longer than t_i. Clearly, $\tau_{\widetilde{t}_i} = \tau_{\widetilde{t}'_i}$, which implies that the sets of $\widetilde{t}_i$- and $\widetilde{t}'_i$-words sets are equal. Furthermore, $[\widetilde{r}/\widetilde{t}'_i]$ results from $[\widetilde{r}/\widetilde{t}_i]$ by this elementary j-reduction. By induction, we obtain the desired t'_i. □

Proposition 3.33 *Assume that t_i is disjoint from s, and that t_i and s_k are $\pm$ homotopic in the cylinder or pair of pants defined by s containing t_i. Then, there exists a cycle t'_i, homotopic to and not longer than t_i, which is disjoint from r, and which is $\pm$ homotopic to r_k in the cylinder or pair of pants defined by r containing t'_i.*

PROOF. Let K be the cylinder or pair of pants defined by s containing t_i. The proof is trivial if s_j is not a boundary of K. If s_j is $\pm$ homotopic to t_i in K, then either $s_j = s_k$, or K is the cylinder bounded by s_j and s_k. In both cases it is easy to conclude: indeed, t_i can be chosen to be simple by Proposition 3.22; it follows that $|r_j| = |t_i|$, and we can take for t'_i a slightly translated copy of r_j or r_j^{-1}.

There remains the case where s_j is not $\pm$ homotopic to t_i in K (and thus K is a pair of pants). P_j contains t_i; one boundary of P_j is a cycle γ homotopic to r_j in P_j, and another one is r_k. The cycles r_j and γ bound a cylinder K' in P_j, and r_j is optimal in K'. Then, using Proposition 3.25, any component of t_i in K' can be swapped into the complementary part K of P_j, thus removing the crossings with r_j. □

We now conclude the proof of Theorem 3.7.

PROOF OF THEOREM 3.7. If s and $s' \in \mathrm{shrt}_i(s)$ are two cut systems by cycles considered in the course of the process, we will write by abuse of notation $s' = \mathrm{shrt}_i(s)$. We let $\widetilde{s}^0$ be the geometric lifts of s^0, enumerated in an arbitrary way. By induction on $n \in \mathbf{N}$, we construct an enumeration $\widetilde{s}^n$ of the geometric lifts of s^n, this enumeration being chosen as in the beginning of Section 3.3.3.3.

Let $\widetilde{t}_i^0$ be a lift of a shortest cycle t_i^0 homotopic to s_i^0. By Proposition 3.21, $[\widetilde{s}^0/\widetilde{t}_i^0]$ reduces to $[\varepsilon]_{\widetilde{t}_i^0}$. By Proposition 3.32, we can construct a sequence $(\widetilde{t}_i^n)_{n\in\mathbf{N}}$ of lifts of shortest homotopic cycles such that the length of $[\widetilde{s}^n/\widetilde{t}_i^n]$ strictly decreases until it becomes empty at some stage n. By Proposition 3.28, t_i^n and a cycle s_k^n are homotopic in the cylinder or pair of pants defined by s^n containing t_i^n. By $k-1$ applications of Proposition 3.33, and then using Proposition 3.22, $|s_k^{n+1}| = |t_i^n|$. The cycles s_k^{n+1} and s_i^{n+1} are $\pm$ homotopic, hence, by Proposition 3.27, there is a finite sequence of p cylinders separating s_k^{n+1} and s_i^{n+1}. It implies that, after p new applications of shrt, all the cycles in these cylinders must have the length of s_k^{n+1}; in particular, $|s_i^{n+1+p}| = |t_i^{n+1+p}|$. From this discussion,

it follows that the length of $(s_i^n)_{n\in\mathbf{N}}$ becomes stationary. It remains to prove that all lengths remain unchanged once s^n and s^{n+1} have the same lengths. Assume s and $s' \in \mathrm{shrt}(s)$ have the same length, and let $i \in [1, N]$; we shall prove that s_i has the same length as t_i (a shortest cycle homotopic to s_i).

$[\widetilde{s}/\widetilde{t}_i]$ reduces to the empty words set; assume that an elementary j-reduction is possible. Let $\hat{t}_i$ and $\hat{s}_j$ be the associated subpaths of $\widetilde{t}_i$ and of the lift of s_j. We will prove that both subpaths have the same length. It will follow that we can modify t_i without changing its length nor its homotopy class to proceed to the j-reduction in $[\widetilde{s}/\widetilde{t}_i]$; hence by induction we will be able to assume that $[\widetilde{s}/\widetilde{t}_i] = \varepsilon$.

If $j \neq 1$, only geometric lifts of $(\mathrm{shrt}_1(s))_1$ appear in the word $\mathrm{shrt}_1(\widetilde{s})/\hat{t}_i$; by Lemma 3.10, this word is parenthesized. By Proposition 3.25, we can iteratively remove all the crossings between $\hat{t}_i$ and $\mathrm{shrt}_1(\widetilde{s})$. $\hat{t}_i$ is replaced this way by a path with the same endpoints $\hat{t}'_i$ that does not cross $\mathrm{shrt}_1(\widetilde{s})$. Iterating the process, we get the existence of a path $\hat{t}''_i$ in $\widetilde{\mathcal{M}}$, not longer than $\hat{t}_i$, with the same endpoints as $\hat{t}_i$, and which crosses no lift of $s'' := \mathrm{shrt}_{j-1} \circ \ldots \circ \mathrm{shrt}_1(s)$. Furthermore, $s''_j = s_j$ is optimal in the cylinders and in the pairs of pants it bounds, because s'' has the same length as $\mathrm{shrt}_j(s'')$. It follows, by Proposition 3.25, that $\hat{s}_j$ cannot be longer than $\hat{t}''_i$. Hence $|\hat{s}_j| = |\hat{t}''_i| = |\hat{t}_i|$.

We can thus assume (up to a change of t_i) that $[\widetilde{s}/\widetilde{t}_i] = \varepsilon$. By Proposition 3.28, t_i and a cycle s_k are homotopic in the cylinder or pair of pants defined by s containing t_i. By $k-1$ applications of Proposition 3.33, and then by Proposition 3.22, we may assume that t_i and s_k bound a cylinder whose interior is disjoint from the cycles of $\mathrm{shrt}_{k-1} \circ \ldots \circ \mathrm{shrt}_1(s)$. This implies $|s_k| = |t_i|$, which finishes the proof if $k = i$. If $k \neq i$, there is, by Proposition 3.27, a sequence of $\pm$ homotopic cycles $s_{i_2}, \ldots, s_{i_{p-1}}$ between s_k and s_i; all these cycles must have the same length, for otherwise the length of some cycle would decrease after the application of shrt to s. This concludes the proof. □

3.4 Extension of an embedding to a cut system

In this section, we consider the problem of optimizing embeddings of graph or of cycles, without assuming that these embeddings are cut systems. The idea is to extend such a family of curves to a cut system and to optimize the cut system by one of the theorems of Section 3.2. By these theorems, the resulting cut system contains curves which are, individually, as short as possible in some class (of homotopy or isotopy). In other words, the extension to a cut system creates no obstruction for the shortenings; on the contrary, it even enables to give a simple algorithm for this purpose!

In the following, we assume that the input embedding of graph or of cycles contains no contractible path or cycle. If the input embedding has contractible curves, these curves, being also simple, bound a topological disk by Theorem 1.8, page 28. A preliminary operation thus consists in testing this property and in removing the contractible curves, if any.

3.4.1 Cut systems by graph

We emphasize that the embeddings of graphs considered are without isolated vertices and may have loops and multiple edges. Such an embedding is thus a family of simple, disjoint paths, except possibly at common endpoints.

Theorem 3.34 *Let g be the genus of $\mathcal{M}$, and b be the number of its boundaries. Let $s = (s_1, \ldots, s_n)$ be a regular embedding of graph on $\mathcal{M}$, containing no contractible path. Then it is possible to extend s to a cut system by graph $(s_1, \ldots, s_N)$ (with $N \geq n$) such that each vertex of this cut system is a vertex of s or a point on the boundary of $\mathcal{M}$ (or, if $\mathcal{M}$ is boundaryless and s is empty, with one single vertex), with $N = O(n + g + b)$.*

Let us start with a lemma.

Lemma 3.35 *Let $\mathcal{M}$ be a compact, orientable surface; let $k \geq 1$ be the number of its connected components, g be its genus, and b be the number of its boundaries; let us assume that $b \geq 1$. Let c be a simple path on $\mathcal{M}$ which intersects the boundary of $\mathcal{M}$ exactly at its endpoints; let $\mathcal{M}'$ be the surface obtained by cutting $\mathcal{M}$ along c; let k' be the number of its connected components, g' be its genus and b' be its number of boundaries. Then one of the following assertions holds:*

- *$k' = k + 1$, $g' = g$, and $b' = b + 1$ (case of a separating path);*
- *$k' = k$, $g' = g - 1$, and $b' = b + 1$ (case of a non-separating path whose endpoints are on the same boundary of $\mathcal{M}$);*
- *$k' = k$, $g' = g$, and $b' = b - 1$ (case of a non-separating path whose endpoints are on two different boundaries of $\mathcal{M}$).*

Proof. Let $\chi(\mathcal{M})$ and $\chi(\mathcal{M}')$ be the Euler characteristics of $\mathcal{M}$ and $\mathcal{M}'$. We have $\chi(\mathcal{M}') = \chi(\mathcal{M}) + 1$; indeed, χ does not depend on the chosen triangulation of the surface (it is a topological invariant), and we can always triangulate $\mathcal{M}$ in such a way that c is an edge of $\mathcal{M}$; the cutting operation thus boils down to duplicate c and the two incident vertices (Figure 3.11). The Euler characteristic being equal to the number of vertices and faces minus the number of edges, it increases by one during the cutting. But $\chi(\mathcal{M}) = 2k - 2g - b$ and $\chi(\mathcal{M}') = 2k' - 2g' - b'$. Hence we have:

$$2(k' - k) - 2(g' - g) - (b' - b) = 1. \tag{3.1}$$

Clearly, $k' - k$ is between 0 and 1, and $b' - b$ is between -1 and 1; Formula (3.1) thus proves that $b' - b$ is odd, hence equals either -1 or 1. On the other hand, if $k' = k + 1$, then we cannot have $b' = b - 1$, because each "side" of c will belong to a boundary in the surface after cutting. □

Proof of Theorem 3.34. Let $\mathcal{M}'$ be the surface obtained by cutting $\mathcal{M}$ along paths $s_1, \ldots, s_n$. Let us prove that $\mathcal{M}'$ has at most $n+1$ connected components, has genus at most $g+n$, and has at most $b+3n$ boundaries. For each vertex v of s, we build a disk D_v chosen as in Section 3.3.2.2. Let $\mathcal{M}_1$ be the surface $\mathcal{M}$ where

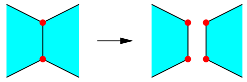


Figure 3.11: Cutting along a path intersecting the boundary of the surface exactly at its endpoints, increases the Euler characteristic of the surface by one.

the disks D_v have been removed. Cutting $\mathcal{M}$ along the paths of s yields a surface $\mathcal{M}'_1$ which is topologically equivalent to the surface $\mathcal{M}'$; in particular, these two surfaces have the same number of connected components, the same genus, and the same number of boundaries. $\mathcal{M}_1$ is connected, has genus g, and has a number of boundaries equal to b plus the number of vertices of s in the interior of $\mathcal{M}$ (which is at most $2n$). The previous lemma shows that $\mathcal{M}'_1$ has at most $n + 1$ connected components, genus at most $g + n$, and at most $b + 3n$ boundaries; the same holds for $\mathcal{M}'$.

Consider a connected component of $\mathcal{M}'$; let g' be its genus and b' be its number of boundaries. Each of these b' boundaries contains at least one point which is, after gluing, either on the boundary of $\mathcal{M}$ or on a vertex of s. It is thus possible to find a polygonal schema with $4g' + 3b'$ edges on this surface so that, after regluing along the paths of s, each vertex of this schema is a vertex of s or on the boundary of $\mathcal{M}$. (If $\mathcal{M}'$ is boundaryless, this means that $n = 0$, and it is possible to find a polygonal schema with one single vertex on $\mathcal{M}'$.) We thus have a regular embedding of graph which cuts $\mathcal{M}$ into a disjoint union of closed disks with $O(n+g+b)$ paths and whose vertices will be, after regluing along the paths of s, either vertices of s or on the boundary of $\mathcal{M}$. To transform this regular embedding of graph into a cut system by graph, it is then sufficient to "double" all paths of this embedding: for each path c of this embedding, we add a path which "goes along" c (the graph thus has multiple edges). The resulting family is a cut system by graph, with $N = O(n + g + b)$ paths. □

The following corollary can be immediately deduced, with Item 2 of Theorem 3.2:

Corollary 3.36 *Let c be a simple path whose intersection with $\partial\mathcal{M}$ is exactly its endpoints. Let C be the set of all paths with minimal length among the paths homotopic to c. Then, there exists an element of C which is simple.*

The following corollary can be deduced from the previous theorem and from Corollary 3.5:

Corollary 3.37 *Let s be a fundamental system of loops which has minimal length among the systems in its homotopy class. Then, for each i, s_i is a loop with minimal length among all simple loops homotopic to s_i.*

From a cut system by graph, it is always possible to obtain a *triangulated* cut system by graph, *i.e.*, a cut system by graph in which each face is incident to at most three paths of the system. N increases but remains asymptotically the same (a triangulation of a polygon with p edges is done with $p - 3$ paths). We will use this fact in the complexity analysis of our algorithm.

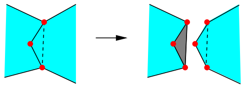


Figure 3.12: Cutting $\mathcal{M}$ along a simple cycle does not change the Euler characteristic of the surface.

3.4.2 Cut systems by cycles

Theorem 3.38 *Let g be the genus of $\mathcal{M}$, and b be the number of its boundaries. We assume that $\mathcal{M}$ is neither a sphere nor a disk. Let $s = (s_1, \dots, s_n)$ be a regular embedding of non-contractible cycles. It is possible to extend s to a cut system by cycles $(s_1, \dots, s_N)$.*

Again, let us begin with a lemma:

Lemma 3.39 *Let $\mathcal{M}$ be a compact, orientable surface; let $k \geq 1$ be the number of its connected components, g be its genus, and b be the number of its boundaries. Let γ be a simple cycle in the interior of $\mathcal{M}$; let $\mathcal{M}'$ be the surface obtained by cutting $\mathcal{M}$ along γ, having k' connected components, b' boundaries, and genus g'. Then one of the following assertions holds:*

- *$k' = k + 1$, $g' = g$, and $b' = b + 2$ (case of a separating cycle);*
- *$k' = k$, $g' = g - 1$, and $b' = b + 2$ (case of a non-separating cycle).*

Proof. Let $\chi(\mathcal{M})$ and $\chi(\mathcal{M}')$ denote the Euler characteristics of $\mathcal{M}$ and $\mathcal{M}'$. A similar argument to the one of the proof of Lemma 3.35 proves that $\chi(\mathcal{M}) = \chi(\mathcal{M}')$, because it is always possible to triangulate $\mathcal{M}$ in such a way that the cycle γ is a cycle of length three in the vertex-edge graph; the cutting boils thus down to duplicate three vertices and three edges (Figure 3.12). Hence:

$$2(k' - k) - 2(g' - g) - (b' - b) = 0. \tag{3.2}$$

Again, $k' - k$ is either 0 or 1; $b' - b$ necessarily equals 2. This is enough to deduce the two indicated possibilities. □

Proof of Theorem 3.38. Cutting $\mathcal{M}$ along the cycles of s yields a set of at most n surfaces, whose sum of generi is at most $g + n$ and whose sum of numbers of boundaries is at most $b + n$.

Let $\mathcal{M}'$ be such a surface; let g' be its genus and b' be the number of its boundaries. $\mathcal{M}'$ is neither a sphere nor a disk. If it is a cylinder or a torus, it admits a decomposition into cylinders with 0 or 1 cycle. Otherwise, it admits a pants decomposition with $3g' + b' - 3$ cycles.

Finally, we obtain a decomposition of $\mathcal{M}$ into cylinders or pairs of pants with $O(g + b + n)$ cycles. To obtain a cut system by cycles, it is then sufficient to "double" all cycles, by creating, for each cycle, a copy which goes along it. □

In particular, a single cycle or a pants decomposition can be extended to a cut system by cycles. We deduce, with Theorem 3.7, the following corollaries:

Corollary 3.40 *Let γ be a simple cycle, and let Γ be the set of the cycles with minimal length among the cycles which are homotopic to γ. There exists an element of Γ which is simple.*

(The previous corollary has been shown in [75] in the case where the curves are drawn on a Riemannian surface.)

Corollary 3.41 *Let s be a pants decomposition which has minimal length among the pants decomposition in its homotopy class. Then, for each i, s_i is a simple cycle with minimal length among the (not necessarily simple) cycles which are homotopic to s_i.*

3.5 Algorithms

This section aims at explaining how it is possible to implement the optimization and extension processes previously described. We will firstly describe how to store efficiently a regular embedding of graph or of cycles on $\mathcal{M}$. Then, we will explain how to translate into effective algorithms the processes described in Section 3.2. After that, we will give their complexity. Finally, we will explain how to extend an embedding of graph or of cycles to a cut system.

Convention 3.42 *Henceforth, we will assume that:*

i. $\mathcal{M}$ is a polyhedral surface, with vertex-edge graph H;

ii. G is the dual graph of H, as defined in Section 3.1.2 (which in particular implies that the faces of G are topological disks);

iii. each vertex of any embedding of graph considered is a vertex of H.

3.5.1 Combinatorial curves

We now describe in detail the combinatorial framework, sketched in Section 3.1.2. The idea is to consider a regular embedding of graph or cycles, and to "forget" its actual geometry, by keeping in memory only the way the curves cross the edges of G. In other words, two embeddings which cross G in the same way will be regarded as identical.

A *path* on the graph H is a sequence of oriented edges $a_1, \dots, a_p$ of H such that, for each k, $k = 1, \dots, p-1$, the target of edge a_k is the source of edge a_{k+1}. An occurrence of an oriented edge of H in this path will be called a *jump* of this path. A jump thus *corresponds* to a unique oriented edge of H, but an oriented edge of H corresponds to zero, one, or several jumps of a path.

A *cycle* on H is a cyclic sequence of oriented edges $a_1, \dots, a_p, a_{p+1} = a_1$ of H such that, for each k, $k = 1, \dots, p$, the target of edge a_k equals the source of edge a_{k+1}. (This distinction between cycle and closed path in a graph is to be compared with the distinction between cycle and closed path on a surface.) An occurrence of an oriented edge of H will still be called a *jump* of this cycle.

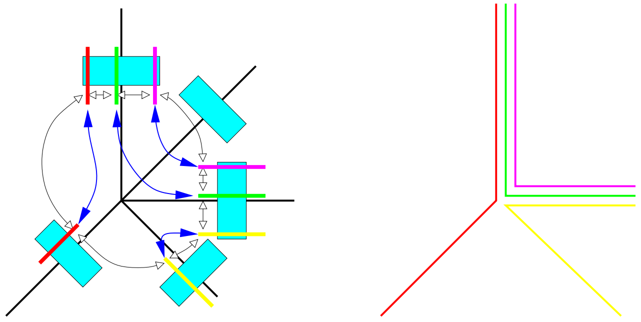


Figure 3.13: The data structure used to store a combinatorial family of cycles on H, in the neighborhood of a vertex v of H of degree 5. Each (non-oriented) edge of H incident to v contains the ordered list of the jumps which are on this edge. Here, no crossing occurs at v. On the right, a "natural" way to see the cycles spread apart on the surface.

3.5.1.1 Combinatorial cycles

Let us begin with the definition of a combinatorial family of cycles, which is the simplest case. We will consider here the *oriented* edges of H; if e is an edge of H, $-e$ denotes edge e with the opposite orientation.

A *combinatorial family of cycles* S is a family of cycles in H with the data, for each oriented edge e of H, of a total order $\preceq_e$ over all the jumps of the set of cycles corresponding to edge e or $-e$. These cycles must be consistent, in the following way: $a \preceq_e b$ if and only if $b \preceq_{-e} a$. These orders $\preceq_e$ represent a way to draw cycles on the surface: schematically, it is possible to represent a combinatorial family of cycles S as in Figure 3.13, each non-oriented edge of H containing an ordered list of jumps. In this representation, $a \preceq_e b$ if and only if jump a is on the left of jump b along the (oriented) edge e.

Let v be a vertex of H, and let $e_1, \ldots, e_n$ be the (cyclic) list of edges of H whose source is v, in clockwise order. Let us define a cyclic order $\preceq_v$ on the jumps of cycles of S incident to v, by enumerating its elements in the order: first, the jumps of cycles of S on e_1 or $-e_1$, in $\preceq_{e_1}$-order; then the jumps of cycles of S on e_2 or $-e_2$, in $\preceq_{e_2}$-order; and so on. In the previous representation, $\preceq_v$ describes the cyclic order of the jumps of S around v.

Let a_1, a_2 and b_1, b_2 be the jumps of two subpaths of size 2 of cycles in S. We will say that these subpaths *cross* if the edges corresponding to a_1 and b_1 have the same target v and if, in the cyclic order $\preceq_v$, a_1 and a_2 separate b_1 and b_2.

It is easy to create a data structure to store a combinatorial family of cycles, in which accessing the predecessor and successor of a jump in a given cycle, and accessing the predecessor and the successor of a jump with respect to some $\preceq_v$-order, takes constant time. Figure 3.13 represents this structure: each rectangular box is a doubly-connected list of pointers to the jumps; each jump knows the edge on which it is, its adjacent jumps in the cycle, and its adjacent jumps in $\preceq_v$-order.

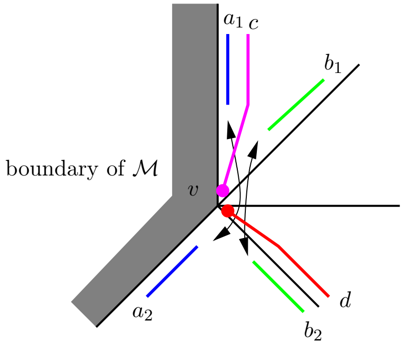


Figure 3.14: Crossings between a vertex v of H which is endpoint of at least one path. Here, there are two subpaths of size 2 at v, a_1a_2 and b_1b_2, and two jumps c and d which are endpoints of paths at v (represented with disks at their endpoints). The cyclic order $\preceq_v$ is, in clockwise order: the "boundary marker" of $\mathcal{M}$, a_1, c, b_1, d, b_2, a_2. The cyclic order $\preceq'_v$ is: the "boundary marker" of $\mathcal{M}$, c, d. Hence crossings occur at v, because of a_1a_2 and also of b_1b_2 (since a_1 and a_2, resp. b_1 and b_2, are not in the same class induced by $\preceq'_v$).

3.5.1.2 Combinatorial curves

We will now extend the definition of combinatorial cycles to the case where there are not only cycles, but also *paths*, drawn on H. A *combinatorial family of curves* S is a family of curves (paths or cycles) on H, with the data, for each oriented edge e of H, of a total order $\preceq_e$ over all the jumps of the set of curves corresponding to edge e or $-e$. These orders must be consistent, in the sense that $a \preceq_e b$ if and only if $b \preceq_{-e} a$. The orders $\preceq_v$ are defined in a slightly different way as above: the definition of $\preceq_v$ is the same as in the previous section if v is not on the boundary of $\mathcal{M}$; if v is on the boundary of $\mathcal{M}$, $\preceq_v$ is defined as above, except that there is an additional element, called "boundary marker", corresponding to the position of the boundary in this cyclic order.

The definition of a crossing must be extended because of the endpoints of the paths. Recall, by Convention 3.42, that a face of G can contain at most one point which is an endpoint of one or several paths. A crossing at some vertex v can be a crossing as defined in the previous section. Another possible type of crossing is the following. Denote by $\preceq'_v$ the restriction of the order $\preceq_v$ to the jumps which are endpoints of paths at v, and to the boundary marker, if it exists at v. This order $\preceq'_v$ naturally partitions the other jumps incident to v, see Figure 3.14. A crossing occurs in this situation if a subpath a_1a_2 of size 2, v being the target of a_1, has its two jumps a_1 and a_2 in different classes of this partition.

The data structure used to store a combinatorial family of curves is slightly more complex than for cycles: it is also necessary to store the $\preceq'_v$-order. Figure 3.15 represents this structure in the case of a family without crossing.

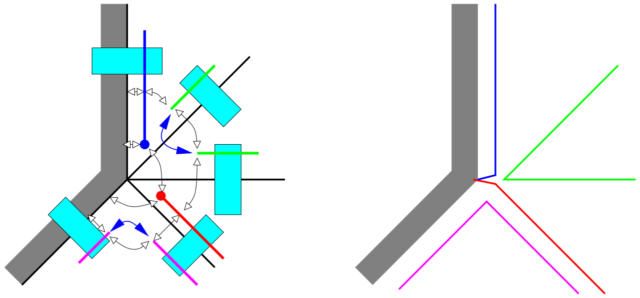


Figure 3.15: The data structure used to store a combinatorial family of curves on H, in the neighborhood of a vertex v of H with degree 5. Here, no crossing occurs. On the right, a representation of the curves spread apart on the surface explains the choice of the definition of the crossings.

3.5.1.3 Correspondence between the topological and combinatorial frameworks

Let s be a regular embedding of cycles on $\mathcal{M}$. By the regularity hypothesis, it is possible to obtain, from s, a combinatorial family of cycles, denoted by $\rho(s)$, in the following way: the order $\preceq_e$ for an edge e of H is given by the ordered list of the crossings between edge e^* of G and the cycles of s (see Figure 3.1, page 62). This combinatorial family of cycles has no crossings (we shall also say that this family is *simple*). Conversely, a combinatorial family of cycles S which is simple can be spread apart to obtain an embedding of cycles in $\rho^{-1}(S)$. Note that if $s = (s_1, \ldots, s_n)$ and $s' = (s'_1, \ldots, s'_n)$ are in $\rho^{-1}(S)$, then, for each i, s_i and s'_i are homotopic by Condition (ii) of Convention 3.42.

The same property holds for embeddings of graphs. Let s be a regular embedding of graph; it is also possible to obtain from s a combinatorial family of curves, denoted by $\rho(s)$, which is simple, that is, without crossing. The converse is still true.

A *combinatorial cut system* (by graph or by cycles) S is a simple combinatorial family of curves such that there exists a cut system (by graph or by cycles) in $\rho^{-1}(S)$. It is then clear that each element of $\rho^{-1}(S)$ is a cut system. (Recall that the map ρ is defined only for embeddings of graphs or cycles.)

The two optimization processes described in Section 3.2 use elementary shortening steps $\mathrm{shrt}_i(s)$. Actually, these shortening processes can be described, not only on cut systems, but also on combinatorial cut systems: $\rho(\mathrm{shrt}_i(s))$ depends only on $\rho(s)$, not on s. This observation justifies the notion of combinatorial family of curves that we have just described: all the computations can be done with this notion, the actual geometry of the curves does not provide any useful information.

