Topics in Low-Dimensional Computational Topology

Arnaud de Mesmay

THÈSE DE DOCTORAT

présentée et soutenue publiquement
le 7 juillet 2014
en vue de l'obtention du grade de

Docteur de l'École normale supérieure

Spécialité : Informatique

par

ARNAUD DE MESMAY

Topics in Low-Dimensional Computational Topology

Membres du jury :

M. Frédéric CHAZAL (INRIA Saclay – Île de France)	rapporteur
M. Éric COLIN DE VERDIÈRE (ENS Paris et CNRS)	directeur de thèse
M. Jeff ERICKSON (University of Illinois at Urbana-Champaign)	rapporteur
M. Cyril GAVOILLE (Université de Bordeaux)	examinateur
M. Pierre PANSU (Université Paris-Sud)	examinateur
M. Jorge RAMÍREZ-ALFONSÍN (Université Montpellier 2)	examinateur
Mme Monique TEILLAUD (INRIA Sophia-Antipolis – Méditerranée)	examinatrice

Autre rapporteur :
M. Eric SEDGWICK (DePaul University)

Unité mixte de recherche 8548 : Département d'Informatique de l'École normale supérieure
École doctorale 386 : Sciences mathématiques de Paris Centre
Numéro identifiant de la thèse : 70791

À Monsieur Lagarde, qui m'a donné l'envie d'apprendre.

Résumé

La topologie, c'est-à-dire l'étude qualitative des formes et des espaces, constitue un domaine classique des mathématiques depuis plus d'un siècle, mais il n'est apparu que récemment que pour de nombreuses applications, il est important de pouvoir calculer *informatiquement* les propriétés topologiques d'un objet. Ce point de vue est la base de la *topologie algorithmique*, un domaine très actif à l'interface des mathématiques et de l'informatique auquel ce travail se rattache. Les trois contributions de cette thèse concernent le développement et l'étude d'algorithmes topologiques pour calculer des décompositions et des déformations d'objets de basse dimension, comme des graphes, des surfaces ou des 3-variétés.

Le premier problème auquel nous nous attaquons traite de déformations : comment peut-on tester si deux graphes dessinés sur une même surface sont isotopes, c'est-à-dire si l'on peut déformer continûment l'un en l'autre ? Ce type de question est relié à des problèmes pratiques que l'on rencontre par exemple dans les systèmes d'information géographique ou les métamorphoses (*morphings*). En nous appuyant sur des concepts de géométrie hyperbolique et de la théorie des *mapping class groups*, nous établissons d'abord un critère combinatoire pour caractériser l'isotopie, ce qui reprouve et améliore un résultat de Ladegaillerie de 1984. Ensuite, en combinant ceci avec des algorithmes antérieurs pour tester l'homotopie de courbes, nous fournissons des algorithmes efficaces pour résoudre ce problème d'isotopie de graphes.

Nous déplaçons ensuite notre étude vers des problèmes de décompositions, en nous intéressant à la découpe de surfaces le long de courbes ou de graphes respectant certaines propriétés topologiques, ce qui est une routine importante en algorithmique des graphes ou en infographie, parmi d'autres domaines. En établissant une forte connexion avec le cas continu, ainsi qu'en étudiant un modèle discret de surfaces aléatoires, nous améliorons les meilleures bornes connues pour plusieurs schémas de découpe. Cela prouve en particulier une conjecture de Przytycka et Przytycki datant de 1993, et fournit également un nouvel algorithme pour calculer des décompositions en pantalons courtes.

Enfin, nous montons d'une dimension, où les meilleurs algorithmes connus pour de nombreux problèmes topologiques (comme le célèbre problème du nœud) sont exponentiels. La plupart de ces algorithmes reposent sur les surfaces normales, un objet omniprésent pour étudier les surfaces plongées dans une 3-variété. Nous étudions une relaxation naturelle de cette notion, les *surfaces normales immergées*, dont la meilleure structure algébrique en fait de bons candidats pour obtenir des algorithmes polynomiaux pour des problèmes topologiques. Dans ce travail, nous montrons qu'utiliser des surfaces normales immergées mène naturellement à un problème de détection de singularités, et nous prouvons que celui-ci est NP-dur ; c'est un résultat notable car l'on dispose de très peu de preuves de difficulté en topologie en 3 dimensions. Notre réduction s'appuie sur une connexion avec une classe restreinte de problèmes de satisfaction de contraintes qui a été partiellement classifiée par Feder.

Abstract

Topology is the area of mathematics investigating the qualitative properties of shapes and spaces. Although it has been a classical field of study for more than a century, it only appeared recently that being able to *compute* the topological features of various spaces might be of great value for many applications. This idea forms the core of the blossoming field of *computational topology*, to which this work belongs. The three contributions of this thesis deal with the development and the study of topological algorithms to compute deformations and decompositions of low-dimensional objects, such as graphs, surfaces or 3-manifolds.

The first question we tackle concerns deformations: how can one test whether two graphs embedded on the same surface are isotopic, i.e., whether one can be deformed continuously into the other? This kind of problems is relevant to practical problems arising with morphings or geographic information systems, for example. Relying on hyperbolic geometry and ideas from the theory of mapping class groups, we first establish a combinatorial criterion to characterize isotopy, reproving and strengthening a result of Ladegaillerie (1984). Combined with earlier algorithms on the homotopy of curves, this allows us in turn to provide efficient algorithms to solve this graph isotopy problem.

We then shift our focus to decompositions, by investigating how to cut surfaces along curves or graphs with prescribed topological properties, which is an important routine in graph algorithms or computer graphics, amongst others domains. By establishing a strong connection with the continuous setting, as well as studying a discrete model for random surfaces, we improve the best known bounds for several instances of this problem. In particular, this proves a conjecture of Przytycka and Przytycki from 1993, and one of our new bounds readily translates into an algorithm to compute short pants decompositions.

Finally, we move up one dimension, where the best known algorithms for many topological problems, like for example unknot recognition, are exponential. Most of these algorithms rely on normal surfaces, a ubiquitous tool to study the surfaces embedded in a 3-manifold. We investigate a relaxation of this notion called *immersed normal surfaces*, whose more convenient algebraic structure makes them good candidates to solve topological problems in polynomial time. We show that when working with immersed normal surfaces, a natural problem on the detection of singularities arises, and we prove it to be NP-hard – this is noteworthy as hardness results are very scarce in 3-dimensional topology. Our reduction works by establishing a connection with a restricted class of constraint satisfaction problems which has been partially classified by Feder.

Table of Contents

Acknowledgments

Research sometimes works in mysterious ways, and if not for all the help I have been offered, I would have probably gotten lost on its tortuous path. To all the people I list here, and to all those I have forgotten, thank you.

It is customary to start this section with kind words to the supervisor, but in this case the first place does not only come out of tradition, it is truly deserved: Éric Colin de Verdière has been a spotless advisor for me throughout my time at the ENS. His expertise, patience and availability have been a tremendous help in the past three years, and his model will be a humbling asymptotics to aim for if it is someday my turn to advise. I would also like to thank Jean Ponce for co-advising me during the first year of this thesis.

I wish to express my gratitude to Frédéric Chazal, Jeff Erickson and Eric Sedgwick who displayed a genuine interest in my work and took the time to review this thesis carefully. I am also very thankful to the other members of my thesis committee: Cyril Gavoille, Pierre Pansu, Jorge Ramírez-Alfonsín and Monique Teillaud.

It was a thorough pleasure to collaborate with seasoned researchers: I am grateful to Benjamin Burton for inviting me to Brisbane and sharing many insights with me, and to Alfredo Hubard, who enlightened these years with his constant enthusiasm and curiosity. My thanks also go to Nati Linial and Uli Wagner who invited me respectively in Jerusalem and Vienna and allowed me entry into their mathematical worlds. This manuscript has also benefited from stimulating discussions with Frédéric le Roux, Frederic Dorn, Ramsay Dyer, Francis Lazarus, Michael Farber and James R. Lee, and with my mathematical friends Olivier, Henri, Quentin and Dimitris. My love for mathematics was ignited by my teachers Jean-Claude Lagarde and Rémi Briançon, and I am glad to show them my appreciation as well.

The École normale supérieure and its Computer Science department have been a delightful place to spend these years, and it has been a pleasure to be a part of the Talgo team with Vincent Cohen-Addad, Zhentao Li, Claire Mathieu and Hang Zhou. I would also like to thank Jean Ponce, Lise-Marie Bivard, Isabelle Delais, Joëlle Isnard, Valérie Mongiat, Jacques Beigbeder, Ludovic Ricardou, Claudie Berrard and the staff of the library for the quality of their work.

On a more personal note, my heartfelt thanks go to my parents Olivier and Nelly and my brother Frédéric for their indefectible support and love.

Last but not least, j'aimerais adresser un merci particulier à Fanny. Je n'irais pas bien loin si elle ne marchait pas à mes côtés.

Paris, June 2014

ARNAUD DE MESMAY

CHAPITRE 1

Introduction en français

En 1860, la plupart des scientifiques étaient convaincus qu'un corps mystérieux appelé l'éther formait la substance qui remplissait l'espace, et qu'il apparaissait dans toutes les structures possibles de notre univers. Pour expliquer les différents éléments que l'on pouvait observer dans la nature, Lord Kelvin proposa la *Théorie du Vortex*, qui postulait que les atomes étaient en fait des vortex d'éther, et qui modélisait donc les éléments chimiques comme des nœuds ou des entrelacs, de telle sorte que les différences topologiques entre les nœuds aient un impact sur les propriétés physiques des éléments correspondants. Cette théorie avait un attrait remarquable car elle permettait d'expliquer certaines propriétés discrètes de la matière que l'on pouvait déjà observer à l'époque, comme les longueurs d'onde auxquelles les atomes émettent et absorbent la lumière. Par exemple, le célèbre doublet d'émission du Sodium pourrait ainsi être hypothétiquement expliqué par sa forme intrinsèque de nœud de Hopf, c'est-à-dire de deux cercle entremêlés, comme le montre la figure 1.1.

Il a fallu peu de temps pour que la célèbre expérience de Michelson et Morley détruisît cette théorie en mettant en évidence qu'il n'y avait probablement pas d'éther du tout. Mais les dégâts avaient déjà été réalisés. Peter Guthrie Tait, en essayant d'étudier cette théorie du vortex, avait déjà établi les premières tables classifiant les nœuds, ainsi que les premières conjectures de la naissante *théorie de nœuds* qui leur correspondait ; il fut bientôt rejoint

FIGURE 1.1 – Le doublet d'émission du Sodium pourrait-il être expliqué par sa forme intrinsèque de nœud de Hopf ?

dans cette aventure par des armées de mathématiciens.

Sautons aux temps modernes, et l'on se rend compte que la théorie des nœuds, malgré son échec initial pour décrire la matière, apparaît dans une multitude de domaines : pour ne citer que quelques exemples, les brins d'ADN se retrouvent noués et dénoués par l'action d'enzymes [240], on sait désormais synthétiser de très petites molécules nouées ou entrelacées [251] et le lecteur est probablement familier avec l'expérience pénible et systématique de devoir dénouer des écouteurs en les sortant d'une poche trop serrée.

Dans une perspective un peu plus large, la théorie des nœuds n'est qu'un domaine de la *topologie*, qui est l'étude mathématique des espaces et des formes. La topologie est un champ de recherche très mûr, dont les premières ébauches comme la formule d'Euler datent au moins du XVIIe siècle avec Descartes. C'est aussi un domaine très actif, comme le montre la récente résolution de la conjecture de Poincaré par Perelman [195, 196].

Mais un nouvel aspect de cette discipline est apparu dans les dernières décennies, avec la révolution accompagnant les technologies de l'information : on ne se satisfait plus de la simple *compréhension* d'un objet topologique – qui est ce que les mathématiques fournissent traditionnellement – mais on veut également pouvoir *calculer* ces propriétés topologiques. Dans le cas de la théorie des nœuds, la question suivante ressort directement de cette perspective : comment peut-on déterminer algorithmiquement si un nœud est noué ? Il s'avère que cette question est fondamentale et illustre bien les techniques et les objectifs de ce nouveau point de vue. Développer des algorithmes systématiques pour s'attaquer à des questions d'ordre topologique est le sujet de la *topologie algorithmique*, un champ de recherche en pleine explosion, visant à la fois à revisiter de vieux problèmes topologiques sous une nouvelle perspective, et à raviver la topologie avec de nouvelles directions de recherche.

1.1 Approche

Cette thèse étudie des problèmes topologiques en basse dimension du point de vue d'un informaticien théorique. Cette perspective nous amène à combiner des techniques de plusieurs domaines à l'interface des mathématiques et de l'informatique.

Topologie. La topologie, et en particulier la topologie algébrique, vise à développer des outils pour acquérir une bonne compréhension des formes et des espaces. Cette étude peut emprunter différentes chemins, comme recomposer des espaces en des pièces plus simples (*théorie de la chirurgie*), analyser les différents sous-espaces qu'ils contiennent (théories de *l'homotopie* et de *l'homologie*), ou bien étudier la connexion entre leurs propriétés topologiques et métriques (*géométrisation*). Ces techniques, rassemblées dans l'optique de créer des algorithmes pour la *classification* et la *décomposition* des espaces de basse dimension, forment la base commune de cette thèse.

Il y a une saveur très différente entre travailler en basse (deux, trois ou quatre) et en haute dimension. Dans le second cas, l'espace supplémentaire permet d'utiliser de nombreux outils pour se déplacer qui sont inaccessibles en petite dimension. D'un point de vue mathématique, cela rend de nombreux problèmes bien plus faciles en grandes dimensions, comme l'illustre le cas de la conjecture de Poincaré. En effet, celle-ci a résisté bien plus longtemps en dimension 3 qu'en dimensions supérieures à cinq, pour lesquelles Smale a apporté une solution en 1961. D'un point de vue algorithmique en revanche, cet espace additionnel est aussi une malédiction : comme nous le verrons dans le chapitre 4, beaucoup de problèmes deviennent indécidables dès que l'on dépasse la dimension quatre, ce qui réduit à néant toute espoir d'obtenir des algorithmes pour nombre de questions. Cette thèse se focalise donc sur les espaces en dimension deux ou trois, c'est-à-dire essentiellement les surfaces et les 3-variétés (espaces ressemblant localement à l'espace euclidien usuel), qui sont le cadre de nombreux problèmes algorithmiques, à la fois résolubles et intéressants.

Structures discrètes. Un aspect important de notre angle d'attaque est qu'il nous amène à considérer essentiellement des structures discrètes. Cela pourrait paraître contradictoire de prime abord, puisque la continuité est une des notions les plus fondamentales de la topologie. Cependant, l'idée n'est pas d'étudier des structures fondamentalement discrètes comme des nuages de points, mais plutôt d'encoder des structures topologiques de façon finie, afin de les manipuler algorithmiquement. Ainsi, au lieu de considérer des surfaces comme des sous-variétés arbitraires d'un espace euclidien, nous préférerons les trianguler, pour pouvoir les traiter comme des morceaux linéaires collés les uns aux autres [1]. Cela donne à notre travail une dimension très combinatoire, et permet d'établir des connexions avec de nombreux domaines des mathématiques discrètes, en particulier avec la théorie des graphes.

Algorithmes. Enfin, notre approche nous amène à nous focaliser fortement sur les algorithmes. Notre objectif est de pouvoir effectuer des calculs efficaces sur les espaces topologiques, et notre point de vue théorique exige que l'on prouve la correction de nos algorithmes, et que l'on étudie leur complexité de la façon la plus précise possible, généralement d'un point de vue asymptotique. Une direction de recherche duale est d'établir la difficulté de certains problèmes, c'est-à-dire de montrer qu'il n'existe pas d'algorithmes efficaces pour effectuer certaines tâches. Dans les deux cas, nous sommes amenés à combiner des outils de topologie et d'informatique théorique pour atteindre ces objectifs, ce qui situe ce travail à l'intersection de ces deux domaines.

Nous remarquons que notre travail se situe au niveau des fondations, et que ces algorithmes restent sur le plan *théorique* et n'ont pas été implémentés. Cependant, ils sont

1. Cette approche n'a rien de nouveau en topologie algébrique, où l'on modélise depuis longtemps les espaces comme des complexes simpliciaux ou des CW-complexes, mais cela convient très bien à notre cadre de travail.

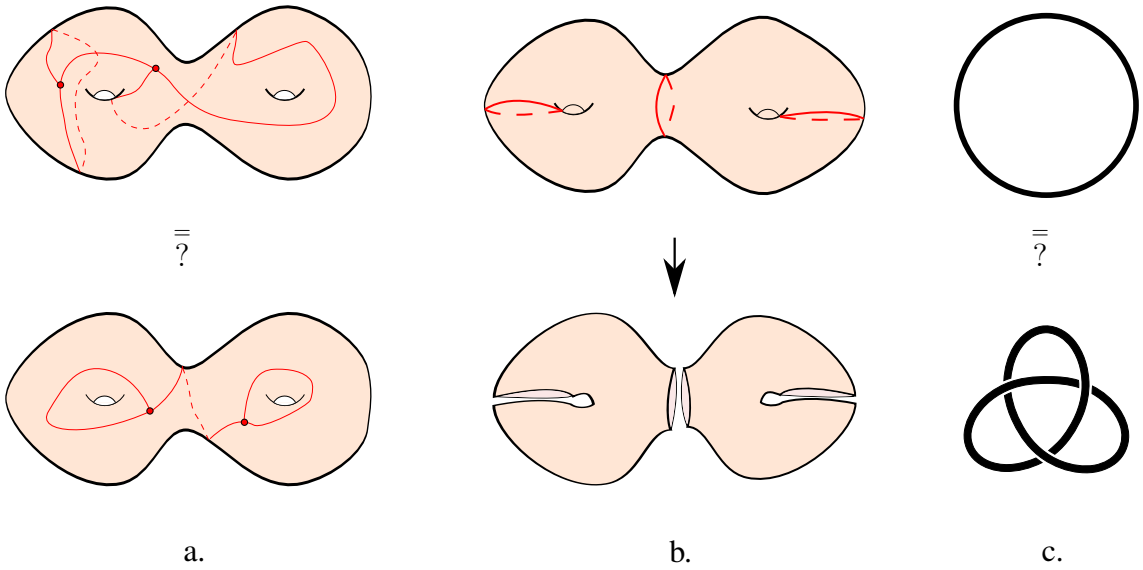


FIGURE 1.2 – a. Existe-t-il une déformation continue entre les deux graphes dessinés sur ce double tore ?
b. Comment découper efficacement une surface en pièces plus simples ?
c. Ces deux nœuds sont-ils les mêmes, c'est-à-dire peut-on passer de l'un à l'autre par une déformation continue ?

mûrs pour être appliqués et utilisés pour des applications où l'on a besoin de travailler avec des formes, comme pour l'infographie, les systèmes d'information géographique, la conception assistée par ordinateur, la robotique ou la bio-informatique. Nous présenterons aussi d'autres applications plus surprenantes d'ingrédients topologiques dans le chapitre 4. Développer soigneusement des implémentations robustes de nos résultats, pour les inclure dans des librairies standards, serait un versant intéressant de notre travail – mais pas celui sur lequel nous nous sommes concentrés jusque-là.

1.2 Contributions de cette thèse

Cette thèse revisite des concepts topologiques classiques d'un point de vue informatique. La première partie de ce travail traite de déformations : Étant donnés deux objets topologiques dans un espace, comment peut-on déterminer si l'un peut être continûment déformé en l'autre ? Résoudre ce genre de questions nous permettrait ensuite de classifier ces objets : la déformation nous fournit une notion précise d'équivalence, et donc une façon de rassembler les objets en catégories ayant des propriétés topologiques similaires. Nous étudions deux occurrences de ce problème de déformation. D'un côté nous nous intéressons aux graphes plongés sur des surfaces, et nous fournissons des algorithmes efficaces pour résoudre la question. D'un autre côté, nous regardons les plongements du cercle dans $\mathbb{R}^3$,

où cela mène au problème du nœud précédemment mentionné. Bien que nous ne résolvions pas le problème dans ce cas-là (c'est une des grandes questions ouvertes du domaine), nous prouvons la difficulté d'une approche naturelle pour s'y attaquer. Le second thème de notre étude tourne autour de décompositions : comment peut-on simplifier efficacement un espace topologique en le coupant en plus petits morceaux ? Dans le cas des surfaces, nous fournissons de nouveaux algorithmes et bornes pour plusieurs instances de cette question.

Tester l'isotopie de graphes sur les surfaces. Le *problème d'isotopies de graphes* pose la question suivante : étant donné un graphe plongé (c'est-à-dire dessiné sans croisements) de deux façons différentes sur une surface, existe-t-il une *isotopie*, c'est-à-dire une déformation continue, entre les deux plongements ? D'un point de vue mathématique, c'est une question fondamentale car l'isotopie est une relation d'équivalence très fine sur les graphes plongés, en cela qu'elle identifie exactement les graphes plongés qui sont intuitivement les mêmes. De plus, les graphes plongés apparaissent sous de très nombreuses formes en topologie de basse dimension, par exemple de par leurs connexions avec les *réseaux ferroviaires*, que l'on décrira succinctement au chapitre 4, ou avec les diagrammes de Heegaard, qui sont un outil fondamental en topologie des 3-variétés [131]. Tester l'isotopie a aussi des applications plus pratiques : les cartes routières peuvent par exemples être modélisées précisément par des graphes plongés, et tester l'isotopie permet de vérifier que deux cartes ont les mêmes propriétés topologiques (cette maison est-elle du bon côté de la route ?). Les domaines liés aux métamorphoses (*morphings*) ou à l'application de textures fournissent également d'autres applications.

À partir de travaux antérieurs à la fois informatiques (algorithmes pour tester l'*homotopie* de courbes plongées) et mathématiques (Ladegaillerie a fourni un critère combinatoire pour tester l'isotopie de graphes), nous fournissons un algorithme efficace pour tester si deux graphes sont isotopes. Un aspect particulier lorsqu'on travaille en basse dimension est qu'il y a une très forte connexion entre la géométrie et la topologie, et nos techniques reposent sur de la géométrie hyperbolique pour étudier la combinatoire de cette relation d'isotopie, ainsi que pour montrer la correction de notre algorithme.