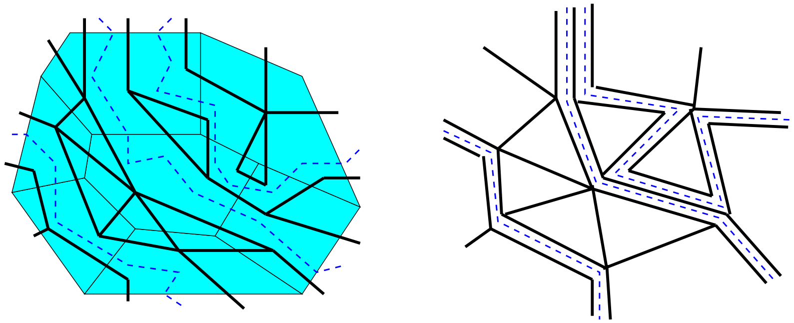


Figure 3.16: The creation of the graph $H(S)$. A: The graph $H(S)$, built from the curves of s. The graph G is depicted in light solid lines, the curves of s in dashed lines, and the graph $H(S)$ in bold lines. B: The same graph built from the combinatorial curves S (retracted on the graph H, not shown here).

3.5.2 Algorithmic study of a shortening operation

The processes of Section 3.2 are based on shortening steps shrt_i. Recall from Convention 3.42 that G is the dual graph of a polyhedral surface $\mathcal{M}$, whose vertex-edge graph is H. Starting with a combinatorial cut system S, we will explain how to compute another combinatorial cut system S' such that, if $S = \rho(s)$, then $S' \in \rho(\mathrm{shrt}_i(s))$. We will prove that the operations shrt_i boil down to finding shortest paths in a graph representing the vertex-edge graph of $\mathcal{M}$ after "cutting" $\mathcal{M}$ along some curves of the current system. We start with the description of this graph.

3.5.2.1 Description of the graph $H(S)$

Let $S = \rho(s)$ be a simple combinatorial family of curves on H. Let us define $H(S)$ as the weighted graph whose vertices are the connected components of the surface $\mathcal{M} \setminus (s \cup G)$, and whose edges join two vertices separated by a piece of an edge e^* of G; such an edge being affected of the weight of e (Figure 3.16A). Intuitively, $H(S)$ is the vertex-edge graph of surface $\mathcal{M}$ cut along the curves of S. It is clear, here again, that $H(S)$ does not depend on $s \in \rho^{-1}(S)$; indeed, the graph $H(S)$ can be computed with the sole data of S, in the following way.

Let e be a (non-oriented) edge of H. Consider the k jumps of S corresponding to e; this yields $k+1$ intervals between these jumps. Each of these intervals corresponds to an edge of $H(S)$, having the same weight as e (Figures 3.16B and 3.17). The endpoints of these edges are identified in the following way: two endpoints of edges of $H(S)$ are the same if they correspond to a same vertex v of H, and if the intervals are not separated by S (that is, the insertion of a path of size two in these two intervals does not introduce any crossing at v).

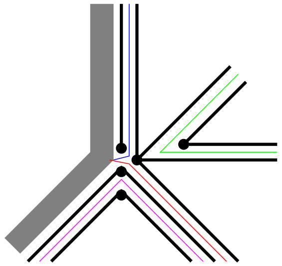


Figure 3.17: The creation of the graph $H(S)$, in bold lines (continuation of Figure 3.15).

In practice, it will never be necessary to explicitely compute the graph $H(S)$: as all operations we will have to do will be local, it will be sufficient, being given some vertex of $H(S)$ (coded by the data of a vertex v of H, by an edge e of H incident to v, and by an interval between two jumps on e), to be able to compute the neighbors of this vertex. This can be done in a time proportional to their number.

To each regular path (resp. cycle) in $\mathcal{M} \setminus s$ corresponds a path (resp. cycle) in $H(S)$, with the same length. Conversely, to each path (resp. cycle) in $H(S)$ corresponds a regular path (resp. cycle) in $\mathcal{M} \setminus s$, and, if the path (resp. cycle) of $H(S)$ is simple, the corresponding path (resp. cycle) of $\mathcal{M} \setminus s$ can be chosen so as to be simple.

3.5.2.2 Shortening operation for cut systems by graph

Let S be a combinatorial cut system by graph on $\mathcal{M}$; let $i \in [1, N]$. Let us compute the graph $H(S \setminus S_i)$ (of course, $S \setminus S_i$ means S where the combinatorial curve S_i has been removed). The path S_i thus corresponds to a path in this graph; let a and b be its endpoints. Let C be a shortest path between a and b in this graph. C is of course simple (it goes at most once through each vertex of $H(S \setminus S_i)$); hence, it defines without ambiguity a path S'_i inserted in $S \setminus S_i$, yielding thus a new combinatorial family of curves S'.

This family S' belongs to $\rho(\mathrm{shrt}_i(s))$, for each $s \in \rho^{-1}(S)$. Indeed, if $s'' \in \mathrm{shrt}_i(s)$, then s''_i defines a path in $H(S \setminus S_i)$ between a and b, which implies that s''_i is at least as long as S'_i.

3.5.2.3 Shortening operation for cut systems by cycles

The optimization of a cut system by cycles is slightly more complicated. Let s be a cut system by cycles. By the proof of Proposition 3.22, computing an element of $\mathrm{shrt}_i(s)$ can be done in the following way. Let P_i be the cylinder or the pair of pants of $s \setminus s_i$ containing s_i.

- If P_i is a cylinder, compute the shortest path q between the two boundaries of this cylinder. This yields a topological disk D_i, where points correspond-

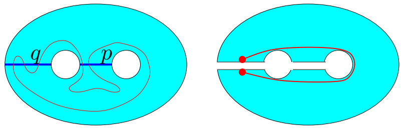


Figure 3.18: An elementary step for cut systems by cycles.

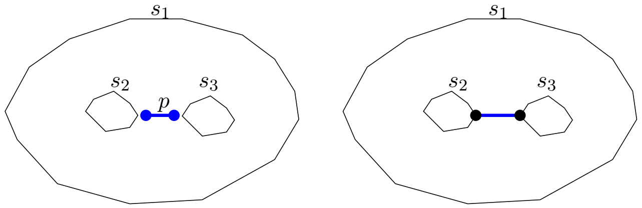


Figure 3.19: Local modifications of the data structure for the elementary step of cut systems by cycles: the path p is "stuck" to the cycles of P_i, thus transforming the cycles s_2 and s_3 to two closed paths.

  ing to points of q before cutting come by pairs. It is sufficient to consider all shortest paths between these pairs, and to take the shortest of these shortest paths.

- If P_i is a pair of pants (see Figure 3.18), one of the boundaries of P_i is homotopic to s_i. We first cut P_i along a shortest path p between the two other boundaries of P_i, thus obtaining a cylinder C. We then compute, as in the previous case, a shortest path q between the two boundaries of C, and then all the shortest paths in the topological disk which has just been built, between the pairs of points corresponding to a point of q.

Let S be a combinatorial cut system by cycles, and let $s \in \rho^{-1}(S)$. From the previous considerations, the computation of $S' = \rho(s')$ with $s' \in \mathrm{shrt}_i(s)$ reduces to computations of shortest paths in surfaces obtained by cutting $\mathcal{M}$ along some cycles. More precisely, if P_i is for example a pair of pants:

- The graph $H(S \setminus S_i)$ has several connected components, one of them, $H_i(S \setminus S_i)$, corresponding to P_i. Some vertices of this graph correspond to boundaries of P_i. We compute a shortest path P between the two "boundaries" of P_i in $H_i(S \setminus S_i)$ which are not homotopic to S_i. This path is simple and can be appended to $S \setminus S_i$ without ambiguity.

- The cycles of these boundaries of P_i, and P, are temporarily transformed into paths (Figure 3.19). This enables to ensure that, when cutting $\mathcal{M}$ along $S \setminus S_i$ and P, a topological disk is obtained (the endpoints of P must be "stuck" on the boundaries of P_i). The resulting combinatorial family thus bounds a cylinder C.

- The same operation is done with a path Q between the two boundaries of C. Q is, again, stuck on the cycles which are on the boundary of C.

- For each vertex of Q in $H(P \cup S \setminus S_i)$, thus corresponding to a pair of vertices (a, b) in $H(P \cup Q \cup S \setminus S_i)$, a shortest path is computed between a and b. The shortest of these shortest paths yields, by reidentification of its endpoints, the desired cycle S'_i.

3.5.3 Complexity of the optimization algorithms

Let K be the complexity of $\mathcal{M}$ (total number of vertices, edges, and faces), g be its genus and b be its number of boundaries. Let α be the ratio between the largest and the smallest weight of the edges of G. Let $S = (S_1, \dots, S_N)$ be a combinatorial cut system on $\mathcal{M}$. Let d be the maximal degree of a vertex in H.[10]

Let R be a family of combinatorial curves on $\mathcal{M}$ which is simple, and let e be an edge of H. The *multiplicity* of R at e is the maximal number of jumps of R on e or $-e$. The multiplicity of R is the maximum of the multiplicities of R, over all edges e of H. In particular, R has a number of jumps bounded from above by K times its multiplicity.

Let μ be the maximum over i of the multiplicity of S_i. Then the number of jumps of a curve at the beginning of the algorithm is $O(K\mu)$ and, as the length of the cut systems can only decrease, the maximal number of jumps of a curve in the course of the algorithm is $O(\alpha K\mu)$.

3.5.3.1 Cut systems by graph

We now consider the case of a cut system by graph S which is triangulated. The triangulation is not necessary at all for the algorithm, but improves its complexity (as we shall see soon, a cut system by graph can be triangulated without modifying, asymptotically, the value of the parameters N and μ, and in a time which is small compared to the time taken by the optimization algorithm).

Let $s \in \rho^{-1}(S)$. Let t_i be a shortest simple path isotopic to s_i in $\mathcal{M}$ minus all vertices of s which are not endpoints of s_i, and as short as possible among all paths having this property. We can assume that each maximal subpath of t_i included in a given face of G crosses each path of s at most $d\mu$ times; furthermore, the number of jumps of $\rho(t_i)$ is $O(\alpha K\mu)$. Hence, the length of the crossing word s/t_i is $O(\alpha dK\mu^2 N)$. It is also a bound on the number of required steps shrt, by the proof of Theorem 3.13. Each topological disk considered in the course of the algorithm can have complexity $O(\alpha K\mu)$, because such a disk is incident to at most three paths of the system, which gives the complexity of the computation of an elementary step shrt_i (dominated by the computation time of the shortest paths, using the algorithm by Henzinger *et al.* [93]): $O(\alpha K\mu)$. (The use of Dijkstra's algorithm, simpler to implement, yields a logarithmic overcost in these parameters.) There are $O(\alpha dK\mu^2 N^2)$ such steps, hence finally:

[10] Let us note that it is possible to describe the complexity of the algorithms under several forms, because numerous parameters must be taken into account. In some cases, it is possible to get rid of the parameter d in the complexity result, at the cost of a more complex definition of the multiplicity, and of an additional multiplicative factor of N.

> **Theorem 3.43** *A triangulated combinatorial cut system by graph $S = (S_1, \dots, S_N)$ being given,*
>
> - *whose maximal multiplicity of a curve is μ,*
> - *on a polyhedral surface $\mathcal{M}$, with complexity K, whose vertex-edge graph has maximal degree d, and whose longest-to-shortest edge ratio is α,*
>
> *a combinatorial cut system isotopic, with fixed vertices, to S and optimal can be computed in $O(\alpha^2 dK^2\mu^3 N^2)$ time.*

This theorem gives thus an upper bound on the complexity of the computation of the result of Theorem 3.2.

3.5.3.2 Cut systems by cycles

The reasoning is similar, but the result is not the same. The number of required steps shrt_i is still $O(\alpha dK\mu^2 N^2)$. In such a step, the complexity of the cylinder or the pair of pants P_i considered is still $O(\alpha K\mu)$. $O(\alpha K\mu)$ shortest paths must be computed on this surface, instead of one single shortest path in the previous case. The complexity is thus $O(\alpha K\mu)$ times the previous complexity:

> **Theorem 3.44** *A combinatorial cut system by cycles $S = (S_1, \dots, S_N)$ being given,*
>
> - *whose maximal complexity of a cycle is μ,*
> - *on a polyhedral surface $\mathcal{M}$, with complexity K, whose vertex-edge graph has maximal degree d, and whose longest-to-shortest edge ratio is α,*
>
> *a combinatorial cut system homotopic to S and optimal can be computed in $O(\alpha^3 dK^3\mu^4 N^2)$ time.*

It is also an upper bound on the complexity of the computation of the result of Theorem 3.7. Note that d, g, and b are bounded from above by K, and that μ is bounded from above by the complexity of the input family of paths. Both algorithms are thus polynomial in their input and in the parameter α.

3.5.4 Extension to a cut system

Our goal is now to explain how to create a cut system, possibly containing a given embedding of graph or of cycles. The previous theorems prove that the algorithms described in Section 3.2 are polynomial in α, d, K, μ, and N. The aim is now to extend the given embedding to a cut system whose values of μ and N are as small as possible.

3.5.4.1 Extension to a cut system by graph

Transformation of a schema into a reduced schema. Let us first consider the following problem: how, starting with a polygonal schema, to obtain a reduced polygonal schema?

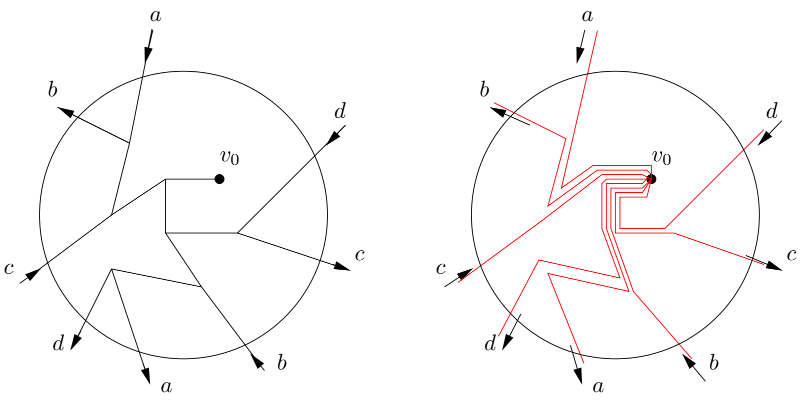


Figure 3.20: Transformation of a polygonal schema into a reduced polygonal schema (here, on a double-torus). On the left, the edges inside the circle denote the edges of the spanning tree T, and the edges a, b, c, and d crossing this disk denote the edges of $H' \setminus T$. On the right, each edge of $H' \setminus T$ is extended into a path going to the basepoint, which gives rise to a reduced polygonal schema in which each loop has multiplicity at most 2.

Lemma 3.45 *Let H' be a subgraph of H which is a polygonal schema for $\mathcal{M}$. Then, it is possible to compute a reduced polygonal schema for $\mathcal{M}$, made of a combinatorial family of paths whose jumps are on the edges of H' and on the boundaries of $\mathcal{M}$, so that:*

- *each path of the reduced schema has multiplicity at most 2;*
- *between any pair of vertices of the polygon corresponding to the reduced schema, there exists a simple path inside the disk corresponding to this schema which has multiplicity at most 4.*

This computation can be done in time $O((g+b)K)$.

Proof. This problem has been considered by Lazarus *et al.* in [105, section 5] in the case of boundaryless surfaces. Their method is as follows. A spanning tree T of H' is computed, rooted at some basepoint v_0. H' being a polygonal schema for $\mathcal{M}$, there are $2g$ edges in $H' \setminus T$. For each such edge $e = a_1a_2$, a loop with basepoint v_0 is created, which is the concatenation of $\gamma_{a_1}^{-1}$, e, and γ_{a_2}, where γ_a denotes the shortest path from a to v_0 in T (Figure 3.20). This construction provides a reduced polygonal schema in which each loop has multiplicity at most 2.

Let us consider two vertices of the reduced polygonal schema (Figure 3.21, left part). One can prove (see [105, Section 5]) that there exists a simple path of multiplicity at most 4 within the disk delimited by the boundary of the schema and which joins these two points. The method is as follows: we choose to run along one of the two simple paths on the boundary of the polygonal schema joinging

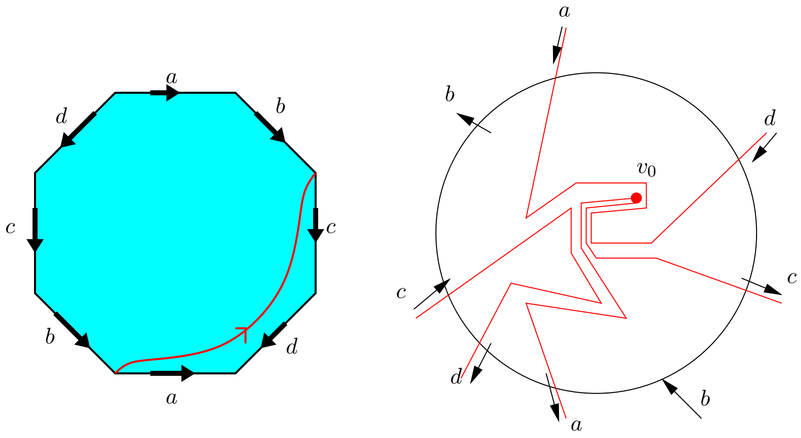


Figure 3.21: Between any pair of vertices of the reduced schema, there exists a path inside the disk which has multiplicity at most 4. This path is obtained by running along the boundary of the reduced polygonal schema and by shortcutting near the basepoint.

these two points (for example $a\bar{d}\bar{c}$ in Figure 3.21, right part). We run along the boundary of the reduced polygonal schema from the chosen initial vertex to the source endpoint of a (this yields a path whose multiplicity is 1). From the target of a to the source of $\bar{c}$, we again run along the reduced polygonal schema, but by shortcutting the return trips to the basepoint v_0 (see on the figure between $\bar{d}$ and $\bar{c}$); this gives a path with multiplicity at most 2; finally, we join the chosen target vertex by running along the reduced polygonal schema. All of this yieds a simple path, with multiplicity at most 4.

In fact, all arguments here extend to the case of surfaces with boundary. (Vertex v_0 is here chosen on the boundary of $\mathcal{M}$, because this will be the case later.) Again, a spanning tree of H' is computed. There are also $2g$ edges in $H' \setminus T$; for these edges, a loop with basepoint v_0 is created as above. These $2g$ loops yield a reduced polygonal schema of the surface $\mathcal{M}$ where the holes have been closed with topological disks. There remains, for each boundary of $\mathcal{M}$ different from the boundary incident to v_0, to introduce in the combinatorial family of curves the shortest path in T between this boundary and v_0. It is also true that each loop has multiplicity at most 2. The same argument as above applies to prove the second part of the lemma. □

A first case. Let $S = (S_1, \ldots, S_n)$ be a simple combinatorial family of paths. In a first stage, we assume that through each vertex of H passes at most one subpath of size 2 of a path of S. Cutting $\mathcal{M}$ along S is thus done in a natural way by removing the adjacencies between faces of H along the edges of H containing a jump of a path of S.

By Theorem 3.34, it is sufficient to extend S to a cut system by graph

$(S_1, \ldots, S_N)$ such that each vertex is a vertex of S or a point on the boundary of $\mathcal{M}$, with $N = O(n+g+b)$: this problem boils down to the one of computing a polygonal schema for each connected component $\mathcal{M}'$ of the surface obtained by cutting $\mathcal{M}$ along the paths of S. For this purpose, one can consider the vertex-edge graph of $\mathcal{M}'$, to which the edges incident to the boundary of $\mathcal{M}'$ are removed; consider a maximal subgraph H' of this graph, such that its complementary part in $\mathcal{M}'$ is connected (it is thus a topological disk). H' is thus a polygonal schema for $\mathcal{M}'$, hence connected, made of $2g'$ independant cycles and b' edges having an incident vertex on boundary of $\mathcal{M}'$ (if $\mathcal{M}'$ has for genus g' and for number of boundaries b'). Then, Lemma 3.45 can be applied.

Finally, it might happen that paths arrive at some vertex v on the boundary of $\mathcal{M}'$ which is not, after regluing, a vertex of S or a vertex on the boundary of $\mathcal{M}$; but, in this case, there exists a vertex v' on the same boundary of $\mathcal{M}'$ which satisfies this property, and it is possible to extend the paths arriving at v to v'. The properties of Lemma 3.45 are still satisfied. It is then possible to triangulate the polygonal schema, by choosing a vertex v_0 of the schema and by creating, if the schema has p sides, a path from v_0 to each of the $p-3$ vertices of the schema which are not incident to v_0 and distinct from v_0: the fact that each subpath of the polygonal schema has a path having the same endpoints and of multiplicity at most 4 proves that we can choose the multiplicity of each of these paths as being at most 4. Furthermore, by appending to S all these triangulated polygonal schemata, and by doubling each path, we obtain a triangulated cut system, with $O(n+g+b)$ paths by Theorem 3.34, each path having multiplicity at most 4.

The general case. Let μ be the multiplicity of S. The general method is analogous to the method we have just described. The only difficulty is to see that all the previous operations can be done with combinatorial families of curves: in the previous case, we have cut $\mathcal{M}$ along the paths of S, but this is in fact not necessary. We will in fact *mimic* the cutting of $\mathcal{M}$ along S, without actually proceeding to the cutting. Another possibility would be to subdivide the surface along an element $s \in \rho^{-1}(S)$ explicitely computed.

The first step consisted in finding topological disks on $\mathcal{M} \setminus S$ such that the complementary part of these disks contains no face of $\mathcal{M}$. The idea is now similar, except that interstices between jumps of S must be taken into account: they create (infinitesimally small) faces. We will in fact, starting with S, add cycles to this combinatorial family of curves, each of these cycles bounding a topological disk. From an algorithmic point of view, we thus maintain, at each step, a simple family of combinatorial curves made of S and also of cycles C_i.

Let us call *faces* of $\mathcal{M} \setminus S$ the union of the faces of H, as well as, for each (non-oriented) edge e of H containing $p \geq 2$ jumps of S, of the $p-1$ interstices between these jumps. These faces are linked by obvious adjacency relations:

- two interstices are adjacent if it is possible to insert a path of size 2 between these interstices, such that this path does not cross S;
- a face of H is adjacent to an interstice if there is a vertex v of H such that it is possible to insert a path of size 2 passing through v, without introducing any crossing, and such that a jump is in the interstice and the other one goes along the face of H;
- two faces of H are adjacent if they are incident, in H, to a same edge e which contains no path of s, or to a same vertex v such that it is possible, without introducing any crossing, to insert in S a path of size 2 going along these two faces.

Let us append to S a cycle C_1 which bounds a face of $\mathcal{M} \setminus S$. As much as possible, extend C_1 by augmenting the interior of C_1 with an incident face of $\mathcal{M} \setminus S$, while maintaining the fact that C_1 bounds a topological disk (the faces of $\mathcal{M} \setminus S$ do not change with this operation). Then, if some faces of $\mathcal{M} \setminus S$ have not been visited, create a second cycle C_2 around this face, and let it grow in the same way; and so on. We finally obtain a simple combinatorial family of curves, which will be denoted (improperly) $S \cup C$.

Another way to view this process is to compute a spanning forest of the graph of faces of $\mathcal{M} \setminus S$ (defined by the adjacency relations above), and, for each tree of the forest, to create a cycle C_i which contains this tree in the topological disk it bounds. In fact, algorithmically, it is easier to proceed to both stages simultaneously: during the creation of the spanning forest, we maintain the "boundary" of the spanning forest with these cycles.

Each connected component $\mathcal{M}'$ of $\mathcal{M} \setminus S$ thus contains a cycle of C, which bounds a topological disk. Hence, the part between this cycle and S cuts $\mathcal{M}'$ into a topological disk: if we consider the connected component of the graph $H(S \cup C)$ comprised in $\mathcal{M}'$ and outside the cycle of C it contains, and remove the edges of this graph going along a boundary of $\mathcal{M}'$, we obtain a polygonal schema for $\mathcal{M}'$. Then, we can proceed in the same way as in the particular case presented above to obtain a reduced schema of $\mathcal{M}'$ whose vertices are, each, either on the boundary of $\mathcal{M}$ or a vertex of S, with $2g'+b'-1$ paths (where g' and b' are the genus and the number of boundaries of $\mathcal{M}'$; if $b'=0$, there are $2g'$ paths). The multiplicity of each of these paths is bounded from above by $4\mu+4$: indeed, each path goes at most twice through each edge of $H(S \cup C)$, and each edge of H can correspond to at most $2\mu+2$ edges of $H(S \cup C)$. We extend this to a triangulated cut system by graph; the multiplicity of each path of the resulting system is at most $8\mu+8$.

All these operations have smaller complexity than the optimization of the cut system. Hence, finally, with Theorem 3.43:

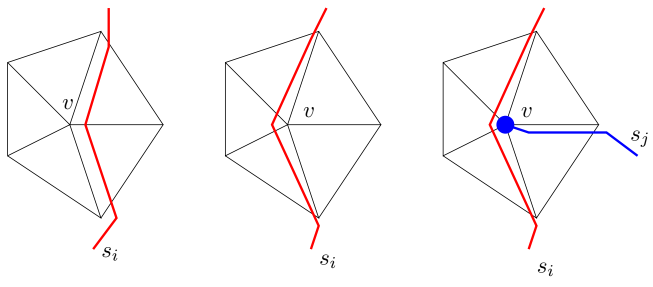


Figure 3.22: A and B: The two paths s_i drawn on the surface correspond to the same combinatorial curve, and are however non-isotopic in $\mathcal{M}$ minus v. C: This ambiguity no longer holds if a path s_j has one endpoint at v.

Theorem 3.46 *A simple combinatorial family of paths $S = (S_1, \ldots, S_n)$, containing no contractible path, being fixed,*

- *whose multiplicity is μ,*
- *on a polyhedral surface $\mathcal{M}$, of complexity K, whose vertex-edge graph has maximal degree d, and whose longest-to-shortest edge ratio is α,*

we can compute a simple combinatorial family of paths $S' = (S'_1, \ldots, S'_n)$ such that, for each i, S'_i is isotopic to S_i in $\mathcal{M}$ minus all vertices of S which are not endpoints of S_i, and as short as possible among the curves satisfying this property. This computation can be done in $O(\alpha^2 dK^2\mu^3(n+g+b)^2)$ time.

This algorithm is thus also polynomial in its input and in the parameter α.

We now have to point out the following subtlety. A simple combinatorial family of paths S being given, the expression "S_i is isotopic in $\mathcal{M}$ minus some vertices" is meaningful only if these vertices are either on the boundary of $\mathcal{M}$, or endpoints of some paths of S. Indeed, imagine that S_i runs along an edge incident to a vertex v of H and that v is interior to $\mathcal{M}$ and endpoint of no path in S. Then the isotopy class of S_i in $\mathcal{M}$ minus v is not determined: we cannot know whether S_i leaves v on its left or on its right when S_i runs along edge e (Figure 3.22A and B). This ambiguity does not remain if v is (for example) an endpoint of a path S_j, because we know that the combinatorial family of paths S is simple (Figure 3.22C). Hence, the expression "S'_i is isotopic to S_i in $\mathcal{M}$ minus all vertices of S which are not endpoints of S_i" is meaningful only because the combinatorial family S' (and not only S'_i) is known.

3.5.4.2 Extension to a cut system by cycles

Let $\mathcal{M}$ be a surface which is neither a sphere nor a disk. The vertex-edge graph of $\mathcal{M}$ is still denoted by H. Let $S = (S_1, \ldots, S_n)$ be a simple family of combinatorial

cycles, whose cycles are non-contractible. If $\mathcal{M}$ is boundaryless and $n = 0$, we first insert in S an essential cycle of $\mathcal{M}$ which is a simple cycle in H, following the paper by Erickson and Har-Peled [68, section 5]. Let $\mu \geq 1$ be the multiplicity of S. We will explain how to extend S into a cut system by cycles $(S_1, \ldots, S_N)$, in such a way that the multiplicity of a cycle is at most $4\mu+2$. In fact, it is sufficient to be able to extend S into a decomposition of $\mathcal{M}$ in cylinders and pairs of pants because, this being done, there only remains to double each cycle of S. We will prove the following lemma:

Lemma 3.47 *Let $\mu \geq 1$ and let S be a simple family of combinatorial cycles on $\mathcal{M}$, containing at least one cycle, such that, for each connected component $\mathcal{M}'$ of the surface $\mathcal{M}$ cut along S and for each edge e of $\mathcal{M}$:*

1. *if $\mathcal{M}'$ is a pair of pants, edge e can be found at most $4\mu+2$ times on the boundary of $\mathcal{M}'$;*
2. *if $\mathcal{M}'$ is not a pair of pants, edge e can be found at most $2\mu+1$ times on the boundary of $\mathcal{M}'$.*

Then one can extend S into a family S', satisfying also the conditions of this lemma, such that S' is a decomposition of $\mathcal{M}$ into cylinders and pairs of pants.

The family S obviously satisfies the hypotheses of this lemma; we obtain in particular that S' is a decomposition into cylinders and pairs of pants whose cyclea have each multiplicity bounded from above by $4\mu+2$.

Proof. Let us assume that S is not a decomposition of $\mathcal{M}$ into cylinders and pairs of pants. We will append to S one or two cycles, thus obtaining a family S' still satisfying the hypotheses of the lemma, which decomposes $\mathcal{M}$ into a disjoint union of topologically simpler surfaces. The result of the lemma will follow by induction.

First case. First assume that there exists a connected component $\mathcal{M}'$ of the surface $\mathcal{M}$ cut along S in such a way that its number of boundaries b' and its genus g' satisfy $b' \geq 4$ and/or ($g' \geq 1$ and $b' \geq 2$). We will append to S a cycle "merging" two boundaries of $\mathcal{M}'$: the new cycle will thus decompose $\mathcal{M}'$ into a pair of pants and a surface with $b'-1$ boundaries and of genus g' (Figure 3.23). Let us remember however that $\mathcal{M}'$ is not necessarily a polyhedral surface: several cycles of S can run along together, hence there might be infinitely thin faces.

We compute a path p in $H(S)$, possibly reduced to one single vertex, which joins, in $\mathcal{M}'$, two distinct boundaries of $\mathcal{M}'$, in such a way that each edge of H used by path p corresponds to no edge of a cycle which still existed in S, nor is on the boundary of $\mathcal{M}$ (to this purpose, it is sufficient to compute a shortest path in $H(S)$ between any boundary of $\mathcal{M}'$ and the union of the other boundaries). Then, we "merge" the two boundaries of $\mathcal{M}'$ which are the endpoints of p using the cycle C created by this path p and going around the two boundaries. Appending C to S, we obtain a family S' which satisfies the hypotheses of the lemma. Indeed, each edge of $\mathcal{M}$ corresponding to an edge of p is travelled exactly twice by C; such an edge then corresponds to no other boundary of the surface obtained by

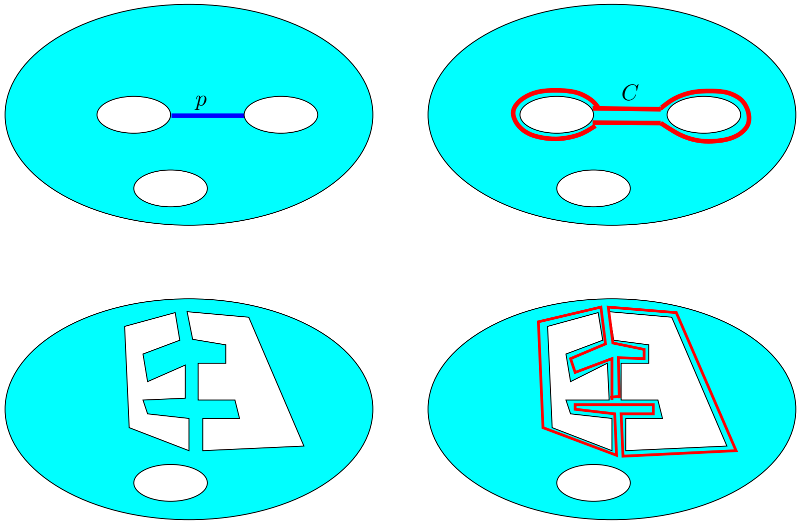


Figure 3.23: Top: cutting $\mathcal{M}'$ with a path p joining two boundaries of $\mathcal{M}'$. Bottom: two boundaries run along together several times by sharing several edges: path p is reduced to one single vertex.

cutting $\mathcal{M}$ along S, by the definition of p. C separates $\mathcal{M}'$ into two connected components:

- a pair of pants, whose boundary is made of the following edges of $\mathcal{M}$: twice each edge of $\mathcal{M}$ corresponding to an edge of path p; and, besides, each of the two merged boundaries, with multiplicity at most twice the multiplicity of the boundaries of the surface $\mathcal{M}'$, that is, at most $4\mu+2$. Hence the first property is satisfied in this pair of pants;

- the complementary part of this pair of pants, which has for boundaries (in terms of edges of $\mathcal{M}$) the set of boundaries of $\mathcal{M}'$ and twice the path p. The second property is thus true in this surface.

Second case. A connected component $\mathcal{M}'$ of $\mathcal{M}$ cut along the cycles of S has exactly one boundary and has non-zero genus (if such a component does not exist, $\mathcal{M}'$ would not have any boundary, which is impossible, or S would be a decomposition into cylinders and pairs of pants). By adapting [68, Section 5] and by using the method of evolution of a cycle bounding a topological disk (as in Section 3.5.4.1), one can compute an essential cycle C in $H(S)$ which is simple in $H(S)$ and which, moreover, runs along the boundary of $\mathcal{M}'$ in the following sense: C is the concatenation of two paths p and q in $H(S)$; p goes along the boundary of $\mathcal{M}'$; the edges of q, viewed as edges of H, are edges containing no jump of S and are not on the boundary of $\mathcal{M}$. (Of course, one of the two paths p and q can be reduced to one single vertex.)

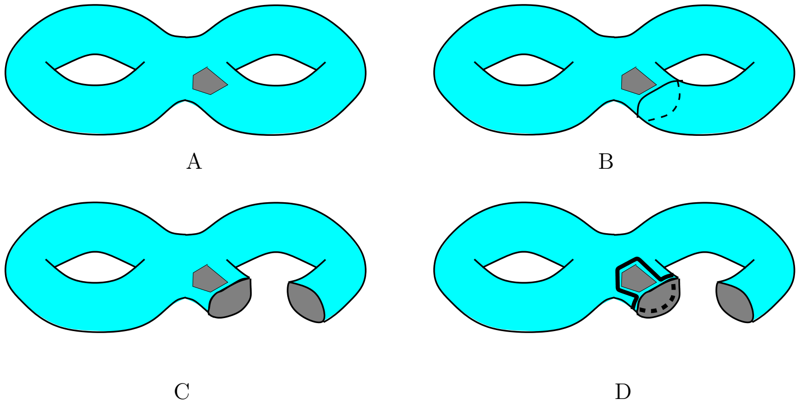


Figure 3.24: Cutting $\mathcal{M}'$ along an essential cycle. A: a double-torus with one boundary. B: Creation of an essential (here, non-separating) cycle adjacent to the boundary. C: Cutting the surface along this cycle. D: Creation of a new cycle enclosing the two adjacent boundaries.

We append C to S. After this operation, the hypotheses of the lemma are not necessarily satisfied since the edges of $\mathcal{M}$ corresponding to p can be found with a higher multiplicity in the resulting surface (Figure 3.24A, B, and C). But we can easily remedy this problem by enclosing the two boundaries incident to p by a new cycle C' (Figure 3.24D): this cycle separates the surface into a pair of pants, whose multiplicity is at most $4\mu+2$ on $\mathcal{M}$ (as in the first case), and into a surface whose boundary is made of q and of the portion of the initial boundary of $\mathcal{M}'$ which does not run along p, hence of multiplicity at most $2\mu+1$. The family S to which C and C' are appended satisfies thus the hypotheses of the lemma. □

We have thus created a decomposition of the surface $\mathcal{M}$ into cylinders and pairs of pants, each of the cycles having multiplicity at most $4\mu+2$. One can check, in the same way as in Section 3.4, that the total number of cycles N is $O(n+g+b)$. Finally, using this result and Theorem 3.44:

Theorem 3.48 *Let $\mathcal{M}$ be a surface which is neither a sphere nor a disk. A simple combinatorial family of cycles $S=(S_1,\dots,S_n)$, containing no contractible cycle, being given,*

- *whose multiplicity is μ,*

- *on a polyhedral surface $\mathcal{M}$, of complexity K, whose vertex-edge graph has maximal degree d, and whose longest-to-shortest edge ratio is α,*

we can compute a simple combinatorial family of cycles $S'=(S'_1,\dots,S'_n)$ such that, for each i, S'_i is homotopic to S_i, and as short as possible among all (non-necessarily simple) cycles homotopic to S_i. This computation can be done in $O(\alpha^3 dK^3\mu^4(n+g+b)^2)$ time.

Again, the algorithm is polynomial in its input and in α.

Conclusion

Implementation

We have presented processes to optimize a graph embedding, and simple, pairwise disjoint cycles, on a surface. A prototype has been implemented for the optimization of a fundamental system of loops on the vertex-edge graph of a boundaryless triangulated surface embedded in $\mathbf{R}^3$, the weight of an edge being equal to its Euclidean length. The implemented algorithm corresponds to the first variation mentioned in Section 3.2.3. The implementation has been done with the polyhedron data structure of the C++ library of computational geometry CGAL[11]. Once the optimal system of loops has been obtained, a smoothing of the curves by local optimization is performed, which enables to obtain paths approximating geodesics on the surface itself: in the star of each vertex of the surface, each piece of loop is replaced by a shortest path in this star; this operation is iterated until the shortening gain in one step is below a given threshold. Of course, this does not modify the homotopy classes of the loops. Figure 3.25 presents an example of result obtained on a double-torus with 1,536 faces. It would be nice to have a full and robust implementation of these algorithms (using for example CGAL), in order to check their adequacy in applications and to experiment possible extensions. The algorithms being purely combinatorial, their implementation does not cause major difficulties. In particular, degeneracy troubles, which are common in geometric algorithms, do not appear here: numerical imprecisions can yield imprecise results, but not inconsistent results (the curves computed are always simple and keep their topological properties).

Complexity study

Our algorithms are polynomial in the complexity of the surface and of the input system and in the parameter α, ratio between the extreme weights of the edges of G. It would be desirable to get rid of this parameter: in the strict sense, the algorithms are not polynomial in the size of the input because of this parameter. Unfortunately, it might be unavoidable: it constitutes the bridge between the "geometric" complexity (length of a curve) and the "combinatorial" complexity (in terms of number of edges of the path in the vertex-edge graph). Anyway, the bounds that we give on the complexity of our algorithms are probably not optimal: one can try to improve them.

Other notions of length

The algorithms we have developed work on the vertex-edge graph of a polyhedral surface; this framework may seem quite restrictive compared to possible more geometric results on polyhedral or Riemannian surfaces. Nevertheless, such a

[11] `http://www.cgal.org`

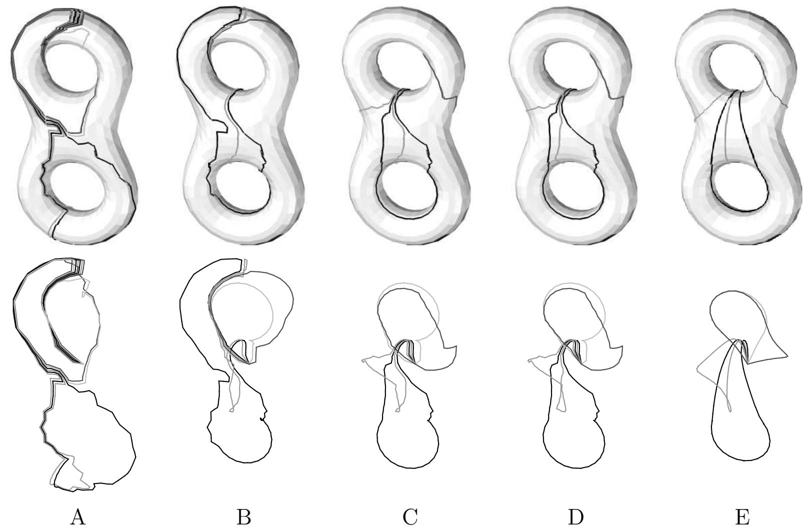


Figure 3.25: A: A canonical fundamental system of loops s, obtained with the algorithm in [105]. The basepoint is behind the surface. B: An element of $\mathrm{shrt}(s)$ (here, shrt means the optimization with the first variation of Section 3.2.3). C: An element of $\mathrm{shrt}^2(s)$. D: An element of $\mathrm{shrt}^3(s)$, which has the same length as an element of $\mathrm{shrt}^4(s)$, is an optimal system. E: The system obtained from an element of $\mathrm{shrt}^4(s)$ by local optimization of the curves on the surface ($4,000$ iterations have been necessary for this smoothing.)

framework is meaningful in some contexts (at least for the purpose of creating a parameterization to remesh a surface, as we emphasized in [2]): a polyhedral surface being given, it may be desirable, in some cases, to compute only paths in its vertex-edge graph. Another possible interest of our framework could be the approximated optimization of curves on a polyhedral surface: some algorithms, as [104], proceed by refinement of the vertex-edge graph of the surface by sampling the edges with a very particular scheme. The framework we introduced may enable, with minor modifications, to optimize in an approximated way curves, guaranteeing the quality of the approximation, in a quite simple way.

It is desirable to extend these results to the case of curves drawn on a polyhedral surface made of polygons which are isometric to Euclidean polygons, taking into account the Euclidean length of the curves. The algorithms of Section 3.2 can be described in this realm, with a few modifications: the curves of a cut system must be allowed to overlap on the surface, but not to cross. The shortening step then amounts to computing shortest paths on a polyhedral surface, and the algorithms in [116] or [31] satisfy this goal. We conjecture that the optimality theorems extend to this situation; the techniques we used seem to generalize to the case of curves drawn on the surface, but difficulties occur in the extension of the notions of cut systems and crossing words (the curves of a cut system can go along each other; in order to define the crossing word, it is necessary to define an order on the pieces of overlapping curves). On the other hand, giving the complexity of the algorithms in this framework seems difficult. The work we have presented, with its algorithms as well as its proof techniques, could thus be used as starting point for this extension.

It would be interesting to generalize this work to the case of curves drawn on a Riemannian surface. But, apart from the difficulties mentioned above, our process crucially uses the fact that the initial cut system and a shortest curve within a given homotopy class cross at a finite number of points. It can happen that it is not true in some pathological cases on a Riemannian surface (think about a fundamental system of loops which winds infinitely many times around the basepoint). However, for the optimality proof, a nice fact, compared to the case of a polyhedral surface, is that, on a Riemannian surface, two maximal geodesics which are tangent at some point are equal: in this sense, the polyhedral case is more problematic, because two geodesics can go along each other for some time and then diverge. We conjecture, for example, that each loop of an optimal fundamental system of loops on a Riemannian surface is as short as possible among the simple loops of its homotopy class.

It seems also natural to search extensions of our results to non-orientable surfaces. However, several difficulties occur in this case. First, our definition of the crossing word uses in a crucial manner the orientability of the surface. Maybe it is sufficient to consider a weakened version of the crossing words where the orientation of the crossings would not be specified, and defining an elementary reduction as the removal of two identical consecutive letters. On the other hand, Theorem 1.7, used for the optimality of a fundamental system of loops, is more complicated for non-orientable surfaces. Finally, the notion of cut system by cycles should be generalized to the case of non-orientable surfaces.

Optimization without fixing the homotopy or isotopy class

Continuations of our works are possible while remaining in the chosen framework, that is, when curves are drawn on the vertex-edge graph of the polyhedral surface. First, we are not able to compute a shortest simple path homotopic to a given simple path (except for loops or paths with endpoints on the boundary of the surface). To this purpose, we could try to understand whether other obstructions as those from the counterexample by Feustel (Figure 1.18, page 27) can happen.

Still in this realm, we have described a way to compute a fundamental system of loops as short as possible within a given homotopy class. But this does not provide any method for computing the shortest fundamental system of loops (among all homotopy classes). An open question is to determine the complexity of this problem. Erickson and Har-Peled proved that it is NP-hard to determine a shortest polygonal schema (even in the case where the weights of the edges equal one), and we have proved that the computation of a shortest fundamental system of loops within a given homotopy class is a polynomial problem (with unit weights). The problem we present now is inbetween these two problems.

Let us imagine, for the sake of simplification, that we only consider systems with the same basepoint. Brahana transformations provide an elementary way to switch from a class of fundamental systems to another one; the homotopy classes of systems can thus be considered as the vertices of a graph whose edges are the Brahana transformations. The study of this graph could give indications regarding how to find the shortest fundamental system of loops: is it sufficient to proceed by successive Brahana transformations which decrease the length of the fundamental system of loops which is optimal in this homotopy class?

Similarly, what is the complexity of computing the shortest pants decomposition of a surface? A greedy algorithm to compute a pants decomposition of the surface is to iteratively cut the surface along a shortest essential cycle; does this method give the shortest pants decomposition? We believe that these questions are important, because this would provide decompositions of the surface which are as canonical as possible. Furthermore, it is interesting in computer graphics applications to decompose a surface with curves as short as possible, independantly from the homotopy class.

In the same vein, one could look for the shortest embedding of a given graph on a given surface. We have shown how to compute the shortest graph embedding within a given isotopy class (with fixed vertices). A generalization is still to be found.

Optimization of non-simple curves

We have not tried to optimize non-simple curves. How, a non-simple curve being given, to find the shortest homotopic curve? In the case of (orientable) surfaces with at least one boundary, we can give a sketch of solution for this purpose. The idea is, with our algorithm, to compute an optimal cut system by paths s (within an arbitrary homotopy class) on this surface. Now, let c be a path on this surface, and let s/c be the crossing word of this path with system s. Let c' be the shortest path homotopic to c. It should be clear, from Section 3.3.1, that

s/c reduces to s/c': indeed, s/c and s/c' reduce to the same irreducible word, and s/c' is irreducible (or, if not, c' can be changed into some path c'' of the same length and same homotopy class, such that s/c'' is obtained from s/c' with this reduction). Hence, the succession of the paths of s crossed by c' is known, and it is thus possible to build a portion of the universal covering space, of size polynomial in the input of the algorithm and in the longest-to-shortest edge ratio, which contains a lift of c'. This method can be extended to the case of cycles. On the contrary, for a boundaryless surface, an algorithm still remains to be found.

Disjoint curves in the vertex-edge graph

In applications, it is desirable to compute a decomposition of the surface with *disjoint* curves on the vertex-edge graph of the surface, that is, each edge of the surface contains at most one edge of the decomposition. This is not always possible in the case of fundamental systems of loops, as we pointed out in Section 2.2.1.1, page 38. But how, requiring that the curves are disjoint (or fixing a limit on the number of edges of the decomposition passing through a given edge of the surface), to find an optimal decomposition? It is certainly possible to find heuristics to favour the families of curves which pass along disjoint edges, using our algorithms: for example, at each shortening step of a curve, the weight of the edges of the surface along which some curves pass can be increased. One can hope that this modification yields good results in practice, but, with such a modification, the convergence of the algorithm is not assured, and there are probably some conditions to be found to ensure its stability. This would deserve a more complete study.

Appendix: Exponential cost of the naïve optimization method

Lemma 3.49 *There exists a surface $\mathcal{M}$, with some graph G (used to compute the lengths of the paths — see Section 3.1), whose edges have unit weights ($\alpha = 1$), and a basepoint v_0, such that the number of lifts of v_0 at distance at most d of a given lift is $2^{\Omega(d)}$.*

Proof. We choose for $\mathcal{M}$ a double-torus, having a canonical fundamental system of loops with basepoint v_0, whose elements have a length smaller or equal than some fixed K. Let us call a, b, c, and d the homotopy classes of these loops. The fundamental group of $\mathcal{M}$ is the free group generated by a, b, c, and d, quotiented by the relation $a.b.a^{-1}.b^{-1}.c.d.c^{-1}.d^{-1} = 1$.

Any word w on the generators a, b, c, and there inverses corresponds to a homotopy class which is representable by a loop of size $O(K.|w|)$. Equivalently, if we fix a lift v_0^ϵ of v_0, such a w corresponds to a lift of v_0, called $R(w)$, at distance $O(K.|w|)$ of v_0^ϵ.

Furthermore, the subgroup generated by a, b, and c is a free group. This implies that, if w and w' are two distinct reduced words, then $R(w) \neq R(w')$. The number of reduced words of length n on the alphabet $\{a, b, c, a^{-1}, b^{-1}, c^{-1}\}$

is $6.5^{n-1} \geq 5^n$, and each of these words correspond to some lift of v_0 at distance $O(Kn)$ of v_0^ϵ. Hence, the number of lifts at distance at most d of v_0^ϵ is $\Omega(5^{d/K})$. □

An integer d being fixed, there always exists a loop ℓ on $\mathcal{M}$ which is as short as possible in its homotopy class, and which has length at least d (it is sufficient to take for ℓ an optimal representant of the homotopy class of $a.b^p$, where p is sufficiently large: such an ℓ can even be chosen so as to be simple). Hence, the "naïve" search of a shortest loop homotopic to ℓ, by computation of the shortest path between the endpoints of one of its lifts in the universal covering space of $\mathcal{M}$, will require $2^{\Omega(d)}$ time.

Chapter 4

Tutte's barycenter method applied to isotopies

This chapter contains the paper [42], written with M. Pocchiola and G. Vegter, initiated in my D.E.A. report [39], with a few modifications (in particular, the proof of Tutte's barycentric theorem has been included here).

Abstract. The topic of this study is Tutte's barycentric embedding theorem [152]. We first give another proof of this theorem; we then present a method to build isotopies of triangulations in the plane, based on Tutte's theorem and on the Maxwell–Cremona correspondence, a result of rigidity theory. Finally, we give a counter-example proving that the analogue of Tutte's theorem in dimension 3 is false.

Introduction

Background on Tutte's barycentric theorem

In this chapter, we will use basic graph theory terminology, see for example [20]. Let $G = (V, E)$ be a planar graph. A *(straight-line) mapping* Γ of G into the plane is a function $\Gamma : V \cup E \to \mathcal{P}(\mathbf{R}^2)$ which maps a vertex $v \in V$ to a point in $\mathbf{R}^2$ and an edge $e = uv \in E$ to the straight line segment joining $\Gamma(u)$ and $\Gamma(v)$. A mapping is an *embedding* if distinct vertices are mapped to distinct points, and the open segment of each edge does not intersect any other open segment of an edge or a vertex. (In this chapter, all mappings and embeddings are straight-line unless otherwise specified.)

In 1963, Tutte [152] gave a way to build embeddings of any planar, 3-connected graph $G = (V, E)$. Let C be a cycle whose vertices are the vertices of a face of G in some (not necessarily straight-line) embedding Γ' of G. Let Γ be a mapping of G into the plane, satisfying the conditions:

i. the set V_e of the vertices of the cycle C is mapped to the vertices of a strictly convex polygon Q, in such a way that the order of the points is respected;

ii. each vertex in $V_i = V \setminus V_e$ is a barycenter with positive coefficients of its adjacent vertices (Tutte assumed all coefficients to be equal to 1, but the proof extends without changes to this case). In other words, the images $\overline{v}$ of the vertices v under Γ are obtained by solving a linear system (S): for each $u \in V_i, \sum_{v|uv\in E} \lambda_{uv}(\overline{u} - \overline{v}) = 0$, where the λ_{uv} are positive reals. It can be shown that the system (S) admits a unique solution, see Section 4.5.

Theorem 4.1 (Tutte's Theorem) Γ *is an embedding of* G *into the plane, with strictly convex interior faces.*

The history of graph embeddings began early. Fáry [69], Stein [144] and Wagner [155] independantly showed that any planar graph admits a (straight-line) embedding. Now, the literature on this subject is abundant; a survey on graph drawing is [55]. See also the books by Ziegler [158] and Richter-Gebert [129] for the important connection between graphs and polytopes by Steinitz' theorem (any 3-connected, planar graph can be realized as the 1-skeleton of a 3D polytope).

Recent works focus on finding embeddings of graphs so that the coordinates of the vertices are integers with absolute value as small as possible; there is a linear algorithm [36] to embed graphs with $n+2$ vertices on the $(n \times n)$-grid with convex faces. Tutte's method with unit coefficients is not a valuable method for this purpose, since it can yield embeddings with exponential area if all coordinates are integers [58]. Any 3-connected planar graph with $n+1$ faces can be embedded on the $(n \times n)$-grid [70]. Other criteria are also interesting, such as controlling the shapes of the faces and/or minimizing the area of the embedding if a minimum distance between two vertices, or between a vertex and a non-incident edge, is imposed [35]. Another topic of interest is also to have an effective version of Steinitz' theorem. This can be done on the cubic grid of size 2^{13n^2}, where n is the number of vertices of the graph [129, p. 143].

Embeddings are not the only way to represent graphs; among others, an alternative approach is to represent the graph with a set of non-overlapping disks in the plane, one for each vertex, so that two vertices are adjacent if and only if the corresponding disks are tangent. This approach is called *circle packing* [136, 22].

Tutte's theorem is important in graph drawing; this is due to its simplicity and its geometric nature. It is the cornerstone of *Floater's parameterization technique* [72] for surface parameterization in computer graphics, used in multiresolution problems [59], texture mapping [109], and morphing [100, 73, 81].

In his paper [152], in addition to showing Theorem 4.1, Tutte simultaneously proves again Kuratowski's planarity criterion [103] of 1930: a graph is planar unless it contains a subdivision of one of the two Kuratowski graphs K_5 and $K_{3,3}$. The proofs of both results are entangled together in Tutte's paper; the consequence is that proving Theorem 4.1 by his method is long and involves quite a lot of graph theory terminology. Later, short proofs of Kuratowski's criterion were given by Thomassen [146], making Tutte's graph-theoretic viewpoint less attractive for the proof of Theorem 4.1.

Other proofs of this theorem exist in the literature, using a more geometric viewpoint. Becker and Hotz [10] use the notion of "quasi-planarity" as the limit case of a planar situation, which yields complicated notations and tedious case analyses; the structure of their paper is non-obvious and the proof is really long. Y. Colin de Verdière [43] shows the result, only for triangulated graphs, on arbitrary surfaces of non-positive curvature using the Gauss-Bonnet formula. More recently, in 1996, Richter-Gebert [129, Section 12.2] has given a simple and transparent proof of this theorem.