Inégalités systoliques discrètes et décompositions de surfaces triangulées. Dans une seconde étape, nous déplaçons notre intérêt vers une problématique de décomposition. Décomposer une surface, c'est la découper en des morceaux plus petits et plus simples, et cela constitue une étape fondamentale pour de nombreuses applications, car cela permet de relier les propriétés topologiques de la surface à celles du plan qui sont très bien comprises. Cette démarche apparaît couramment par exemple pour développer des algorithmes pour les graphes plongés sur des surfaces, où cela se révèle très pratique de se ramener au cas planaire. Similairement, les décompositions de surface sont aussi pertinentes pour réaliser du maillage, de la paramétrisation ou pour de nombreuses autres applications ; nous renvoyons à l'introduction du chapitre 6 correspondant.

Lorsqu'on découpe une surface pour la simplifier, il est important de contrôler la longueur de la découpe : couper le long d'une courbe courte nous permet de limiter les modifications induites par cette décomposition. C'est donc un problème naturel d'ajouter une contrainte géométrique à une décomposition topologique en essayant de trouver la façon la plus courte pour découper une surface et la simplifier. Il existe différents schémas pour découper une surface, et nous étudions trois d'entre eux : les *systoles*, les *décompositions en pantalons* et les *graphes de découpe*. Nous améliorons d'abord les meilleures bornes connues sur les longueurs des systoles, ce qui répond à une conjecture de Przytycka et Przytycki datant de 1993. Nous fournissons aussi un nouvel algorithme pour calculer des décompositions en pantalons courtes, et nous identifions une obstruction à la présence de courts graphes de découpe. Nos résultats reposent sur une connexion explicite avec le cas continu, qui a été étudié en profondeur dans le cadre de la *géométrie systolique*, ainsi que sur l'utilisation d'un modèle aléatoire de surfaces.

Surfaces normales immergées. Enfin, nous fixons notre attention sur des problèmes en dimension 3. Le problème angulaire de la topologie algorithmique en cette dimension est le célèbre *problème du nœud* auquel nous avons déjà fait référence précédemment : Comment peut-on tester si un cercle plongé dans l'espace euclidien peut être isotopé en un cercle planaire, ou en termes profanes si un nœud peut être dénoué ? C'est un des rares problèmes algorithmiques pour lesquels aucun algorithme polynomial n'est connu, mais aucune preuve de difficulté non plus. De façon analogue, de nombreux problèmes en 3 dimensions sont encore assez mal compris et restent l'objet de nombreux efforts. Par exemple, alors que tester si deux surfaces sont homéomorphes (ou topologiquement équivalentes) est très simple, le problème analogue en 3 dimensions, où les espaces sont décrits par des tétraèdres collés les uns aux autres, est extrêmement délicat.

Un des outils principaux pour étudier un espace topologique est d'analyser les sous-espaces qu'il contient. Similairement, les meilleurs algorithmes connus pour des problèmes en 3 dimensions reposent tous sur le concept de *surfaces normales*, qui fournissent une structure algébrique pour représenter et manipuler les surfaces plongées dans un espace tridimensionnel. Dans ce travail, nous étudions une généralisation naturelle des surfaces normales où celles-ci sont autorisées à être immergées et non plus plongées. Cela mène naturellement à un problème de détection de singularités que l'on prouve être NP-difficile, c'est-à-dire qu'on ne peut probablement pas trouver d'algorithme polynomial pour ce problème – ce genre de preuve est particulièrement précieux dans ce domaine puisque très peu de résultats de difficulté y sont connus. Notre réduction s'appuie sur une connexion avec une classe restreinte de problèmes de satisfaction de contraintes qui a été partiellement classifiée par Feder. Nous étudions aussi quelques variantes autour de ce problème et fournissons un algorithme pour le résoudre dans un cas restreint.

1.3 Organisation

Dans le chapitre 3, nous commençons par introduire les différentes notions que nous utiliserons et explorerons tout au long de cette thèse. Notre travail repose sur plusieurs domaines mathématiques, que ce soit de la géométrie, de la topologie ou des mathématiques discrètes, et le lecteur est encouragé à survoler les sujets avec lesquels il est familier et à se focaliser sur les autres.

Le chapitre 4 dresse un état de l'art de différentes facettes de la topologie algorithmique. Il est pensé comme une introduction approfondie aux problèmes que nous étudions dans cette thèse, ainsi que plus largement au domaine de recherche auquel ils se rattachent. Il est divisé en deux parties, suivant que l'on insiste plus sur l'aspect algorithmique ou topologique.

Les chapitres 5, 6 et 7 exposent ensuite respectivement nos résultats sur les isotopies de graphes, les décompositions de surfaces et les surfaces normales immergées. Ils sont indépendants. Les résultats que nous y présentons ont été respectivement publiés dans les articles [A, B, C]. Par rapport à ces articles, nous avons changé la présentation en de multiples occasions pour simplifier la lecture, et nous renvoyons aux débuts des chapitres pour discuter précisément des modifications réalisées.

Enfin, nous concluons au chapitre 8 en récapitulant nos résultats principaux et en exposant de futures directions de recherche en lien plus ou moins direct avec notre travail.

Pour alléger la lecture du chapitre 5, quelques arguments n'ont été qu'ébauchés et les preuves complètes ont été renvoyées à l'annexe A.

CHAPTER 2

Introduction

In 1860, it was widely believed that a mysterious substance called æther constituted the fabric of space and pervaded every possible structure in the universe. To explain the different shapes we could observe in nature, Lord Kelvin advanced the *Vortex Theory*, postulating that atoms were merely vortices of æther, and therefore that chemical elements were knots or links, with differences in the topology of the knots impacting the physical properties of the corresponding elements. This theory had a strong appeal since it could explain some of the discrete features of matter that could already be observed, such as the wavelengths at which atoms emit and absorb light. The famous Sodium doublet for example could be hypothetically explained by its intrinsic structure as a Hopf link, i.e., two intertwined circles, as in Figure 2.1.

Soon enough though, the Michelson-Morley experiment shattered this model by displaying strong evidence that there was no æther at all. But the damage had already been done. Peter Guthrie Tait, trying to investigate the vortex theory, established the first tables classifying knots as well as the first conjectures of the nascent corresponding *knot theory*, which would then be developed by countless mathematicians.

Fast forward to modern times, knot theory, despite its initial failure to describe matter, appears in a wide variety of domains: DNA strands get knotted and unknotted through the action of enzymes [240], much smaller linked and knotted molecules have been synthe-

FIGURE 2.1: Could the emission doublet of the Sodium be explained by its intrinsic nature as a Hopf link?

sized [251], and the reader is probably familiar with the painful untying one has to undergo when pulling headphones out of a tight pocket.

Zooming out a little bit, knot theory is but a subfield of *topology*, which is the mathematical study of spaces and shapes. Topology is a mature mathematical field, with some of the foundational ideas, like Euler's formula, dating back to the 17th century with Descartes. It is also a very active one, as is showcased by the recent resolution of the Poincaré Conjecture by Perelman [195, 196].

However, a new focus has appeared in the last decades, with the revolution accompanying information technologies: we are not satisfied anymore with the plain *understanding* of a topological object – which is what mathematics traditionally provide – we also want to *compute* its topological properties. In the case of knot theory, the following question immediately arises from this perspective: how can one test algorithmically if a knot is knotted? This question turns out to be fundamental and exemplifies the tools and the aim of this approach. Devising systematic algorithms to tackle topological questions is the topic of *computational topology*, a blooming area of research aiming both at revisiting old problems from this fresh perspective and refueling topology with intriguing new research directions.

2.1 Approach

This thesis investigates some low-dimensional topological problems from the point of view of a theoretical computer scientist. This perspective leads us to combine tools from various subfields at the interface of mathematics and computer science.

Topology. Topology, and especially algebraic topology, revolves around building tools to acquire a good understanding of shapes and spaces. This study may take various paths, like recombining shapes into simpler pieces (*surgery theory*), investigating the subspaces contained therein (*homotopy* and *homology theory*), or looking at how the topological features impact the metric properties of the space (*geometrization*); all of these approaches being naturally highly intertwined. These techniques, brought together towards algorithms on the *classification* and the *decomposition* of low-dimensional shapes, form the backbone of this thesis.

There is a very different flavor between working in low (two, three, or four) and high dimensions. In the latter case, the additional space allows to use several tools to move spaces around. From a mathematical point of view, this makes some problems much easier in higher dimensions, as is exemplified by the Poincaré conjecture: it resisted much longer in three dimensions than in dimensions larger than four, for which Smale provided a solution as early as 1961. From a computational point of view however, this additional space is also a curse: as we will survey in Chapter 4, many problems become undecidable as soon as we hit dimension four, removing any hope to design algorithms for numerous questions.

Therefore, in this thesis, we focus on spaces in dimensions two and three, that is, mostly on surfaces and 3-manifolds (spaces looking locally like the ambient Euclidean space), which are host to plenty of algorithmic questions both tractable and interesting.

Discrete structures. An important feature of our approach is the heavy focus on discrete structures. This might seem contradictory at first, since continuity is one of the most fundamental notions of topology. Notwithstanding, the point is not to study the topology of truly discrete sets like points clouds, but rather to find ways to encode topological structures in finite space, and manipulate them algorithmically. For example, instead of thinking of surfaces as arbitrary sub-manifolds of the usual Euclidean space, we will prefer to triangulate them, and manipulate them as linear pieces glued together [1]. This gives a strong combinatorial flavor to our work, and unveils important connections with classical topics in discrete mathematics, and, in particular, graph theory.

Algorithms. Another key aspect of this work is the focus on algorithms. Our goal is to be able to carry out efficient computations on topological spaces. The theoretical viewpoint puts an emphasis on proving the correctness of the algorithms we design, as well as analyzing their complexity in the most accurate way possible, generally from an asymptotic angle. The dual research direction is to establish the hardness of some problems, i.e., that no efficient algorithm exists to carry out specific tasks. In both cases, we need to combine tools from topology and from computer science to achieve our objectives, which places this work at the interface of both fields.

We remark that these algorithms stay in the *theoretical* realm and have not been implemented, as we are working at the level of foundations. However, this groundwork is ripe to be exploited for further applications where one needs to deal with shapes, such as computer graphics, geographic information systems, computer aided design, robotics, or bioinformatics. We will also survey some more surprising applications of topological ingredients in Chapter 4. Developing careful and robust implementations of our results in widely used libraries would be an interesting side of this work – though not one we chose to focus on for now.

2.2 Contributions of this thesis

This work revisits some standard topological concepts from a computational viewpoint. The first aspect of our work deals with deformations: given two topological objects in a space, how to determine whether one can be continuously deformed into the other? Solving this kind of problems allows in turn to classify these objects: deformation gives a precise

1. We do not claim novelty on this approach: modeling spaces as simplicial or CW-complexes is standard in algebraic topology – but it fits very well within our framework.

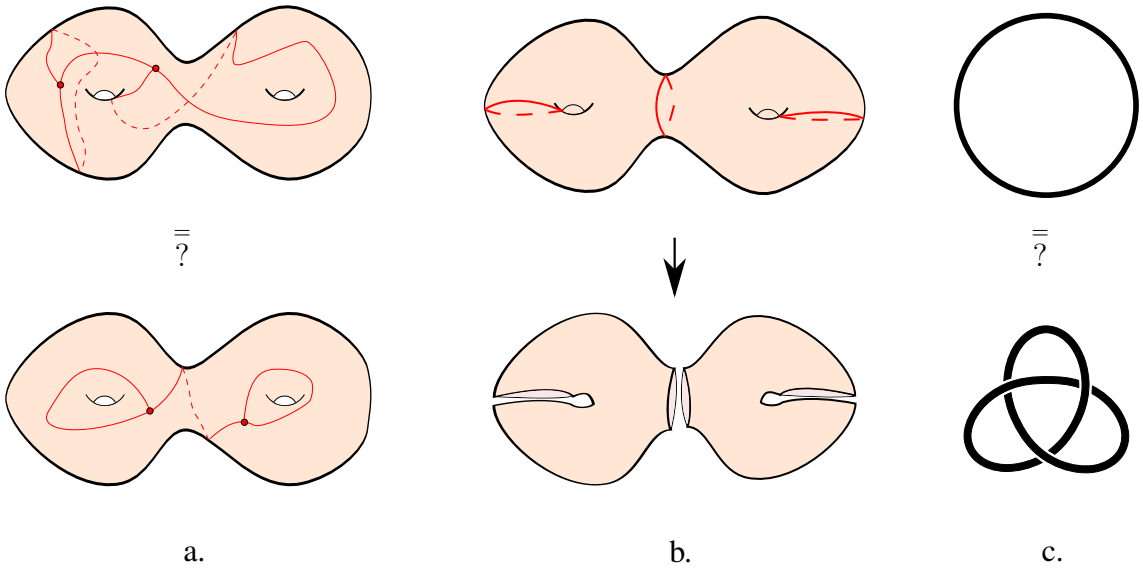


FIGURE 2.2: a. Is there a continuous deformation between the two graphs embedded on this double torus?
b. How can one efficiently cut a surface into smaller pieces?
c. Are these two knots the same, i.e., is there a continuous deformation between one and the other?

notion of equivalence, and being able to test for deformation allows to classify objects into collections sharing similar topological properties. We investigate two different occurrences of these problems. On one hand we study graphs embedded on surfaces, where we provide efficient algorithms to solve this question. On the other hand we look at embeddings of the circle in $\mathbb{R}^3$, where this leads to the aforementioned unknot problem. Although we do not obtain an efficient algorithm in this case – this is a prominent open question of the field – we prove the hardness of a natural approach to attack it. The second theme of this thesis revolves around decompositions: how can one efficiently simplify a topological space by cutting it into smaller pieces? In the case of surfaces, we provide new algorithms and bounds for several instances of this problem.

Testing graph isotopy on surfaces. The *graph isotopy problem* asks the following question: given a graph embedded in two different ways on a given surface, is there an *isotopy*, i.e., a continuous deformation, between them? From a mathematical point of view, it is a fundamental question, since isotopy is a very precise relation on embedded graphs, corresponding well to the intuitive notion of being alike. Furthermore, embedded graphs are pervasive structures in low-dimensional topology, for example through their connections with train tracks, which we will introduce in Chapter 4, or Heegaard diagrams, a fundamental tool in 3-manifold topology [131]. Testing isotopy also has more practical purposes, as

it is relevant for many applications: for example road maps can be accurately described using embedded graphs, and testing isotopy allows to certify whether two maps feature the same topological properties (is this house on the right side of the road?). Other applications abound in morphing or texture mapping.

Building upon earlier works both computational (algorithms to test the *homotopy* of embedded curves) and mathematical (Ladegaillerie provided a combinatorial criterion to test the isotopy of graphs), we devise an efficient algorithm to test efficiently whether two graphs are isotopic. One of the flavors of working in low dimensions is that there is a strong interplay between geometry and topology, and our techniques rely on hyperbolic geometry to investigate the combinatorics of the isotopy relation, as well as to prove the correctness of our approach.

Discrete systolic inequalities and decompositions of triangulated surfaces. We then shift our focus from classification to decomposition. Decomposing a surface is the process of cutting it into smaller and simpler pieces, and this constitutes a fundamental step for multiple purposes, as it relates a surface to the plane, of which topological properties are very well understood. In turn, this allows for example to design algorithms for graphs embedded on surfaces by relating them to planar graphs in a controlled way. Similarly, surface decompositions are also relevant for surface meshing, parameterization, and many other applications – we refer to the introduction of the corresponding Chapter 6.

When cutting a surface to simplify it, it is key to control the length of the cut: cutting along a short curve will limit the modifications induced by this decomposition. It is therefore a natural problem to add a geometric constraint to the topological decomposition and to try to find the shortest way to cut a discrete surface to simplify it. There exist several different cut patterns, and we investigate three of them, namely *systoles*, *pants decompositions* and *cut-graphs*. We first improve the best known bounds on the lengths of the systoles, which answers a longstanding conjecture of Przytycka and Przytycki from 1993. We also provide a new algorithm to compute a short pants decomposition and identify a new obstruction to short cut-graphs. This line of work relies on an explicit connection with the continuous case, which leads to the heavily investigated theory of *systolic geometry*, as well as the use of a random model for surfaces.

Immersed normal surfaces. Finally, we move up one dimension. The iconic problem of computational topology in 3 dimensions is the famous *unknot problem* that we hinted at in this introduction: How can one test whether an embedded circle in Euclidean space can be isotoped to a planar circle, or in layman's terms whether it can be unknotted? It is one of the few algorithmic problems for which no polynomial-time algorithm is known, but no proof of hardness either. Many other 3-dimensional problems are also poorly understood and are the object of much scrutiny. For example, while testing whether two discrete surfaces are topologically the same is very easy, the analogous problem in 3 dimensions (where spaces

are modeled by gluing tetrahedra together) turns out to be extremely delicate.

One of the main ways to study a topological space is to investigate the subspaces it contains. Similarly, the best algorithms for 3-dimensional problems all rely on *normal surfaces*, which provide a compact algebraic structure to represent and manipulate surfaces embedded in a 3-dimensional space – they are now a ubiquitous tool in 3-dimensional computational topology. In this work, we investigate a natural generalization of normal surfaces, where we allow them to be immersed instead of embedded. This naturally leads to a problem about detecting singularities, which we show to be NP-hard, that is, no polynomial-time algorithm is to be expected for this problem – this is especially relevant in this field since very few hardness results are known. Our reduction works by establishing a connection with a restricted class of constraint satisfaction problems which has been partially classified by Feder. We also investigate some variants of this problem and provide a polynomial-time algorithm for a restricted case.

2.3 Organization

In Chapter 3, we begin by introducing the various notions we will be using and exploring throughout this thesis. Our work uses tools from various mathematical subfields, be it geometry, topology, or discrete mathematics, and the reader is encouraged to only skim over the topics he is familiar with and focus on the other techniques.

Chapter 4 is a survey on various facets of computational topology. It is intended as a thorough introduction to the problems that are investigated in this thesis, as well as more generally to the research area they belong to. It is split into two different parts, depending on whether the focus is more on the algorithmic or the topological side of computational topology.

Chapters 5, 6 and 7 then respectively expose our results on graph isotopies, decompositions of surfaces and immersed normal surfaces. They can be read independently. The results presented therein have been respectively published in the articles [A, B, C]. Compared to these articles, we have changed the presentation in many occasions to ease the reading, unify or draw a better global picture, the precise modifications being described at the start of every chapter.

Finally, to conclude in Chapter 8, we recapitulate our main results and expose some future avenues of research arising from our work.

To lighten the reading of Chapter 5, some of the arguments there have only been sketched and the full proofs have been deferred to Appendix A.

CHAPTER 3

Preliminaries

In this chapter, we introduce the main notions that will be used throughout the thesis. By essence, the exposition will be rather formal and rigorous, the reader might therefore want to skip ahead on the first pass, and come back here to grasp the necessary concepts when needed.

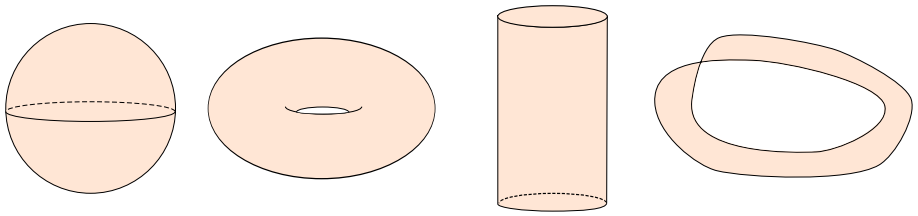
3.1 Geometric topology

Topology, and in particular algebraic topology, is both the main mathematical inspiration and tool of this thesis. Although this section introduces all the necessary notions, getting used to them might require a more lengthy exposition, which the reader can find in the standard textbook of Hatcher [129], or in the one of Stillwell [238] for a more combinatorial perspective.

3.1.1 Topology of surfaces

A *surface* S is a compact topological space such that an open neighborhood of every point is locally homeomorphic to the plane or a closed half-plane. The points with boundary homeomorphic to a half plane form the *boundary* of the surface. Usual examples of surfaces include the sphere, the torus, the cylinder and the Möbius strip, which are pictured in Figure 3.1. A surface is *non-orientable* if it contains a sub-surface homeomorphic to a Möbius strip, and *orientable* otherwise.

Throughout this thesis, we only consider the surfaces up to homeomorphism, and our denominations use this implicitly. For example, a *disk* means a surface homeomorphic to the usual disk $\mathbb{B}^2 \subseteq \mathbb{R}^2$. This is motivated by the following fundamental classification result about surfaces.



a. b. c. d.

FIGURE 3.1: A sphere, a torus, a cylinder and a Möbius strip.

Theorem 3.1.1 (Classification of surfaces)**.** *Every connected surface is homeomorphic to either*

- *The orientable surface of genus $g \geq 0$ with $b \geq 0$ boundaries, which is obtained by gluing g handles to a sphere and removing b open disks with disjoint closures.*
- *The non-orientable surface of genus $g \geq 0$ with $b \geq 0$ boundaries, which is obtained by removing g open disks from a sphere, attaching g Möbius strips to the resulting boundaries, and removing b open disks with disjoint closure.*

Therefore, connected surfaces are entirely described by their genus g, the number of boundaries b and their orientability. Throughout this thesis, we will, unless specified otherwise, only consider connected and orientable surfaces, which we denote by $S_{g,b}$. As an example, the surfaces in Figure 3.1 are respectively $S_{0,0}, S_{1,0}, S_{0,2}$ and the non-orientable surface with one boundary and no genus. Similarly, Figure 3.2 pictures the surface of genus 2 and no boundaries, also known as a double torus. The *Euler characteristic* of an orientable surface is defined by $\chi(S_{g,b}) = 2 - 2g - b$.