Our work

Proof of Tutte's theorem

We wrote in 2000 another proof of Tutte's theorem, without being aware of the existence of Richter-Gebert's proof at that time. Our proof appeared in a preliminary version of [42], and is included in this chapter. It ressembles Richter-Gebert's proof in the fact that the key idea of counting all angles of the mapping in two different ways has also been used in his proof, but some arguments are different.

Our proof is transparent and progressive; it consists of two clearly delimited stages. First, we show Tutte's result without effort under two additional restrictions:

iii. the graph G is triangulated: every face of Γ', except possibly the face corresponding to the cycle C, is a triangle;

iv. the images of these triangles under the mapping Γ are non-degenerate, *i.e.*, their interior is non-empty.

After that, we deal with the degeneracies, which are the core of the problem (in our proof as well as in other proofs): we show that under hypotheses (i) and (ii), three vertices belonging to a face are not on the same line. This step uses the non-planarity of $K_{3,3}$ together with simple geometric ideas. Then, the generalization to arbitrary 3-connected graphs comes easily.

Isotopies

Tutte's theorem yields a method, described by Floater and Gotsman [73] and Gotsman and Surazhsky [81], to morph two triangulations, the boundary being the same convex polygon in both embeddings. One can compute coefficients $\lambda_{uv} > 0$, for each interior vertex u and each neighbor v of u, so that u is the barycenter with coefficients $(\lambda_{uv})_v$ of its neighbors in the initial embedding. Doing the same for the final embedding and interpolating linearly the coefficients yields an isotopy (a continuous family of embeddings) by Tutte's theorem. This method leaves some freedom for the computation of the barycentric coefficients of the vertices in both embeddings. Hence, we study the following natural question: is it possible to apply the same technique, with the additional restriction that the coefficients are symmetric ($\lambda_{uv} = \lambda_{vu}$)? The interest is that this has a clear and appealing physical interpretation: fix the exterior vertices and edges and replace each interior edge joining two vertices u and v by a spring with rigidity λ_{uv}; then the equilibrium state of this physical system is the solution of the system (S).

The problem of computing such symmetric coefficients is solved with Maxwell–Cremona's theorem from rigidity theory. The drawback of our method is that these coefficients are not always positive, hence Tutte's theorem does not apply in all cases. After small experiments (with 20 vertices or so), we thought that our method always yielded an isotopy, even if some weights were negative. This is not the case, and we have small examples refuting this conjecture. However, our method gives positive coefficients if both embeddings are in the rather general class of regular triangulations (recall that a regular subdivision is the projection of the lower faces of a polytope generated by a family of points). This idea of replacing edges of a graph by springs has been used in several other contexts: in mechanics [156], for graph connectivity computation [110], in an algorithmic study of operations on polyhedra [96]. Force-directed algorithms (see [55]) are an important class of graph drawing methods that use springs (with, additionally, electric and/or magnetic forces). In [77] is described a tool for the visualization of evolving embeddings of graphs.

Generalization to 3D space

Finally, we study the extension of Tutte's theorem to three dimensions. We present an overview of the proof that there exist two triangulations of a tetrahedron which are combinatorially equivalent but for which there is yet no linear isotopy from one to the other, a fact which is specific to spaces of dimension ≥ 3. This result has been stated by Starbird in [141]; we give an outline of the proof and explain parts of the proof not written in his paper and required to show this theorem. Then we show that the natural generalization of Tutte's barycentric embedding theorem is false in 3D. The translation of Tutte's hypotheses (in the triangulated case) from 2D to 3D is as follows: consider an embedding of a simplicial 3-complex K into $\mathbf{R}^3$, the boundary being a convex polyhedron. If a mapping of K into $\mathbf{R}^3$, with the same boundary, is so that each interior vertex is barycenter with positive coefficients of its neighbors, then we would expect that it is an embedding. It turns out that this fact is false. To our knowledge, this attempt of generalizing Tutte's theorem for 3D complexes is new, and our refutation of this extension raises interesting open questions, in the context of isotopies as well as in view of embedding 3-complexes.

4.1 Proof of Tutte's theorem

We prove here Tutte's theorem ([152], relying on [151]).

We first rephrase condition (ii) in more compact terms. Define the *strict convex hull* of a set of points to be the interior, in the space affinely generated by these points, of the convex hull of these points. It is then easy to see that condition (ii) is equivalent to the following:

ii'. each $v \in V_i$ lies in the strict convex hull of its adjacent vertices.

A proof of this equivalence is provided for completeness in Section 4.4. We thus need not use System (S) anymore; the proof of its invertibility is easy and

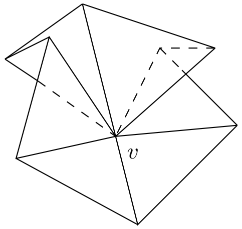


Figure 4.1: The triangles with v as a vertex, involved in the computation of $\sigma(v)$.

not necessary for the proof of Tutte's theorem, we defer it to Section 4.5.

Recall that Γ' is a (not necessarily straight-line) embedding of G with facial cycle C. In the sequel, Γ' is our reference embedding, and we shall call the *faces of* G (or *triangles* if (iii) is satisfied) the faces of the embedding Γ' except the face bounded by C. We first state a preliminary lemma, assuming the hypotheses of Theorem 4.1:

Lemma 4.2 *Each $v \in V_i$ is mapped by Γ into the interior of the polygon Q.*

Proof. First, each vertex in V_i is mapped, by Γ, into the interior or the boundary of the polygon Q. For if this is not the case, by convexity of Q, there is a vertex $v \in V_i$ so that Q and $\overline{v}$ are separated by a line D. Among the vertices whose images under Γ lie on the same side of D as v, consider those which are the farthest from D. Obviously, at least one of these vertices cannot be in the strict convex hull of its adjacent vertices.

Suppose that a vertex $v \in V_i$ is mapped into the boundary of Q, on a line D which contains an edge of Q. Because the image of any vertex lies in the same (closed) half-plane bounded by D, and by condition (ii'), the vertices in V_i which are adjacent to v are also contained in D. Thus, all vertices of the connected component of v in $G - V_e$ lie on D. This contradicts the 3-connectivity of G, because removing the two vertices of V_e which are on D destroys the connectivity of G. $\square$

4.1.1 Proof of the theorem in a special case

Our intermediate goal is now to prove the theorem under the additional assumptions (iii) and (iv). We first show a local planarity property for Γ.

Lemma 4.3 *Under assumptions (iii) and (iv), the interiors of the images of two distinct triangles of G which share a common vertex do not overlap.*

Proof. By (iv), the angles of a triangle are well-defined and positive. We first introduce some terminology. For each vertex v, let $\alpha(v)$ be equal to 2π if $v \in V_i$, and to the angle of the polygon Q at v if $v \in V_e$; let $\sigma(v)$ be the sum, over all the triangles incident to v, of the angle of the image of such a triangle at v under the mapping Γ (Figure 4.1). Our aim is to show that, for each vertex v, $\sigma(v) = \alpha(v)$, that is, there is no "folding" at v in the mapping Γ. Because all faces of G are triangles, the structure of G in the neighborhood of a vertex v is quite simple: the vertices adjacent to v form a cycle (if $v \in V_i$) or a path (if $v \in V_e$). Let us call $v_1, \ldots, v_p$ the neighbors of v in the order of this cycle (or path).

A key ingredient of the proof of Lemma 4.3 is the following fact: $\sigma(v) \geq \alpha(v)$, with equality if and only if the triangles incident to v do not overlap. To prove this, let $\theta(uvw)$ be the geometric angle (between 0 and π) of the triangle uvw at v in the mapping Γ, and assume v to be in V_i (the proof is easier for v in V_e). By (ii') and (iv), $\overline{v}$ lies in the interior of the convex hull of $\overline{v}_1, \ldots, \overline{v}_p$. It also lies in the interior of the convex hull of three of these vertices, as can easily be shown by hand (this is Carathéodory's Theorem in dimension two). Hence, there exist i, j and k so that $1 \leq i < j < k \leq p$ so that the sum of the angles $\theta(v_i v v_j)$, $\theta(v_j v v_k)$ and $\theta(v_k v v_i)$ equals 2π. We have $\sum_{q=i}^{j-1} \theta(v_q v v_{q+1}) \geq \theta(v_i v v_j)$, and similar relations between j and k and between k and i. Adding these three inequalities, we obtain that $\sigma(v) \geq 2\pi$. Moreover, equality holds if and only if there is equality in all previous sums, that is, the ordering of the vertices $v_1, \ldots, v_p$ around v is preserved in Γ, which is also equivalent to the fact that the triangles incident to v do not overlap.

Now, let t be the total number of triangles. We have:

$$(2|V_i| + |V_e| - 2)\pi = \sum_{v \in V} \alpha(v) \leq \sum_{v \in V} \sigma(v) = \pi t. \tag{4.1}$$

The first equality is a consequence of the fact that the sum of the angles of the polygon Q is $(|V_e| - 2)\pi$, the inequality has been shown above, and the second equality is true because the sum of the angles of a triangle in the plane equals π. We claim that, on the other hand, the leftmost and rightmost members of (4.1) are equal. Indeed, Euler's formula, applied to the planar graph G, yields (if e is the number of interior edges):

$$(|V_i| + |V_e|) - (|V_e| + e) + (t + 1) = 2. \tag{4.2}$$

The fact that every face of G has three edges is expressed by:

$$3t = 2e + |V_e|. \tag{4.3}$$

Combining equations (4.2) and (4.3) to eliminate e leads to equality of the extreme members of (4.1), as claimed. Since also $\sigma(v) \geq \alpha(v)$, we get: for each $v \in V, \sigma(v) = \alpha(v)$. Using then the equality case in the fact given above, we obtain that the triangles incident to v do not overlap, which concludes the proof. $\square$

We now show global planarity, that is, Γ is an embedding.

Lemma 4.4 *Under restricting assumptions (iii) and (iv), Theorem 4.1 holds.*

Proof. First note that the vertices, edges and triangles of G define an (abstract) 2-dimensional simplicial complex K. In fact, one can view K as a 2-manifold with boundary: each point of this manifold is defined by its barycentric coordinates in a triangle of G, with the obvious identifications of points on edges or vertices. Γ induces a map from this manifold K into Q: for each point p in K, determined by its barycentric coordinates in a triangle uvw, its image $\Gamma(p)$ is the point in the triangle $\overline{uvw}$ with the same barycentric coordinates. In this setting, Γ is continuous and even a local homeomorphism: each point p has a neighborhood $N(p)$ so that $\Gamma|_{N(p)}$ is a homeomorphism on its image set. This is clear for points p in the interior of a triangle and Lemma 4.3 proves this fact if p belongs to an edge or is a vertex of a triangle. If a is a point inside or on the boundary of Q, $n(a) = |\Gamma^{-1}(a)|$ is finite; for otherwise, by compactness, there would be an accumulation point of the set $\Gamma^{-1}(a)$, contradicting the local homeomorphism property.

Let a be a point in or on the boundary of Q, and $p = n(a) \geq 0$; we show that $n(b) = p$ for b sufficiently close to a (that is, the function n is locally constant). Let $N_1, \ldots, N_p$ be disjoint open neighborhoods of each of the points in $\Gamma^{-1}(a)$, chosen small enough so that $\Gamma|_{N_i}$ is a homeomorphism for each i. Let $N = \Gamma(N_1) \cap \cdots \cap \Gamma(N_p)$. N is a neighborhood of a; $F = \Gamma(K \setminus (N_1 \cup \cdots \cup N_p))$ is a compact set which does not contain a; hence $N' = N \setminus F$ is a neighborhood of a. Each $b \in N'$ has exactly p preimages in $N_1 \cup \cdots \cup N_p$ because $b \in N$ and no preimage outside this set because $b \notin F$. Thus, by connectivity of Q, n is constant; its value is 1 on the boundary of Q by Lemma 4.2, hence Γ is a homeomorphism. The proof is complete. □

4.1.2 The general case

We have proved the theorem in a particular case; we will use this result in the sequel. From this point, unless stated otherwise, we do not assume conditions (iii) and (iv) anymore, but only the hypotheses of Theorem 4.1. The goal is to show that some degenerate cases cannot occur, using the 3-connectivity of G. We first state a quite general lemma, inspired by Tutte [152], which we call the Y–lemma in view of the geometry of the problem. The situation is depicted in Figure 4.2. Note that, in this section, any path in a graph is supposed to be simple and non-degenerate.

Lemma 4.5 (Y–lemma) *Let v_1, v_2, v_3 and v be pairwise distinct vertices of a graph H. Assume, for $i = 1, 2, 3$, that there is a path P_i from v_i to v which avoids the v_j's (for $j \neq i$). Then there exist three paths P_i', from v_i to a common vertex v', which are pairwise disjoint (except at v').*

Proof. First, using P_1 and P_2, we easily get a (simple) path R from v_1 to v_2, so that R and P_1 have the same first edge $v_1 z$. Then we consider the path P_3. If this path P_3 intersects R, let v' be the first vertex of intersection on P_3. v' splits R in two parts, which we call P_1' (from v_1 to v') and P_2' (from v_2 to v'); P_3' is the part of P_3 going from v_3 to v', with loops removed (if any). The P_i''s satisfy the property stated in the lemma. If P_3 does not intersect R, we call v' the last vertex on P_1 (when going from v_1 to v) which is also on R. Such a vertex exists

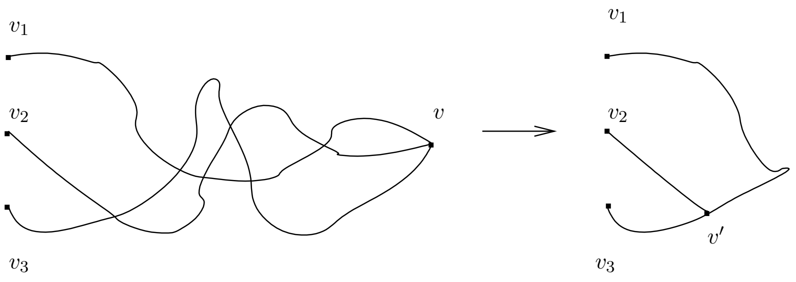


Figure 4.2: The situation in the Y–lemma.

and is different from v_1 because $v_1 z$ is the first edge of R and P_1. Let P_3' be the path defined by P_3 followed by the part of the path P_1 which goes from v to v', with loops removed (if any). v' splits R in two parts, which we call P_1' and P_2'. The paths P_i''s satisfy the desired property. □

We now come back to the situation of Theorem 4.1, and we introduce some geometric definitions, partially taken from [152]. We will represent a line in the plane by the zero set of a non-constant affine form. Henceforth, φ is such an affine form. A vertex v of G is called *φ-active* if there is a vertex v' adjacent to v so that $\varphi(\overline{v}) \neq \varphi(\overline{v}')$, *$\varphi$-inactive* otherwise. The *φ_+-poles* are the vertices $v \in V$ so that $\varphi(\overline{v})$ is maximal; the definition for the *φ_--poles* is analogous. The *φ-poles* are the φ_+-poles and the φ_--poles. By Lemma 4.2, a φ-pole must be in V_e. It is then clear that there are exactly one or two φ_+-poles and that, in the latter case, they are connected by an edge of Q. If $\overline{v}_1, \ldots, \overline{v}_k$ lie on the line $\varphi = 0$, $G(\varphi_+, v_1, \ldots, v_k)$ is the graph induced by the vertices lying in the half-plane $\varphi > 0$, to which we add the vertices $v_1, \ldots, v_k$ and all edges from one of these vertices to a vertex in $\varphi > 0$. Let $G(\varphi)$ be the subgraph of G induced by the vertices v lying on the line $\varphi = 0$. The following lemma was also shown in [152].

Lemma 4.6 *Let v be a φ-active vertex so that $\varphi(\overline{v}) = 0$; assume that v is not a φ_+-pole. Then there exists a path in $G(\varphi_+, v)$ from v to a φ_+-pole of G.*

Proof. The problem boils down to this: given a φ-active vertex w, which is not a φ_+-pole, prove that it is possible to find a neighbor of w which has a greater value of φ and is also φ-active. If $w \in V_e$, there exists in V_e a vertex adjacent to w which has a greater value of φ; this vertex is also φ-active. If $w \in V_i$, then w has neighbors in both increasing and decreasing directions of φ, because a vertex is in the strict convex hull of its adjacent vertices (hypothesis (ii')) and because w is φ-active. It is therefore possible to find an adjacent vertex with a greater value of φ. This vertex is also φ-active. □

The two following lemmas show that some degenerate cases cannot occur. The first one is along the lines of [152], contrary to the second one which uses another argument.

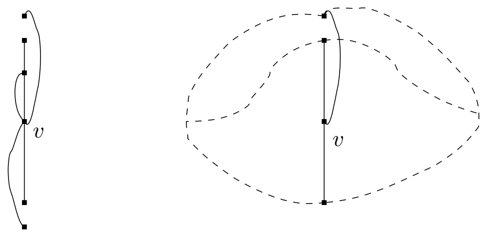


Figure 4.3: A summary of the proof of Lemma 4.7.

Lemma 4.7 *For any φ, G has no φ-inactive vertex.*

PROOF. Suppose that there is a φ-inactive vertex v. Figure 4.3 summarizes the proof: we show that the planar graph G contains a subdivision of the bipartite graph $K_{3,3}$, which is impossible (see for example [148]). Using the fact that G is 3-connected, we can see the existence, in $G(\varphi)$, of three distinct φ-active vertices $v_i, i = 1, 2, 3$, and three paths P_i joining v to v_i, so that, for any i, the path P_i does not contain any vertex v_j for $j \neq i$. Indeed, let w be a vertex of G so that $\varphi(\overline{w}) \neq 0$. By connectivity of G, take a path from v to w and, on this path, take the first φ-active vertex and call it v_1. Do the same in $G - \{v_1\}$ and choose v_2 (use 2-connectivity). Finally, use 3-connectivity to select v_3 in $G - \{v_1, v_2\}$.

Applying then the Y–lemma in $G(\varphi)$, we get the existence of a vertex v' in $G(\varphi)$, together with three distinct paths (except at v') P_i from v_i to v' in $G(\varphi)$. We now use Lemma 4.6. We have the existence, in $G(\varphi_+, v_1, v_2, v_3)$, of three paths Q_i joining v_i to a vertex x so that $\varphi(\overline{x}) > 0$. Then, the Y–lemma allows us to assume, by changing x and the Q_i's if necessary, that these three paths are disjoint (except at x). Similarly, in $G(\varphi_-, v_1, v_2, v_3)$, we have three disjoint paths R_i joining v_i to a vertex y so that $\varphi(\overline{y}) < 0$. Using the paths P_i, Q_i and R_i, which are all pairwise disjoint except at their endpoints, and the vertices x, v', y and v_1, v_2, v_3, we get a subdivision of the graph $K_{3,3}$. This contradicts the planarity of G. □

Lemma 4.8 *Let v_i, $i = 1, 2, 3$, be three vertices of a face of G. Then, under Γ, the v_i's are not collinear.*

PROOF. Suppose the v_i's are on the line $\varphi = 0$. Figure 4.4 gives the essential ideas of the proof: we again find a subdivision of $K_{3,3}$. By Lemma 4.7, the v_i's are φ-active. Since the v_i's are collinear, at least one of them is in V_i, so none of them is a φ-pole. Again, Lemma 4.6 and the Y–lemma show the existence of a vertex x and three disjoint paths (except at x) Q_i joining v_i to x in $G(\varphi_+, v_1, v_2, v_3)$. Using a similar argument on the other side of the line $\varphi = 0$, we finally obtain the existence of x, y and six disjoint (except at their endpoints) paths joining x or y to the v_i's. Let G' be the graph G to which we add a vertex w linked to the v_i's. Because the v_i's belong to a common face, G' is planar. But it also contains a subdivision of $K_{3,3}$ (with the six paths described above and the three new paths joining w to the v_i's), which is impossible. □

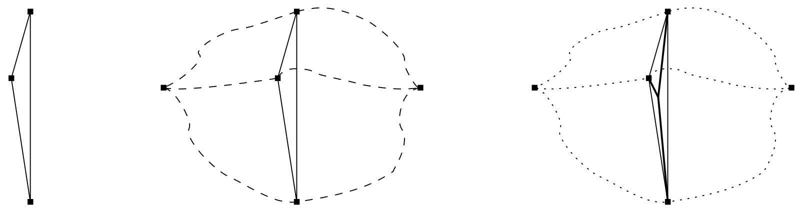


Figure 4.4: A summary of the proof of Lemma 4.8.

The two previous lemmas have been shown merely under the hypotheses of Theorem 4.1. Lemma 4.7 will be used to deal with the non-triangulated case. But here, as a straightforward consequence, we get the proof of the theorem in the triangulated case:

Corollary 4.9 *Theorem 4.1 holds when (iii) is satisfied, that is, in the particular case where each face of G is a triangle.*

PROOF. Indeed, assuming condition (iii), Lemma 4.8 states that condition (iv) is also true. We can thus apply Lemma 4.3. Therefore, Γ is an embedding. □

We can now prove Theorem 4.1 in full generality. We first triangulate G. More precisely, this means that edges are added to split the faces of G in triangles, without adding vertices (this is done in a purely combinatorial way: no geometry is involved here). Let G_1 be this planar augmented graph. Adding the same edges in the mapping Γ gives us a mapping Γ_1 of the graph G_1. We now check that we can apply Corollary 4.9 to G_1 and Γ_1. In fact, this boils down to checking condition (ii') to Γ_1. By condition (ii') and Lemma 4.7 applied to Γ, the neighbors of an interior vertex are not all on a line (under Γ), and such a vertex is in the interior of the convex hull of its neighbors. Because Γ_1 is obtained from Γ by adding extra edges, condition (ii') also holds for Γ_1. Thus, by Corollary 4.9, Γ_1 is an embedding. Deleting the edges we added earlier to Γ, we obtain that Γ is an embedding as well. It is clear that the faces are strictly convex.

4.2 Isotopies in the plane

Now, we detail the construction of the isotopy outlined in the introduction. Let $G = (V, E)$ be a 3-connected planar graph, and let Γ_0 and Γ_1 be two embeddings of G into the plane. We look for an isotopy between Γ_0 and Γ_1, restricting ourselves to the following situation: the boundary cycle C of the exterior face of Γ_0 is a convex polygon, it bounds also the exterior face of Γ_1, and the corresponding vertices of C are at the same location in Γ_0 and Γ_1. During the isotopy, the vertices of C have to remain at the same position. In addition, we will require the graph G to be triangulated. See Figure 4.5.

A natural idea arising to solve this problem is the following: try to deform Γ_0 into Γ_1 by keeping the exterior vertices at the same place and moving the

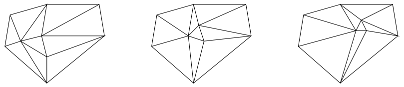


Figure 4.5: An isotopy Γ_t ($t \in [0,1]$) in our framework: here Γ_0, $\Gamma_{1/2}$ and Γ_1 are depicted.

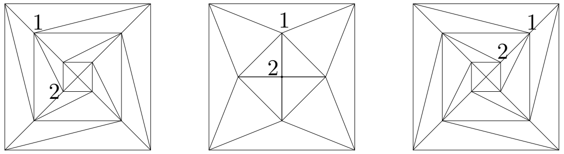


Figure 4.6: An example showing that the naive approach does not work. The figure shows Γ_0 (left) and Γ_1 (right). The two inner squares are "twisted" to the left (resp. right) under Γ_0 (resp. Γ_1), and the innermost square must rotate by an angle of π in the whole motion. With the linear motion, the vertices of the inner square would collapse at $t = 1/2$, as shown in the picture in the middle. Therefore, this motion does not yield an isotopy.

interior vertices linearly. That is, $\Gamma_t(v) = (1-t)\Gamma_0(v) + t\Gamma_1(v)$ for an interior vertex v and t in $[0,1]$. It turns out that this approach does not always yield an isotopy, as Figure 4.6 demonstrates. Bing and Starbird [15], generalizing a result by Cairns [26], showed the existence of an isotopy in the context described above; if the cells are strictly convex, one can ensure that they remain strictly convex during the deformation [147]. A series of more mathematical papers study the topological space of embeddings of a given triangulation (with boundary fixed), also called the set of homeomorphisms of a (2D) simplicial complex K that are affine linear on each simplex of K and are the identity on the boundary of K: in [18], it is proved that (if the outer boundary is convex) it is homeomorphic to $\mathbf{R}^{2k}$ where k is the number of interior vertices. See also the references in that paper for further reading on this topic.

However, these papers do not provide an algorithmic solution to this problem. As explained in the introduction, Gotsman et al. [73, 81] gave a method, based on Tutte's theorem, to solve this isotopy problem, representing a vertex as barycenter of its neighbors. We will use the following definitions in order to study the case where the barycentric coefficients are symmetric. Let E_i be the set of (undirected) interior edges (the edges for which at least one incident vertex is in V_i). A *weight function* on Γ, or *stress*, is a map $\omega : E_i \to \mathbf{R}$; hence $\omega_{uv} = \omega_{vu}$. ω is *positive* if $\omega_{uv} > 0$ for each interior edge uv. If ω and the positions of each $v \in V_e$ are fixed, the *equilibrium state* is defined by the system: for each $u \in V_i$, $\sum_{v|uv \in E} \omega_{uv}(\overline{u} - \overline{v}) = 0$. In these conditions, ω is an *equilibrium stress* for Γ.

Here is a summary of our approach: compute equilibrium stresses ω^0 (resp. ω^1) of embeddings Γ_0 (resp. Γ_1); then, for $t \in [0,1]$, compute the equilibrium state of $\omega^t = (1-t)\omega^0 + t\omega^1$. The difficulty resides in computing an equilibrium stress for a given embedding Γ: our method relies on Maxwell–Cremona's correspondence,

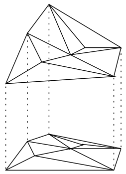


Figure 4.7: A lift of an embedding.

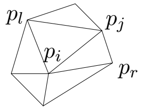


Figure 4.8: The notations for the computation of ω_{ij}.

a theorem well-known in rigidity theory (see Hopcroft and Kahn [96] for details on this theorem, and [85] for a general introduction to rigidity theory). Think of Γ as being in the plane $z = 0$ of $\mathbf{R}^3$. Take any *lift* of Γ, by adding to each vertex $\bar{v} = p_v = (x_v, y_v, 0)$ of Γ a third coordinate, leading to $q_v = (x_v, y_v, z_v)$. Consider the polyhedral terrain whose vertices are the q_i's and which has the same incidence structure as Γ (Figure 4.7). Now, let ij be an interior edge of Γ; let l and r be the left and right neighbor of the (oriented) edge ij (Figure 4.8) and φ^L_{ij} (resp. φ^R_{ij}) the affine form which takes the value z_i, z_j, z_l (resp. z_r) at points p_i, p_j, p_l (resp. p_r). We will define an equilibrium stress for Γ determined by this lift.

If $a_0, \ldots, a_k$ are $k+1$ points of $\mathbf{R}^k$, written as column vectors, we introduce the multi-affine bracket operator $[a_0, \ldots, a_k]$, defined by

$$[a_0, \ldots, a_k] = \begin{vmatrix} a_0 & a_1 & \ldots & a_k \\ 1 & 1 & \ldots & 1 \end{vmatrix}$$

(this quantity being proportional to the signed volume of the convex hull of the a_i's).