In many situations, the surfaces are the host spaces, and we are actually interested in the elements that live on the surfaces. A *path* on a surface S is a continuous map $p : [0; 1] \to S$. A *loop* is a path whose endpoints coincide, they form the *basepoint* of the loop. An *arc* is a path whose endpoints lie on the boundary, and a *cycle* is a continuous map $\gamma : S^1 \to S$, where S^1 is the usual circle. Finally, a *curve* denotes either a cycle or a path. A curve is *simple* if it is injective.

We now introduce the essential concept of *cutting* a surface S along a simple curve γ. It corresponds to the intuitive definition of cutting along γ with a pair of scissors: We say that S_γ has been obtained by *cutting* S along a simple curve γ if S_γ is equipped with a homeomorphism h between two of its boundary components such that:

1. The quotient space $S_\gamma/(h(x) \sim x)$ is a surface homeomorphic to S,
2. The image of these distinguished boundary components under this quotient map is γ.

Proving that such a surface S_γ always exists is easy when S and γ are smooth or piecewise-linear. It is however a bit more intricate in the purely topological case, we refer to Epstein [81, Appendix] to see how to reduce the problem to the piecewise-linear setting.

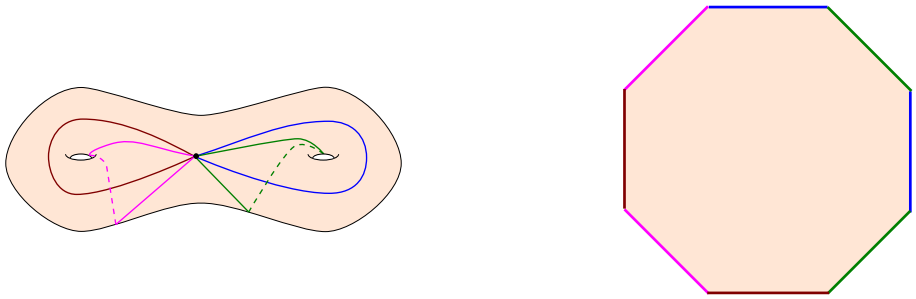
In contrast to surfaces that are mainly considered up to homeomorphism, we need a finer notion to distinguish between curves, for example because all the simple paths are homeomorphic. The notion of homotopy formalizes the intuitive idea of a continous deformation. The paths p and p' are *homotopic* if there is a continuous map $h : [0;1] \times [0;1] \to S$ such that $h(0,t) = p(t)$, $h(1,t) = p'(t)$ for all $t \in [0;1]$, and $h(\cdot,0)$ and $h(\cdot,1)$ are constant maps. Similarly, two cycles γ and γ' are *(freely) homotopic* if there is a continuous map $h : [0;1] \times S^1 \to S$ such that $h(0,t) = p(t)$, $h(1,t) = p'(t)$ for all $t \in S^1$. A cycle is *contractible* if it is homotopic to a constant map, and *essential* if it is not homotopic to a point or a boundary component.

This definition of homotopy allows the curves we consider to self-intersect during the deformation. Disallowing it leads to the neighborly concept of *isotopy*: The paths p and p' are *isotopic* if there is a continuous map $h : [0;1] \times [0;1] \to S$ such that $h(0,t) = p(t)$, $h(1,t) = p'(t)$ for all $t \in [0;1]$, $h(\cdot,0)$ and $h(\cdot,1)$ are constant maps, and for every t, $h(t,\cdot)$ is a simple path. One defines similarly isotopic cycles. One can also define an *ambient isotopy* $i : S \to S$, which is a homeomorphism homotopic to the identity: there exists a continuous family of homeomorphisms $h : [0,1] \times S$ such that $h(0)$ is the identity and $h(1) = i$. Then two curves are ambient isotopic if there exists an ambient isotopy sending one to the other. Using vector fields, it can be shown that these two notions of isotopy are equivalent for curves disjoint from the boundary, see Farb and Margalit [96, Proposition 1.11] or Hirsch [138, Theorem 1.3]. See also Appendix A where we will investigate how to extend isotopies of graphs to ambient isotopies.

Furthermore, in many circumstances, the following theorem of Epstein allows us to switch between isotopy and homotopy at will.

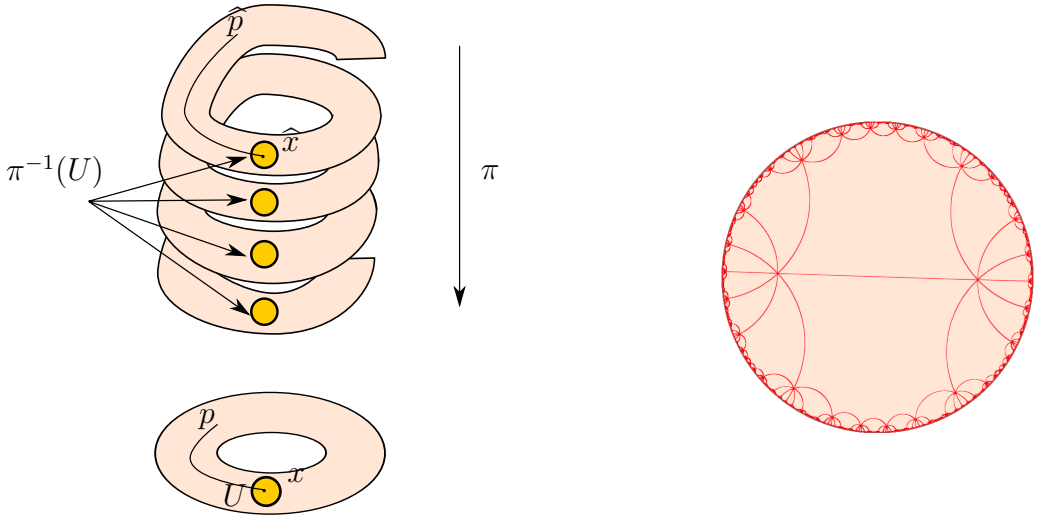
Theorem 3.1.2 ([81])**.** *Let γ and γ' be two essential cycles. Then γ and γ' are isotopic if and only if they are homotopic.*

Loops can easily be *inverted* and *concatenated*. If γ is a loop with basepoint x, the inverse loop γ^{-1} is defined by the map $t \mapsto \gamma(1-t)$, and for γ_1 and γ_2 two loops with basepoint x, the concatenation $\gamma_1 \cdot \gamma_2$ is defined by the map $t \mapsto \gamma_1(2t)$ for $t \in [0,1/2]$, and $t \mapsto \gamma_2(2(t-1/2))$ for $t \in [1/2,1]$. With these operations, the set of homotopy classes of loops with fixed basepoint $x \in S$ constitutes a group called the *fundamental group* of S and denoted by $\pi_1(S,x)$. The homotopy class of a loop c in $\pi_1(S,x)$ is denoted by $[c]$. It is easy to see that for different $x \in S$, the groups $\pi_1(S,x)$ are all isomorphic, and when we are only interested in the algebraic structure of the fundamental group, we denote it simply by $\pi_1(S)$. By definition, a homotopy of a loop fixes its basepoint. Forgetting this basepoint x amounts to seeing loops as cycles, and one can show that two such loops – or cycles with a point x in common – c and c' are freely homotopic if and only if they are conjugated in the fundamental group $\pi_1(S,x)$. The fundamental group of S_g can be defined



a. b.

FIGURE 3.2: a. A canonical system of loops on a genus 2 surface. b. The octagon we obtain after cutting along this system of loops.



a. b.

FIGURE 3.3: a. A covering of the annulus, and one lift of a path p b. The universal cover of a genus 2 surface is an octagonal tiling [1]of the hyperbolic plane $\mathbb{H}^2$.

using $2g$ generators and one relation, giving the presentation $\pi_1(S) = \{a_1, b_1 \dots a_{g,b_g} \mid [a_1, b_1] \dots [a_g, b_g]\}$ where $[x, y]$ denotes a commutator. These generators, called *canonical*, are pictured in Figure 3.2. One intuition for this group presentation is that when one cuts along the canonical generators, one obtains a $4g$-gon, and one can read the unique relation out of the sequence of the boundaries.

There is a strong connection between the fundamental group of a surface and its covering spaces, that we will only touch upon. A *covering space* of S is a space $\widehat{S}$ together with a continuous surjective map $\pi : \widehat{S} \to S$ such that for every $x \in S$, there exists an open neighborhood U of x such that $\pi^{-1}(U)$ is a disjoint union of homeomorphic copies of U. The reader scared by this definition should look at the example of the annulus in Figure 3.3.

A covering space allows to *lift* a path: if p is a path on S such that $p(0) = x = \pi(\widehat{x})$ for some $\widehat{x} \in \widehat{S}$, there is a unique path $\widehat{p}$ on $\widehat{S}$ starting at $\widehat{x}$ such that $p = \pi \circ \widehat{p}$, once again this is pictured in Figure 3.3a. Two paths are homotopic on S if and only if they have homotopic lifts. Similarly, cycles have a unique lift passing through a given point in a cover, but the lift might be a path instead of a cycle. Homotopic cycles have homotopic lifts, and furthermore, any lift of a contractible cycle is a cycle.

Every surface has a unique covering space $\widetilde{S}$ in which every cycle is contractible, it is called the *universal cover* of S. One convenient property of universal covers is that a cycle on S is contractible if and only if its lifts in the universal cover are contractible as well. Figure 3.3b. shows the universal cover of a genus 2 surface, where the canonical system of loops induces an octagonal tiling on this cover (see also the next paragraph for some introduction to the hyperbolic disk and the Poincaré model).

3.1.2 Geometry of surfaces

In this section, we introduce the different notions pertaining to the geometry of surfaces, and more precisely to the metric structure one can endow a surface with. A good general reference on Riemannian geometry is the book of Do Carmo [73].

A *Riemannian metric* on a surface S is an inner product g_x on the tangent spaces T_xS that varies smoothly when x moves on S. Although we will only consider the corresponding *Riemannian surfaces* (not to be confused with Riemann surfaces), the concepts introduced in this section apply in any dimension to other *manifolds*. To this inner product corresponds an Euclidean norm $||\cdot||$ on T_xS, and this naturally defines a way to measure the length of the curves drawn on the surface. For example if p is a path on S, its length is defined by $\ell(p) = \int_0^1 ||p'(t)||$. This then gives a metric structure on S, where the distance between two points x and y in S is the infimum of the lengths of the paths joining x to y. A *geodesic* on a Riemannian surface is a curve that is locally minimal, i.e., such that any local perturbation increases its length. The reader should be wary that geodesics are in general *not* shortest paths, although the converse is true – this situation is analogous to the one of local versus global minima of a function.

The easiest example of a Riemannian metric is the one of a surface embedded in $\mathbb{R}^3$, where the Euclidean structure of $\mathbb{R}^3$ naturally defines the inner product, and the distance is the one naturally associated with the embedding. It is an intricate theorem of Nash [125] that any Riemannian metric can be obtained this way. Although this point of view is enlightening for the intuition, it is generally more convenient to work with the intrinsic definition.

To a Riemannian surface S is naturally associated a notion of curvature. Recall that the curvature of a planar curve measures quantitatively how far from being a straight line it is. This can be generalized to surfaces in the following way. For a point x in the interior of S,

1. The hyperbolic tiling has been generated using Dmitry Bryant's software *Tessellation*.

any curve γ_x going through x has a *normal curvature* at that point, which is the curvature of the curve γ_x projected onto the plane containing the tangent of γ_x at x and the normal of the surface at x. The principal curvatures k_1 and k_2 on x are the infimum and the supremum of all the normal curvatures at x of possible curves going through x. The *curvature* of S at x, denoted by K_x is the product $k_1 k_2$.

A fundamental insight lying at the origin of geometric topology is that there is a strong interplay between the metric and topological properties of surfaces, and manifolds in general. An easy illustration of this is given by the Gauss-Bonnet theorem, linking the curvature of a surface to its Euler characteristic.

Theorem 3.1.3 (Gauss-Bonnet Theorem)**.** *Let S be a surface without boundary, then*

$$\int_S K dA = 2\pi\chi(S).$$

Amongst surfaces, some of particular interest are the surfaces with constant curvature, because they exhibit the richest symmetries. Up to rescaling the metric, this constant curvature can be $1, 0$ or -1, which correspond respectively to *spherical*, *Euclidean* and *hyperbolic* manifolds. Following the Gauss-Bonnet theorem, we see that spherical, respectively euclidean and hyperbolic structures can only exist on surfaces with a respectively positive, zero or negative Euler characteristic – variants of the Gauss-Bonnet theorem provide the same classification for surfaces with boundary. Since the curvature is only defined locally, a cover of a surface with a Riemannian metric is naturally endowed with a Riemannian metric as well. In the spherical, Euclidean and hyperbolic case, the universal covers are respectively the usual sphere $\mathbb{S}^2$, the Euclidean plane $\mathbb{E}^2$ and the hyperbolic plane $\mathbb{H}^2$. The last one may be a newcomer to some: an easy way to visualize it is to picture its geometry using the Poincaré disk model. In this geometry, the hyperbolic plane is pictured in the open disk $\mathbb{D}^2$, and a geodesic between two points is a portion of a (usual) circle meeting the boundary circle with straight angles. For example, the lines in Figure 3.3b. are geodesics on the Poincaré disk, which together form an octagonal tiling. For a further introduction to hyperbolic geometry, we refer to the book [42] or the beginning of the lecture notes of Thurston [246].

In the other direction, it is easy to see that the corresponding surfaces can be endowed with a constant curvature metric. For example, for hyperbolic surfaces, one can start with a geodesic $4g$-gon with sides of equal length and interior angle sum 2π in the hyperbolic plane $\mathbb{H}^2$ and glue opposing sides to get a hyperbolic metric on S_g.

3.1.3 3-manifolds and knots

Similarly to surfaces, a *3-manifold* is a compact topological space such that every point is locally homeomorphic to $\mathbb{R}^3$ or to the closed half space $\{x, y, z \mid x, y, z \in \mathbb{R}, z \geq 0\}$. The notions of homotopy, fundamental group and ambient isotopy that we introduced for

surfaces readily generalize to 3-manifolds (and any topological space). We will describe 3-manifolds using *triangulations* – or sometimes *tetrahedrizations* when we want to distinguish them from their 2-dimensional analogue: A triangulation T is a topological space obtained from a disjoint set of t tetrahedra $T = (T_1, \ldots, T_t)$ by (combinatorially) gluing some pairs of two-dimensional faces of these tetrahedra; a gluing between two faces is specified by a bijection from the vertex set of the first face to the vertex set of the second face. As a result of these gluing, edges and vertices of tetrahedra are also identified; it is also allowed to glue two zero-, one-, or two-dimensional faces of the same tetrahedron. A *face* of a triangulation T is a two-dimensional simplex, incident to one or two tetrahedra in T. The *link* of a vertex v in a triangulation is the surface obtained as the frontier of a small regular neighbourhood of v.

Since in a 3-manifold, the neighborhood of every point has to be an open ball or a half ball, the following conditions are necessary for a triangulation T to be a 3-manifold (possibly with boundary):

1. Each vertex has a neighborhood homeomorphic to $\mathbb{R}^3$ or to the closed half-space;

2. After the gluings, no edge is identified to itself in the reverse orientation.

Conversely, it is known [189] that any 3-manifold M is the underlying space of such a triangulation. Henceforth, T denotes a triangulation of a 3-manifold M. A *normal isotopy* is an ambient isotopy of M that is fixed on the 2-skeleton of T.

The mathematical theory of knots, as pictured for example in the classic textbook of Burde and Zieschang [28], is a daunting collection of intricate invariants. However, we will use very little of this theory to study the relevant algorithmic problems, and everything we need is introduced here and in Chapter 7.

A *knot* is a an embedding of the circle $\mathbb{S}^1$ in $\mathbb{R}^3$. Knots are generally considered up to deformations, that is, ambient isotopies of $\mathbb{R}^3$, and we will say that two knots are *equivalent* is they are ambient isotopic, or by a slight abuse of language simply consider them to be the same knot. The *unknot* is the usual embedding of a circle into $\mathbb{R}^3$. For compacity purposes, and since it does not change the definition nor the equivalence classes, knots are generally considered in the one-point compactification of R^3, that is, the 3-dimensional sphere S^3.

A *polygonal knot* is a knot such that the embedding is realized by a finite set of line segments. A *tame knot* is a knot equivalent to a polygonal knot, and since the focus on this thesis is on discrete and computational mathematics, all the knots we will consider will be tame knots from now on.

A theorem of Seifert [232] states that if K is a knot, there exists a surface with boundary S embedded in $\mathbb{S}^3$ such that $\partial S = K$. This surface is called the *Seifert surface* of the knot, and the proof of its existence is algorithmic. It is easy to see that a knot K is the unknot if and only if it bounds a disk, then called its *Seifert disk*: in one direction the ambient isotopy preserves the topological nature of the disk, and in the other the disk provides an isotopy

untying the knot. The *genus* of a knot is the genus of its Seifert surface with the smallest genus. By the previous observation, the only knot of genus 0 is the unknot.

Knots are often studied via their *complement*. By taking a small open neighborhood U of a knot K and considering $\mathbb{S}^3 \setminus U$, one obtains a 3-manifold with boundary. This 3-manifold is a solid torus $\mathbb{D}^2 \times \mathbb{S}^1$ if and only if the original knot is the unknot. Similarly, for any knot K, the topology of K is entirely described by the topology of its complement in $\mathbb{S}^3$. This key result is the Gordon-Luecke theorem [111], generally summed up as "The knots are determined by their complements". This justifies the study of the complement of a knot as a means to study the knot itself.

3.2 Embedded graphs and maps

3.2.1 Embedded graphs

We generally use the classical terminology for graphs, and refer to Diestel [72] for background on graph theory. By a slight abuse of language, most of the graphs we consider will actually be multi-graphs, i.e. multiple edges and loops are allowed. When this is not the case, we will say that the graph is simple. The vertices and the edges will always be denoted by V and E, and the complete graph on n vertices is denoted by K_n while the complete bipartite graph on n and m vertices by $K_{n,m}$.

A graph $G = (V, E)$ can naturally be seen as a topological space, for example by representing it in $\mathbb{R}^3$ and taking the induced topology. An *embedding* of a graph $G = (V, E)$ on a surface S is a continuous injective mapping from G, considered as a topological space, to S. Therefore, it maps vertices to distinct points and edges to disjoint paths intersecting only at their endpoints, and we will generally identify a graph with its embedding. The specific case where $S = \mathbb{R}^2$ or equivalently $S = \mathbb{S}^2$ gives rise to *planar graphs*. In more generality, the study of embedded graphs is the starting point of *topological graph theory*, and we refer to the book of Mohar and Thomassen [188] for an extensive study of their properties. In this section, we only recall the basic notions that we will use. The *genus* of a graph is the smallest possible genus of a surface on which it can be embedded. As an illustration, the complete graph K_n has genus $\Theta(n^2)$.

For a graph G embedded on a surface S, a *face* is a connected component of the complement of the image of G on S. The set of faces is denoted by $F(G)$, or F when there is no ambiguity. An embedding is *cellular* if every face is homeomorphic to a disk. The *degree* of a face is the number of edges adjacent to it, and a *triangulation* is a cellularly embedded graph where every face has degree exactly three. Similarly to the case of 3-manifolds mentioned before, by a result of Rado [206], every surface can be triangulated (this is not hard

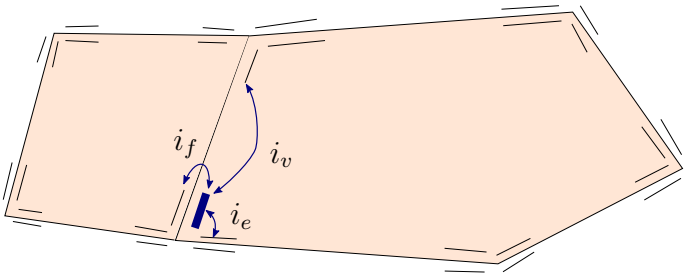


FIGURE 3.4: A picture of a combinatorial map, with the three involutions i_v, i_e and i_f for a flag (in bold).

to show, but there are pitfalls hiding behind the obvious feeling of this result; for example *not* every 4-manifold can be triangulated [207]).

It is an old observation that when a graph G is cellularly embedded, it satisfies Euler's formula $|V| - |E| + |F| = \chi(S)$. In particular, if G is a simple graph (i.e, with no loops or multiple edges), it satisfies $|E| = |V| + O(g)$. This justifies the somewhat abusive expression of the *complexity of an embedded graph* that will stand for $O(|E|)$: it also controls the genus, the number of vertices and the number of faces.

3.2.2 Combinatorial maps

When a graph G is cellularly embedded on a surface S, the precise location of the vertices and the edges is unimportant: for most purposes, we only care about the "look" of the graph. This is encoded in the *combinatorial map* of the embedding which essentially describes the circular order of the edges around each vertex of the surfaces. Then gluing disks on the faces allows to recover an embedding of G on S. One also encounters the morally equivalent notions of *rotation systems* [188], *fat graphs* [117], or *ribbon graphs* [78].

Many data structures exist to represent combinatorial maps [79, 153, 173], we describe here a variant of the *gem representation*. The gem representation stores four *flags* per edge; intuitively, if the edge is oriented, two flags close to its head, one to its left and one to its right, and similarly for the tail. More formally, a flag represents an incidence between a vertex, an edge, and a face of the embedding. Three involutions allow to move from a flag to an "incident" flag in the graph: The first one, i_v, keeps the same edge-face incidence and moves to the opposite vertex; the second one, i_e, keeps the same vertex-face incidence and moves to the opposite edge; the last one, i_f, keeps the same vertex-edge incidence and moves to the opposite face. Also, each flag has a pointer to the underlying vertex, edge, and face of G. This data structure is pictured in Figure 3.4

The model of computation that will be always used implicitly in this thesis is the real RAM mode [3], as is customary in computational geometry. Therefore, pointers are assumed to be stored in constant space and accessed in constant time. This shows that our

data structure has complexity linear in the number of edges of the graphs. Furthermore, any reasonable operation one would like to perform on an embedded graph can be carried out efficiently (i.e. in linear time) with this data structure, like traversing the edges adjacent to a given face or even traverse all the vertices, edges and faces of the graph.