Lemma 4.10 *For each interior edge ij and any $p \in \mathbf{R}^2$,*

$$\varphi^L_{ij}(p) - \varphi^R_{ij}(p) = \frac{[p_i, p_j, p]}{[p_i, p_j, p_l]}(\varphi^L_{ij}(p_l) - \varphi^R_{ij}(p_l)).$$

PROOF. It is a consequence of Cramer's formula. Let φ be an affine form on $\mathbf{R}^k$ and $a_0, \ldots, a_k$ be $k+1$ affinely independent points, $a \in \mathbf{R}^k$. Let $\alpha_0, \ldots, \alpha_k$ be the barycentric coordinates of a with respect to the a_i's, that is, by definition:

$$\begin{array}{ccccccc} \alpha_0 a_0 & + & \ldots & + & \alpha_k a_k & = a \\ \alpha_0 & + & \ldots & + & \alpha_k & = 1. \end{array}$$

Cramer's formula now implies:

$$\alpha_i = \frac{[a_0, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_k]}{[a_0, \ldots, a_k]}.$$

So (if $k = 2$, and because φ is an affine form):

$$\varphi(a) = \frac{[a, a_1, a_2]}{[a_0, a_1, a_2]}\varphi(a_0) + \frac{[a_0, a, a_2]}{[a_0, a_1, a_2]}\varphi(a_1) + \frac{[a_0, a_1, a]}{[a_0, a_1, a_2]}\varphi(a_2).$$

It is now easy to conclude. □

Define, for any interior edge ij and for a point p not on the line $(p_i p_j)$:

$$\omega_{ij} = \frac{\varphi^L_{ij}(p) - \varphi^R_{ij}(p)}{[p_i, p_j, p]}.$$

This definition does not depend on the point p, by Lemma 4.10. Furthermore, $\omega_{ij} = \omega_{ji}$. In practice, there is an intrinsic formula (recall that the q_i's are the lifts of the points p_i's, which are the images of the vertices under Γ):

Lemma 4.11 $\omega_{ij} = \dfrac{[q_i, q_j, q_l, q_r]}{[p_i, p_j, p_l][p_i, p_j, p_r]}$.

PROOF. By definition of ω_{ij}:

$$\omega_{ij}[p_i, p_j, p_l][p_i, p_j, p_r] = (z_l - \varphi^R_{ij}(p_l))[p_i, p_j, p_r]. \tag{4.4}$$

By Cramer's formula, as in the proof of Lemma 4.10:

$$\varphi^R_{ij}(p_l)[p_i, p_j, p_r] = z_i[p_l, p_j, p_r] + z_j[p_i, p_l, p_r] + z_r[p_i, p_j, p_l].$$

Thus the left member of Equation (4.4) equals

$$z_l[p_i, p_j, p_r] - z_i[p_l, p_j, p_r] - z_j[p_i, p_l, p_r] - z_r[p_i, p_j, p_l],$$

which equals $[q_i, q_j, q_l, q_r]$ (by developping this determinant with respect to the third line). □

Theorem 4.12 *ω is an equilibrium stress for Γ.*

PROOF. For any point p in the plane, $i \in V_i$, we have:

$$\sum_{j|ij\in E} \omega_{ij}[p_i, p_j, p] = \sum_{j|ij\in E} (\varphi^L_{ij}(p) - \varphi^R_{ij}(p)) = 0,$$

because the affine form φ corresponding to a face incident to p_i appears twice in this sum, once counted positively, once negatively. As $[p_i, p_j, p] = \det(p_j - p_i, p - p_i)$, this implies

$$\det\Big(\sum_{j|ij\in E} \omega_{ij}(p_i - p_j), p - p_i\Big) = 0,$$

for each point p in $\mathbf{R}^2$. Therefore

$$\sum_{j|ij\in E} \omega_{ij}(p_i - p_j) = 0.$$

□

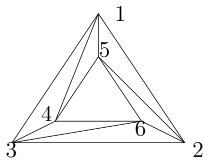


Figure 4.9: An embedding which is not a regular subdivision. Indeed, assuming it is possible to lift it to a lower convex hull, we can suppose, by adding a suitable affine form to all the z_i's, that $z_4 = z_5 = z_6 = 0$. If this graph were a regular subdivision, we would have $z_1 > z_2 > z_3 > z_1$, which is impossible.

Thus, each lift of the embedding Γ determines an equilibrium stress on Γ. Conversely, it is possible to show that an equilibrium stress determines a unique lift of Γ, up to the choice of an affine form of $\mathbf{R}^2$ (Maxwell's theorem, shown for example in [96] in a slightly different context).

If we have *positive* equilibrium stresses ω^0 and ω^1 of Γ_0 and Γ_1 respectively, we have a method to compute an isotopy between Γ_0 and Γ_1: by Tutte's theorem, because $\omega^t = (1-t)\omega^0 + t\omega^1$ is a positive stress for each $t \in [0, 1]$, the corresponding mapping Γ_t is an embedding, and $(\Gamma_t)_{t\in[0,1]}$ is clearly continuous (the map which associates to each invertible matrix its inverse, is continuous), hence an isotopy. Furthermore, it is easy to characterize the set of embeddings which admit a positive equilibrium stress: an edge ij has a positive weight if and only if the line $q_i q_j$ (with the notations above) is under the line $q_l q_r$. Recall that a *regular triangulation* is a triangulation which is the projection of the lower faces of a polytope generated by a family of points, see [158]. Hence an embedding has a positive stress if and only if it is a regular triangulation. Therefore, we have:

Theorem 4.13 *If Γ_0 and Γ_1 are regular triangulations, then we can compute an isotopy between Γ_0 and Γ_1.*

Testing whether Γ is a regular subdivision, and, if so, computing a positive lift, can be done easily using linear programming; indeed, we have a convex lift for Γ if and only if, for each interior edge ij and with the notations above, $[q_i, q_j, q_l, q_r] < 0$, which is a linear inequality in the z_k's. Not all triangulations are regular subdivisions, as shown in Figure 4.9 (see [158, p. 132]), but a large class of embeddings are regular subdivisions, including Delaunay triangulations for example (because the Delaunay triangulation of a set of points is the projection of the edges of the convex hull of the points lifted on the standard paraboloid, see [19, p. 437] or [60, p. 303]); this remark might be useful because of the wide use of these triangulations in computational geometry.

In practice, we tried to build an isotopy between a random triangulated embedding and the "canonical" embedding of the same graph (that is, the embedding obtained by Tutte's method when all weights equal 1). We lift Γ_0 to the standard paraboloid $z = x^2 + y^2$, compute the equilibrium stress ω_0, and use linear interpolation between ω^0 and the unit weights ω^1. Although the initial stress is not necessarily positive, it turns out that, in many (not too big) cases, this

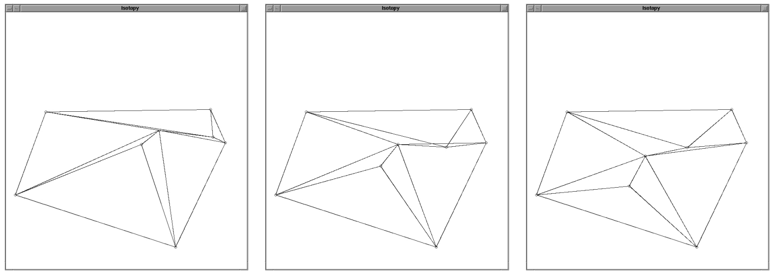


Figure 4.10: An example of non-planarity with the linear interpolation between the weights of a lift on the standard paraboloid, and unit weights.

method yields an isotopy; long experiments have been necessary to find a small counterexample like Figure 4.10. See Section 4.6 for numerical coordinates. Our smallest counterexample uses 4 outer vertices and 2 inner vertices, but the failure is very hard to see on the screen and can only be proved by computation. Lifting on the paraboloid may give an isotopy even if the considered triangulation is non-regular, like in Figure 4.6, but can also fail with regular triangulations (the initial and final triangulations in Figure 4.10 are regular). This method has been programmed in C++ using Numerical Recipes and the LEDA library, and also in Mathematica for exact computations.

Several other approaches could be done in the same spirit to try to find a method which would work for a larger class of embeddings than the regular subdivisions. One could attempt to study the space of stresses which yield an embedding (thus an isotopy corresponds to a path in this space). If we restrict ourselves to the linear interpolation between the weights, an important question is: are there two embeddings Γ_0 and Γ_1 so that, for any lifts of Γ_0 and Γ_1, the interpolation $\omega^t = (1-t)\omega^0 + t\omega^1$ of the corresponding weights does not yield an isotopy? If it is not the case, how to compute the lifts?

We have seen that using linear interpolation from the weights of a lift on the standard paraboloid to unit weights does not always yield an isotopy. Nevertheless, we have the following conjecture (checked during all our experiments): during this interpolation, the matrix involved in the computation of the positions of the points is symmetric positive definite.

If it is the case, it has the following interesting consequence. If ω is a stress on G, let us denote by M_ω the matrix involved in the inversion of System (S). It can be shown (see the proofs of Lemma 4.18) that M_ω is symmetric positive definite if ω is positive; moreover, $\omega \mapsto M_\omega$ is linear. If M_{ω^0} and M_{ω^1} are symmetric positive definite, so is $M_{(1-t)\omega^0+t\omega^1} = (1-t)M_{\omega^0} + tM_{\omega^1}$, and uniqueness of the positions of the vertices is guaranteed during the motion (which may fail to be an isotopy). Similarly, if M_{ω^0} is symmetric positive definite and ω^1 is a positive stress, since multiplying ω^1 by a positive number does not affect the equilibrium state, we can assume $\omega^1 \geq \omega^0$ (this notation simply means that for

each interior edge ij, $\omega^1_{ij} \geq \omega^0_{ij}$). Each nondecreasing family ω^t of stresses from ω^0 to ω^1 yields a family M_{ω^t} of symmetric positive definite matrices; indeed, $M_{\omega^t} = M_{\omega^0} + M_{\omega^t-\omega^0}$; the first matrix of the right term is symmetric positive definite, the second one is positive because the corresponding stress is non-negative on each interior edge. Thus, if this conjecture is true, the positions of the vertices are uniquely determined for many choices of the interpolation between the weights.

4.3 Generalization to 3D space

We explain here why the analogue of Tutte's theorem is false in 3D space, thus making it difficult to build isotopies in 3D. Here, it is convenient to use combinatorial simplicial complexes (all simplicial complexes considered here are combinatorial, not geometric; see for example [153]).

We introduce some other definitions, generalizing those in 2D. A *mapping* f from a simplicial complex C into $\mathbf{R}^d$ is a map from all the simplexes of C into $\mathcal{P}(\mathbf{R}^d)$ satisfying: if $\{v_1,\ldots,v_p\}$ is a simplex of C,

$$f(\{v_1,\ldots,v_p\}) = \mathrm{Conv}\ \{f(v_1),\ldots,f(v_p)\}.$$

An *embedding* of C into $\mathbf{R}^d$ is a mapping so that, for any two simplexes $\sigma,\tau \in C$, $f(\sigma\cap\tau) = f(\sigma)\cap f(\tau)$. As usual, an *isotopy* $(h(t))$ $(t\in[0,1])$ of C into $\mathbf{R}^d$ is a continuous family of embeddings of C into $\mathbf{R}^d$. Finally, the *image* of a simplicial complex C by a mapping f is the union of the sets $f(\tau)$, over all simplexes τ of C.

In this section, we will often manipulate complexes whose embeddings have to be fixed on the "boundary" of these complexes. A *3-complex with tetrahedral boundary* (C,B,b) is a simplicial 3-complex C with a subcomplex $B\subset C$ so that B is simplicially equivalent to the boundary of a 3-simplex, together with an embedding b of B into $\mathbf{R}^3$. An *embedding* f of (C,B,b) into $\mathbf{R}^3$ is an embedding of C so that $f|_B = b$ and the image of f is exactly the tetrahedron bounded by the image of b. An *isotopy* of a 3-complex with tetrahedral boundary is a continuous family of embeddings.

The goal of this section is to show:

Theorem 4.14 *There exist a complex with tetrahedral boundary (C,B,b), and two mappings f and j of (C,B,b) into $\mathbf{R}^3$, such that:*

1. *f is an embedding,*
2. *$j|_B = f|_B$,*
3. *each vertex in $C\setminus B$ is, under j, barycenter with positive coefficients of its neighbors,*
4. *but j is not an embedding.*

This theorem is a counterexample to the analogue of Tutte's theorem in three dimensions: the first condition is the analogue of planarity, the second condition fixes the images of the exterior vertices by j and the third one is the condition for the interior vertices.

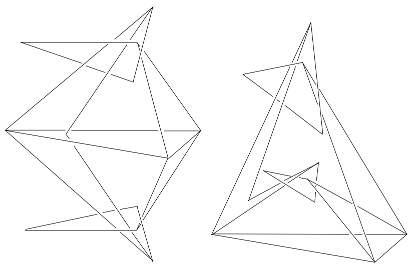


Figure 4.11: Starbird's embeddings f_1 and g_1 of C_1.

The cornerstone for the proof of Theorem 4.14 is the description by Starbird [141] of a graph C_1, embedded into $\mathbf{R}^3$ in two different ways f_1 and g_1, so that it is impossible to deform one embedding to the other without bending the edges. Yet, if bending the edges is allowed, such a deformation becomes possible. These embeddings are depicted in Figure 4.11, copied from his paper. We found coordinates for the vertices of these embeddings, available in Section 4.7. In the lemma below, we rephrase the properties stated by Starbird.

Lemma 4.15 *The following holds:*

1. *There are a 3-complex with tetrahedral boundary (C, B, b), so that C contains C_1, and two embeddings f and g of (C, B, b) extending respectively f_1 and g_1.*
2. *If C, f and g satisfy the preceding condition, there is no isotopy of (C, B, b) taking f to g.*

The first part of Lemma 4.15 expresses the fact that f and g are combinatorially equivalent triangulations (tetrahedralizations for purists) of a tetrahedron, with the same boundary. Despite this, as stated in the second part, there is no isotopy from f to g. It is to be noted that the analogue of this lemma is false in 2D by Tutte's theorem.

The proof of the second part of this lemma is given in detail in Starbird's paper, we shall not explain the argument here. Shortly said, the author uses properties of piecewise linear curves embedded in 3D space to show that the embeddings f_1 and g_1 cannot be deformed from one to the other while keeping the edges of C_1 straight, for otherwise at some stage of the isotopy there would be a degeneracy which would prevent to have an embedding. Then, because f (resp. g) extends f_1 (resp. g_1), there cannot be any isotopy between those embeddings as well.

We will give a detailed summary of the proof of the first part of Lemma 4.15, because it is stated in Starbird's paper but not all details of the proof are supplied. The main idea for the proof is the following "fundamental extension lemma" enabling to extend an isotopy of a complex to an isotopy of a complex with tetrahedral boundary containing this complex. It is proved in [15, Theorem 3.3]; we rephrase it here for convenience in our framework (it holds in fact in arbitrary dimension):

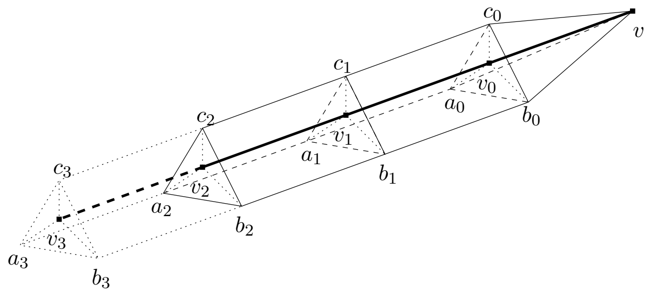


Figure 4.12: How an edge vw of C_1 (in bold) is protected by a skinny flexible tube. The vertices $v_0, \ldots, v_n$ are spread uniformly on the edge of C_1 which is considered, to make the edge flexible during the isotopy. An equilateral triangle $a_ib_ic_i$ is drawn around v_i, and the vertices of these triangles are linked as shown in the figure. Note the special treatment at the end of the edge (vertex v). The space between the triangles $a_ib_ic_i$ is also triangulated (not all edges are shown in the figure). Thus, a 3-dimensional simplicial complex protects each edge of C_1.

Lemma 4.16 *Let C be a simplicial 3-complex and $(h(t))$ be an isotopy of C into $\mathbf{R}^3$. Then there are a 3-complex with tetrahedral boundary $(\widetilde{C}, \widetilde{B}, \widetilde{b})$ so that $\widetilde{C}$ contains C and an isotopy $(\widetilde{h}(t))$ of $(\widetilde{C}, \widetilde{B}, \widetilde{b})$ into $\mathbf{R}^3$ extending $(h(t))$.*

We shall not give the proof here. The two key ingredients are that slightly perturbing an embedding still yields an embedding, and the use of refinements of triangulations in $\mathbf{R}^3$.

Proof of Lemma 4.15, first part. We first express the fact that it is possible to deform $f(C_1)$ to $g(C_1)$ if bending the edges is allowed: there is a refinement C_2 of C_1 (by adding vertices on the edges of C_1) and an isotopy $(h(t))$ of C_2 into $\mathbf{R}^3$ taking f_2 to g_2. Here, f_2 is to be understood in the following manner (and similarly for g_2): if v is a vertex in C_1, then $f_2(v) = f_1(v)$; and if an edge $e = vw$ of C_1 is subdivided with vertices $v_0, \ldots, v_n$ inserted on e, then $f_2(v_0), \ldots, f_2(v_n)$ are spread uniformly on $f_1(v)f_1(w)$. It is easy to see that this fact is true, as written in the paper, if you build a model of $f_2(C_2)$ with strings (or small bars) and deform it to $g_2(C_2)$.

No argument apart from the fact that such a deformation is possible is given in Starbird's paper to complete the proof. We thus suggest the following: In fact, we extend a bit more C_2 by protecting each edge of C_1 (split in C_2) by a 3-complex looking like a skinny tube (Figure 4.12). Define f_2 and g_2 naturally on these tubular protections; the images of f_2 and g_2 are just thickened versions of the images of f_1 and g_1. By Lemma 4.16, extend C_2 to a 3-complex with tetrahedral boundary (C_3, B_3, b_3), extending the isotopy $(h(t))$ to an isotopy $(\widetilde{h}(t))$ of (C_3, B_3, b_3). Now, considering $\widetilde{h}(0)$ and $\widetilde{h}(1)$, the complex (C_3, B_3, b_3) nearly satisfies the conditions required in the first part of Lemma 4.15, except that C_3 does not contain exactly C_1 because the edges of C_1 have been subdivided.

Thus, in f_3 and g_3, the only thing we have to do is to retriangulate compatibly the tubular protections of each (split) edge vw of C_1, removing the vertices $v_0, \ldots, v_n$ splitting this edge and restoring the initial edge vw. Since the tubular protections of vw look alike under f_3 and g_3 (the v_i's are on a line, and similarly for the a_i's, b_i's and c_i's), this retriangulation is easy: the compatibility will be automatically satisfied. See [14, pp. 4–6] for similar retriangulation problems: first retriangulate the 2D region which is the convex hull of v, w, and the a_i's by removing the v_i's and linking each of the a_i's to v. Do the same with the b_i's and the c_i's. Now, we have to retriangulate three thirds of the tubular protection of edge vw. To retriangulate the region which is the convex hull of v, w, the a_i's and the b_i's, simply insert a new vertex p in the interior of this region; since its boundary is still triangulated, it is sufficient to insert in the complex the simplexes which are on the boundary of this region with p adjoined ("coning" the boundary of this region from p). Do the same for the other thirds. The resulting complex (C, B, b) and embeddings f and g satisfy the hypotheses. $\square$

Proof of Theorem 4.14. First notice that, under f and g, all interior vertices are barycenter with positive coefficients of their adjacent vertices. For otherwise a vertex i would be on a face of the polytope generated by the neighbors of i, hence i would have no neighbor on a half-space whose boundary passes through the image of i; this contradicts the fact that i is a vertex interior to the triangulation. Let i be an interior vertex, and let λ_{ij}^f (resp. λ_{ij}^g) be the barycentric coefficients of i with respect to its neighbors j in the embedding f (resp. g). Note that the coefficients may be non-symmetric: we follow the approach of [73] to ensure we have positive coefficients. Then, for $t \in [0,1]$, consider $\lambda_{ij}^t = (1-t)\lambda_{ij}^f + t\lambda_{ij}^g > 0$. Fix the positions p_i of the vertices $i \in B$, and look for the positions of the other vertices i satisfying the equations: $\sum_{j|ij\in E} \lambda_{ij}^t (p_j - p_i) = 0$, where E is the set of edges of C. This system admits a unique solution for each $t \in [0,1]$ (exactly the same proof holds as in Section 4.5). Let us call the resulting family of mappings $(\bar{h}(t))$. By Lemma 4.15, second part, $(\bar{h}(t))$ cannot be an isotopy: there is a $t_0 \in [0,1]$ such that $\bar{h}(t_0) = j$ is not an embedding. (C, B, b), f, and j satisfy the conditions of Theorem 4.14. $\square$

This theorem is a counterexample to the generalization of Tutte's theorem in 3D, described in introduction. In fact, the result is slightly stronger: j is not an embedding, but even the restriction of j to the 1-skeleton of C is not an embedding (two edges must cross). This also implies that constructing isotopies of complexes in 3D is much more difficult than in 2D. Starbird [142, 143] showed the following theorem which might be a clue to find a solution: if there are two embeddings f and g of a complex K with tetrahedral boundary into $\mathbf{R}^3$ (or more generally if the boundary is a convex polyhedron), then there might be no isotopy from f to g, but there is always a suitable refinement K' of the complex K for which there is an isotopy between f and g. The problem is now to realize algorithmically the refinement and the isotopy; unfortunately, it is unclear how to proceed. Another track would be to try to find more restrictive conditions under which a barycentric method would work; for example, if some subcomplexes are forbidden, or if the

complex is sufficiently refined, does Tutte's barycentric method always yield an embedding?

Recently, another counter-example for the analogue of Tutte's theorem in 3D has been described by Ó Dúnlaing [122].

4.4 Appendix: The strict convex hull

Recall that the strict convex hull of a set of points is the interior, in the space affinely generated by this set of points, of the convex hull of these points. The following lemma shows that conditions (ii) and (ii') are equivalent.

Lemma 4.17 *Let $A = \{a_1, \ldots, a_n\} \subset \mathbf{R}^d$. Then the strict convex hull of A is the set of barycenters with positive coefficients of the points in A.*

Proof. Suppose $p \in$ Str Conv A. Take $k \in \{1, \ldots, n\}$. There is an $\varepsilon_k > 0$ so that $p + \varepsilon_k(p - a_k) \in$ Conv A. Therefore it is possible to write $p = \sum_{i=1}^n \mu_i^k a_i$, where $\sum_{i=1}^n \mu_i^k = 1$, $\mu_i^k \geq 0$ and $\mu_k^k > 0$. Taking $\lambda_i = \frac{1}{n}\sum_{k=1}^n \mu_i^k$ yields that p is a barycenter with positive coefficients of the points in A.

For the opposite inclusion, suppose $p = \sum_{i=1}^n \lambda_i a_i$, with $\lambda_i > 0$ and $\sum_{i=1}^n \lambda_i = 1$. If $|\mu_1|, \ldots, |\mu_n|$ are sufficiently small, $p + \sum_{i=1}^n \mu_i(a_i - p)$ is in Conv A. This shows that $p \in$ Str Conv A. $\square$

4.5 Appendix: Invertibility of System (S)

Lemma 4.18 *If the coefficients λ_{ij} are positive, System (S) admits a unique solution.*

Before showing this lemma, we must explicitely compute the entries of the matrix involved in System (S). For convenience, note $v_1, \ldots, v_m$ the interior vertices and $v_{m+1}, \ldots, v_n$ the exterior ones. The matrix involved in System (S) is square, of size m, and defined, if $1 \leq i, j \leq m$ and with the convention $\lambda_{ij} = 0$ if ij is not an edge, by:

$$m_{ij} = -\lambda_{ij}, \text{ if } i \neq j;$$
$$m_{ii} = \sum_{k=1}^{n} \lambda_{ik}.$$

Several proofs of this lemma exist in the literature. We first give the most straightforward proof in the general case. It uses the well-known "diagonal dominant property" of matrices and can be found in [72, p. 237].

Proof. We show that the kernel of M is $\{0\}$. If $M \cdot y = 0$ for a column vector y with m entries, then: for each $i \in \{1, \ldots, m\}$, $\sum_{j=1}^n \lambda_{ij}(y_i - y_j) = 0$, where $y_j = 0$ if $j > m$ by definition. Consider an index i such that $|y_i|$ is maximal. As λ is positive, the preceding equation yields $y_j = y_i$ for every j neighbor of i. Because G is connected, and because $y_j = 0$ if $j > m$, we get $y_i = 0$. Therefore, M is invertible. (In fact, the same argument shows that M is symmetric definite positive, for it cannot have a nonpositive eigenvalue.) $\square$

We now prove Lemma 4.18 in the special case where the coefficients are symmetric, using the physical interpretation with the springs. E_i denotes the set of interior edges.

PROOF. The energy of the system made of the springs is defined by

$$\mathcal{E} = \frac{1}{2} \sum_{ij \in E_i} \lambda_{ij} |p_j - p_i|^2.$$

Consider that the positions of the exterior vertices are fixed; $\mathcal{E}(p_1, \dots, p_m)$ is a polynomial function of degree two. If at least one interior vertex p_i goes to infinity, $\mathcal{E}$ tends to $+\infty$ by connectivity of G and positivity of the coefficients. Thus, the homogeneous polynomial of degree two in the coordinates $p_1, \dots, p_m$ of $\mathcal{E}$ is a quadratic form which is symmetric definite positive. But the matrix of this quadratic form is exactly the matrix M, as it can be checked easily using the fact that the coefficients are symmetric. Thus M is symmetric definite positive and (S) admits a unique solution. □

Finally, we indicate that Lemma 4.18 is a consequence of the *matrix tree theorem* (see Brualdi and Ryser [23, p. 324], Chaiken [29], Orlin [123] or Zeilberger [157]), a theorem interpreting combinatorially the determinant of certain matrices in terms of arborescences of graphs.

PROOF. Let $(n_{ij})_{1 \leq i \neq j \leq m+1}$ be real numbers. Consider the complete directed graph (without loops) $\bar{G}$ with $m+1$ vertices, each edge (ij) having, by definition, weight n_{ij}. Let P be the square matrix of size $m+1$ defined by:

$$p_{ij} = -n_{ij}, \text{ if } i \neq j;$$

$$p_{ii} = \sum_{k=1}^{m+1} n_{ik}.$$

The matrix P is called the *Laplacian matrix* of $\bar{G}$. A *spanning arborescence* of $\bar{G}$ rooted at i is a subgraph of $\bar{G}$ covering all vertices of $\bar{G}$ so that it has no directed cycle and all vertices $j \neq i$ have, in $\bar{G}$, outdegree equal to one. The *matrix tree theorem* asserts that the cofactor of the ith diagonal element of matrix P is exactly the sum, over all spanning arborescences of $\bar{G}$ rooted at i, of the product of the weights of the edges of this arborescence.