It is natural to wonder when two combinatorial maps have rigorously the same properties, and then what can be said about two embedded graphs with the same combinatorial maps. This leads to the following definition. Two combinatorial maps of the same graph G are *isomorphic* if there is a bijection φ between their sets of flags F_1 and F_2, commuting with the three involutions, and such that the underlying vertex (resp., edge) of a flag f in F_1 is the same as that of $\varphi(f)$; such a bijection is called a *map isomorphism*. One can easily test in linear time whether two combinatorial maps are isomorphic.

What information does the combinatorial map reveal about the embedding of a graph? By the following lemma, it turns out to exactly coincide with the homeomorphism class of the embedded graph.

Lemma 3.2.1. *The combinatorial maps representing two cellular embeddings G_1 and G_2 of the same abstract graph G on S are isomorphic if and only if there exists a homeomorphism of S mapping G_1 to G_2.*

Proof If there exists a homeomorphism of S mapping G_1 to G_2, then obviously the combinatorial maps of G_1 and G_2 are isomorphic. Conversely, any isomorphism of combinatorial maps extends naturally to a homeomorphism between the tubular neighborhoods of the graphs. (Informally, one can build a disk for each vertex and a strip for each edge of G, and attach these disks and strips as prescribed by the combinatorial map of G_1, so that their union forms a tubular neighborhood of G_1. And one can similarly do the same for G_2, the gluings being combinatorially the same since the combinatorial maps are isomorphic.) This homeomorphism between the two tubular neighborhoods extends by radial extension in every disk, and since all the faces are disks, we obtain therefore a homeomorphism of the whole surface S. □

3.2.3 Combinatorial and cross-metric surfaces

We now present the models that we will use to describe surfaces and objects embedded on them. Both Chapters 5 and 6 deal with cycles or graphs embedded on surfaces, and they occur either as input or output of the algorithms presented therein. Therefore, we need to explain how these objects are stored and manipulated algorithmically. For embedded graphs, and in particular for cellularly embedded graphs, it might be tempting to just use the combinatorial map: indeed we just showed that it characterizes the graph embedding up to a homeomorphism of the surface. But homeomorphism is quite a coarse measure of similarity for embedded graphs, and we will want to distinguish homeomorphic but non isotopic graphs, for example in Chapter 5. Even more, the homeomorphism class of an

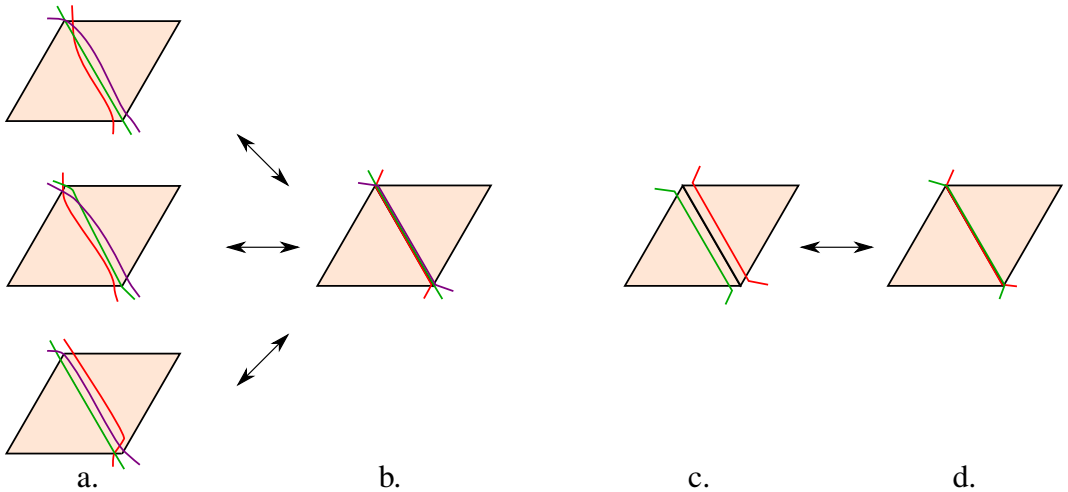


FIGURE 3.5: First problem: the three different configurations in a. will be represented in the same way b. on a combinatorial map. Second problem: the two close curves in c. will be represented by the touching curves in d., although cutting along these curves does not yield the same topological space in both cases.

embedded simple cycle gives little to no information on this cycle, for example because all the nonseparating simple cycles on a surface are homeomorphic (this is a version of the "change of coordinates"principle, see Farb and Margalit [96, Section 1.3.2]). Therefore, we will use a more precise description of discrete surfaces, by the means of an embedded cellular graph. Depending on the use, this setting will be declined in two flavors which are dual to each other: *combinatorial* or *cross-metric* surfaces.

3.2.3.1 Combinatorial surfaces

Combinatorial surfaces are perhaps the more natural description one can think of. A combinatorial surface is a surface S coupled with a cellularly embedded graph G with nonnegative weights on its edges. The curves we consider in this model are the walks on the graph G, and the length of the curve is simply the sum of the length of the edges that the walk follows. We emphasize that the walks do not have to be self-avoiding: in order to model non-simple curves we may have crossings, or portions of a walk following the same edge.

3.2.3.2 Cross-metric surfaces

The combinatorial model is the easier way to model embedded cycles on the surface and is the dominant one in the literature, but it is not well adapted for some purposes. We illustrate their shortcomings with two difficulties that naturally arise:

1. Combinatorial surfaces are inaccurate regarding crossings of curves. When several curves run along the same edge and end up crossing, the information of where the

crossings take place and in which order is not encoded. This is pictured in Figure 3.5a. and b.

2. For many purposes, it is important to be able to cut a surface along a cycle, or a set of cycles. On a combinatorial surface, two non-crossing but very close cycles will be represented as running along the same edge. Cutting along these is then topologically unsound, since the small space lying in-between disappears. This is pictured in Figure 3.5c. and d.

It is possible to circumvent the second problem by awkwardly saying that the curves actually lie in a ε-neighborhood of their representation; the first problem remains though. Instead, it was noted by Colin de Verdière and Erickson [56] that resorting to the dual setting leads to the model of cross-metric surfaces which lifts these issues in a convenient way.

A *cross-metric surface* is a surface S together with a cellularly embedded graph G^* with nonnegative weights on its edges. The curves that we consider in this model are the one that are in *general position* with respect to G^*, i.e., the ones that intersect the edges of G^* transversely and away from the vertices. The length of a curve is simply its *crossing weight*, which is the sum of the weights of the edges of G^* that it crosses. One can think of a cross-metric surface as a topological surface endowed with a discrete notion of metric: the length of a curve jumps when it crosses an edge.

Cross-metric surfaces and combinatorial surface are naturally associated by duality, as the notation suggests. Starting from a combinatorial surface G, one can transform it to the cross-metric setting by putting a vertex on each face, and edges between adjacent faces, as pictured in Figure 3.6 [1]. This gives the *dual graph* G^*, and combinatorial curves, that is, walks on (S, G), become cross-metric curves on (S, G^*), up to choosing the location of the crossing points when they run along the same edge. Note that the dual of a triangulation will yield a graph where every vertex has degree 3, which we call a *trivalent* graph.

A family of (possibly self-intersecting) curves on a cross-metric surface (S, G^*) is encoded by maintaining the combinatorial *arrangement* of G^* and the curves, which is the combinatorial map associated with the superimposition of the graph and the curves. This encodes the crossings and the relative positions of the graph and the curves unambiguously. In this setting, it is easy to describe the *cutting* of a surface S along a simple curve γ: since parallel portions of curves do not run along the same edge, it is simply the space $S \setminus \gamma$, where one glues two disjoint copies of γ along the two new open ends. It is straightforward to update the arrangement of the graph and the curves after such a cutting.

1. When the surface has a boundary, the construction of the dual graph gets more intricate, but the reader can guess the general construction out of Figure 3.6.

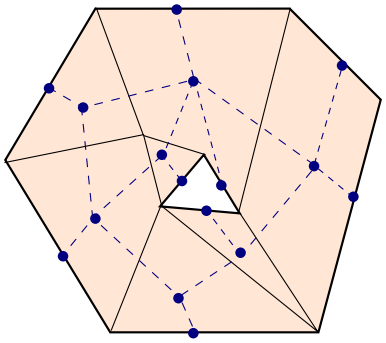


FIGURE 3.6: A graph embedded on an annulus, and the associated dual graph in dashed lines.

3.3 Compressed structures

3.3.1 Normal curves

One striking feature of curves embedded on surfaces is that, when the surface is described by a triangulation, they can be represented very compactly. The first natural way that comes to mind, in order to represent a curve transversely embedded on a triangulation is to encode the sequence of intersections with the edges of the triangulation. This is equivalent to storing the arrangement of the curve with the triangulation that we introduced in the last subsection. But in some cases, one can do much better by just storing the number of intersections with each edge of the triangulation. This is the idea behind normal curves and normal coordinates, which we now introduce. Although we will not manipulate normal curves per se in this thesis, some of their properties will be discussed in the survey in Chapter 4 and serve as a good introduction to normal surfaces around which Chapter 7 revolves.

Let S be a surface and T be triangulation of this surface. A curve γ embedded on S is *normal* with respect to T if it is transverse to T, and the intersection of γ with every triangle in T is comprised of a disjoint union of *normal arcs*, i.e., distinct paths whose endpoints lie on distinct sides of the triangle, see Figure 3.7. One sees readily that there are three possible types of normal arcs in each triangle (one for each pair of sides), and the *normal coordinates* of γ are the number of each type of normal arc in every triangle. Therefore, if the triangulation T consists of t triangles, a curve γ is described by $3t$ nonnegative integers, that we view as a vector in $\mathbb{Z}_+^{3t}$. Normal coordinates satisfy the *matching equations*, that stipulate that the number of normal arcs hitting one side of an edge of the triangulation equals the number of normal arcs hitting the other side.

The point of normal curves is that it is very easy to recover an embedded curve from its normal coordinates if they satisfy the matching equations. Indeed, one can just draw the normal arcs in every triangle, and the matching equations guarantee that there will be a

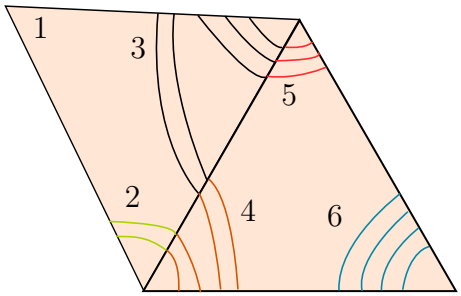


FIGURE 3.7: Normal curves on two triangles. There are 6 different types of normal arcs on two triangles, traced in different colors, and this normal curve can be encoded by the vector $(0, 2, 5, 4, 3, 4)$.

unique way to pair them to the neighborly segments, and to obtain in the end a closed multicurve. One important drawback is that the resulting multicurve needs not be connected, and it is a nontrivial problem to detect when it is (we refer to the discussion in Chapter 4).

Normal curves have two main advantages compared to intersection sequences. The first one is their compactness. Counting intersections allows for an exponential compression in some cases: for example if a curve spirals n times around a vertex of degree d, the sequence of intersections has complexity $O(nd)$ while the number of intersections has complexity $O(d \log n)$. The second one is the algebraic structure they confer: normal coordinates correspond to a vector in $\mathbb{R}^{3t}$, and matching equations define a cone in this space. Normal surfaces can then be added or multiplied by a scalar. We note that the sum of two normal curves does *not* in general correspond to the disjoint union of the two normal curves. For example, for the normal curves pictured in Figure 3.7, we have $(0, 2, 5, 4, 3, 4) = (0, 2, 0, 0, 2, 0) + (0, 0, 5, 4, 1, 4)$, but the reader can check that the decomposition does not correspond to a disjoint union.

3.3.2 Normal surfaces

A *normal surface* in T is a properly embedded surface in T that meets each tetrahedron in a possibly empty collection of triangles (cutting off a vertex) and quadrilaterals (separating a pair of vertices), which are called *normal disks*. In each tetrahedron, there are 4 possible types of triangles and 3 possible types of quadrilaterals, pictured in Figure 3.8. The intersection of an embedded normal surface with a face of the triangulation gives rise to a *normal arc*. There are 3 possible types of normal arcs within each face: the type of a normal arc is defined according to which vertex of the face it separates from the other two.

To each embedded normal surface, one can associate a vector in $(\mathbb{Z}_+)^{7t}$, where t is the number of tetrahedra in T, by listing the number of triangles and quadrilaterals of each type in each tetrahedron. This vector provides a very compact and elegant description of that surface.

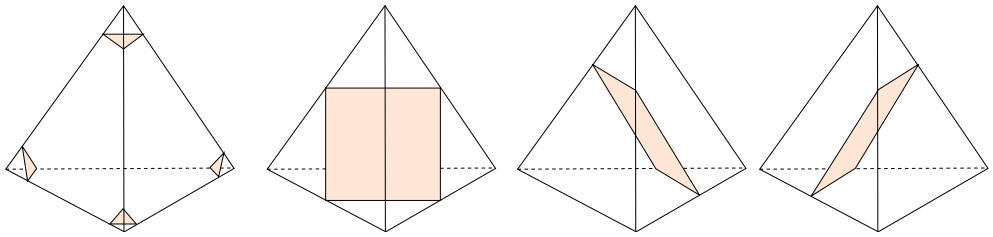


FIGURE 3.8: The seven types of normal disks within a given tetrahedron: Four triangles and three quadrilaterals.

An embedded normal surface corresponding to a vector $(\mathbb{Z}_+)^{7t}$, called its *normal coordinates* , satisfies two types of conditions:

- The first type of conditions is the *matching equations*, which generalize the ones for normal curves. Consider a normal arc type in a given non-boundary face f of T. This normal arc type corresponds to exactly one triangle normal coordinate, $v_{t,1}$, and one quadrilateral normal coordinate, $v_{q,1}$, in a tetrahedron incident with f. Similarly, let $v_{t,2}$ and $v_{q,2}$ be the triangle and quadrilateral normal coordinates corresponding to that arc type in the opposite tetrahedron. The matching equation for that arc type is, by definition, $v_{t,1} + v_{q,1} = v_{t,2} + v_{q,2}$.

- The second type of conditions, the *quadrilateral conditions*, stipulates that, within any tetrahedron, at most one of the three quadrilateral coordinates must be non-zero. Indeed, two quadrilaterals of different types within the same tetrahedron must cross, and therefore this condition is needed to ensure that the surface does not self-intersect.

Conversely, if T is a triangulation of size t and $v \in (\mathbb{Z}_+)^{7t}$. Then v corresponds to an embedded normal surface if and only if the matching equations and the quadrilateral conditions are fulfilled. The reconstruction process can be depicted as follows:

- In each tetrahedron, by the quadrilateral conditions, there is at most one type of quadrilateral. One places as many parallel copies of this quadrilateral as needed in the tetrahedron, and then place the parallel triangles next to every vertex of the tetrahedron. It is straightforward to do so without having any intersection between triangles and quadrilaterals.

- One glues the faces on the triangulation together, and in the process, one needs to glue normal arcs, i.e., triangles or quadrilaterals on the one side to triangles and quadrilaterals on the other side. By the matching equations, the numbers fit, and the gluing is imposed by the order in which the normal disks are placed in the tetrahedra.

Therefore, an embedded normal surface is represented up to a normal isotopy by a vector in $(\mathbb{Z}_+)^{7t}$ satisfying the matching equations and the quadrilateral conditions. Moreover, given a triangulation and normal coordinates, checking that the matching equations or the quadrilateral conditions hold can trivially be done in linear time.

From this construction, one sees moreover that every vector of normal coordinates corresponds to a unique normal surface, up to a normal isotopy.

CHAPTER 4

Some background on computational topology

In this chapter, we survey various facets of computational topology. Most of the results I am interested in can be split into two different subareas. In one case, we start with an algorithmic problem, and we use a topological tool or notion to solve it, or classify objects according to it. For example, solving or approximating optimization problems for graphs embedded on surfaces fits well within this "topology for algorithms" framework. On the other side, one also encounters purely topological tools or invariants that beg for an algorithm to compute them. The problems of deciding if two cycles are homotopic, or classifying 3-manifold are good examples of this "algorithms for topology" approach. This survey follows this dichotomy, and is split into two independent parts. The first half of the survey, in Section 4.1, is more tailored for computer scientists, and they might find it natural to read it before Section 4.2. Mathematicians, on the other hand, could prefer reading it the other way around. This chapter focuses on explaining how the various concepts fit together through various examples instead of being exhaustive, though we generally mention the relevant results adjacent to those presented.

Both algorithms and topology are fields of which one cannot overstate the depth nor the width, therefore this exposition focuses on selected topics. Since this thesis revolves around the computational topology of surfaces and 3-manifolds, our attention here is mostly drawn on low-dimensional results, with a strong emphasis on subjects with connections to our work, which we mention where appropriate. However, a survey is ontologically quite digressive, and various other topics appear as well – in Section 4.1.3, we glimpse at other concepts involving a mix of topology and combinatorics/computer science, and we quickly present a couple of applications outside the realm of low-dimensional topology.

There are now many references introducing and discussing computational topology under different premises. We refer the curious reader to the textbooks of Edelsbrunner and Harer [76], Zomorodian [255] and Dey, Edelsbrunner and Guha [70] for a broad exposition, and to the survey of Colin de Verdière in [55] as well as the lecture notes of Erickson [83].

Although reading this chapter is not necessary to understand the following ones, it forms a good introduction to the problems we consider. In particular, Chapters 5, 6 are respectively related to the contents of Sections 4.2.1 and 4.1.2, while Chapter 7 is connected to Sections 4.2.2 and 4.2.3.

4.1 Topology for algorithms

4.1.1 Planar graphs

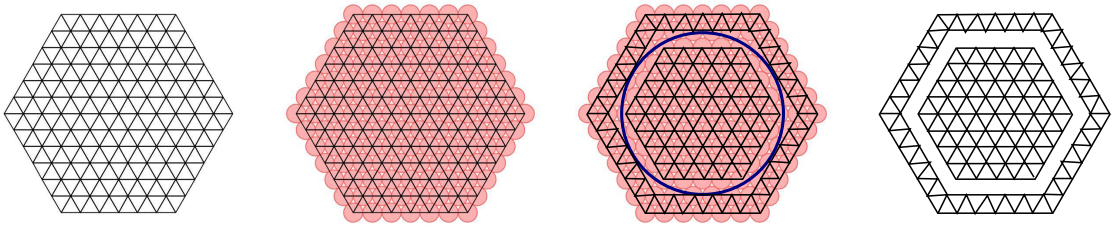
Graphs are an ubiquitous structure in theoretical computer science, as they can model a seemingly endless number of situations, and therefore solving algorithmic problems on graphs has been the focus of the algorithmic community since its inception, and will probably remain so for a while. In this context, studying restricted classes of graphs allows to grasp a better understanding of the general case, as well as devising specific algorithmic techniques tailored to the problem at hand. We will be particularly interested in graphs with a topological constraint, among which the prominent example is the case of planar graphs. In this whole section, n designates the number of vertices of a graph, and, when appropriate, g its genus.

Euler's formula, in the planar case, provides the first nontrivial results on planarity: if v, e and f are respectively the number of vertices, edges and faces of a planar simple graph, we obtain that the number of edges is linearly bounded by the number of vertices, and more precisely that the average degree of a planar graph is smaller than 6. Another important topological feature of the plane is the Jordan curve theorem [123], stipulating that every simple cycle separates it into two connected components (the interior and the exterior). In this section, we will illustrate how these topological ingredients can be used to give two interesting properties of planar graphs: we will first see that they can be cut along small *separators*, and then that they have a characterization using forbidden minors. We mention other results on planar graphs more quickly at the end of the section. The reader might find this section unconvincing, since the topology of the plane is rather dull. However, planar graphs are the first step towards graphs embedded on surfaces, which are presented in the next section. They feature deeper topological properties, and we will showcase that the planar case is actually the first building block or the inspiration for many algorithms on them.

4.1.1.1 Planar separators

The basic definitions for this section have been introduced in Section 3.2.1.

The Jordan curve theorem shows that closed curves are a natural way to separate regions on the plane. This suggests that planar graphs might be the right class of graphs to consider when trying to partition a graph into smaller parts. This is the point of the *pla-*



a. b. c. d.

FIGURE 4.1: a. A planar graph b. Its circle-packing [1] c. A circle partitioning the circle packing. d. The partition of the planar graph.

nar separator theorem of Lipton and Tarjan [174]. A basic paradigm in algorithm design is the so-called "Divide and Conquer" method, consisting in cutting a problem into two fairly independent sub-problems and solving these recursively. One of the difficulties in this approach generally lies in the "fairly", where one wants to control accurately how the two sub-instances are correlated. In the case of graphs, one example of the corresponding questions is the following: In a graph G, how many vertices (and their adjacent edges) does one need to remove to obtain two disconnected components of controlled size (say not containing more than two thirds of the vertices) ? For general graphs, no nontrivial answer to this question exists: the example of the complete graph shows that one might not even be able to disconnect the graph by doing so. But when the graph is specified to be planar, one can do much better. Indeed, the aforementioned planar separator theorem shows that if n is the number of vertices, there always exists a set of $O(\sqrt{n})$ vertices giving a balanced separation. Moreover, this set of vertices can be computed in linear time. One of the proofs of this theorem, though not very efficient algorithmically, uses the beautiful Koebe-Andreev-Thurston Theorem (see Stephenson [237] for an entire book devoted to the subject) that shows that any simple planar graph can be represented by a family of touching circles in $\mathbb{R}^2$, where each circle corresponds to a vertex and the edges represent the adjacency between circles; this is called a *circle packing*. In this representation, one possible separator can be found by choosing a circle C in $\mathbb{R}^2$ (but not one of those of the circle packing) of well chosen size, and taking as separator the set of vertices corresponding to the circles intersected by C. With carefully chosen parameters, this gives a separator of size $\sqrt{n}$, see Figure 4.1 for an illustration of this technique.

One might feel fooled by the motivation using the Jordan curve theorem, since the separators produced by the Lipton-Tarjan algorithm need not have a circular structure. The corresponding refinement is the purpose of Miller's cycle separator theorem [185], which shows that one can assume that all the vertices that we remove to partition the graph lie on a Jordan curve that does not cross any edge. Various proofs and extensions [186, 187] once again feature strong connections with the circle packing theorem. Several other variants of

1. The circle-packings have been generated using Kenneth Stephenson's software *CirclePack*.

the planar separator theorem have also been devised since then. Among them, by applying repeatedly the planar separator theorem, one can obtain partitions of a planar graph into arbitrarily small pieces, yielding the so-called *r-division* [104] of a planar graph. The planar separator theorem and its variants are pervasive in the algorithmic literature on planar graphs, with applications in computing shortest paths [132], distance queries [245], flows and cuts [143], and this is just a very restrictive picture.

4.1.1.2 Minors

Another combinatorial implication of being planar lies within the theory of *graph minors*. A graph H is a *minor* of another graph G if H can be obtained from G by removing edges and contracting edges. A family of graphs is *minor-closed* if every minor of a family is also in this family. Since deletion and contraction preserves planarity, planar graphs are a minor-closed family, and it is easy to see, but not that easy to prove, that the graphs K_5 and $K_{3,3}$ are not planar. It is more striking to realize that the converse also holds. This is the celebrated Wagner's Theorem [250]: A graph is planar if and only if it does not contain K_5 or $K_{3,3}$ as a minor. The following generalization is even more striking: Every minor-closed family of graphs is characterized by a **finite** set of excluded minors. This is the main result of the Robertson-Seymour theory [215], and the proof is spread over more than twenty papers and as many years. We will survey some aspects of the proof later on in this chapter.

Algorithmic applications of graph minors may not seem immediate at first. Wagner's theorem gives an algorithm to test whether a given graph is planar, but it is painfully slow and for all purposes, a linear planarity testing algorithm by Hopcroft and Tarjan [139] is favored. The strength of minors lies more in the structural insight that they give on graphs, and this insight gives rise to powerful algorithmic techniques. Most notably, the study of graph minors led to the discovery of the concept of *tree-width* (or the related *branch-width*), which measures quantitatively how similar to a tree a graph is. It turns out that this captures exactly the right way of decomposing a graph to do dynamic programming; this gave birth to a flurry of algorithmic results, see for example the survey of Bodlaender [21] for an introduction.

4.1.1.3 Other results

Planar graphs have been a thriving research area for several decades, and therefore we now understand a vast amount of their properties. For example, it can be shown [103] that any planar graph with no face of degree 1 or 2 has a vertex or degree at most four, or a vertex of degree at most five incident to two vertices of degree at most six. This can be used to prove that every graph is 5-colorable, and the infamous Four Color Theorem [11] actually shows that four colors suffice.

On the algorithmic side, very efficient algorithms have been developed for the basic problems: minimum spanning trees can be computed in linear time [182], as well as short-

est path trees [209], which provides an $O(n \log n)$ time algorithm to compute minimum cuts. Furthermore, some hard, problems on general graphs have polynomial solutions when restricted to planar graphs. These include max-cut [121], graph isomorphism [140] and computing the branch-width [233]. There are also a vast number of results both from the viewpoint of approximation algorithms and parameterized algorithms. We refer to the upcoming textbook of Klein and Mozes [154] for in-depth study of the algorithmic world of planar graphs.

It is an important question to quantify how far from being planar a given graph is. The genus of the graph, discussed thereafter, can be one indicator of this, and another one is the *crossing number*, which is the minimum number of crossings of a drawing of a graph in the plane. Determining the crossing number of a graph is NP-hard, even for $1-$planar graphs, i.e., planar graphs with one additional edge [41]. A number of open questions on crossing numbers and their variants remain unsolved; as an example the crossing number of the complete graphs is not even known. The dynamic survey of Schaefer [222] discusses the various questions of this theory.

4.1.2 Surface-embedded graphs

We now turn our attention to the case of surface-embedded graphs. Since every graph can be embedded on a surface (intuitively, one can for example put every vertex on a ball and connect them with tubes corresponding to the edges), these are not to be thought of as a restricted family of graphs, and more as a *parameterized* one, where the parameter is naturally the genus of the surface. In this section, we explain various aspects of the algorithmic theory of embedded graphs, and how they are tied to the topology of the underlying surface, as well as their connection with the planar case. A discussion on other aspects of surface embedded graphs is presented at the end of this section.

4.1.2.1 On the genus of a graph

The first question one encounters is how to actually compute the genus of a graph. This problem has been shown to be NP-hard by Thomassen [241], and from this result one can deduce with simple gadgets [51] that it is also unapproximable to an additive error of n^{ε} for any $0 \leq \varepsilon < 1$ unless P=NP. On the upper bound side, when the integer g is fixed, Kawarabayashi, Mohar and Reed [151] gave a linear time algorithm to compute either an embedding of a graph into a surface of genus at most g, or a certificate of non embeddability. Recent progress has been made in approximating the genus in the specific case of bounded-degree graphs. A result of Chen, Kanchi and Kanevsky [51] allowed to compute an $O(\sqrt{n})$-approximation of the genus in this case, and it has been recently improved by Chekuri and Sidiropoulos [50], who provide an algorithm that embeds a bounded degree graph of genus g with n vertices on a surface of genus $O(g^{14} \log^{19/2} n)$. Without delving into the details, this algorithm is a prime example of the algorithmic applications of the

graph minors theory, where one devises two different algorithms depending on whether the tree-width of the input graph is small or large. This line of work has surprising algorithmic consequences: algorithms designed for surface embedded graphs have traditionally always assumed that one was given an embedding of a genus g graph on a surface of genus g. Approximating the genus and the embedding allows to bypass this assumption and obtain algorithms that work for any graph. As an example, using this technique, the recent polynomial algorithm by Erickson and Sidiropoulos [90] to approximate a variant of the Traveling Salesman Problem on embedded graphs of bounded genus computes an $O(\log g / \log \log g)$-approximation of the optimal solution even when no embedding is known a priori.

4.1.2.2 Graph structure theorem

The minor characterization of planar graphs extends to surface embedded graphs: for every surface S, there exists a finite family of forbidden minors characterizing the graphs embeddable on S, see for example Diestel [72, Section 12.5] for a self contained proof. We note that although the precise family of forbidden minors is known for the plane and the projective plane, the lists are unknown for any other surface, and the number of minors grows very fast with the genus. One might see this result as a consequence of the Robertson-Seymour Theorem on minor-closed family, but actually it goes the other way around: graphs embedded on surfaces are a building block on this theorem. More precisely, the way Robertson and Seymour prove their theorem is by first proving a finite forbidden-minor theorem for graphs on surfaces [215], and then obtaining a *structure theorem* for minor-closed families [214]. To state this last theorem we introduce the following definitions. A *clique-sum* of order k of two graphs G and H is the graph obtained by identifying two cliques of size k in G and H, and a *vortex* of a graph G in a subgraph with a path-like structure embedded in a particular way (see for example Kawarabayashi and Mohar [150] for a precise definition). We say that a graph is *k-nearly embedded* on a surface S if one can remove at most k vertices (called *apices*) and k vortices from this graph, such that the resulting graph is embeddable on S. Now, the structure theorem says that for any graph H, there exists an integer k such that any graph excluding H as a minor can be obtained as a clique sum of at most k graphs that can be k-nearly embedded on a surface on which H can not be embedded. This shows that surface embedded graphs, which form an a priori purely topological classification of graphs, are fundamental tools in the study of the more abstract minor-closed families of graphs. In more practical terms, this leads to the following classical evolution in algorithm design for minor-closed graphs. First a specific technique is introduced to deal with a problem on planar graphs, then the techniques are extended to surface embedded graphs, and finally, they are generalized to minor-free graphs; by the structure theorem, it just amounts to dealing with the apices, the vortices, and the clique-sums. Examples of successful algorithms following this path include small separators [152], shortest paths, and bigger classes of algorithmic problems including for example vertex cover and dominating set [67, 220]. In the end, this is a surprising case where a topological condition bears

the fruit of a much more general mathematical understanding, with very strong algorithmic implications.

4.1.2.3 Graph Planarization

But before going to minor-free families, how does one generalize an algorithm for planar graphs to one for graphs on surfaces ? We illustrate this question in the case of the planar separator theorem. Gilbert, Hutchinson and Tarjan [110] (see also Eppstein [79]) generalized the planar $O(\sqrt{n})$ bound to a $O(\sqrt{gn})$ bound for graphs embedded on a surface of genus g. To obtain their result, they identified a set of $O(\sqrt{gn})$ vertices, such that removing them makes the graph planar. Then one can simply apply the planar separator theorem to conclude. This framework, based on simplifying a graph embedded on surfaces to make it planar, is central in algorithm design for embedded graphs, and has been applied for example to connectivity problems like Steiner Tree or Subset TSP [23], approximate distance oracle [149], matchings [62] or expansion parameters [193]. For example, one natural way to simplify an embedded graph is to cut it along a non-contractible cycle, which will either disconnect the graph or reduce its genus. This is where the topology of surfaces intervenes: the various topological properties of the surface and the graph embedding will impact the techniques to use to simplify them. We present here one important *surface decomposition*, namely the *cut-graph* of a surface, where one cuts along a graph instead of a single cycle.

For a surface S, a *cut-graph* on S is an embedded graph G such that cutting S along G results in a planar surface, as pictured for example in Figure 4.2. A natural algorithmic problem that corresponds to this is the minimization variant: What is the length [1] of the shortest cut-graph for a surface S? This problem has been introduced by Erickson and Har-Peled [86], where they provide a NP-hardness proof, an exact but exponential algorithm to solve this problem, as well as an $O(\log^2 g)$-approximation algorithm. Furthermore, it is known that finding a shortest cut-graph when the set of vertices is prescribed can be done in polynomial time [54]. This suggests that the difficulty of the problem comes from the location of the vertices, for which there are $\binom{O(n)}{O(g)}$ choices, and it has been open since then whether the problem is fixed-parameter tractable when parameterized by the genus, i.e. whether there exists an algorithm of complexity $f(g)poly(n)$, where g is the genus of S, f is any function, and n is the complexity of the cross-metric surface S. One possible evidence in this direction is that the hardness proof proceeds by a reduction to the Steiner tree problem, which is fixed parameter tractable with respect to the number of terminals – the connectivity structure of a cut-graph is however more complicated, and the algorithms for Steiner trees do not translate easily.

In addition to the purely algorithmic applications, such a cut-graph has a wide range of applications in computer science. As an example we explain its role in computer graphics. A *parameterization* [69, 254] of a surface S is a correspondence between S and a domain

1. We define the length of the graph via either the combinatorial or the cross-metric models defined in Section 3.2.3.

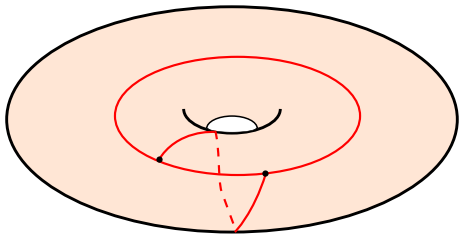


FIGURE 4.2: A cut-graph of a torus.

of the plane. A natural way to obtain one is to cut along a cut-graph, and compute a homeomorphism between the planar region we obtain and the target domain of the plane. In order to have a faithful representation of the surface, it is necessary to have a cut-graph that is not too invasive on the surface: taking the shortest one is the natural criterion. Parameterizing a surface has many practical applications: for example it provides a tool to visualize a surface (it is a practical version of the atlas and maps used in differential geometry), it also provides a way to map some planar texture on it, as well as a means to *mesh* it, i.e. discretize it in pieces of some prescribed form.

With the same goal of simplifying a surface by cutting it along a cycle or a graph, a vast literature has been devoted to optimization problems with a topological flavor, for example the problems of computing shortest cycles with prescribed topological properties [38, 86] (non contractible, non separating or splitting [46]), or tightening a cycle in a given homotopy class [56]. The tools used in these endeavors are quite diverse, and we refer to [55] for a survey on these.

In Chapter 6, we present new results on the lengths of different kinds of graph decompositions: we investigate the lengths of cycles with prescribed topological properties, as well as the lengths of cut-graphs and *pants decompositions*. We refer to this chapter for the relevant definitions, results, and further discussion.

4.1.2.4 Flows and homology

In this section we present another interesting algorithmic insight regarding graphs on surfaces: the strong connection between homology and flows. Flows are a classical tool in algorithm design that depicts how a fluid would propagate in a graph: there is a source and a sink node in a graph, symbolizing the entry and exit point of the fluid, and each edge is assigned a value representing the amount of fluid it contains, and this value must not exceed a given *capacity*, which is specified for every edge. A feasible flow is an assignment of values to the edges such that at every vertex, a Kirchhoff-like law is respected: the amount of flow arriving is equal to the amount of flow leaving. The goal of a *maximum flow* problem is to find a feasible flow maximizing the amount of fluid transiting between the source and the sink. It turns out that this concept is very similar to the one of a *homology cycle*, introduced by Poincaré [199] at the end of the nineteenth century, and which is one

of the fundamental invariants of algebraic topology. A 1-dimensional homology cycle [1] on a surface described by a cellularly embedded graph is an assignment of integer values to its edges such that the values satisfy the same Kirchhoff law at the vertices as the flows. Therefore, a flow is just a homology cycle violating this law at the source and the sink node. Homology was originally designed to study topological spaces by the means of linear algebra, and this algebraic structures has very practical algorithmic applications: using linear programming on these homology spaces, Chambers, Erickson and Nayyeri [48] obtained efficient algorithms to compute maximum flows and different variants for embedded graphs. Furthermore the max-flow min-cut theorem [102] shows that flows and cuts are intimately connected, and therefore these algorithmic techniques apply to computing cuts as well [85, 88].

4.1.2.5 Other results on surface embedded graphs

The study of the structural properties of embedded graphs is the subject of *topological graph theory*, for which Mohar and Thomassen [188] and Gross and Tucker [117] are the main references.

One important theme [216, 242] is that if an embedded graph is subdivided enough to not see the topology of a surface locally (for example if the length of the shortest non-contractible cycle is big enough), it shares many properties with planar graphs. For example, under such an assumption, 3-connected graphs have a unique embedding on a given surface S. Another application of this idea deals with graph coloring: while in general, graphs embedded on surfaces might need an arbitrary number of colors (surprisingly, finding the tight number of colors for a given surface is easier than in the planar case, see Ringel [211]), for such locally planar graph, five colors suffice [243].

The dichotomy of our survey leaves aside the purely combinatorial aspects of surfaces, and consequently some were omitted in this discussion. Most notably, there is a well developed theory around the enumerative and bijective theory of maps [25] initiated by Tutte [248], which leads to practical applications such as encoding [6, 7]. In another direction, the theory of random maps [165] has recently exploded into a myriad of results.

As we mentioned before, concepts stemming from minor graph theory like tree-width and branch-width have a strong algorithmic impact on the theory of surface embedded graphs. Most notably, in this direct lineage, the recent theory of *bidimensionality* [68] provides a framework to obtain approximations or parameterized algorithms for a vast class of problems on surface embedded graphs (and sometimes minor-free graphs as well), including domination, vertex cover, matching and feedback vertex set problems [67, 101].

Different techniques have also been introduced to simplify a surface: much progress has been made on *stochastic planarization* [24, 142, 234], which aims at finding a random

1. Technically, homology is defined relatively to a ring, this informal discussion corresponds to the $\mathbb{Z}$ case.

mapping from an embedded graph to a planar graph, such that in average, the distances are not too much distorted; the theory of *random partitions* [167] has been fruitful as well.

4.1.3 Other applications of topology

Finally, with an illustrative purpose, we quickly present two additional applications of topology in computer science which use tools of a different flavor than graph theory. In particular, the underlying topological spaces are not restricted to low dimensions. Other topics in theoretical computer science and combinatorics where topology plays a crucial role include fair-division problems [19, 146], Kneser's conjecture [175], evasiveness of graph properties [212], embeddability problems [179]; and in the wider world of computer science, directed algebraic topology and its applications to concurrency [94] and the slow revolution surrounding homotopy type theory [249].

4.1.3.1 Persistence

The theory of *topological persistence* is a very important emerging field dealing with data analysis, for more information we refer the interested reader to the surveys of Ghrist [109] or Edelsbrunner and Harer [75]. We illustrate how it works on a specific problem, the one of recognizing a topological shape out of a point cloud sampled from this shape. One possible idea is to consider balls of radius r centered at each point, and to grow r continuously. For small r, the topology of the union of these balls will be uninteresting, i.e., the same as the point cloud, while for big r, the size of the balls will obfuscate the whole picture, thus making an observer unable to read any topological information about the shape. The insight of the persistence theory is that a pertinent way to deal with this issue is to record the lifetime of a topological feature, for example k-dimensional holes: we record both the first r, the *birth*, at which a feature appears, and the r where its disappears, its *death*. The relevant features of our shape will be the ones that survive for a long time. Instead of homotopy (although some recent progress [169] in this direction has been made), of which computation is delicate, the theory of persistence is mostly applied with homology [1]; by means of linear algebra, one can compute the evolution of the homology with the parameter r very efficiently [256].

Persistent homology is a tool of strong generality, which makes it very relevant for practical purposes: linear algebra is very efficient, and techniques have been developed to deal with noise and outliers [49], or to use different parameters instead of the size of the balls (this is formalized through the concept of a *filtration*). While the range of applications of persistence is hard to predict, it is already a common tool in data analysis, and some surprising results have already appeared. For example, a work of Carlsson, Ishkhanov, de Silva and Zomorodian [43] has investigated a point cloud obtained by taking the high-contrast 3

1. The reader unfamiliar with k-dimensional homology should just think of it as a computably tractable way to count the k-dimensional holes of a topological space for any k.

by 3 pixel blocks out of a collection of *natural images*, i.e., random outdoor scenes. After suitable normalization, the points look like they are uniformly distributed, but surprisingly, further analysis using topological persistence reveals that the shape they represent is very close to the one of a Klein bottle. They also provide a theoretical explanation for this fact.

4.1.3.2 Distributed computing

Finally, we quickly present a strong connection between algebraic topology and distributed computing, which was discovered by Herlihy and Shavit [135]; there is an upcoming book on the subject [133]. Instead of the framework of manifolds in which this thesis takes mostly place, this connection is better exhibited in the terms of *simplicial complexes*, which are obtained by abstractly gluing simplices. In 1 dimension, this gives rise to graphs, while in the 2-dimensional case, these can be thought of as the spaces obtained by gluing triangles together, without the natural rule that only two triangles are glued along a single edge. This gives rise to branching points, or edges, and spaces with more topological richness than surfaces. In this setting, a *simplicial map* is a continuous map between two simplices sending each vertex to another vertex such that the simplices are preserved. We now explain how to connect these spaces with classical problems of distributed computing. Typically, distributed computing deals with independent *processes* that try to perform a *task* together by following a *protocol*. Example tasks are the consensus problem, where the processes start with a set of values and try to all agree on an output [1], or the k-set agreement where they must output at most k different results (consensus is 1-set agreement). Consensus is trivially solvable by a majority computation, but distributed systems are prone to failure, therefore one wants a *wait-free* protocol, which is a protocol resilient to the failure of every process but one.

One possible way to represent this is to model the input and the output as simplicial complexes I and O, and the task as a map Δ carrying every input simplex to a set of output simplices. The input (respectively output) complex will consist of vertices representing the processes and their input (respectively output) value. Every possible input or output corresponds to a simplex linking the corresponding vertices. For example, for the consensus problem with n processes, the input complex consists of vertices (P, v) where P is the name of a process and v the value it is assigned as input. The simplices are put between all the sets of vertices $(P_1, v_1), \dots, (P_n, v_n)$ where all the P_i are different. The output complex has the same vertices, but the simplices are only put between the sets $(P_1, v_1), \dots, (P_n, v_n)$ where all the P_i are different and all the v_i are equal (this is the condition for a consensus). Finally, the task is the map that carries each possible input, therefore each input simplex, to the set of possible outputs that it can produce, and therefore to a set of output simplices. This representation is pictured in Figure 4.3 for the consensus problem with two values and

1. The *validity* assumption stipulates that if they all start with the same value, they have to agree on this one.

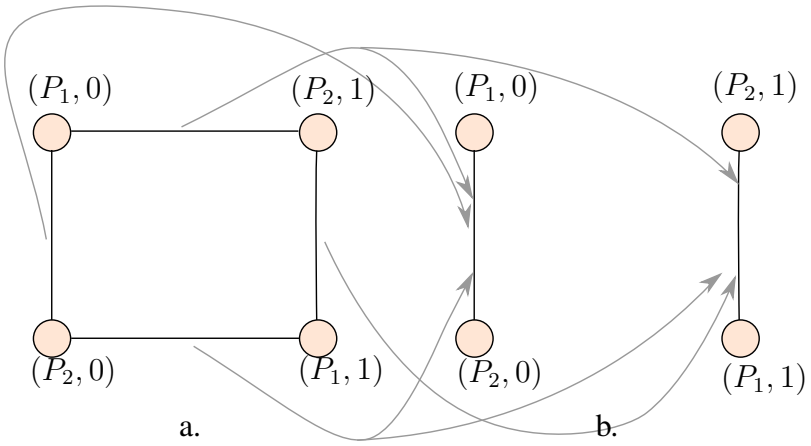


FIGURE 4.3: The 2-process consensus: a. The input complex b. The output complex. The gray arrows represent the map Δ, by the validity assumption, some arrows are forbidden.

two processes, the simplicial complexes for this problem are simply graphs. Note that in this case, the input process is a sphere, which is actually true for every value of n.