Apply this theorem to our particular case: let $n_{ij} = \lambda_{ij}$ if $1 \leq i \neq j \leq m$; if $i \leq m$, let $n_{i,m+1} = \sum_{k=m+1}^{n} \lambda_{ik}$ and $n_{m+1,i} = 0$. The $(m+1)$th cofactor of P is exactly the determinant of the matrix M and also equals the sum, over all spanning arborescences of $\bar{G}$ rooted at vertex $m+1$, of the product of the weights of the edges of this arborescence. There is at least one spanning arborescence yielding a nonzero contribution to this sum: to see this, take a spanning tree of the graph induced by the inner vertices of G, and add one directed edge from a vertex in G which, in G, is linked to an exterior vertex, to vertex $m+1$. Since the weights of the edges are nonnegative, the contribution of any spanning arborescence is nonnegative, hence the cofactor is positive and M is invertible. □

4.6 Appendix: Counter-examples

We present here the data sets of embeddings which present a failure of the method presented in Section 4.2 (by lifting the embedding on the standard paraboloid to compute the initial weights, and then using linear interpolation between these weights and the unit weights).

The data format is as follows: each line corresponds to a vertex of the embedding, and contains, in this order, the vertex number, its x- and y-coordinates, and the list of its neighbors.

4.6.1 The smallest counter-example found

In this counter-example, the situation is close to a degeneracy, but one can check by numerical computation that this mapping is indeed an embedding, and that this does not yield an isotopy. It is made of four exterior vertices and two interior vertices.

```
1 -500 900 2 5 6 4
2 -850 900 1 3 5
3 -950 -900 4 6 5 2
4 0 -400 1 6 3
5 -900 -699 6 1 2 3
6 -800 -300 1 5 3 4
```

4.6.2 Counter-example presented in Figure 4.10

```
1 -681.67 314.31 5 2 8 6
2 -938.19 -391.67 7 8 1 3
3 419.75 -833.89 4 8 7 2
4 841.39 52.42 5 6 8 3
5 712.91 332.73 1 6 4
6 733.43 99.34 5 1 8 4
7 128.62 38.94 8 2 3
8 277.47 156.82 1 2 7 3 4 6
```

4.7 Appendix: Coordinates for Starbird's embeddings

We present here two data sets in OOGL format (to be viewed for example with Geomview[1]), which are Starbird's embeddings presented in Figure 4.11. The format of the main part of each data set is as follows: each line denotes a vertex, with its x-, y- and z-coordinates. Each pair of lines denotes an edge.

[1] http://www.geomview.org

First embedding:

```
VECT

17 34 17
2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1
####################
# Center part

0   -20 0   # 7
-4  20  0   # 5

-4  20  0   # 5
4   20  0   # 6

4   20  0   # 6
0   -20 0   # 7

# Upper part

4   20  0   # 6
-6  0   16  # '4

-6  0   16  # '4
-16 -6  14  # '3

-16 -6  14  # '3
2   6   8   # '2
```

```
2   6   8   # '2
-10 6   20  # '1

-10 6   20  # '1
0   -20 0   # 7

-10  6  20  # '1
0    -12 0  # 8

-4  20  0   # 5
-6  0   16  # '4

# Lower part (symm.
# in Z of upper part)

4   20  0   # 6
-6  0   -16 # 4

-6  0   -16 # 4
-16 -6  -14 # 3

-16 -6  -14 # 3
2   6   -8  # 2

2   6   -8  # 2
-10 6   -20 # 1

-10 6   -20 # 1
0   -20 0   # 7
```

```
-10  6   -20# 1
0    -12 0  # 8

-4  20  0   # 5
-6  0   -16 # 4
####################
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
```


Second embedding:

```
VECT

17 34 17
2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1
####################
# Center part

0   -20 0   # 7
-4  20  0   # 5

-4  20  0   # 5
4   20  0   # 6

4   20  0   # 6
0   -20 0   # 7

# Upper part

4   20  0   # 6
-6  0   24  # '4

-6  0   24  # '4
-16 -6  21  # '3

-16 -6  21  # '3
2   6   12  # '2
```

```
2   6   12  # '2
-10 6   30  # '1

-10 6   30  # '1
0   -20 0   # 7

-10  6  30  # '1
0    -12 0  # 8

-4  20  0   # 5
-6  0   24  # '4

# Lower part (symm.
# in X, shrink in Z
# of upper part)

-4  20  0   # 6
6   0   8   # 4

6   0   8   # 4
16  -6  7   # 3

16  -6  7   # 3
-2  6   4   # 2

-2  6   4   # 2
10  6   10  # 1

10  6   10  # 1
0   -20 0   # 7
```

```
10   6   10 # 1
0    -12 0  # 8

4   20  0   # 5
6   0   8   # 4
####################
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
1 0 0 1
```

Chapter 5

Conforming Delaunay triangulations in 3D

This chapter contains the paper [38], written with D. Cohen-Steiner and M. Yvinec, with only very slight modifications.

Abstract. We describe an algorithm which, for any piecewise linear complex (PLC) in 3D, builds a Delaunay triangulation conforming to this PLC.

The algorithm has been implemented, and yields in practice a relatively small number of Steiner points due to the fact that it adapts to the local geometry of the PLC. It is, to our knowledge, the first practical algorithm devoted to this problem.

Introduction

In the following, the term *faces* denotes objects in 3D space which are either 0-dimensional faces called vertices, 1-dimensional faces called edges or 2-dimensional faces called 2-faces. The vertices are just points, the edges are straight line segments, and the 2-faces are polygonal regions possibly with holes and isolated edges or vertices included in their interior. A piecewise linear complex, called for short PLC, is a finite set $\mathcal{C}$ of faces such that:

- the boundary of any face of $\mathcal{C}$ is a union of faces of $\mathcal{C}$;
- the intersection of any two faces of $\mathcal{C}$ is either empty or a union of faces of $\mathcal{C}$.

A triangulation $\mathcal{T}$ is said to *conform* to a PLC $\mathcal{C}$ if any face of $\mathcal{C}$ is a union of faces of $\mathcal{T}$. In this chapter, we propose an algorithm which, given a PLC $\mathcal{C}$, finds a set of points $\mathcal{P}$ whose Delaunay triangulation conforms to $\mathcal{C}$. The set $\mathcal{P}$ includes the vertices of $\mathcal{C}$ and a certain number of additional points which are usually called Steiner points.

This question is motivated by problems in mesh generation and geometric modeling: in these fields, it is crucial to decompose the space into a set of simplices which conforms to a given PLC, with the additional restriction that the shape of the cells must satisfy certain properties. Delaunay triangulations present several features (see, *e.g.*, [19]) which can be exploited to solve this problem, and many mesh generation algorithms make use of this concept.

The problem of computing a conforming 2D Delaunay triangulation was solved by Saalfeld [133] and Edelsbrunner and Tan [64]. The algorithm by Edelsbrunner and Tan [64] guarantees an $O(n^3)$ bound on the number of generated Steiner vertices, if n is the size of the input. Most of the further works on the subject are based on the Delaunay refinement approach pioneered by Ruppert [132] and Chew [32]. Shewchuk [140] gave an algorithm in 3D which builds a conforming Delaunay triangulation under restrictive conditions on the angles of the PLC. Murphy, Mount, and Gable [119] found a solution which works under no restriction, but, as written in the conclusion of their paper, produces far too many points in practice. The main interest of their paper is to show the existence of a conforming Delaunay triangulation with a finite set of vertices for any 3D PLC.

Our algorithm uses the Delaunay refinement approach. Initially, the set $\mathcal{P}$ is the set of vertices of the complex $\mathcal{C}$. Points are then added to $\mathcal{P}$ until each edge and each face of the complex $\mathcal{C}$ is a union of simplices which are in the Delaunay triangulation of $\mathcal{P}$.

The main difficulty with such a strategy is to ensure termination. Indeed, it is known that sharp edges and corners may induce cascading additions of Steiner points. To avoid this effect, we first define a protected area around edges and vertices of the PLC with a special refinement process. Outside the protected area, the PLC can be refined using Ruppert's process and the interaction between refinements in both areas can be controlled. Murphy, Mount, and Gable use a similar approach, but do not take into account the local geometry of the complex: the existence of a pair of vertices or a vertex and a non-incident edge very close together implies a high number of output vertices in the neighborhood of all edges of the input complex. The main difference between our approach and their work lies in the definition of the protected area: in our case, this area adapts to the local geometry of the input PLC, implying the creation of fewer points in practice.

The algorithm is presented in Section 5.1 and proved to be correct in Section 5.2. In Section 5.3, we present the details of the construction of the initial protected area, skipped in Section 5.1. Section 5.4 presents some refinements to improve the running time of the algorithm and to lower the number of vertices in the output conforming triangulation. At last, we end with experimental results in Section 5.5.

5.1 The algorithm

After a few definitions, we describe the protected area (Subsections 5.1.2 and 5.1.3). We then define the refinement process used for this area (Subsections 5.1.4 and 5.1.5). Finally, we describe the main procedure and summarize the whole algorithm.

5.1.1 Definitions and notations

The *circumball* of a segment ab is the ball admitting the segment ab as diameter. The *circumball* of a triangle abc is the ball admitting the circumscribing circle of abc as great circle.

An edge (resp. a triangle) is said to have the *Gabriel property* if its circumball contains no point of $\mathcal{P}$ in its interior. A point in the interior of the circumball of an edge (resp. a triangle) is said to *encroach upon* this edge (resp. this triangle).

In the following, we note $\mathrm{bd}(B)$ the boundary of a ball B, $\mathrm{int}(B)$ the interior of B and $\mathrm{circum}(ab)$ (resp $\mathrm{circum}(abc)$) the circumball of the segment ab (resp. of the triangle abc).

5.1.2 Protecting balls

The *1-skeleton* Sk of the complex $\mathcal{C}$ is the union of the 0- and 1-dimensional faces of $\mathcal{C}$. The protected area is defined by means of a set $\mathcal{B}$ of closed balls, called protecting balls, satisfying the following requirements:

i. the union of the balls in $\mathcal{B}$ covers the 1-skeleton Sk of the complex $\mathcal{C}$;

ii. the balls are centered on points which are in Sk;

iii. if two balls intersect, their centers belong to the same edge of the complex $\mathcal{C}$;

iv. if a face of $\mathcal{C}$ intersects a ball, then it contains the center of this ball;

v. the intersection of any three balls in $\mathcal{B}$ is empty;

vi. any two balls are not tangent;

vii. the center of any ball is inside no other ball.

(i) and (iv) imply that any vertex in $\mathcal{C}$ is the center of a ball in $\mathcal{B}$. We show in Section 5.3 how to build a set of balls satisfying these requirements. Furthermore, in Section 5.4, we show that there is in fact no need to cover all the edges.

5.1.3 Center-points, h-points, p-points, and *SOS*-points

We describe here a few subsets of points, included in the balls of $\mathcal{B}$, that we need to add first in the set $\mathcal{P}$. See Figure 5.1.

Let B be a ball in $\mathcal{B}$ with center o. Let $\mathcal{B}_B$ be the set of balls in $\mathcal{B}$ that intersect B. By condition (v), the intersections of B with the elements of $\mathcal{B}_B$ are disjoint.

We first add the center o of B. Such a point will be called a *center-point*. Then, for each element B_i of $\mathcal{B}_B$, consider the radical plane of B and B_i. It intersects the line joining the centers of B and B_i at a point h_i, which is on an

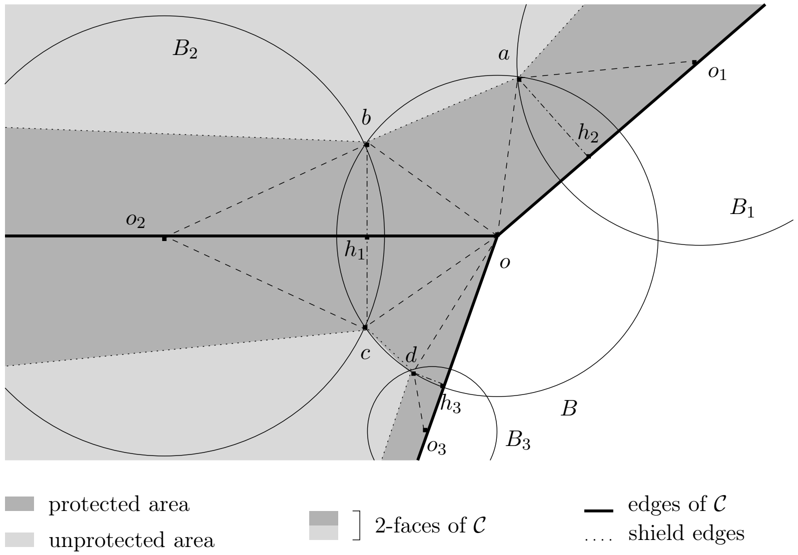


Figure 5.1: The situation in the neighborhood of a ball B, incident to three other balls B_1, B_2 and B_3. There are two faces in the complex, limited by three edges, in the plane of the figure. Point h_i is added on the radical plane of B and B_i. p-points $a, b, c,$ and d belong to the boundary of two balls and to a face, they are therefore also inserted in $\mathcal{P}$. Incident to o are four right-angled triangles (*e.g.*, oh_2a) and two isosceles triangles (*e.g.*, oab). The shield edges are ab and cd.

edge of $\mathcal{C}$ by condition (iii). The point h_i is added to the set $\mathcal{P}$. Such points will be called *h-points*.

By condition (iv), any face of $\mathcal{C}$ which intersects $B \cap B_i$ contains the centers of B and B_i, and thus can be either the edge including the segment oo_i (o_i is the center of B_i) or a 2-face incident this edge. For each 2-face F of $\mathcal{C}$ intersecting $B \cap B_i$, we add to $\mathcal{P}$ the intersection points of F with the circle $\text{bd}(B) \cap \text{bd}(B_i)$. We called those points *p-points*.

Consider the plane Q of a 2-face of $\mathcal{C}$ intersecting B (and thus containing o). The edges of $\mathcal{C}$ split the disk $Q \cap B$ into one or several sectors. We focus on sectors which are included in $\mathcal{C}$. The p-points further split these sectors in subsectors. We call *right-angled subsectors* the subsectors limited by an edge of $\mathcal{C}$ and a p-point and *isosceles subsectors* the subsectors limited by two p-points.

If some isosceles subsectors form an angle $\geq \pi/2$, we add some points on their bounding circular arcs to subdivide them in new subsectors forming an angle $< \pi/2$. For reasons that will be clear in Subsection 5.1.4, these points are called SOS-*points*. The new subsectors with angle $< \pi/2$ are still called *isosceles subsectors*.

Center-points and h-points are the only categories of points added in the interior of protecting balls. p-points and SOS-points lie on the boundaries of protecting balls. SOS-points belong to a single protecting ball while p-points belong to the intersection of two balls.

Isosceles subsectors are defined by the center o of a ball B and by two points a and b (either p-points or SOS-points) on $\text{bd}(B)$. Line segments such as ab, joining two points that define an isosceles subsector, are called *shield edges*. In the following, triangles defined by center-points and shield edges such as oab are referred to as *isosceles triangles*. Triangles spanned by a center-point, a h-point and a p-point on the boundary of some right-angled subsector are referred to as *right-angled triangles*.

Definition 5.1 *The* protected area *is the union of the isosceles and right-angled triangles. See the dark gray area in Figure 5.1. In particular, the protected area is included in the union of the protecting balls.*

Definition 5.2 *The* unprotected area *is the complex* $\mathcal{C}$*, minus the protected area.*

5.1.4 The "split-on-a-sphere" strategy

During the process, it will be necessary to split shield edges. Since we do not want to add more points inside the balls in $\mathcal{B}$, we use a special treatment to split such a shield edge, called the "split-on-a-sphere" strategy (SOS for short). See Figure 5.2.

Let ab be a shield edge to be split, in a ball B. We distinguish two cases: a and b are both SOS-points and belong to a single ball B, or at least one of these two points (for example a) is a p-point and belongs also to another ball B'.

If a and b belong only to B, let c be the midpoint of the shortest geodesic arc ab on $\text{bd}(B)$. To refine edge ab, we add c to $\mathcal{P}$ and replace the shield edge ab by two shield edges ac and cb.

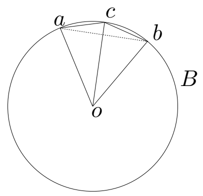


Figure 5.2: The SOS strategy: We split the shield edge ab by inserting the point c on the boundary of the ball.

If a is a p-point belonging to $\text{bd}(B) \cap \text{bd}(B')$, the idea is quite similar; however, if we do not take care, the SOS strategy could lead to cascading insertions of points, because refining an edge on B would lead to refinement of an edge on B', and so on. We thus use a strategy "à la Ruppert" [132], using circular shells. We consider the length of the segment ab, divided by two, and round it to the nearest distance d which is of the form $2^k, k \in \mathbf{Z}$ (the unit distance has been chosen arbitrarily at the beginning of the algorithm). Let c be the point of the shortest geodesic arc ab on $\text{bd}(B)$ at distance d from a. We split the shield edge ab using the point c.

In both cases, the added point c belongs to the category of SOS-points. Note that, due to the SOS refinement strategy, the protected and unprotected areas, still defined as in Subsection 5.1.3, will slightly evolve during the algorithm. Each SOS refinement increases the protected area and decreases the unprotected area.

5.1.5 The protection procedure

This procedure adds some points to set $\mathcal{P}$ to ensure that shield edges and isosceles triangles have the Gabriel property. It uses recursively the SOS strategy and works as follows: While there is an encroached shield edge ab or an encroached isosceles triangle oab, refine the edge ab using the SOS strategy.

5.1.6 The whole algorithm

Let us recall that the algorithm works by adding points to set $\mathcal{P}$. We note $Dt_3(\mathcal{P})$ the 3D Delaunay triangulation of points in $\mathcal{P}$. For each plane Q of a 2-face in $\mathcal{C}$, we note $Dt_2(\mathcal{P} \cap Q)$ the 2D Delaunay triangulation of points in $\mathcal{P} \cap Q$. These triangulations are updated upon each insertion of a point in $\mathcal{P}$.

The algorithm performs the initialization step and the main procedure described below.

The Initialization Step:

- Construct and initialize the protected area (as described in 5.1.2 and 5.1.3);
- execute the protection procedure.

We will see later that the Delaunay triangulation of $\mathcal{P}$ conforms to the part of $\mathcal{C}$ which is inside the protected area. Because the algorithm maintains the Gabriel property of shield edges, in each plane Q of a 2-face F of $\mathcal{C}$, the 2D triangulation $Dt_2(\mathcal{P} \cap Q)$ conforms to the shield edges in this plane and thus to the unprotected part F_u of F. The main procedure ensures that the triangles of $Dt_2(\mathcal{P} \cap Q)$ included in F_u appear in the 3D triangulation $Dt_3(\mathcal{P})$.

The Main Procedure:
The Main Procedure consists in executing the following loop: While there is a triangle T in the 2D Delaunay triangulation $Dt_2(\mathcal{P} \cap Q)$ of the plane Q of a 2-face F of $\mathcal{C}$ such that:

a. T is included in the unprotected part F_u of F,

b. T does not appear in $Dt_3(\mathcal{P})$,

refine T trying to insert its circumcenter c, that is:

- if c encroaches upon no shield edge, insert it;
- otherwise, split all the shield edges encroached upon by c using the SOS strategy, and then execute the protection procedure.

5.2 Proof of the algorithm

Two steps are involved for the proof of this algorithm. First, we prove invariants of the algorithm concerning the positions of the points added and the Gabriel property of some triangles and edges. After that, we are able to prove termination.

5.2.1 Properties maintained in the algorithm

Lemma 5.3 *At the beginning (and the end) of each execution of the main loop, the shield edges have the Gabriel property.*

PROOF. Indeed, this is true before the first execution of the main loop, because the protection procedure, which has just been executed, ensures this property; for the same reason, this also holds after an execution of the loop leading to the split of shield edges. At last, a circumcenter is inserted in $\mathcal{P}$ only if it does not violate this property. □

In the following, we define an *added* circumcenter to be a circumcenter inserted in the set $\mathcal{P}$, and a *rejected* circumcenter to be a circumcenter considered in the algorithm but not inserted because it encroaches upon some shield edge.

Lemma 5.4 *Any circumcenter (added or rejected) considered by the algorithm lies in the unprotected area, outside the protecting spheres. In particular, no point is added inside the protecting spheres after the initialization step, and $\mathcal{P}$ is included in $\mathcal{C}$.*

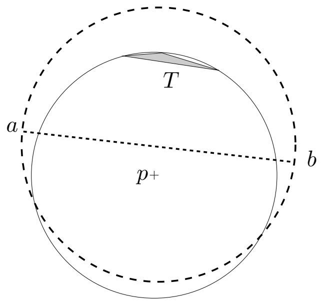


Figure 5.3: The circumcenter p of a triangle T lies in the unprotected area.

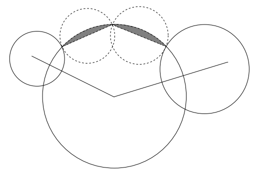


Figure 5.4: The intersection of the unprotected area with the union of protecting balls is included in the circumballs of shield edges.

PROOF. Let T be a triangle whose circumcenter is considered at some step of the algorithm. T lies in the unprotected area, and belongs to the 2D Delaunay triangulation $Dt_2(\mathcal{P} \cap Q)$ of the plane Q of some 2-face in $\mathcal{C}$. Let p be the circumcenter of T. Assume for contradiction that p lies outside the unprotected area. Let m be a point in T. Since shield edges enclose the connected component of the unprotected area which contains T, the segment pm must intersect a shield edge ab. The vertices a and b cannot be inside circum(T) because T belongs to $Dt_2(\mathcal{P} \cap Q)$. Hence (Figure 5.3), triangle T belongs to the circumball of ab, which is impossible by Lemma 5.3.

Moreover, since the circumballs of shield edges cover the intersection of the unprotected area with the protecting balls (see Figure 5.4), any added circumcenter is actually outside the protecting spheres. □

Proposition 5.5 *At the beginning (and the end) of each execution of the main loop, the isosceles triangles have the Gabriel property.*

PROOF. The proposition is obvious after the initialization step because the protection procedure is called and enforces the Gabriel property of isosceles triangles. For the same reason, it is also the case when a circumcenter has just been rejected because it encroaches upon some shield edge.

It remains to see that this proposition is still true when a circumcenter has just been inserted: such a circumcenter lies outside the protecting spheres (by

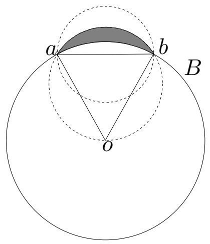


Figure 5.5: The balls B, circum(oab), and circum(ab).

Lemma 5.4) and outside the circumball of any shield edge (otherwise it is not inserted in $\mathcal{P}$). Let ab be such a shield edge, belonging to ball B. We note that the boundaries of B, circum(ab), and circum(oab) belong to a pencil of spheres. Because the angle $\widehat{aob}$ is smaller than $\pi/2$, we have circum(oab) $\subset$ circum(ab) $\cup B$ (Figure 5.5). The result follows. $\square$

Lemma 5.6 *Let B be a ball with center o, and p be a point on the boundary of B. If, at some stage of the algorithm, the segment op is encroached upon, the encroaching point is a h-point h_i on the radical plane of B and B_i, and p belongs to* $\text{bd}(B) \cap \text{int}(B_i)$.

PROOF. The circumball of op is inside B. Therefore, op can only be encroached upon by a vertex in this ball, and not by the center of B, hence only by a h-vertex in B. Suppose that op is encroached upon by a vertex h_i, belonging to B and B_i. The encroachment condition can be rewritten $\widehat{oh_ip} > \pi/2$. Because points q in $\text{bd}(B)$ that satisfy $\widehat{oh_iq} > \pi/2$ lie in $\text{bd}(B) \cap \text{int}(B_i)$, p belongs to $\text{int}(B_i)$. $\square$

Proposition 5.7 *At each stage of the algorithm, the right-angled triangles have the Gabriel property.*

PROOF. Suppose that a right-angled triangle oh_jp does not have the Gabriel property at some stage of the algorithm: h_j is on the radical plane between B and B_j, and p is on the boundary of B and B_j. Because the circumball of oh_jp is the circumball of op, by Lemma 5.6, the encroching point is a h-point, and p has to belong to the interior of a third ball B_i, which is impossible by condition (v). $\square$

Center points and h-points cut the edges of $\mathcal{C}$ in subedges. Note that Proposition 5.7 implies that these subedges are edges of $Dt_3(\mathcal{P})$.

5.2.2 Termination proof

Proposition 5.8 *The protection procedure always terminates.*

The proof is a straightforward consequence of the following lemma.

Lemma 5.9 *For each call to the protection procedure, there exists $\theta > 0$ such that no isosceles triangle with angle at the center of the ball less than θ will be split.*

PROOF. Let oab be an isosceles triangle with shield edge ab in a protecting ball B. We consider in turn three kinds of possible encroaching points: points on the boundary of B (case 1), points in the interior of B (case 2), and points outside B (case 3). In each case k, we prove the existence of a value θ_k, such that neither oab nor ab can be encroached upon by a point of type k if $\widehat{aob} < \theta_k$.

Recall that the three balls B, circum(ab) and circum(oab) belong to a pencil of spheres. Because the angle $\widehat{aob}$ is smaller than $\pi/2$, we have circum(oab) $\subset$ $B \cup$ circum(ab) and circum(ab) $\cap B \subset$ circum(oab) (see Figure 5.5). Therefore, it is enough to check that points on the boundary of B or outside B (cases 1 and 3) do not encroach upon ab and that points in B (case 2) do not encroach upon oab.

1. For a plane Q of a 2-face of $\mathcal{C}$ intersecting B, we consider the circle bounding $B \cap Q$ and we denote by $S(Q, B)$ the union of arcs on this circle spanned by the isosceles triangles in Q. Notice that all the SOS-points inserted on B are located on such a set $S(Q, B)$.

   If Q is the plane containing oab, no point of $S(Q, B)$ encroaches upon ab. If Q' is another plane, the distance between $S(Q, B)$ and $S(Q', B)$ is strictly positive, so there is a value $\theta_1(B, Q, Q')$ such that ab is not encroached upon by a point on $S(Q', B)$ if $\widehat{aob} < \theta_1$. Setting $\theta_1 = \min\{\theta_1(B, Q, Q')\}$ achieves the proof of case 1.

2. The only points in a ball B which can encroach upon an isosceles triangle oab in B are the h-points in B. Suppose that a point h_i (on the radical plane of B and B_i) encroaches upon oab.

   If h_i is in the plane Q of oab, we prove that encroachment is not possible. Indeed, if h_i encroaches upon oab, h_i encroaches either upon oa or upon ob. Thus a or b would belong to $\text{bd}(B) \cap \text{int}(B_i)$, by Lemma 5.6, which is impossible because a and b are either p-points or SOS-points.

   Let us now deal with the case where h_i does not belong to the plane Q. Let $c \in S(Q, B)$; c does not belong to B_i, for otherwise h_i would belong to Q. Let us prove that h_i is not in the closed ball circum(oc). If h_i is in the interior of circum(oc), this means that oc is encroached upon by h_i, hence, by Lemma 5.6, c belongs to $\text{int}(B_i)$, which is not the case. Similarly, if h_i is on the boundary of circum(oc), c belongs to B_i.