One can see, but we will not explain how, that the execution of a protocol can also be modeled using a simplicial complex P, and for an input simplex S, there is an associated subcomplex of P denoted by $P(S)$. This leads to the following rephrasing of our distributed problem: The existence of a protocol to solve a task (I, O, Δ) is equivalent to the existence of a protocol complex P and a simplicial map $\delta : P \to O$ such that for all $S \in I$, and for all $T \in P(S)$, $\delta(T) \in \Delta(S)$. Putting restrictions on the protocol amounts to putting restrictions on the complex P and the simplicial map δ one is looking for.

Although the simplicial formulation of a distributed computing problem might seem like a pompous obfuscation of the underlying computational ideas, it allows to use the whole machinery of algebraic topology to prove the nonexistence of maps, and therefore the impossibility of a task. For example, Herlihy and Shavit [135] used this framework to show that when processes are only allowed to write and to read on shared variables, the topological conditions to solve the consensus problem are too stringent and make it impossible. Although the impossibility of consensus has been known for some time [99], this theory gives additional tools and has a wide applicability. For example, for more intricate protocols, further study shows that there is a direct connection between the homology of the associated protocol complex, and the number of processes for which one can solve the consensus problem [136]. This theory has been applied to many more distributed computing problems and led to a tight classification of solvable problems for some protocol classes [135]. The optimist reader might conclude that the right tool has been found and that everything in this branch of distributed computing should be classified soon enough, but it has been proved by Gafni and Koutsoupias [105] that no algorithm exists for deciding whether 3-processes tasks are decidable. Furthermore, in another intriguing display of the

connection to topology, the tools used in this proof are also topological in nature: using the same formalism, they built a reduction linking the solvability of a task to the contractibility of a loop in a simplicial complex, which, as we will see in Section 4.2.1, is undecidable. We also refer to Herlihy and Rajsbaum [134] for other results along these lines. It is an exciting research perspective to explore the reverse direction, as to whether topological notions could give insight on upper bounds on distributed protocols. Some work has appeared along these lines, taking inspiration from the equivariant Hopf Theorem [12, 44].

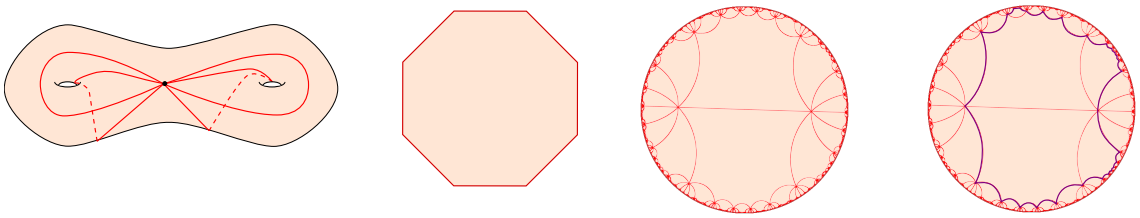
4.2 Algorithms for topology

4.2.1 Decision problems on surface groups and variants

While the study of computers for themselves and algorithmics as a genuine field of interest only really started in the second half of the twentieth century, various questions about how to systematically compute things have appeared way earlier. Moreover, by its combinatorial nature, algebraic topology has attracted algorithmic questions from the very start. As early as 1911, Max Dehn [65] introduced the *word problem*, i.e., the problem of deciding systematically whether two words in a finitely generated groups represent the same element, or equivalently whether a given word is equivalent to the empty word, as well as the sister problems of *conjugacy* (are two elements conjugate?) and *isomorphism* (are two groups isomorphic?). The next year, in [66], he solved the word and the conjugacy problems for fundamental groups of surfaces of genus at least two. Extending what is now called Dehn's algorithm, as well as on the other side, studying the undecidability arising from the word problem, have been major motivations in the development of algorithms in general, and in particular computational topology.

4.2.1.1 Dehn's problems on surfaces

The word problem is about deciding whether given a word w in a group, there are local cancellations among the subwords of w that allow to simplify w up to the empty word. When the group is the fundamental group of a surface, letters of a word correspond to loops, and local cancellations to contractions of these loops. Therefore, Dehn's main insight is that the word problem for fundamental groups of surfaces is the same thing as the *contractiblity problem* on the surface, i.e., deciding whether a given cycle can be contracted to a point. Now, as described in Section 3.1.1, the universal cover of a surface of genus at least two is the hyperbolic plane, and a canonical system of loops lifts to a *tiling* of this hyperbolic plane, as pictured in Figure 4.4. Therefore, a contractible cycle on the surface lifts to a cycle in this tiling of the universal cover. One can now use the combinatorial properties of the hyperbolic plane to solve the contractibility problem. The key observation is that a cycle in this hyperbolic tiling satisfies the following *subpath property*: either it contains a spur,



a. b. c. d.

FIGURE 4.4: a. A canonical system of loops of a genus 2 surface b. The surface obtained after cutting along the loops c. The hyperbolic tiling [1] obtained by lifting the loops d. A cycle on this tiling satisfying the subpath property.

i.e., it backtracks at some point, or it contains more than half of the edges of the boundary of a polygon in succession, see Figure 4.4.c. In both cases, we get a computational way to simplify the cycle to a point, certifying the contractibility. For the conjugacy problems, one can see that it similarly relates to the *homotopy problem* for cycles on surfaces. Similar tools, that are also dubbed Dehn's algorithm, give a solution for this variant.

4.2.1.2 Generalizations of Dehn's algorithms

The insight of Dehn's algorithm can be rephrased in more abstract terms: the transformation to a geometric problem on the hyperbolic plane provides a subpath property, that when pulled back to the original word problem, gives a *cancellation property*, that is, an easily testable property on a word that leads to an algorithm to simplify them mechanically. One can wonder whether such a cancellation property is specific to groups arising from topological settings. Identifying precisely what kind of cancellation property, relatively to the presentation of the group, allows for an algorithm for the word problem, led to the *small cancellation theory*. Initiated by Greendlinger [113, 114], it is a far-reaching generalization of this observation to a wider class of groups, and became one of the cornerstones of combinatorial group theory [176].

In a different direction, the graph we obtained by lifting the canonical polygon to the universal tiling is an example of a *Cayley graph*. In general, given a group G with a set of generators S, the Cayley graph is the graph obtained by taking the elements of G as vertices, and every generator s as an edge between g and gs, for every $g \in G$. This gives a geometric realization for any group, and in the case of surface groups, the subpath property shows that the Cayley graph verifies a strong *isoperimetric inequality*: any cycle of length ℓ encloses a number of disks bounded by some function of ℓ. For example, in the case pictured in Figure 4.4.d., one case see that the isoperimetric inequality is actually a linear one. The theory of *automatic groups* [82] shows that when the Cayley graph of a group possesses an isoperimetric inequality, there is a natural way to endow it with an

1. The hyperbolic tiling has been generated using Dmitry Bryant's software *Tessellation*.

automatic structure, i.e., with automata computing the group multiplication and testing equality. This allows to solve the word problem in automatic groups in quadratic time. Studying the geometry of Cayley graphs has been an active mathematical topic for a long time now, and therefore this theory applies to a wide class of groups, essentially to all the groups acting nicely enough on hyperbolic manifolds. More examples include mapping class groups [190], and some small cancellation groups [108] [1]. To tackle the conjugacy problem, a variant called *biautomatic groups* have been introduced, but many questions about them remain open [82].

Algorithms improve in two directions, and in addition to generalizing them, one also wants to improve their efficiency. The quest for optimal algorithms for the contractibility and homotopy problems on surfaces seemingly ended with Dey and Guha [71] who provided linear time algorithms for both problems using small cancellation theory. But a flaw was recently discovered in their approach by Lazarus and Rivaud [164], who proposed an alternative, more geometric, linear time solution. The main difficulty to handle the homotopy problem is that unlike in the contractibility problem where we reach the empty word/cycle, there is no canonical objective one wants to attain by doing simplifications. One of Lazarus and Rivaud's main contribution is to define *canonical cycles* in a homotopy class, and showing how to reach them in very limited time. Naively, the shortest cycles homotopic to a given one can be ordered from the left to right, and one canonical choice is to pick the rightmost cycle in this class. Erickson and Whittlesey [92] subsumed both approaches by showing how small cancellation theory also naturally led to these canonical cycles.

The topological notion of homotopy captures fairly well the intuition of a continuous deformation between two curves, except that, as mentioned in Section 3.1.1, in the process of deformation, we may observe additional self-intersections. This motivates the similarly looking *isotopy problem*, where one wants to find an isotopy between two curves. For simple non-contractible cycles, Epstein's Theorem 3.1.2 shows that both problems are equivalent, and contractible cycles do not have much interest. On the other hand, generalizing the problem to graphs gives a strong similarity criterion for embedded graphs, with many algorithmic applications. The study of the corresponding *graph isotopy problem* is the object of Chapter 5, where we give efficient algorithms both in a combinatorial model and a more geometric one in the plane.

4.2.1.3 Limitations

Finally, we note that as soon as we deviate from surfaces, the word and conjugacy problems very quickly become intractable. In general, the word problem has been shown to be undecidable by Novikov [192], and since every finitely presented group can be realized as the fundamental group of a smooth $4-$manifold, the word problem is undecidable for those

1. In a way, it has been shown that the theory of automatic groups supersedes small cancellation theory [95].

as well. One can reduce the isomorphism problem to the word problem, and building on this, Markov [178] showed that the *homeomorphism problem*, i.e., deciding whether two manifolds are homeomorphic, is undecidable as well for $4-$manifolds (see [239] for a self-contained proof). However, in the intermediate $3-$manifold case, most problems are still solvable, and the tale of some of these algorithms is the subject of the next section.

4.2.2 Topological problems with 3-manifolds

We advise the reader to be familiar with the notions introduced in Sections 3.1.3, 3.3.1 and 3.3.2 before reading this section and the next one.

When provided with a discrete description of a manifold, generally a triangulation or a simplicial complex, the first question one has in mind is to recognize to which manifold it corresponds, or equivalently being able to tell whether two descriptions correspond to the same manifold. In the case of surfaces, this can be trivially solved by computing the Euler characteristic and the orientability, which allows to classify every surface in linear time. This "miracle" only appears in 2 dimensions though, as one can show using Poincaré duality that the Euler characteristic of 3-manifolds is always zero [239, Section 8.2.2], and therefore gives no information on its topology. In addition to being a natural problem, recognizing specific 3-manifolds has immediate applications. For example, recall that a knot embedded in S^3 is unknotted if an only if its complement is a solid torus; therefore being able to identify a solid torus gives a solution to the unknotting problem. As another motivation, being able to recognize a 3-sphere is the fundamental step to test whether a given 4-dimensional complex is a manifold, since it is enough to test that all the vertex links are 3-spheres. We will survey the known algorithms for these two problems.

4.2.2.1 Knot recognition

The problem of detecting whether a knot is unknotted has a long history, and has been intriguing researchers for quite a long time. At first sight, it is unclear whether is it decidable at all, as Turing stated in a famous paper [247] as early as 1954. The first algorithm was proposed shortly after by Haken [122], and was subsequently improved by Hass, Lagarias and Pippenger [128]. As a high level overview, their algorithm works as follows. Taking the complement of a small neighborhood of a piecewise linear knot K and triangulating it, we obtain a triangulated 3-manifold with boundary, such that there is a disk spanning this boundary if and only if the initial knot is unknotted – this is the Seifert disk we introduced in Section 3.1.3. The idea is to somehow enumerate all the possible surface spanning this boundary, and test their topology by computing their Euler characteristic. Of course, there are a priori an infinite number of surfaces, but Haken's insight is that it is enough to consider the normal surfaces: by putting the disks in general position and chopping off undesired parts, one can safely assume that the disk we are looking for crosses the triangulation transversely, and that these intersections are only normal disks. This allows to use the

full strength of normal surfaces: seeing them as vectors in $\mathbb{R}^{7t}$, with the associated scalar multiplication and addition, the normal surfaces having boundary K define a cone. One can show that if it exists, the disk we are looking for lies on one of the extremal rays of the cone, and that it does not have too big a complexity. It is then just a matter of enumerating those [1]. From a complexity viewpoint, this shows that the unknot problem is in NP, i.e. that one can certify in polynomial time that a knot is unknotted: the certificate is the normal coordinates of the disk.

There has also been work in the reverse direction, to certify whether a knot is knotted. A surprising connection was made by Greg Kuperberg when he proved that, assuming the Generalized Riemann Hypothesis [63], the unknotting problem is in co-NP [159], i.e. there is a polynomial certificate of knottedness. The idea behind the proof is the following. It is well known [28, Section 3.B] that the fundamental group of the complement of a knot is abelian if and only if the knot is unknotted. Furthermore, Kronheimer and Mrowka [158] showed that the fundamental group of a non trivial knot has a nonabelian representation [2] into $SU(2) \subseteq SL(2,\mathbb{C})$, and that this representation can be described with polynomials. It is then a matter of algebraic complexity to transform this representation into one into $SL(2,\mathbb{Z}/p\mathbb{Z})$, and it results from work of Koiran [157] that it is doable with a polynomial bound of p if the Generalized Riemann Hypothesis holds. We also note that Agol announced an unconditional proof that unknotting is in co-NP, but no article nor preprint has appeared yet [1].

For the unknot problem, new results have appeared recently in a direction apparently disjoint from the previous works. Any knot can be projected onto a plane, giving rise to a *knot diagram*, where one indicates which strand of a knot lies above the other one at ever crossing. Then, one can try to simplify a knot diagram by just doing local moves which obviously do not change its isotopy class. A possible choice of local moves is the *Reidemeister moves*, pictured in Figure 4.5, and one can quite easily show that a knot is unknotted if and only if there is a sequence of Reidemeister moves that realize this unknotting. Now, let us observe that any bound on the size of this sequence immediately gives an algorithm to test the unknot: it suffices to try all the possible sequences of length smaller than this bound. In a recent breakthrough [161], Lackenby showed that for a knot diagram with k crossings, there exists a sequence of Reidemeister moves using only a number of moves polynomial in k. This gives an immediate algorithm, and furthermore a polynomial-sized certificate of unknotting, and therefore an alternative proof that the problem is in NP: one only needs do write down the sequence of Reidemeister moves. Although this result is conceptually simpler than the use of normal surfaces, it is actually a refinement of the previous algorithms, since the proof heavily relies on normal surface theory as a means to "compress" the number of Reidemeister moves one needs to perform

1. Actually, using stronger tools coming from operational research [33], one can greatly optimize the efficiency of the algorithm, experimentally reaching a polynomial-time behavior on all practical instances.

2. A representation of a group G in a linear group H is just a group homomorphism $\varphi : G \to H$. It is nonabelian if there exist two noncommutative elements under the image of φ.

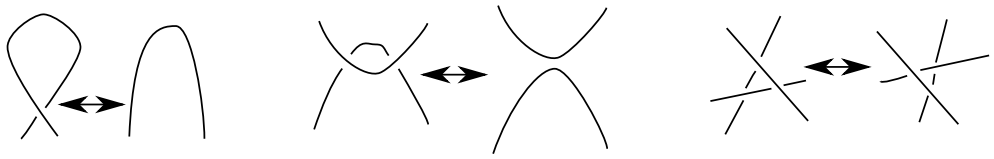


FIGURE 4.5: The three Reidemeister moves to simplify a knot.

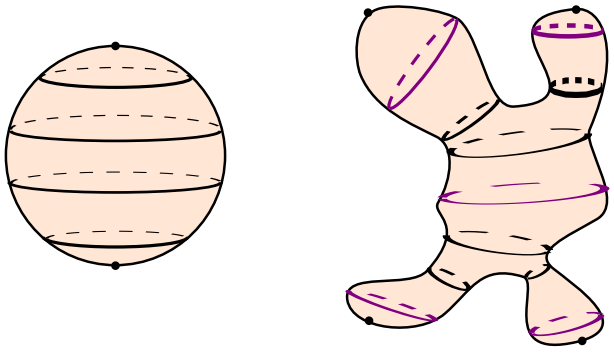
to simplify a knot diagram.

4.2.2.2 Sphere recognition

We now turn our attention to 3-sphere recognition. This problem is intimately related to the famous Poincaré conjecture, stating that every simply connected, closed 3-manifold is homeomorphic to the 3-sphere, and it has been noted that many attempts at 3-sphere recognition would actually imply a combinatorial proof of the Poincaré conjecture [126]. The first algorithm for 3-sphere recognition is due to Rubinstein [217], and was greatly simplified by Thompson [244], we present its main ideas following the exposition of Hass [126]. A natural way to identify a topological n dimensional sphere is to use the fact that it is foliated by $(n-1)$ spheres, i.e., that one can grow a continuous family of disjoint $(n-1)$ spheres between two poles of the sphere, see Figure 4.6.a. for a 2-dimensional example. When the sphere is a quite distorted, this foliation might take a different, more tree-like shape, like on Figure 4.6.b. The idea is to use these $(n-1)$-spheres as a certificate to recognize the n-sphere.

More precisely, in the 3-dimensional case we are interested in, we only look for the shortest and the longest 2-spheres of a foliation. The insight is that in the piecewise linear setting, one can show that the shortest spheres correspond to *normal* spheres, while the longest ones are *almost normal* spheres. Almost normal surfaces are slight variations of normal surfaces where one allows one additional octagonal shape in the way the surface intersects the triangulation; for our purpose, they share the same properties as normal surfaces, most notably in terms of algebraic structure. Once again one can show that both the normal and almost normal spheres we are looking for lie on the extremal rays of the (almost) normal surface cone. Therefore, the algorithm just [1] needs to look at a maximal family of disjoint normal and almost normal spheres on these extremal rays, which can be done by enumeration. Then, checking some conditions [126] on this family of spheres allows to certify that the underlying 3-manifold is a sphere. Careful analysis of this procedure shows that 3-sphere recognition lies in NP [230] (see also the discussion in the next section), and tools similar to the ones used for the unknot can be used to show that it is in co-NP as well [127], assuming the generalized Riemann hypothesis. We note that these results imply that both the unknot and the 3-sphere recognition problems are likely not NP-

1. For technical reasons, one also needs to test the additional property that the 3-manifold has trivial homology, but this is easy to compute.



a. b.

FIGURE 4.6: a. The usual sphere foliated by circles. b. A distorted sphere with its corresponding shortest and longest geodesics.

complete, and are thus good candidates to be either NP-intermediate, or simply polynomial time solvable.

4.2.2.3 Other problems

We only touch a few words about the general problem of recognizing 3-dimensional manifolds, and refer to the lecture notes of Jaco [144], the book [17, Section 1.4] or the article [231] for additional reference. It follows from Thurston's Geometrization conjecture, which has been proved alongside the Poincaré conjecture, that every 3-manifold can be "cut" into several pieces, each of which has a geometric structure, of which there are only 8 possible types. In contrast, in the two dimensional case, there are only 3 possible geometries, as we saw in Section 3.1.2: the Euclidean, the spherical and the hyperbolic one. For each specific type (the hyperbolic one being the most prevalent), specific algorithms have been designed to exploit the geometric structure to identify the manifold. The proof of the Geometrization conjecture showed that this set of algorithms is exhaustive, which coupled with an algorithm to actually compute the decomposition, gives a complete algorithm to identify a 3-manifold, or test whether there exists a homeomorphism between two of them. The complexity of this algorithm cannot be overstated, and drastic improvements still need to be found before an implementation can be dreamed of.

This *geometrization approach* to deal with algorithmic problems in $3-$manifold theory is actually commonplace, and to close the parallel with the previous sections, we note without delving into details that the known solutions to the word [252] and the conjugacy [201] problems in 3-manifold groups work in the same way: start by decomposing the 3-manifold in geometric pieces, and design an algorithm using specific tools depending on the geometry. For these two problems, no algorithm is known that does not use the Geometrization Conjecture.

4.2.3 Compressed structures in topology

In this section, we investigate how the study of curves embedded on surfaces, or surfaces embedded in $3-$manifolds, naturally leads to compact structures, and how to deal with them algorithmically.

Recall that *normal coordinates* allow for a compact representation of embedded curves on surfaces, when the surface is described a triangulation. One dimension higher, the neighborly concept of *normal surfaces* gives a concise way to describe surfaces embedded transversely to a tetrahedrization.

But as is usual with compressed data structures, the compactness of the representation leads to challenges regarding their manipulation. In particular, the previously mentionned results about the complexity of the unknotting and 3-sphere recognition problems already contain some intricate issues, that are inherent to the manipulation of algorithms involving normal curves or surfaces. We illustrate this with the problem of connectivity of normal curves.

4.2.3.1 Connectivity of normal curves

We saw that any normal coordinates satisfying the matching equations naturally correspond to an embedded 1-manifold. But this manifold needs not be connected. And due to the compactness of the representation, it is a nontrivial problem to check connectivity in time polynomial in the input, i.e. in the normal coordinates. For instance, a naive algorithm would be to actually build the manifold, and count the number of components, but "drawing" n normal arcs of the same type has exponential complexity in the input of size $O(\log n)$. The connectivity problem has been open for some time before Agol, Hass and Thurston introduced the first polynomial algorithm to test it in [2]. Their algorithm actually deals with the following more abstract problem: Given a collection of k bijections between subintervals of an interval $[1, N] \subseteq \mathbb{Z}$, they compute the number of orbits, i.e., the number of equivalence classes given by the action of the bijections, in time $O(k \log N)$. We illustrate the connection of this problem with normal curves in Figure 4.7: We label the intersections of the normal arcs with the triangulation by 1 to N, and normal arcs give a bijection between subintervals of $[1, N]$. Counting the number of orbits under the action of this bijection amounts exactly to counting the number of connected components of the corresponding normal curve. The orbit counting algorithm is quite technical and we will not delve into the description, but just remark that despite of the very combinatorial nature of the problem, the idea behind the algorithm is a geometric one, and the exponential speedup can be interpreted with hyperbolic geometry [2, Remark at the end of Section 4]. We remark that this algorithm works exactly the same way for normal surfaces and that this orbit counting algorithm is fundamental in the proof that 3-sphere recognition lies in NP. It also allows (and was initially designed) to show that the following *knot genus* problem is in NP: What is the smallest genus of a surface having a knot K as a boundary ?

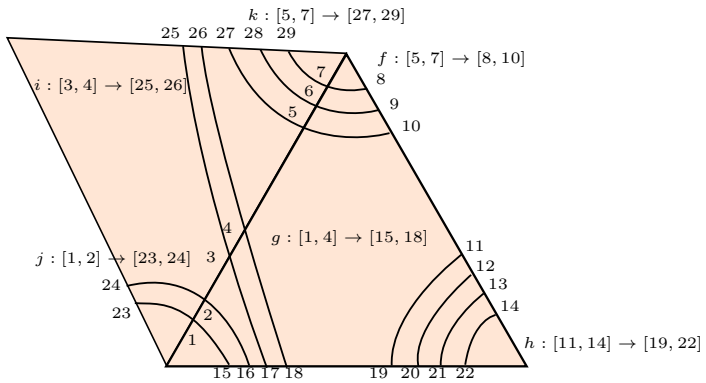


FIGURE 4.7: A triangle with normal curves. The normal arcs induce the bijections f to k between the intersection points of the arcs with the triangulation, described by subintervals of $[1, 29]$.

4.2.3.2 Straight-line programs

Another fruitful connection appeared while studying normal curves and surfaces. The natural way to describe a simple curve embedded on a triangulated surface, such that the intersections with the edges of the triangulation are transverse, is to remember the successive intersections of the curve with the triangulation. This corresponds to a word that we call the *intersection sequence* of the simple curve. Now, there is an established theory studying the different ways to compress a given word to make it more compact, the best known compression method being probably the Lempel-Ziv encoding [168]. We will be using the theory of straight line programs instead, which are easier to manipulate. A *straight line program* over an alphabet $\sum$ is a sequence of assignments to variables $w_1 \dots w_n$ such that the right hand side of the assignment for w_i only involves letters of $\sum$ or variable w_j for $j < i$. The word generated by a straight line program is the word corresponding to the last variable, and its length is the sum of the lengths of the assignments. In some cases, the size of a straight line program is exponentially more compact than the length of the word it generates. For example, the word a^n for n a power of 2 can be encoded by the straight line program $w_1 = a, w_2 = w_1 w_1, \dots w_{\log n} = w_{\log n-1} w_{\log n-1}$.

It is an important insight of Schaefer, Sedgwick and Štefankovič [227, 236] that normal curves and straight line programs representing intersection sequences are roughly equivalent, i.e., that the sizes of both representations are polynomially related, and that one can switch between them in polynomial time. This allows to use the theory of straight line programs to deal with the algorithmic challenges posed by normal surfaces. For example, the problem of deciding the connectedness of a normal curve can be phrased as a word equation over straight line programs, and one of a special structure that we know how to solve efficiently. Related problems, like deciding whether two connected normal curves are homotopic, or computing the intersection numbers of two normal curves, can also be solved using the same tools [224, 226]. To illustrate the strength of this connection, this

theory has been used to prove that the problem of recognizing string graphs, i.e., graphs derived from the intersection of curves (strings) on the plane, is in NP [227, 228], while it was not even known before whether it was decidable. We note that the relationship between normal curves and straight line programs can be carried over to normal surfaces for a handful of problems, giving for example an alternate algorithm to decide the connectivity of a normal surface [226].

4.2.3.3 Tracing normal curves

It is quite intuitive that, when drawing on a curve on the plane or a surface with exponentially many crossings with a given triangulation, one ends up always repeating the same patterns, for example spiraling, or being stuck between two other portions of curves and following them for a long time. This intuition, and its limitations, have been studied in [223, 225], and have led Erickson and Nayyeri [89] to devise a polynomial time algorithm to *trace* a normal curve embedded on a surface. The idea of this tracing is that although the intersections of a normal curve with a triangulation may have exponential size in the input, for many applications, like testing connectedness, one actually does not care about the initial triangulation. If one is allowed to modify it, it becomes possible to output a new cellular graph embedded on the surface, such that the normal curves are embedded in its 1-skeleton, and this new graph can have polynomial complexity. To be able to carry out this computation in polynomial time, the main insight is that one actually does not need to follow a normal curve throughout its exponential number of intersections to guess its whereabouts: if it starts to spiral, we can skip to the point where it leaves it, and similarly when it is stuck behind two adjacent subcurves. This allows for the exponential speedup in this tracing. Then, one can read a lot of information on the cellular graph, like the connectedness or the contractibility, and this can all be done efficiently.

4.2.3.4 Different representations

Finally, a few other ways to describe simple curves on surfaces have been used in the mathematical community. Thurston introduced *train tracks* [191, 194] as a tool to study the *mapping class group* of a surface, which describes the homeomorphisms of a surface to itself. Train tracks can be obtained from an embedded curve on the surface by blurring it, so that parallel portions merge and give rise to a graph looking like a train track, as is picture in Figure 4.8: every portion of a train track is labeled by an integer representing the number of arcs merged into it, and there are natural matching equations every time a train track branches [1]. The nice structure and drawability of train tracks make for a convenient analysis of their combinatorics from a mathematical point of view, but it is quite straightforward to switch back and forth between them and normal curves, therefore they pose no different algorithmic challenge. Generalizing train tracks to surfaces gives rise to

1. Note the close analogy with the matching equations of a normal curve.

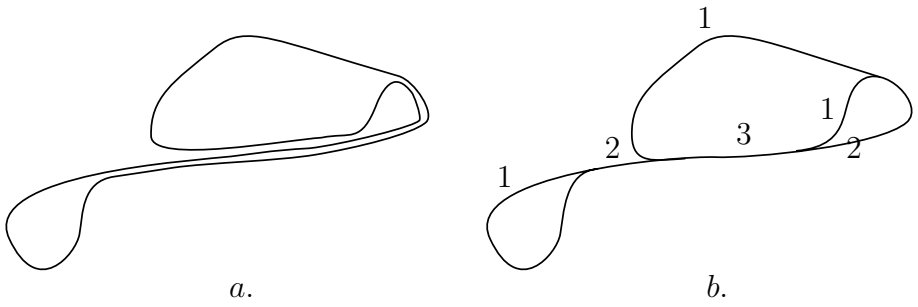


FIGURE 4.8: A closed curve (a) represented by a train track (b).

branched surfaces [100], which have been used to study embedded surfaces in 3-manifold topology [161]. Another way to encode curves is by the means of the *Dehn-Thurston coordinates* [97]: if we take a maximal family Γ of disjoint non homotopic and non contractible simple cycles on a surface (such cycles form a *pants decomposition*, see Chapter 6), it is natural to expect that counting the number of intersection of a given simple cycle with each of the cycles in Γ should give an accurate picture of its homotopy class. It turns out that if we additionally store the *twisting* of the cycle around every cycle of Γ, i.e. how many times they run parallelwise, this characterizes the homotopy class, and provides a compact representation of curves.

In Chapter 7, we study a relaxed variant of normal surfaces where quadrilaterals are allowed to intersect each other. We show that, unlike in the previous examples where the issues raised with compactness could be avoided with clever algorithmic techniques, the main problem on this class of normal surfaces, *immersibility*, is NP-hard, and therefore there is no hope to solve it efficiently.

CHAPTER 5

Isotopy of graphs on surfaces

> In this chapter, we present efficient algorithms for the following problem: Given two embeddings G_1 and G_2 of the same abstract graph G on an orientable surface S, decide whether G_1 and G_2 are isotopic; in other words, whether there exists a continuous family of embeddings between G_1 and G_2.

The results of this chapter, obtained with Éric Colin de Verdière, have been published in Discrete and Computational Geometry [A], following an earlier conference version at the Symposium on Computational Geometry. The techniques used are the same as in these articles, except in the section about isotopies with fixed vertices (Section 5.6). In order to showcase the more meaningful sides of our results, we added precisions and discussions where needed, and some technical results are deferred to Appendix A.

5.1 Introduction

This chapter follows the line of work presented in Chapter 4, and particularly in Section 4.2 where we introduced various algorithms to solve topological problems. As we saw there, topological problems on surfaces have been thoroughly investigated, since they are at the same time non-trivial, interesting, and tractable, and since the underlying mathematics, notably in the field of combinatorial group theory, are well-understood.

In this chapter, we investigate deformations of embedded graphs: Given two embeddings of a graph G on a surface, can we deform one continuously to the other without introducing intersections between edges during the process? In other words, does there exist an *isotopy* between these two graph embeddings?

This problem is motivated both by mathematical and practical purposes. On the mathematical side, it is a natural variant of the homotopy problem. Instead of testing the similarity

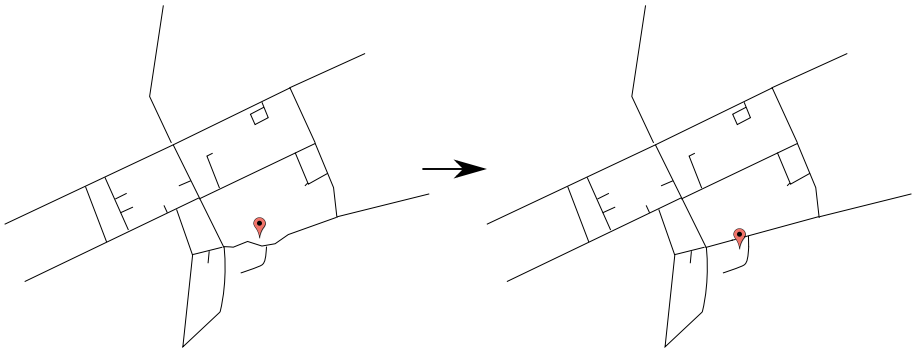


FIGURE 5.1: To design a visually appealing road network, a natural step is to straighten the roads, but one should be careful while doing so and not place meaningful features of the map (exemplified here by the pointer) on the wrong side of the road.

between two curves, we consider the more complicated structure of a graph, for which testing isotopy is a natural criterion of similarity. Graphs embedded on surfaces are a pervasive structure in geometric topology, underlying for example the description of train tracks (see Section 4.2.3.4) and Heegard diagrams [131], and therefore it is key to be able to classify them. From the point of view of topological graph theory, we also note that testing isotopy is a much finer criterion than just testing graph isomorphism, in that it looks precisely at the embedding, whereas isomorphism is just an abstract combinatorial notion.

For more practical applications, as a motivating special case, consider a finite set of obstacle points P in the plane and a graph G embedded in $\mathbb{R}^2 \setminus P$ in two different ways, G_1 and G_2. Does there exist a "morph" between G_1 and G_2 (possibly moving the vertices and bending the edges) that avoids passing over any obstacle? This is relevant for morphing applications: To compute a morphing between two images, it is helpful to first build a deformation between compatible graphs representing the most salient features of the images. In such applications, it is sometimes desirable to add some topological requirements on the morphing, e.g., to force some area of the deforming image to always cover a fixed point of the plane during the deformation. Such requirements can be encoded using obstacle points, since a face of the graph containing an obstacle point has to contain it during the whole deformation.

Another motivation for this problem comes from geographic information systems and map simplification. When simplifying a road network, it is crucial that the features of the map (cities, mountains) stay on the same side of the roads, as pictured on Figure 5.1. This can be tested by considering these features as obstacle points and testing whether the road networks obtained before and after straightening are isotopic; see, e.g., Cabello, Liu, Mantler and Snoeyink [39].

More generally, assume that we have a triangulated surface in $\mathbb{R}^3$, and two embeddings G_1 and G_2 of the same graph G on that surface (not necessarily on the skeleton of the triangulation). Each graph G_i is encoded by its combinatorial arrangement with the trian-

gulation. Can we continuously move G_1 to G_2? In this setting, the graphs G_1 and G_2 might represent textures on the surface, and the question is whether one can continuously move one texture so that it coincides with the other.

5.1.1 Related work

For a general purpose introduction to algorithmic problems stemming from topology, we refer the reader to the survey in Chapter 4. In this section, we quickly discuss the additional literature related to the specific problem of isotopy testing.

As we already saw, the related problem of testing homotopy of cycles embedded on a surface S has been investigated by Dey and Guha [71], and later Lazarus and Rivaud [164] as well as Erickson and Whittlesey [92]. They provided an algorithm to solve this problem in optimal linear time if the input curves are represented as walks in a graph embedded on S. Cabello et al. [39] give efficient algorithms for testing homotopy of paths in the special case where the surface S is a punctured plane (a plane minus a finite set of obstacle points) and the input paths are represented by polygonal paths in the plane. We will be using the algorithms in both settings as black boxes.

A neighbourly problem, to which a large body of research is devoted, is that of computing shortest homotopic paths or cycles. The study of this problem was initiated by Hershberger and Snoeyink [137] for a triangulated surface where the vertices lie on the boundary, and revisited by Efrat, Kobourov and Lubiw [77] and Bespamyatnikh [15] for paths in a punctured plane. These algorithms, in particular, allow to decide whether two paths are homotopic, and have a better complexity than the algorithm by Cabello et al. [39] in some cases. Results on the shortest homotopic path/cycle problem are also known for combinatorial surfaces [57, 58]: it is solvable in polynomial time [56].

The *mapping class group* of a surface (without boundary) S is, roughly, the set of isotopy classes of all orientation-preserving homeomorphisms from S to S – see, e.g., Farb and Margalit [96] for a recent and exhaustive survey on this topic. Although we use little of this vast theory, it is quite connected to our problem: If G_1 and G_2 are cellularly embedded on S, a homeomorphism of S that maps G_1 to G_2 represents a unique element of the mapping class group, and testing isotopy amounts to testing whether this element is the identity. Hence it is closely related to the *word problem* in mapping class groups, which can be solved in quadratic time [124, 190]. However, these algorithms take an input vastly different to ours, which hinders any possible comparison. Note that if S is a n-punctured sphere, the mapping class group of S corresponds to another classic mathematical object called the pure braid group with n strands, which has garnered considerable algorithmic attention in recent years, in particular due to its possible applications to cryptography [156].

5.1.2 Our results

The input to our algorithm is a description of the surface S, the graph G, and the graph embeddings G_1 and G_2. All surfaces are assumed to be compact, connected, and orientable, but they may have boundary. An embedding maps each vertex (or edge, or halfedge) of G to the corresponding feature on the surface. In particular, the correspondence between the vertices (or edges, or halfedges) of G_1 and G_2 is given.

Our algorithmic results come in two flavors, depending on the model used.

In the general model, G_1 and G_2 are embeddings of G on an arbitrary surface S. To represent them, we use a model similar to the cross-metric surface model introduced in Section 3.2.3.2, except that we disregard the underlying metric structure since we are only interested in the topological features of the embedding. Thus, a graph embedding is represented by the intersections it forms with a given cellularly embedded graph. We emphasize that this input does not consider the crossings betweeen G_1 and G_2.

The first result is a linear-time algorithm to decide whether G_1 and G_2 are isotopic:

Theorem 5.1.1. *Let S be an orientable surface, possibly with boundary. Let H be a fixed graph cellularly embedded on S. Let G_1 and G_2 be two graph embeddings of the same graph G on S, each in general position with respect to H. Given the combinatorial map of the arrangement of G_1 with H (resp., G_2 with H), of complexity k_1 (resp., k_2), we can determine whether G_1 and G_2 are isotopic in $O(k_1 + k_2)$ time.*

We note that the surface S is not fixed in this result; the constant in the $O(\cdot)$ notation does not depend on S.

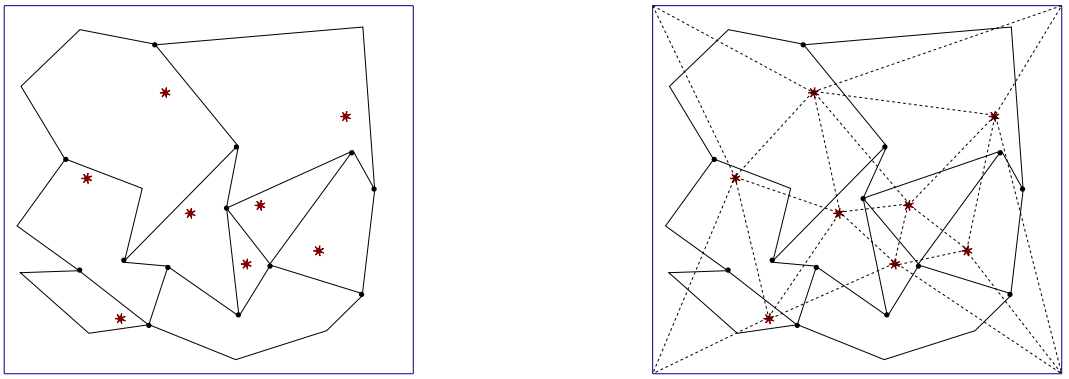
We also study the complexity of the problem in the case where S is the plane minus a finite set P of obstacle points, and G_1 and G_2 are piecewise-linear graph embeddings of G in $\mathbb{R}^2 \setminus P$. In this case, the input is the point set P together with the embeddings G_1 and G_2, where each edge of each embedding is represented as a polygonal path – again, the embeddings G_1 and G_2 may intersect arbitrarily.

In this setting, we obtain the second result:

Theorem 5.1.2. *Let P be a set of p points in the plane, and let G_1 and G_2 be two piecewise-linear graph embeddings of the same graph G in $\mathbb{R}^2 \setminus P$, of complexities (number of segments) k_1 and k_2 respectively. We can determine whether G_1 and G_2 are isotopic in $\mathbb{R}^2 \setminus P$ in time $O(n^{3/2} \log n)$ time, where n is the total size of the input. In more detail, the running time is, for any $\varepsilon > 0$,*

$$O\Big((k_1+p)\log(k_1+p)+(k_2+p)\log(k_2+p) +\min\Big\{(k_1+k_2)p\,,\; p^{1+\varepsilon}+(k_1+k_2)\sqrt{p}\log p\Big\}\Big).$$

We note that the isotopy is a continuous family of topological embeddings; one may assume that all these embeddings are piecewise-linear, but we claim no upper bound on their complexities.



a. b.

FIGURE 5.2: A graph embedded on the plane with obstacles (stars) in the two models. The cross-metric model (b.) requires an underlying cellularly embedded graph (here a triangulation) which makes it more cumbersome.

The discrepancy between the complexities of our algorithms might be startling at first: the case of planar graphs is clearly included in the case of surface-embedded graphs, yet the complexity is better in the latter case. This is justified by the difference of inputs: storing the point set of a planar embedding is more compact than the cross-metric model, as is pictured in Figure 5.2, and therefore efficient algorithms are harder to design – this is similar to the situation with compressed structures we discussed in Section 3.3.

5.1.3 Overview of the techniques

If two graph embeddings G_1 and G_2 of the same graph G are isotopic, then clearly:

1. There is an oriented homeomorphism of the surface that maps G_1 to G_2 [1];
2. if γ is a cycle in G (possibly with repeated vertices and edges), then its images in G_1 and G_2 are homotopic.

It was shown by Ladegaillerie [162] that such necessary conditions are, in fact, sufficient. However, the second condition is not algorithmic, since there are infinitely many cycles in G. A close inspection of Ladegaillerie's proof reveals that $O(g + b)$ pairs of cycles need to be tested for homotopy, where g and b denote the genus and number of boundary components of the surface. In Ladegaillerie's construction, the complexity of the family of cycles is not explicitly given. With some work, and using some of our techniques, one might be able to obtain an explicit algorithm using his construction, but the running

1. assuming that G_1 and G_2 are in the interior of S, which is true by our general position assumption.

time would certainly be larger, by at least an additional $O(g)$ factor (since Ladegaillerie's decomposition contains a *pants decomposition*, see Chapter 6), if not more.

Using a very different method, we reprove Ladegaillerie's characterization in a strengthened form, in Theorem 5.4.1 below: We provide an explicit set of cycles Λ in G of linear overall complexity of which homotopy in the two embeddings of G, in addition to the oriented homeomorphism between G_1 and G_2, suffices to ensure the isotopy of G_1 and G_2. Our algorithmic results follow, since one can perform efficiently the homeomorphism test (this essentially amounts to checking equality of two combinatorial maps, see Section 5.2.2), as well as the test of homotopy between two given cycles using the aforementioned black boxes.

We note that it is not straightforward to find a suitable set Λ in G of overall linear complexity satisfying the above condition. In particular, a natural candidate for Λ would be the set of all facial cycles in G_1 (and thus in G_2). However, Figure 5.3 shows that this family does not ensure isotopy of the graphs, even in the case where the surface is the sphere with four punctures.

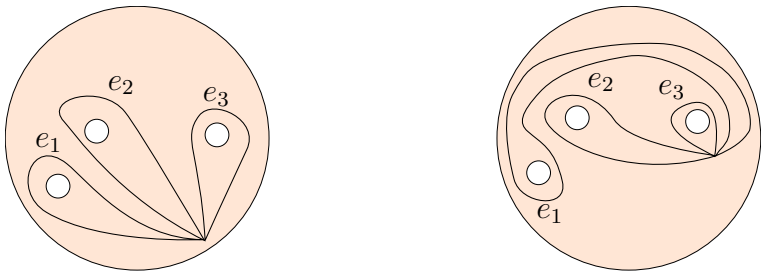


FIGURE 5.3: Two embeddings G_1 and G_2 of a one-vertex graph with three loop edges on the sphere with four punctures. These two embeddings are not isotopic, although there exists an oriented homeomorphism mapping one to the other, and the four cycles following the boundaries of the faces are homotopic in G_1 and G_2

This chapter is organized as follows. We start with some quick preliminaries (Section 5.2) introducing the few notions specific to this chapter. Then the ideas behind the computation of the family Λ and the algorithms are cut into three parts.