   Hence, the distance between h_i and the ball circum(oc) is strictly positive. Let $\delta(B, Q, h_i)$ be the minimum (strictly positive) of this distance for $c \in S(Q, B)$. Let $\delta'(B, \theta)$ be the Hausdorff distance between circum(oc) and circum($oa'b'$) where $oa'b'$ is an isosceles triangle with a' and b' on $\text{bd}(B)$, axis oc and $\widehat{a'ob'} = \theta$. As $\delta'(B, \theta)$ goes to 0 when θ goes to 0, there exists $\theta_2(B, Q, h_i)$ such that $\delta'(B, \theta) < \delta(B, Q, h_i)$ for any $\theta \leq \theta_2(B, Q, h_i)$. It follows that oab cannot be encroached upon by h_i if $\widehat{aob} < \theta_2(B, Q, h_i)$. Setting $\theta_2 = \min\{\theta_2(B, Q, h_i)\}$ achieves the proof of case 2.

3. Consider now the case where edge ab is encroached upon by a point p outside the ball B. At each call of the protection procedure, the set of points outside the protecting spheres is fixed. Also, the distance between two sets $S(Q_1, B_1)$ and $S(Q_2, B_2)$ which do not share a p-point is bounded from below. Thus, there is a value θ_3' such that, if $\widehat{aob} < \theta_3'$, edge ab cannot be encroached upon by p except if p belongs to $S(Q, B')$ where Q is the plane of oab and B' intersects B. Therefore, the only case remaining to be considered is the case where a is a p-point in $Q \cap \mathrm{bd}(B) \cap \mathrm{bd}(B')$ and ab is encroached upon by a point p of $S(Q, B')$. However, in this case, we split edges incident to a using circular shells. Hence, after a few splits, the edges incident to a will have the same lengths and will be unable to encroach upon each other. Therefore, we get a value $\theta_3 \leq \theta_3'$ satisfying the desired requirement.

□

Theorem 5.10 *The algorithm terminates, and, once it is the case, the Delaunay triangulation of $\mathcal{P}$ conforms to the complex $\mathcal{C}$.*

PROOF. It is sufficient to prove that the main procedure terminates: indeed, once it is the case, Propositions 5.5 and 5.7 show that the Delaunay triangulation of $\mathcal{P}$ conforms to the protected area of $\mathcal{C}$, and the fact that the algorithm ends precisely means that the Delaunay triangulation of $\mathcal{P}$ also conforms to the unprotected area of $\mathcal{C}$. We prove the termination of the main procedure by proving first that the number of added circumcenters is finite and second that the number of shield edges encroached upon by rejected circumcenters is finite. Because the protection procedure is already known to terminate, these two facts imply the termination of the main procedure.

By construction of the protecting spheres, the unprotected area is a disjoint union of plane regions. Let F_u be such a region. As previously noticed, owing to the SOS strategy, these unprotected regions slightly evolve during the algorithm; however, they are always shrinking. Consequently, the distance between F_u and the other regions as well as the distance between F_u and the set of center-points and h-points added in the interior of the protecting balls can be bounded from below by a constant δ_F. Let T be a triangle in F_u whose circumcenter has to be inserted in $\mathcal{P}$ and let C_T be the circumcircle of T. As T does not belong to $Dt_3(\mathcal{P})$, its circumball circum(T) contains a point in $\mathcal{P}$ which is not in the plane of F_u. Such a point can be inside a protecting ball (a center-point or a h-point), on the boundary of a protecting ball (and thus on the boundary of another region), or an added circumcenter (in another region by Lemma 5.4). Therefore circum(T) either contains a point added in the interior of a protecting sphere or intersects another unprotected region, and the radius of C_T is thus larger than δ_F. Because T belongs to the 2D Delaunay triangulation in the plane of F_u, C_T encloses no point of $\mathcal{P}$. The area of F_u being finite, this shows that the number of added circumcenters is bounded.

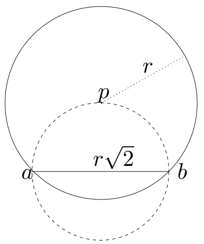


Figure 5.6: The shortest shield edge ab which may be encroached upon by a rejected circumcenter p.

Let us now show that the total number of edges encroached upon by rejected circumcenters is finite. For this purpose, consider a shield edge encroached upon by the center p of a circumcircle C in a region F_u. C being empty and of radius larger than δ_F, it is easy to show that the shield edge has length at least $\delta_F\sqrt{2}$ (see Figure 5.6). Thus the number of those edges is finite. □

5.3 Construction of the protecting balls

We have to build the set $\mathcal{B}$ of protecting balls satisfying the conditions described in Subsection 5.1.2. The efficiency of the algorithm really depends on this construction: the less balls there are, the less points will be produced in $\mathcal{P}$.

Definition 5.11 *Let $\mathcal{C}$ be a PLC. The* local feature size *of a point p with respect to $\mathcal{C}$ is the distance between p and the union of faces of $\mathcal{C}$ that do not contain p.*

Let $lfs(p)$ denote the local feature size of point p with respect to the PLC which is given as input of the algorithm. We address the following construction of the enclosing balls. Let α be a real, $0 < \alpha < \frac{1}{2}$ (typically $\alpha = 0.4$).

First, for each vertex v of the PLC, construct a ball of radius $\alpha \cdot lfs(v)$.

Then, on each edge e, do the following. While e is not completely covered by balls, consider a maximal open line segment a_1a_2 in e and outside the union of the balls in the current set $\mathcal{B}$. Point a_i $(i = 1, 2)$ is an intersection of ball B_i (with center o_i and radius r_i) with edge e. We will insert a ball between B_1 and B_2. Let o be the midpoint of a_1a_2. Insert a new ball B in $\mathcal{B}$, of center o and radius r, with:

$$r = \min\left\{\alpha \cdot lfs(o), oa_1 + \frac{r_1}{2}, oa_2 + \frac{r_2}{2}\right\}.$$

To ensure condition (vi), if $r = oa_1$, we replace r by $(1 - \varepsilon)r$ where ε is a small positive constant.

Lemma 5.12 *This construction terminates.*

Proof. Consider an edge e, whose vertices have just been protected by two spheres. Let A be the union of the (open) line segments which are in e minus the union of the current set of balls. Call A_0 the set A just after the protection of the endpoints of e. The distance $d = \min\{lfs(p) | \; p \in \overline{A_0}\}$ is strictly positive (the lfs function is continuous on $\overline{A_0}$, and lfs does not vanish on $\overline{A_0}$). The insertion of a new ball:

- either increases by one the number of connected components of A and decreases the measure of A by at least $2(1-\varepsilon)\cdot\alpha\cdot d$ (hence this case can happen only a finite number of times),
- or decreases by one the number of connected components of A (without increasing the measure of A).

The result follows. □

Conditions (i), (ii), (iv), (vi) and (vii) are obviously satisfied. (iii) follows from the fact that if two points o and o' do not belong to the same edge, oo' is larger than or equal to $lfs(o)$ and $lfs(o')$. If two balls B and B', centered at o and o' with radii r and r', are in $\mathcal{B}$, then $r < \frac{1}{2} lfs(o)$ and similarly for r'. Thus $r + r' < oo'$, hence the balls cannot intersect.

(v) is also true. Indeed, if three balls intersect, their centers must be vertices of a triangle in $\mathcal{C}$. But it follows from our construction that two balls centered on vertices of the PLC cannot intersect because $\alpha < \frac{1}{2}$.

Hence we have:

Proposition 5.13 *This construction of $\mathcal{B}$ is correct.*

5.4 Improvements

5.4.1 Speeding up the protection procedure

The following proposition shows that when the protection procedure is called from the main procedure, there is no need to check whether isosceles triangles have the Gabriel property.

Proposition 5.14 *After the initialization process, enforcing Gabriel property for shield edges in the protection procedure is enough to ensure Gabriel property for isosceles triangles.*

Proof. Upon termination of the initialization step, all isosceles triangles have the Gabriel property. Suppose that, at some stage of the algorithm, a point encroaches upon some isosceles triangle oab without encroaching ab. Let B be the ball containing oab. Since circum(oab) is included in the union of B and circum(ab) (Figure 5.5), the encroaching point must be inside B.

Hence it is sufficient to show that no isosceles triangle is encroached upon by a vertex inside its protecting ball during the algorithm. By contradiction, let $T = oab$ be the first isosceles triangle encroached upon by a vertex in B. Since no

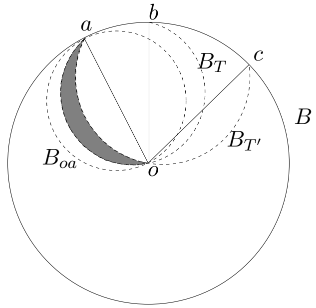


Figure 5.7: circum$(oab) \subset$ circum$(oac) \cup$ circum(oa).

point is inserted inside the balls during the main procedure, T must be a triangle which results from the splitting of some triangle $T' = oac$. The encroaching point can thus only be a h-point h_i lying inside B. Arguing that circum(oab), circum(oac), and circum(oa) belong to a sphere pencil and comparing their radii, we deduce (Figure 5.7) that circum$(oab) \subset$ circum$(oac) \cup$circum(oa). However, h_i does not belong to circum(oac) because $T' = oac$ was not encroached upon by h_i, nor to circum(oa) (by Lemma 5.6). Therefore h_i does not belong to circum(oab), which yields the contradiction. □

5.4.2 Restricting the area where balls are required

In 5.1.2, the set $\mathcal{B}$ is constructed so that the balls cover the whole 1-skeleton Sk of $\mathcal{C}$. We explain here that this is not always necessary. Indeed, the balls are introduced to avoid troubles with small angles; they are thus not required at places where faces intersect with an angle large enough. This remark enables to put less balls in $\mathcal{B}$, hence to reduce the size of the output $\mathcal{P}$. We first describe the modification in the construction of the balls, and then prove that, despite this slight modification, the algorithm is still correct.

Let $e = o_1o_2$ be an edge of the PLC so that all angles between faces incident to e are $\geq \pi/2$. We modify the algorithm in the following way. Still construct balls B_1 and B_2 centered at the vertices o_1 and o_2. In $\mathcal{P}$, insert o_1, o_2, and the two intersections p_1 and p_2 of e with the boundaries of B_1 and B_2.

Consider p_1p_2 as a shield edge in the main procedure. In other words, whenever this edge would be encroached upon by the insertion of a point in $\mathcal{P}$, split this edge in the middle, to keep it protected at each stage of the algorithm. The original edge of $\mathcal{C}$ is thus not in the protected area, but the process is exactly like in the standard algorithm.

There are only minor modifications for the proof of the algorithm. The unprotected area is still bounded with shield edges. The proof of termination of the protection procedure is analogous: Lemma 5.9 can be adapted without difficulty to show that there also exists a length $\delta > 0$ such that the protection procedure

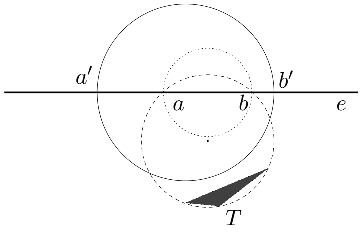


Figure 5.8: In the half-space H^+ (above the edge e in the figure), the part of circum(T) is included in the part of circum(ab) which is, in turn, contained in the part of circum($a'b'$).

never splits a shield edge which is a part of an edge and with length less than δ. The only difficulty is to show the following proposition.

Proposition 5.15 *The modified version of the main procedure always terminates.*

PROOF. Let F_u be a region, in a plane Q, incident to edge e. The distance between F_u and the regions non-incident to e as well as the distance between F_u and the set of center-points and h-points outside Q can be bounded from below by a constant $\delta_F > 0$. Let p be the circumcenter of a triangle T in F_u, added to $\mathcal{P}$. We will show that the circumball of T cannot contain a vertex of another face incident to e, which implies that the radius of this circumball is larger than δ_F, like in the proof of Theorem 5.10.

Suppose for contradiction that T is encroached upon by a point p' of $\mathcal{P}$ on a face incident to e. Necessarily, because the angles of the faces of $\mathcal{C}$ are obtuse at e, the circumball of T must intersect e. Let a and b be the intersection points of the boundary of circum(T) with e. Let $a'b'$ be the unique shield edge included in e which is intersected by circum(T). (The uniqueness follows from the fact that points in $\mathcal{P}$, like a' and b', cannot lie in circum(T).) Let H be the plane orthogonal to F_u and containing e, and H^+ be the half-space bounded by H and not containing T. Clearly, circum$(T) \cap H^+ \subseteq$ circum$(ab) \cap H^+ \subseteq$ circum$(a'b') \cap H^+$ (see Figure 5.8). The point p' is in circum$(T) \cap H^+$, hence in circum($a'b'$), which means that p' encroaches upon the shield edge $a'b'$ and yields the contradiction.

The remaining part of the proof of termination of the main procedure is exactly the same as in the proof of Theorem 5.10. $\square$

5.5 Experimental results

The algorithm has been implemented and tested using the Computational Geometry Algorithms Library CGAL[1]. Results for several models are displayed in Figures 5.9, 5.10, 5.11, 5.12, and 5.13.

[1] `http://www.cgal.org/`

	geological data	triceratops	umbrella
nb input vertices	7,566	2,832	16
nb non Delaunay faces	1,045	2,194	5
nb output vertices	25,793	27,947	122
running time (s)	83	570	0.7

Figure 5.9: Experimental data.

Figure 5.9 gives for each model, the number of vertices of the input PLC (*nb input vertices*), the number of 2-faces to which the Delaunay triangulation of input vertices does not conform (*non Delaunay faces*), and the number of vertices of the conforming output triangulation (*nb output vertices*). In those examples and in most cases, the number of vertices in the output conforming triangulation and the number of input vertices are in a ratio comprised between 3 to 1 and 10 to 1.

The running times, measured on a PC with 500Mhz processor, do not include the computations of local feature size values, because the current implementation uses a very slow brute force algorithm for it. We are currently designing a data structure devoted to speed up these computations.

Conclusion

We have presented an algorithm for computing a conforming Delaunay triangulation of any three-dimensional piecewise linear complex. The most important innovation, compared to the paper by Murphy et al. [119], is to enclose critical places by balls whose radii fit the local complexity of the complex, with the use of the local feature size. Our experimental results show that it is valuable in practice. The algorithm could be easily modified to guarantee in the resulting mesh the Gabriel property for any triangle included in a constraint. The next step currently under work is to investigate how conforming meshes with guarantees on the shape and size of the elements can be obtained. Several questions remain open: we did not try to find the time complexity of our algorithm. It would also be interesting, as in [64] in the plane, to find a bound on the output depending on the size of the initial complex and/or (like in [132]) the *lfs* function. Even from a theoretical point of view (see [61, p. 171]), no nice upper bound is known for the size of a conforming Delaunay triangulation in terms of the size of the input complex and the *lfs* function.

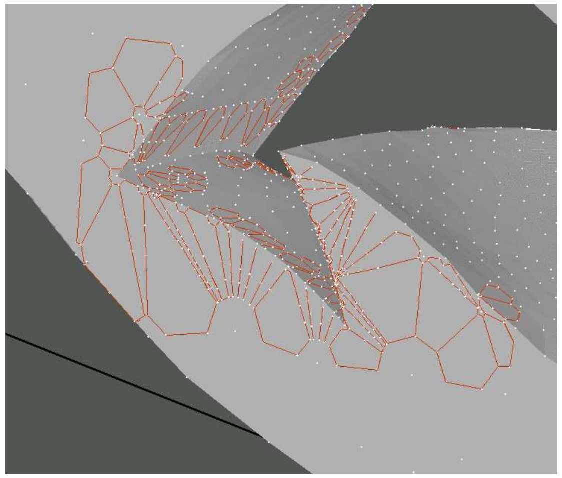


Figure 5.10: Detail of a geological formation (Courtesy of T-surf and Mr. Reinsdorff). Solid line segments stand for shield edges.

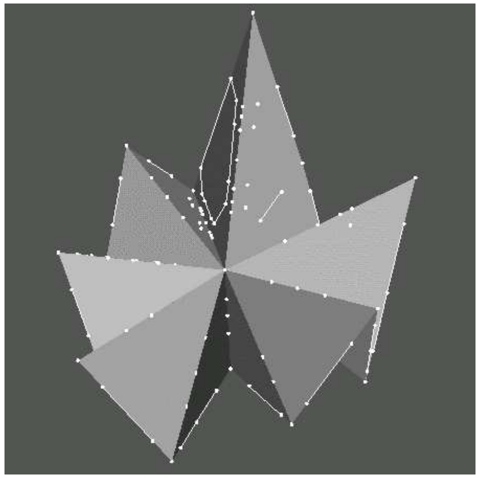


Figure 5.11: Umbrella. Solid line segments stand for shield edges.

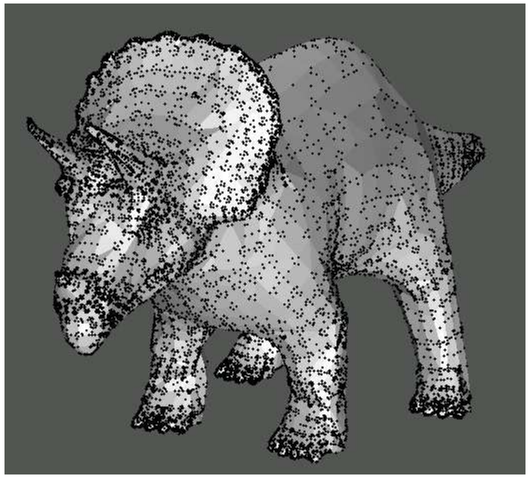


Figure 5.12: Triceratops.

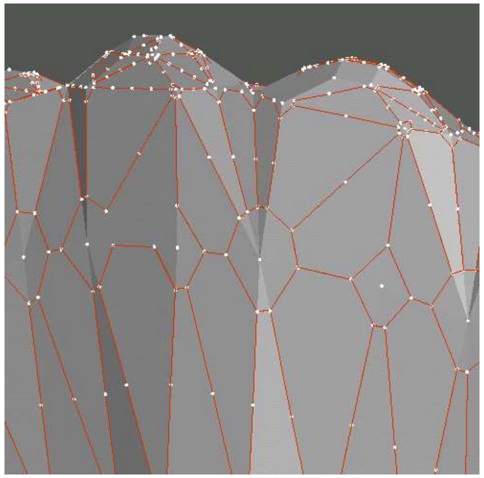


Figure 5.13: Detail of the triceratops. Solid line segments stand for shield edges.

Conclusion

The main theme of this dissertation is the study of operations which preserve and/or give prominence to the topology of geometric objects, by decomposition, deformation, or shortening. These theoretical works are motivated by questions arising in some applications. We aimed at optimizing curves on surfaces: families of simple, pairwise disjoint cycles, or graphs embeddings (keeping their vertices fixed). We have also worked on the shortening and the deformation of rectilinear embeddings of graphs in the plane, this time by moving the vertices. Then, we have given an algorithm to decompose the space $\mathbf{R}^3$ into Delaunay simplexes conforming to polyhedral constraints. We now sum up each of our contributions, trying to deduce lessons from this, then we sketch future research directions.

Summary of the contributions

Optimization of curves on surfaces

We have first considered the problem of shortening an embedding of graph or of cycles on a surface, while maintaining the topological properties of these embeddings. We have introduced the notion of cut system, which is a set of curves splitting the surface into topologically elementary surfaces. We have described quite natural iterative processes to shorten such cut systems. The analysis of these algorithms, sometimes tricky, turned out to be very fruitful. The consequences of this analysis are of several natures:

- on the *complexity* of the algorithms: they converge in a finite number of steps; their complexity is polynomial in the complexity of the surface and of the input system, and in a geometric characteristic of the surface (ratio between the largest and the smallest weight of the edges);
- on the *resulting system*: after stabilization, the cut system is optimal, *i.e.*, as short as possible among all cut systems having the same topological properties;
- from these properties arise new *methods* to optimize a set of simple, disjoint curves on a surface: it is sufficient to extend this family into a cut system, to optimize this system, and to remove the curves corresponding to the introduced curves.

This work was initiated further to the paper [105], which provides algorithms for computing canonical polygonal schemata. The fundamental systems of loops

obtained in this paper, of size *asymptotically* optimal, are visually not satisfying because they are longer than necessary, which is harmful for applications; our initial goal was simply to remedy this problem. We have thus developed an optimization algorithm of a fundamental system of loops (very close to the optimization algorithm of a cut system by graph, in the case where the graph has one single vertex). Even without a theoretical analysis of the result of this algorithm, it was clear that it would give good results in practice, sufficient for most applications, and we could have given up at that point. But the analysis has shown the consequences mentioned above, and has led to simplifications of this algorithm (the complexity of the algorithms we presented here is better than the one given in [40]), and to extensions (shortening of embeddings of graphs and of cycles).

Tutte's barycenter method applied to isotopies

In Chapter 4, we have studied a classical theorem to embed graphs, Tutte's barycentric theorem, dating back to 1963 [152], and we have exploited it to study isotopies of graphs embeddings, in view of applications like metamorphosis. It seemed interesting to give a more modern proof of this theorem, admitting the results that Tutte simultaneously proved and which are now well-known (Kuratowski's theorem). It turned out later that a proof using some arguments similar to ours had been given in 1996 by Richter-Gebert [129, section 12.2]. Some differences remain between both proofs, and it would be interesting to compare them in detail.

We have then described a method, based on Tutte's theorem, to create an isotopy between two embeddings of a given triangulated planar graph. The interest of the use of Tutte's theorem is that it has a simple physical interpretation: we have tried to go as far as possible with this interpretation in the creation of isotopies. The results tend to show that we can create an isotopy only under quite restrictive conditions on the initial and final embeddings.

Finally, the statement of Tutte's theorem admits a natural generalization in higher dimensions, and we have proved that this statement becomes false in dimension 3. The method we used is quite complex, since far simpler counter-examples have been found later, but is interesting on the theoretical viewpoint. Indeed, the fact that this theorem does not generalize to dimension three is directly connected to a known topological obstacle arising when trying to go from dimension two to dimension three, described by Starbird [141], specifically, the existence of some simplicial complexes embedded in $\mathbf{R}^3$ which are combinatorially equivalent but non-isotopic, even if their boundaries are the same. This also illustrates the link between this theorem by Tutte and the isotopies of simplicial complexes.

Conforming Delaunay triangulations in 3D

In Chapter 5, we have given an algorithm to compute a set of points in $\mathbf{R}^3$, whose Delaunay triangulation conforms to a given set of polyhedral constraints;

the problem comes from mesh generation and geometric modeling. The main interest of our algorithm, compared to previous approaches, is that the number of resulting points is reasonable in practice. The method consists in "protecting" the constrained vertices and edges by spheres whose size fits the local geometry of the constraints. An important question is to know whether this algorithm, as it is or slightly modified, is usable in practice to create volumic meshes.

Future works

Our works can be extended and improved in several directions.

Tetrahedral meshes

Our work on conforming Delaunay triangulations lies in the more general theme of the creation of *tetrahedral meshes*. In domains such as scientific computing, numerical simulation, geometric modeling, and visualization, a preliminary requirement is to decompose the space or the studied object into elementary objects, tetrahedra, triangles, edges, and vertices. The properties of the mesh: number, shape, and size of its simplexes, have great impact on the speed and efficiency of a numerical computation or an algorithmic treatment. Delaunay triangulations constitute a privilegiated way to create meshes, due to their well-studied properties and to the shape of its elements. In two dimensions, conforming and constrained Delaunay triangulations can both be used for this purpose. In three dimensions, the constrained Delaunay triangulation of a polyhedral object exists only under some conditions (see [139]). It is necessary to conduct research in this field, in order to possibly mix constrained and conforming Delaunay triangulations together. Obtaining theoretical guarantees on the quality of the output mesh of an algorithm constitutes a difficult but interesting problem. The creation of anisotropic meshes, and meshes on curved objects, are long-term goals.

Embeddings and isotopies of 3D complexes

There are several possible extensions of our results regarding Tutte's barycentric theorem. First of all, to apply the theorem, the fixed vertices must be on the boundary of a convex polygon. Is it possible to generalize this theorem to the case where other vertices, located inside the polygon, have also their positions fixed? This problem seems interesting from the point of view of constrained parameterization. On the other hand, one can wish to create an isotopy between two embeddings of graphs while requiring that some vertex is fixed throughout the deformation, or follows a prescribed move.

But the potential extensions of this work mainly lie in the 3D case. We have proved a negative result: Tutte's barycentric method does not work in dimension three. Thus, there remains to find algorithmic methods to embed a volumic simplicial complex into $\mathbf{R}^3$, if possible as natural and leaving as much flexibility as Tutte's method. This problem is certainly difficult (the existence of knots in dimension three and higher is an example of obstructions that exist when

dimension increases), but quite fundamental (it is only a generalization of the problem of graph drawing to higher dimension).

We could then work on the problem of computing isotopies between two such embeddings, allowing if necessary the refinement of the simplicial complex: the existence of such an isotopy has been discussed ([15, 143]), but never studied under an algorithmic viewpoint. The theory of realization spaces of polytopes in dimension 4 and Mnëv's universality theorem [130] are related to these questions and may provide some useful consequences. All these problems are very important in the domain of three-dimensional modeling: 3D parameterization and volumic metamorphoses are two examples of possible applications of these theoretical questions.

Topological decompositions and shortenings

Our work related to the shortening of curves on surfaces can be extended to several directions, presented in the conclusion of Chapter 3; we refer the reader to this section. We now give research tracks which are less directly connected to this work.

Chapters 3 and 4 are both concerned with the shortening of graphs embeddings:

- in Chapter 4, we want to move the vertices of a rectilinear graph embedding in the plane, in order to minimize the sum of the squares of the lengths of the edges (in the case of unit weights);
- in Chapter 3, we shorten the edges of an embedding of a graph on a surface, keeping the vertices fixed. We also achieve, in particular, the minimization of the sum of the squares of the lengths of the edges among some class of embeddings.

An attractive idea is to mix both aspects, working on graphs embeddings on a surface whose vertices are allowed to move. It can be desirable to compute the graph embedding (possibly within a given homotopy class) which minimizes the sum of the squares of the lengths of the edges. On the other hand, how to create an isotopy between two isotopic graphs embeddings on a surface (without using an embedding as short as possible)? All these questions raised in Chapter 4 in a planar realm can be also asked on a surface.

A triangulation of a surface can be simplified by iterative edge contractions, when this contraction does not modify the topology of the surface. The triangulations of a given surface which do not admit any valid contraction are called the *minimal triangulations* of a surface (see [61, Chapter 4], and [8, 9, 120, 52]). A goal of these papers is to count the minimal triangulations of a given surface and to compute them. The existence of a pants decomposition of a surface yields an indication of its structure, and may allow to deduce a way to bound and/or generate its minimal triangulations: is it possible to define the notion of minimal triangulation of a surface with boundary, like a pair of pants, and how can the minimal triangulations of a surface be obtained from a gluing of minimal triangulations of pairs of pants?

Some works [94, 65, 13] have considered the problem of shortening curves in the plane within a given homotopy class. In this sense, our work constitutes a generalization of these works to not necessarily planar surfaces. Is it possible to generalize other known results in the plane? A recent work [66] considers the problem of computing separators in a graph embedded on a not necessarily planar surface, using a polygonal schema of the surface. The literature on planar graphs is abundant; how can such results be generalized to graphs embedded on surfaces?

Most surfaces used in practice are embedded in $\mathbf{R}^3$. But our works do not consider this embedding at all. A track for future research is to determine the types of decompositions which are more natural than other ones with respect to the embedding in the ambiant space. For instance, it seems desirable to split the surface along closed curves which enclose the (volumic) object or its complementary part (that is, the curve shall be the boundary of a disk included in the object or in its complementary part). How to create a pants decomposition or a fundamental system of loops satisfying this condition? How can tools such as homology in three dimensions, or the contour graph, be used? Are other types of decompositions (by Morse complexes, for example) more adequate? What are the applications concerned by this question (metamorphosis between volumic objects, three-dimensional mesh recognition, ...)?

A probably more difficult problem is to work on the decomposition, shortening, and deformation of geometric objects in higher dimensions. How to shorten a surface in the space $\mathbf{R}^3$ minus some obstacles, keeping the homotopy class of the surface? A series of questions in algorithmic knot theory can also be raised: how to represent a knot in a form as canonical as possible [134], to realize algorithmically the unknotting of a knot, or to deform a knot into another one?

List of Figures

Bibliography

[1] P. K. Agarwal, B. Aronov, J. O'Rourke, and C. A. Schevon. Star unfolding of a polytope with applications. *SIAM Journal on Computing*, 26(6):1689–1713, 1997.

[2] P. Alliez, É. Colin de Verdière, O. Devillers, and M. Isenburg. Isotropic surface remeshing. In *Proceedings of the International Conference on Shape Modelling and Applications (SMI'03)*, pages 49–58, 2003.

[3] P. Alliez and C. Gotsman. Recent advances in compression of 3D meshes. In *Proceedings of the Symposium on Multiresolution in Geometric Modeling*, 2003.

[4] P. Alliez, M. Meyer, and M. Desbrun. Interactive geometry remeshing. *ACM Transactions on Graphics. Special issue for SIGGRAPH conference*, 21(3):347–354, 2002.

[5] P. Angelier and M. Pocchiola. A sum of squares theorem for visibility complexes and applications. In B. Aronov, S. Basu, J. Pach, and M. Sharir, editors, *Discrete and Computational Geometry: The Goodman–Pollack Festschrift*, volume 25 of *Algorithms and Combinatorics*, pages 79–139. Springer-Verlag, 2003.

[6] M. A. Armstrong. *Basic topology.* Undergraduate Texts in Mathematics. Springer-Verlag, 1983.

[7] J.-M. Autebert. *Théorie des langages et des automates.* Masson, 1994.

[8] D. W. Barnette and A. L. Edelson. All orientable 2–manifolds have finitely many minimal triangulations. *Israel Journal of Mathematics*, 62:90–98, 1988.

[9] D. W. Barnette and A. L. Edelson. All 2–manifolds have finitely many minimal triangulations. *Israel Journal of Mathematics*, 67:123–128, 1989.

[10] B. Becker and G. Hotz. On the optimal layout of planar graphs with fixed boundary. *SIAM Journal on Computing*, 16(5):946–972, 1987.

[11] M. Bern, D. Eppstein, P. K. Agarwal, N. Amenta, P. Chew, T. Dey, D. P. Dobkin, H. Edelsbrunner, C. Grimm, L. J. Guibas, J. Harer, J. Hass,

A. Hicks, C. K. Johnson, G. Lerman, D. Letscher, P. Plassmann, E. Sedgwick, J. Snoeyink, J. Weeks, C. Yap, and D. Zorin. Emerging challenges in computational topology. ACM Computing Research Repository, 1999. Report of an NSF seminar on computational topology.

[12] J. Berstel. *Transductions and context-free languages.* Teubner Studienbücher Informatik, 1979.

[13] S. Bespamyatnikh. Computing homotopic shortest paths in the plane. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'03)*, pages 609–617, 2003.

[14] R. H. Bing. *The geometric topology of 3-manifolds.* American Mathematical Society, 1983.

[15] R. H. Bing and M. Starbird. Linear isotopies in E^2. *Transactions of the American Mathematical Society*, 237:205–222, 1978.

[16] G. D. Birkhoff. Dynamical systems with two degrees of freedom. *Transactions of the American Mathematical Society*, 18:199–300, 1917.

[17] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G. Ziegler. *Oriented matroids.* Cambridge University Press, 1993.

[18] E. D. Bloch, R. Connelly, and D. W. Henderson. The space of simplexwise linear homeomorphisms of a convex 2-disk. *Topology*, 23(2):161–175, 1984.

[19] J.-D. Boissonnat and M. Yvinec. *Géométrie algorithmique.* Ediscience international, 1995. Published in English under the title *Algorithmic Geometry*, Cambridge University Press, 1998. Translated by Hervé Brönimann.

[20] B. Bollobás. *Modern graph theory*, volume 184 of *Graduate Texts in Mathematics.* Springer-Verlag, 1998.

[21] T. Brahana. Systems of circuits on 2-dimensional manifolds. *Annals of Mathematics*, 23:144–168, 1921.

[22] G. R. Brightwell and E. R. Scheinerman. Representations of planar graphs. *SIAM Journal on Discrete Mathematics*, 6(2):214–229, 1993.

[23] R. A. Brualdi and H. J. Ryser. *Combinatorial matrix theory*, volume 39 of *Encyclopedia of Mathematics and its Applications.* Cambridge University Press, 1991.

[24] P. Buser. *Geometry and spectra of compact Riemann surfaces*, volume 106 of *Progress in Mathematics.* Birhäuser Verlag, 1992.

[25] S. Cabello, Y. Liu, A. Mantler, and J. Snoeyink. Testing homotopy for paths in the plane. In *Proceedings of the 18th Annual ACM Symposium on Computational Geometry (SOCG'02)*, pages 160–169, 2002.

[26] S. S. Cairns. Deformations of plane rectilinear complexes. *American Mathematical Monthly*, 51:247–252, 1944.

[27] J. Canny and J. Reif. New lower bound techniques for robot motion planning problems. In *Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science (FOCS'87)*, pages 400–409, 1987.

[28] D. E. Cardoze. An optimal algorithm for the minimum edge cardinality cut surface problem. In *Proceedings of the 19th Annual ACM Symposium on Computational Geometry (SOCG'03)*, pages 338–343, 2003.

[29] S. Chaiken. A combinatorial proof of the all minors matrix tree theorem. *SIAM Journal on Algebraic and Discrete Methods*, 3:319–329, 1982.

[30] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete and Computational Geometry*, 6:485–524, 1991.

[31] J. Chen and Y. Han. Shortest paths on a polyhedron. In *Proceedings of the 6th Annual ACM Symposium on Computational Geometry (SOCG'90)*, pages 360–369, 1990.

[32] L. P. Chew. Guaranteed-quality triangular meshes. Technical Report TR-89-983, Department of Computer Science, Cornell University, Ithaca, NY, April 1989.

[33] D. R. J. Chillingworth. Simple closed curves on surfaces. *Bulletin of the London Mathematical Society*, 1:310–314, 1969.

[34] D. R. J. Chillingworth. An algorithm for families of disjoint simple closed curves on surfaces. *Bulletin of the London Mathematical Society*, 3:23–26, 1971.

[35] M. Chrobak, M. T. Goodrich, and R. Tamassia. Convex drawings of graphs in two and three dimensions. In *Proceedings of the 12th Annual ACM Symposium on Computational Geometry (SOCG'96)*, pages 319–328, 1996.

[36] M. Chrobak and G. Kant. Convex grid drawings of 3-connected planar graphs. *International Journal on Computional Geometry and Applications*, 7(3):211–223, 1997.

[37] M. Cohen and M. Lustig. Paths of geodesics and geometric intersection numbers. I. In *Combinatorial group theory and topology (Alta, Utah, 1984)*, volume 111 of *Annals of Mathematical Studies*, pages 479–500. Princeton University Press, 1987.

[38] D. Cohen-Steiner, É. Colin de Verdière, and M. Yvinec. Conforming Delaunay triangulations in 3D. *Computational Geometry: Theory and Applications*, 2004(?). To appear. A preliminary version appeared in the *Proceedings of the 18th Annual ACM Symposium on Computational Geometry (SOCG'02)*.

[39] É. Colin de Verdière. Isotopies de graphes planaires avec applications à la métamorphose. D.E.A., École normale supérieure (Paris), 2000. `http://www.di.ens.fr/~colin/textes/dea.ps.gz`.

[40] É. Colin de Verdière and F. Lazarus. Optimal system of loops on an orientable surface. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'02)*, pages 627–636, 2002. A preliminary version appeared in the *Proceedings of the 18th European Workshop on Computational Geometry (EuroCG'02)*.

[41] É. Colin de Verdière and F. Lazarus. Optimal pants decompositions and shortest homotopic cycles on an orientable surface. In *Proceedings of the 11th Symposium on Graph Drawing (GD'03)*, 2003. To appear. A preliminary version appeared in the *Abstracts of the 19th European Workshop on Computational Geometry (EuroCG'03)*.

[42] É. Colin de Verdière, M. Pocchiola, and G. Vegter. Tutte's barycenter method applied to isotopies. *Computational Geometry: Theory and Applications*, 26(1):81–97, August 2003. A preliminary version appeared in the *Proceedings of the 13th Canadian Conference on Computational Geometry (CCCG'01)*.

[43] Y. Colin de Verdière. Comment rendre géodésique une triangulation d'une surface ? *L'Enseignement Mathématique*, 37:201–212, 1991.

[44] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to algorithms*. MIT Press, 1990.

[45] R. Courant and H. Robbins. *What is mathematics?* Oxford University Press, 1979. An elementary approach to ideas and methods.

[46] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 1997.

[47] M. de Berg, M. van Kreveld, and S. Schirra. Topologically correct subdivision simplification using the bandwidth criterion. *Cartography and GIS*, 25:243–257, 1998.

[48] M. de Graaf and A. Schrijver. Making curves minimally crossing by Reidemeister moves. *Journal of Combinatorial Theory, Series B*, 70(1):134–156, 1997.

[49] M. Dehn. Transformation der Kurven auf zweiseitigen Flächen. *Mathematische Annalen*, 72:413–421, 1912.

[50] M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes. In *Eurographics conference proceedings*, pages 209–218, 2002.

[51] T. K. Dey, H. Edelsbrunner, and S. Guha. Computational topology. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry - Proc. 1996 AMS-IMS-SIAM Joint*

Summer Research Conf. Discrete and Computational Geometry: Ten Years Later, number 223 in Contemporary Mathematics, pages 109–143. American Mathematical Society, 1999.

[52] T. K. Dey, H. Edelsbrunner, S. Guha, and D. V. Nekhayev. Topology preserving edge contraction. *Publications de l'Institut Mathématique (Beograd)*, 60(80):23–45, 1999. Geometric combinatorics (Kotor, 1998).

[53] T. K. Dey and S. Guha. Transforming curves on surfaces. *Journal of Computer and System Sciences*, 58:297–325, 1999.

[54] T. K. Dey and H. Schipper. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. *Discrete and Computational Geometry*, 14(1):93–110, 1995.

[55] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph drawing.* Prentice Hall, 1999.

[56] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik 1*, pages 269–271, 1959.

[57] P. H. Doyle and D. A. Moran. A short proof that compact 2-manifolds can be triangulated. *Inventiones Mathematicae*, 5:160–162, 1968.

[58] P. Eades and P. Garvan. Drawing stressed planar graphs in three dimensions. In *Proceedings of the 3rd International Conference on Graph Drawing (GD'95)*, volume 1027 of *Lecture Notes in Computer Science*, pages 212–223. Springer-Verlag, 1996.

[59] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. *Computer Graphics*, 29(Annual Conference Series):173–182, 1995.

[60] H. Edelsbrunner. *Algorithms in combinatorial geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science.* Springer-Verlag, 1987.

[61] H Edelsbrunner. *Geometry and topology for mesh generation.* Cambridge University Press, 2001.

[62] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-Smale complexes for piecewise linear 3-manifolds. In *Proceedings of the 19th Annual ACM Symposium on Computational Geometry (SOCG'03)*, pages 361–370, 2003.

[63] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete and Computational Geometry*, 30:87–107, 2003.

[64] H. Edelsbrunner and T. S. Tan. An upper bound for conforming Delaunay triangulations. *Discrete and Computational Geometry*, 10(2):197–213, 1993.

[65] A. Efrat, S. G. Kobourov, and A. Lubiw. Computing homotopic shortest paths efficiently. In *Proceedings of the 10th Annual European Symposium on Algorithms (ESA'02)*, pages 411–423, 2002.

[66] D. Eppstein. Dynamic generators of topologically embedded graphs. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'03)*, pages 599–608, 2003.

[67] D. B. A. Epstein. Curves on 2-manifolds and isotopies. *Acta Mathematica*, 115:83–107, 1966.

[68] J. Erickson and S. Har-Peled. Optimally cutting a surface into a disk. In *Proceedings of the 18th Annual ACM Symposium on Computational Geometry (SOCG'02)*, pages 244–253, 2002.

[69] I. Fáry. On straight line representations of planar graphs. *Acta scientiarum mathematicarum(Szeged)*, 11:229–233, 1948.

[70] S. Felsner. Convex drawings of planar graphs and the order dimension of 3-polytopes. *Order*, 18(1):19–37, 2001.

[71] C. D. Feustel. Homotopic arcs are isotopic. *Proceedings of the American Mathematical Society*, 17:891–896, 1966.

[72] M. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14(3):231–250, 1997.

[73] M. S. Floater and C. Gotsman. How to morph tilings injectively. *Journal of Computational and Applied Mathematics*, 101:117–129, 1999.

[74] A. T. Fomenko and T. L. Kunii. *Topological modeling for visualization.* Springer-Verlag, 1997.

[75] M. Freedman, J. Hass, and P. Scott. Closed geodesics on surfaces. *Bulletin of the London Mathematical Society*, 14:385–391, 1982.

[76] P. J. Frey and P.-L. George. *Maillages. Applications aux éléments finis.* Hermès Science Publications, 1999.

[77] C. Friedrich and P. Eades. The Marey graph animation tool demo. In *Proceedings of the 8th International Symposium on Graph Drawing (GD'00)*, volume 1984 of *Lecture Notes in Computer Science*, pages 396–406, 2001.

[78] M. E. Gage. Curve shortening on surfaces. *Annales Scientifiques de l'École normale supérieure, Quatrième série*, 23(2):229–256, 1990.

[79] M. Garland, A. Willmott, and P.S. Heckbert. Hierarchical face clustering on polygonal surfaces. In *Proceedings of the ACM Symposium on Interactive 3D Graphics*, pages 49–58, 2001.

[80] J. E. Goodman. Pseudoline arrangements. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 5, pages 83–110. CRC Press, 1997.

[81] C. Gotsman and V. Surazhsky. Guaranteed intersection-free polygon morphing. *Computers and Graphics*, 25(1):67–75, 2001.

[82] M. A. Grayson. The heat equation shrinks embedded plane curves to round points. *Journal of Differential Geometry*, 26(2):285–314, 1987.

[83] M. A. Grayson. Shortening embedded curves. *Annals of Mathematics. Second Series*, 129(1):71–111, 1989.

[84] M. D. Greendlinger. On Dehn's algorithm for the conjugacy and word problems with applications. *Communications on Pure and Applied Mathematics*, 13:641–677, 1960.

[85] P. M. Gruber and J. M. Wills, editors. *Handbook of convex geometry*. North-Holland, 1993.

[86] X. Gu, S. Gortler, and H. Hoppe. Geometry images. *ACM Transactions on Graphics. Special issue for SIGGRAPH conference*, 21(3):355–361, 2002.

[87] D. Halperin, L. Kavraki, and J.-C. Latombe. Robotics. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 41, pages 755–778. CRC Press, 1997.

[88] J. Hass and P. Scott. Intersections of curves on surfaces. *Israel Journal of Mathematics*, 51(1–2):90–120, 1985.

[89] J. Hass and P. Scott. Shortening curves on surfaces. *Topology*, 33(1):25–43, 1994.

[90] A. Hatcher. Pants decompositions of surfaces. `http://www.math.cornell.edu/~hatcher/Papers/pantsdecomp.pdf`, 2000.

[91] A. Hatcher. *Algebraic topology*. Cambridge University Press, 2002. Available at `http://www.math.cornell.edu/~hatcher/`.

[92] M. Henle. *A combinatorial introduction to topology*. Dover Publications, 1994.

[93] M. Henzinger, P. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *Journal of Computer and System Sciences*, 55(1, part 1):3–23, 1997.

[94] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Computational Geometry: Theory and Applications*, 4:63–98, 1994.

[95] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM Journal on Computing*, 28(6):2215–2256, 1999.

[96] J. E. Hopcroft and P. J. Kahn. A paradigm for robust geometric algorithms. *Algorithmica*, 7(4):339–380, 1992.

[97] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

[98] I. M. James, editor. *History of topology*. North-Holland, 1999.

[99] T. Kanai and H. Suzuki. Approximate shortest path on a polyhedral surface based on selective refinement of the discrete graph and its applications. In *Geometric Modeling and Processing*, pages 241–250. IEEE Computer Society Press, 2000.

[100] T. Kanai, H. Suzuki, and F. Kimura. Metamorphosis of arbitrary triangular meshes. *IEEE Computer Graphics and Applications*, 20(2):62–75, 2000.

[101] R. Kimmel, N. Kiryati, and A. M. Bruckstein. Multi-valued distance maps in finding shortest paths between moving obstacles. *IEEE Transactions on Robotics and Automation*, 14(3):427–436, 1998.

[102] W. Klingenberg. *Lectures on closed geodesics*. Grundlehren der mathematischen Wissenschaften. Springer-Verlag, 1978.

[103] K. Kuratowski. Sur le problème des courbes gauches en topologie. *Fundamenta mathematicae*, 15:271–283, 1930.

[104] M. Lanthier, A. Maheshwari, and J.-R. Sack. Approximating shortest paths on weighted polyhedral surfaces. *Algorithmica*, 30(4):527–562, 2001.

[105] F. Lazarus, M. Pocchiola, G. Vegter, and A. Verroust. Computing a canonical polygonal schema of an orientable triangulated surface. In *Proceedings of the 17th Annual ACM Symposium on Computational Geometry (SOCG'01)*, pages 80–89, 2001.

[106] F. Lazarus and A. Verroust. 3D metamorphosis: a survey. *The Visual Computer*, 8(3):131–146, 1998.

[107] D. T. Lee and F. P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14(3):393–410, 1984.

[108] C. E. Leiserson and F. M. Maley. Algorithms for routing and testing routability of planar VLSI layouts. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing (STOC'85)*, pages 69–78, 1985.

[109] B. Lévy and J.-L. Mallet. Non-distorted texture mapping for sheared triangulated meshes. In *SIGGRAPH 1998*, pages 343–352, 1998.

[110] N. Linial, L. Lovász, and A. Wigderson. Rubber bands, convex embeddings and graph connectivity. *Combinatorica*, 8(1):91–102, 1988.

[111] L. Lusternik and L. Schnirelmann. Sur le problème de trois géodésiques fermées sur les surfaces de genre 0. *Comptes-rendus de l'Académie des Sciences de Paris*, 189:269–271, 1929.

[112] M. Lustig. Paths of geodesics and geometric intersection numbers. II. In *Combinatorial group theory and topology (Alta, Utah, 1984)*, volume 111 of *Annals of Mathematical Studies*, pages 501–543. Princeton University Press, 1987.

[113] J. Maillot, H. Yahia, and A. Verroust. Interactive texture mapping. In *SIGGRAPH 1993*, pages 27–34, 1993.

[114] W. S. Massey. *Algebraic Topology: An Introduction*, volume 56 of *Graduate Texts in Mathematics*. Springer-Verlag, 1977.

[115] J. S. B. Mitchell. Geometric shortest paths and network optimization. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of computational geometry*, pages 633–701. Elsevier, 2000.

[116] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, 1987.

[117] B. Mohar and C. Thomassen. *Graphs on surfaces*. Johns Hopkins Studies in the Mathematical Sciences. John Hopkins University Press, 2001.

[118] D. M. Mount. The number of shortest paths on the surface of a polyhedron. *SIAM Journal on Computing*, 19:593–611, 1990.

[119] M. Murphy, D. M. Mount, and C. W. Gable. A point-placement strategy for conforming Delaunay tetrahedralization. *International Journal of Computational Geometry and Applications*, 11(6):669–682, 2001.

[120] A. Nakamoto and K. Ota. Note on irreducible triangulations of surfaces. *Journal of Graph Theory*, 20(2):227–233, 1995.

[121] M. Neumann-Coto. A characterization of shortest geodesics on surfaces. *Algebraic and Geometric Topology*, 1:349–368, 2001.

[122] C. Ó Dúnlaing. Flat-face embeddings of certain 2-complexes. In *Abstracts of the 18th European Workshop on Computational Geometry (EuroCG'02)*, pages 75–79, 2002.

[123] J. Orlin. Line-digraphs, arborescences, and theorems of Tutte and Knuth. *Journal of Combinatorial Theory. Series B*, 25(2):187–198, 1978.

[124] E. Papadopoulou. k-pairs non-crossing shortest paths on a simple polygon. *International Journal of Computational Geometry and Applications*, 9(6):533–552, 1999.

[125] G. Perelman. The entropy formula for the Ricci flow and its geometric application. `http://www.arxiv.org/abs/math.DG/0211159`, November 2002.

[126] G. Perelman. Ricci flow with surgery on three-manifolds. `http://www.arxiv.org/abs/math.DG/0303109`, March 2003.

[127] D. Piponi and G. Borshukov. Seamless texture mapping of subdivision surfaces by model pelting and texture blending. In *SIGGRAPH 2000*, pages 471–478, 2000.

[128] D. Poulalhon and G. Schaeffer. Optimal coding and sampling of triangulations. In *Proceedings of the 30th International Colloquium on Automata, Languages and Programming (ICALP'03)*, pages 1080–1094, 2003.

[129] J. Richter-Gebert. *Realization spaces of polytopes*, volume 1643 of *Lecture Notes in Mathematics*. Springer-Verlag, 1996.

[130] J. Richter-Gebert. The universality theorems for oriented matroids and polytopes. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry — Proceedings of the 1996 AMS-IMS-SIAM Join Summer Research Confererence Discrete and Computational Geometry: Ten Years Later*, Contemporary Mathematics, pages 269–292, 1999.

[131] G. D. Rose. No assembly required. *The Sciences*, 36:26–31, 1996.

[132] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *Journal of Algorithms*, 18:548–585, 1995.

[133] A. Saalfeld. Delaunay edge refinements. In *Proceedings of the 3rd Canadian Conference on Computational Geometry (CCCG'91)*, pages 33–36, 1991.

[134] R. G. Scharein. *Interactive topological drawing*. PhD thesis, Department of Computer Science, The University of British Columbia, 1998.

[135] H. Schipper. Determining contractibility of curves. In *Proceedings of the 8th Annual ACM Symposium on Computational Geometry (SOCG'92)*, pages 358–367, 1992.

[136] O. Schramm. How to cage an egg. *Inventiones mathematicae*, 107:543–560, 1992.

[137] A. Schrijver. *Combinatorial optimization. Polyhedra and efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, 2003.

[138] J. A. Sethian. *Level set methods and fast marching methods*. Cambridge University Press, 1999.

[139] J. R. Shewchuk. A condition guaranteeing the existence of higher-dimensional constrained Delaunay triangulations. In *Proceedings of the 14th Annual ACM Symposium on Computational Geometry (SOCG'98)*, pages 76–85, 1998.

[140] J. R. Shewchuk. Tetrahedral mesh generation by Delaunay refinement. In *Proceedings of the 14th Annual ACM Symposium on Computational Geometry (SOCG'98)*, pages 86–95, June 1998.

[141] M. Starbird. A complex which cannot be pushed around in E^3. *Proceedings of the American Mathematical Society*, 63(2):363–368, 1977.

[142] M. Starbird. Linear isotopies in E^3. *Proceedings of the American Mathematical Society*, 65(2):342–346, 1977.

[143] M. Starbird. The Alexander linear isotopy theorem in E^3. In *Continua, decompositions, manifolds*, pages 243–267. University of Texas Press, Austin, 1983.

[144] S. K. Stein. Convex maps. *Proceedings of the American Mathematical Society*, 2:464–466, 1951.

[145] J. Stillwell. *Classical topology and combinatorial group theory.* Springer-Verlag, 1980.

[146] C. Thomassen. Kuratowski's theorem. *Journal of Graph Theory*, 5(3):225–241, 1981.

[147] C. Thomassen. Deformations of plane graphs. *Journal of Combinatorial Theory, Series B*, 34:244–257, 1983.

[148] C. Thomassen. The Jordan–Schönflies theorem and the classification of surfaces. *The American Mathematical Monthly*, 99(2):116–130, 1992.

[149] W. Thurston. *Three-dimensional geometry and topology, volume 1.* Princeton University Press, 1997.

[150] J. R. Tristano, S. J. Owen, and S. A. Canann. Advancing front surface mesh generation in parametric space using a Riemannian surface definition. In *Proceedings of the 7th International Meshing Roundtable*, pages 429–445, 1998.

[151] W. T. Tutte. Convex representations of graphs. *Proceedings London Mathematical Society*, 10:304–320, 1960.

[152] W. T. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, 13:743–768, 1963.

[153] G. Vegter. Computational topology. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 28, pages 517–536. CRC Press, 1997.

[154] G. Vegter and C. K. Yap. Computational complexity of combinatorial surfaces. In *Proceedings of the 6th Annual ACM Symposium on Computational Geometry (SOCG'90)*, pages 102–111, 1990.

[155] K. Wagner. Bemerkungen zum Vierfarbenproblem. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 46:26–32, 1936.

[156] W. Whiteley. Motions and stresses of projected polyhedra. *Structural Topology*, 7:13–38, 1982.

[157] D. Zeilberger. A combinatorial approach to matrix algebra. *Discrete Mathematics*, 56(1):61–72, 1985.

[158] G. M. Ziegler. *Lectures on polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, 1994.

[159] Živaljević. Topological methods. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 11, pages 209–224. CRC Press, 1997.

Index

Abstract

This dissertation is concerned with the algorithmic study of three operations on geometric objects: their decomposition, their deformation, and their shortening.

The main theme is the shortening of curves on a surface and the decomposition of a surface into topologically elementary surfaces. Let us consider a set of curves drawn on a surface, without intersections or self-intersections. We wish to shorten them as much as possible while preserving their topology, that is, by deforming them continuously without introducing intersections beween them. This is in particular useful in geometric modeling and computer graphics, where finding short topological decompositions is necessary.

We present classical results of topology of surfaces and an overview of the previous works in computational topology of surfaces which are related to this problem. We then provide algorithms to solve it, for embeddings of graphs with fixed vertices and sets of cycles without intersections, in a setting where the curves are drawn on the vertex-edge graph of a polyhedral surface. These algorithms are polynomial in their input and in the ratio between the extreme weights of the edges of the vertex-edge graph. We prove optimality results of each of the resulting curves.

Another work, motivated by the creation of metamorphoses (morphings) between objects, deals with the deformation of triangulations in the plane. We reprove and use Tutte's barycentric embedding theorem to build such deformations, and we prove that its analogue in dimension three does not hold.

A conforming Delaunay triangulation is a Delaunay triangulation of the space which fits the shape of a given polyhedral object. This concept is used in mesh generation. We give an algorithm which computes such a triangulation in dimension three, with a relatively small number of vertices, due to the fact that it adapts to the local geometry of the object.

Keywords: computational topology, homotopy, shortest path, topological decomposition, embedding, graph, parameterization, Delaunay triangulation.