1. First we show that when two families of cycles are *stable*, which is a topological criterion we introduce, then testing homotopy and homeomorphism is enough to ensure isotopy between the families (Section 5.3).

2. We then compute from G_1 and G_2 two stable arrangements of cycles Λ_1 and Λ_2 in their tubular neighborhoods, which are isotopic if and only if G_1 and G_2 are isotopic. With the previous result about stable families, this proves the aforementioned strengthened form of Ladegaillerie's result (Section 5.4) [1].

1. There are some difficulties lurking behind this because the isotopy we obtain will not be pointwise –

3. We then deduce our main computational results in Section 5.5.

Finally, in Section 5.6, we briefly indicate how our algorithms and results extend to the case where we require some vertices of the graph to be fixed throughout the isotopy.

5.2 Preliminaries

In this chapter, we assume that the reader is familiar with the concepts introduced in the preliminaries, especially in Sections 3.1.1, 3.1.2 and 3.2. We introduce a handful of additional notions that will be used only in this chapter.

5.2.1 Background

Henceforth, S is a compact, connected, orientable surface with genus g and b boundary components. In this chapter, the curves we will be dealing with will mostly be *cycles*, i.e., (not necessarily simple) closed curves.

5.2.1.1 General position

A (finite) family of cycles on S is in *general position* if the cycles are in the interior of S, there are finitely many (self-)intersection points, and each intersection is a transverse crossing between exactly two pieces of cycles. Similarly, a (finite) family of graph embeddings on S is in *general position* if all embeddings are in the interior of S, there are finitely many intersection points between two different embeddings, and each intersection is a transverse crossing of the interiors of exactly two edges. Classical approximation techniques, see for example Epstein [81, Appendix], allow us to approximate every edge in a graph embedding by a piecewise linear edge using an ambient isotopy. In particular, all graph embeddings in this paper can be assumed to be piecewise-linear. By doing small perturbations if necessary, this allows us to assume that, moreover, all the graph embeddings we consider are in general position, and we will always make this assumption unless stated otherwise.

5.2.1.2 Homeomorphisms and isotopies

We will use homeomorphisms of the surface S at several occasions in this chapter, and we need to outline some subtle nuances in the terminology we use. We will often consider a homeomorphism h that *maps* a cycle $c : S^1 \to S$ into another one, $c' : S^1 \to S$. Unless stated otherwise, this expression means that $h \circ c = c'$, namely, h *pointwise* maps c to c'. However, we will sometimes need weaker concepts. We say that h *maps* c to c' not necessarily pointwise, but only *as sets*, if they do so up to reparameterization; namely, if there

see the discussion in these sections for more detail.

is a homeomorphism $\varphi : S^1 \to S^1$ such that $h \circ c \circ \varphi = c'$. If furthermore the homeomorphism φ is increasing (intuitively, h maps c to the cycle c' not necessarily pointwise, but the orientations of $h(c)$ and c' are the same), we say that h *maps* c to c' not necessarily pointwise, but *preserving the orientations* of the cycles. We also need to introduce *oriented homeomorphisms* – sometimes also called *orientation-preserving homeomorphisms*– which are homeomorphisms that preserve the orientation of a surface, i.e. such that the image of any contractible loop is a loop turning in the same direction.

Often we will also have an abstract G and two embeddings G_1 and G_2 of G on a surface S; we say that a homeomorphism h *maps* G_1 to G_2 if it maps each edge of G_1 (pointwise) to the corresponding edge of G_2. By a slight abuse of language, we will say that two graphs, or two families of cycles, are homeomorphic if there exists a homeomorphism of the surface mapping one to the other.

We recall that two homeomorphisms are *isotopic* if there exists a continuous family of homeomorphisms between them, and that an *ambient isotopy* is a homeomorphism isotopic to the identity; it follows that an ambient isotopy is oriented. For $A \subseteq S$, we say that a homeomorphism h is an ambient isotopy *relatively to* A if there is a continuous family of homeomorphisms between h and the identity such that each homeomorphism is the identity on A. This notion of isotopy naturally translates to objects embedded on surfaces: we say that two curves or graphs are *ambient isotopic* if there exists an ambient isotopy mapping one to the other.

In Section 3.1.1, we introduced a more local notion of isotopy for cycles, and this can also be done for graphs. Two embeddings G_0 and G_1 of the same abstract graph G on S are *isotopic* if there is a continuous family of embeddings $(G_t)_{t\in[0,1]}$ between G_0 and G_1. In more detail, the data of an embedding is given by the choice of a point for each vertex and a path connecting the appropriate vertices for each edge (with some conditions asserting that no crossing occurs); a family (G_t) of embeddings is continuous if all these maps vary continuously over t. (The vertices may, in particular, move.) When we talk about isotopies between families of cycles, we implicitly consider the families of cycles are graphs and use the previous definition.

It is natural to ask whether, as for cycles, the two notions of isotopy coincide. It turns out to be a byproduct of our techniques, which we obtain in Corollaries 5.4.2 and 5.4.3. See also the discussion at the end of the chapter for a pointer to another possible proof.

The following lemma will be used repeatedly in this chapter, it is one of the fundamental tools in the topology of surfaces.

Lemma 5.2.1 (Alexander's Lemma; see, e.g., Farb and Margalit [96, Lemma 2.1])**.** *Let D be a disk and $h : D \to D$ be a homeomorphism fixed on the boundary of D. Then h is an ambient isotopy relatively to the boundary of D.*

Proof We can assume that D is the unit closed disk in the plane. The continuous family

of embeddings between h and the identity can be explicitly defined by

$$F(x,t) = \begin{cases} (1-t)h(\frac{x}{1-t}) & \text{if } 0 \leq |x| \leq 1-t \\ x & \text{if } 1-t \leq |x| \leq 1 \end{cases}$$

for $0 \leq t < 1$, and $F(x,1) = x$ for each $x \in D$. □

5.2.1.3 Hyperbolic properties

Recall from Section 3.1.2 that if S has a negative Euler characteristic and has no boundary, it can be provided with a hyperbolic metric; this naturally induces a hyperbolic metric on its universal cover $\widetilde{S}$, which is then isometric to the open hyperbolic disk $\mathbb{H}^2$. Since is is often more convenient to work with compact spaces, we introduce the *compactification* of $\mathbb{H}^2$, which just amounts to adding a boundary $\partial\mathbb{H}^2$ so that the disk becomes a closed one, which we denote by $\overline{\mathbb{H}^2}$. We refer to [96, Chapter 1] for more details on this construction and we just state the properties that we will use. A non-contractible cycle on S lifts into an arc in $\mathbb{H}^2$, and adding the accumulation points on the boundaries gives a lift in $\overline{\mathbb{H}^2}$. We call its intersections with the boundary the *endpoints* of the lift or, according to the orientation of the lift, its *source* and its *target*. One can see that two such lifts in $\overline{\mathbb{H}^2}$ have common endpoints if and only if they stay at a bounded distance from each other. If S has boundary, $\widetilde{S}$ is isometric to a totally geodesic subspace of $\mathbb{H}^2$, that is, a subspace $M \subseteq \mathbb{H}^2$ such that every geodesic of M with its induced Riemannian metric is also a geodesic of $\mathbb{H}^2$.

We will be using the two following folklore facts of hyperbolic geometry, which hold for any surface S with negative Euler characteristic:

- If two cycles on a surface S are homotopic, they have lifts in $\mathbb{H}^2$ with the same endpoints [35, Lemma 1.6.5].
- Every non-contractible cycle is homotopic to a unique geodesic [96, Proposition 1.3].

5.2.2 Combinatorial maps for non-cellular embeddings

We saw in Section 3.2.2 a convenient data structure to manipulate the combinatorial maps associated to cellularly-embedded graphs. In this chapter, we will also need to handle non cellularly-embedded graphs, which leads to the following variant of combinatorial maps.

We adapt the gem representation (see Figure 5.4, it is also worthwhile to compare it to Figure 3.4 in the preliminaries) to handle graphs that are not cellularly embedded, by adding the following information. (For simplicity of exposition, we only consider graphs without isolated vertices; it is not hard to extend the data structure to handle this case.) First, each face may be incident to several cycles of G, and stores a list containing one flag of each such cycle. Conversely, each flag has a pointer to the face it belongs to. Also, a

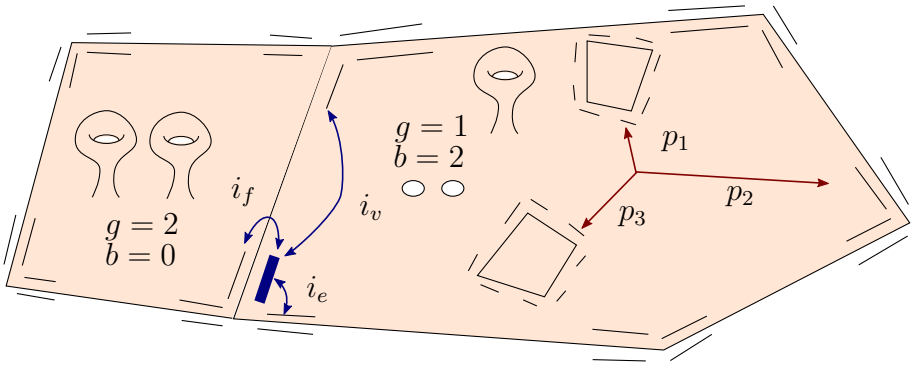


FIGURE 5.4: An example of extended combinatorial map, with the three involutions i_v, i_e, and i_f for a given flag (in bold), and the pointers p_1, p_2 and p_3 between a face and one flag for each incident cycle

face may have non-zero genus and contain some boundary components of S, so we store this genus and number of boundary components of S.

Two such *extended combinatorial maps* [1] are *isomorphic* if, in addition to the conditions above for standard combinatorial maps, the corresponding faces have the same genus and the same number of boundary components of the surface. The corresponding bijection will be called an *extended map isomorphism*. As in the cellular case, one can check whether two extended combinatorial maps of two embeddings G_1 and G_2 of G are isomorphic in linear time in their complexity. And as in the standard case, we then have the following lemma.

Lemma 5.2.2. *Two extended combinatorial maps of G_1 and G_2 are isomorphic if and only if there exists a homeomorphism of the surface mapping G_1 to G_2.*

Proof The proof is similar to that of Lemma 3.2.1. The isomorphism extends to a homeomorphism between the neighborhoods of the graphs G_1 and G_2; since each face of G_1 has the same topology as the corresponding one in G_2, this homeomorphism extends naturally to the whole surface. □

Our construction does not depend on whether S is orientable. If S is orientable, which will always be the case in this chapter, a flag has a natural *orientation*, depending on whether we turn clockwise or counterclockwise around the vertex of the flag when starting on the edge of the flag, close to the vertex, and moving towards the face of the flag. Furthermore, each of the three involutions reverses the orientation. An *orientation* of an extended combinatorial map assigns an orientation "clockwise" or "counterclockwise" to each flag such that each involution reverses the orientation of the flag. There exists an *orientation-preserving isomorphism* between two oriented extended combinatorial maps if

1. We will drop the adjective "extended" when it is obvious from the context.

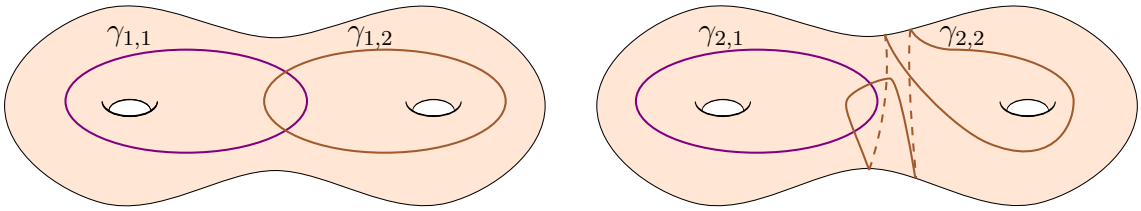


FIGURE 5.5: Two families of cycles $\Gamma_1 = (\gamma_{1,1}, \gamma_{1,2})$ and $\Gamma_2 = (\gamma_{2,1}, \gamma_{2,2})$ such that Γ_1 and Γ_2, and each cycle is homotopic to the corresponding cycle in the other family, yet Γ_1 and Γ_2 are not isotopic.

they are isomorphic as extended combinatorial maps and the bijection between the flags preserves the orientation. This can also be tested in linear time. Equivalently, there exists an oriented homeomorphism of the surface mapping one graph embedding to the other.

5.3 Isotopies of stable families of cycles

We saw that given two families of cycles Γ_1 and Γ_2 embedded on a surface S, one can efficiently test whether they are homeomorphic, and whether every cycle in Γ_1 is homotopic to the corresponding cycle Γ_2. This section addresses the problem of finding an appropriate hypothesis on the families of cycles so that these two tests are enough to ensure that there exists an isotopy between Γ_1 and Γ_2. To see that an additional hypothesis is indeed needed, we depict in Figure 5.5 an example of two families that satisfy both tests, yet are not isotopic. To circumvent these examples, we introduce the notion of a *stable* family of cycles, which leads to Theorem 5.3.1. However technical difficulties will appear: this hypothesis will not allow us to ensure isotopy pointwise, but only preserving the orientation of cycles, which in turn will lead to difficulties in the next section.

Let Γ be a family of cycles in general position on S. A *k-gon* in Γ, for $k \geq 1$, is an open disk on S whose boundary is formed by exactly k subpaths of Γ. A *0-gon* is a disk whose boundary is a single simple cycle in Γ. In general, k-gons may contain and may be crossed by other pieces of the arrangement of Γ; if this is not the case, we say that the k-gon is *empty* .

We say that Γ is *stable* if its arrangement contains no *empty* k-gon for $k \leq 3$. In this section, we prove the following result.

Theorem 5.3.1. *Let S be an orientable surface and let $\Gamma_1 = (\gamma_{1,1}, \ldots, \gamma_{1,n})$ and $\Gamma_2 = (\gamma_{2,1}, \ldots \gamma_{2,n})$ be two* stable *families of cycles on S in general position such that:*

1. *there exists an oriented homeomorphism h of S mapping each cycle $\gamma_{1,j}$ of Γ_1 to the corresponding cycle $\gamma_{2,j}$ of Γ_2 not necessarily pointwise, but preserving the orientations of the cycles, and*

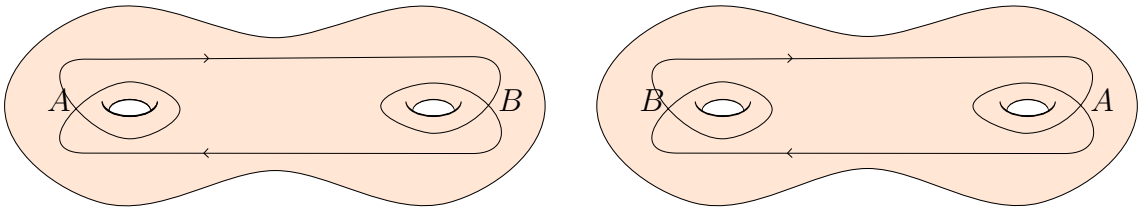


FIGURE 5.6: A stable family comprised of a single cycle $\gamma : S^1 \to S$, drawn in two different ways on a double torus. Letter A denotes the image of the same point of S^1 in both embeddings, and similarly for B. Both cycles are homotopic, some oriented homeomorphism of S maps one to the other pointwise, and there exists an isotopy of S (namely, the identity) that maps one to the other, preserving their orientation; however, there exists no isotopy of S that maps one to the other pointwise

2. *each cycle of* Γ_1 *is homotopic to the corresponding cycle of* Γ_2.

Then there is an isotopy of S *that maps each cycle of* Γ_1 *to the corresponding cycle of* Γ_2*, not necessarily pointwise, but preserving the orientations of the cycles.*

We note that conversely, if there exists an ambient isotopy of S mapping Γ_1 to Γ_2, then conditions (1) and (2) are satisfied.

As a side remark, we will actually use this result in a setting where we know that h maps each cycle of Γ_1 *pointwise* to the corresponding cycle of Γ_2; however, even under this stronger hypothesis, it does not always hold that there exists an isotopy of S that maps each cycle in Γ_1 *pointwise* to the corresponding cycle of Γ_2; see Figure 5.6.

The remaining part of this section is devoted to the proof of Theorem 5.3.1 if the surface S has negative Euler characteristic. The idea of the proof is to push the families of cycles to an arbitrarily close neighborhood of their corresponding geodesics. This is where the stability hypothesis intervenes: using an argument of de Graaf and Schrijver [64], we show that when the families are stable, this pushing can be done with isotopies. Then, the homotopy hypothesis shows that the geodesics corresponding to both families are the same, since there is a unique geodesic in every homotopy class on hyperbolic surfaces. The last ingredient is the oriented homeomorphism, which shows that in the small neighborhoods of the geodesics, which we call *corridors*, the cycles in Γ_1 and Γ_2 are in the same order. All in all, this provides an isotopy between Γ_1 and Γ_2 – but not a pointwise one.

The proof for the remaining surfaces uses slightly different tools and is deferred to Appendix A.2.

5.3.1 Basic consequences of Euler's formula

We start with simple consequences of Euler's formula that show that excluding empty k-gons is the same as excluding k-gons.

Lemma 5.3.2. *Assume just for this lemma that S is a sphere. Let G be a connected (hence cellularly embedded) graph on S, such that every vertex has degree four. Then there are at least three faces in G with degree smaller than four.*

Proof Let v, e, and f denote the number of vertices, edges, and faces of G. The sum of the degrees of all faces is $2e$, which, by Euler's formula $v - e + f = 2$ and double-counting of the vertex-edge incidences $4v = 2e$, equals $4f - 8$. Since every face has degree at least one, if at most two faces have degree smaller than four, the sum of the degrees of all faces is at least $4f - 6$. This is a contradiction. □

A trivial corollary is the following:

Corollary 5.3.3. *Assume just for this corollary that S is a sphere, a disk, or an annulus. Let G be a connected graph embedded on S, such that every vertex has degree four. Then there is at least one face in G with degree smaller than four.*

Proof Let $\bar{S}$ be the sphere obtained from S by attaching a disk to each of the boundary components of S. The graph G is embedded on $\bar{S}$, and by Lemma 5.3.2 it contains at least three faces (on $\bar{S}$) with degree smaller than four. At least one of these faces does not contain the disks attached to S, since there are at most two such disks. □

Lemma 5.3.4. *Let Γ be a stable family of cycles in a surface S. Then no k-gon can exist in Γ for $k \leq 3$.*

Proof Consider a hypothetical k-gon D for Γ, with $k \leq 3$. We prove that D strictly contains another k'-gon with $k' \leq 3$, which is a contradiction, since by induction this would give an infinite family of k-gons in a finite graph.

Assume first that D has exactly k vertices on its boundary; in other words, no cycle crosses the boundary of D. Since D cannot be an empty k-gon, there must be a connected component Γ' of the arrangement of Γ that is entirely inside D. Corollary 5.3.3 implies that Γ' contains empty k'-gons with $k' \leq 3$ inside D, and thus D contains a smaller k'-gon, as desired.

Assume now that D has at least $k + 1$ vertices on its boundary, i.e., it intersects at least another cycle in Γ. Consider the restriction of the arrangement of Γ to the closed disk D, and let G be the connected component of that restriction that contains the boundary of D. To prove the lemma, it suffices to prove that G has an interior face of degree at most three. We now assume that every interior face of G has degree at least four, and will reach a contradiction.

G has two types of edges: e_{ext} *external* edges lying on the boundary of the k-gon, and e_{int} *internal* edges. Similarly, it has three types of vertices: v_{int} vertices in the interior of

the k-gon, v_{flat} degree-three vertices on the boundary of the k-gon, and $v_{extr} = k$ degree-two vertices on the boundary of the k-gon. Let f denote the number of interior faces of G in the plane. By double-counting arguments, we obtain:

$$\begin{aligned} v_{extr} + v_{flat} &= e_{ext} && (5.1)\\ 4f &\leq 2e_{int} + e_{ext} && (5.2)\\ 4v_{int} + v_{flat} &= 2e_{int}. && (5.3) \end{aligned}$$

Euler's formula implies

$$\begin{aligned} 4 &= 4(v_{extr} + v_{flat} + v_{int}) - 4(e_{ext} + e_{int}) + 4f \\ &\overset{(5.1)}{=} 4v_{int} - 4e_{int} + 4f \\ &\overset{(5.3)}{=} -v_{flat} - 2e_{int} + 4f \\ &\overset{(5.2)}{\leq} -v_{flat} - 2e_{int} + 2e_{int} + e_{ext}. \end{aligned}$$

With equation (5.1), this gives $v_{extr} \geq 4$ which yields a contradiction and concludes the proof. □

5.3.2 Following the geodesics

We recall that $\Gamma_1 = (\gamma_{1,1}, \ldots, \gamma_{1,n})$ and $\Gamma_2 = (\gamma_{2,1}, \ldots \gamma_{2,n})$ are two stable families of cycles on S in general position satisfying the hypotheses of Theorem 5.3.1. Since we assume that the Euler characteristic of S is negative, we can endow S with a hyperbolic metric and identify $\widetilde{S}$ with a totally geodesic subspace of the open hyperbolic disk. For each j, if $\gamma_{1,j}$ is not null-homotopic, let g_j be the unique geodesic homotopic to it. (Some g_j's may be identical.)

The starting idea of our proof is that by shortening a simple cycle locally, i.e., in arbitrary small balls, one can push it arbitrarily close to its corresponding geodesic, and this pushing can be done via an isotopy. But when one tries to do that with a family of cycles, some may stand in the way of others, for example it is impossible to untie an empty 2-gon just by using isotopies. Notwithstanding, it is a result of de Graaf and Schrijver [64] that if one is further allowed to perform the subset of Reidemeister moves pictured in Figure 5.7, then one can push a family of cycles to an arbitrarily close neighborhood of their geodesics. The proof of this result is based on a slight refinement of Ringel's theorem [210] for arrangements of curves in the disk. In our case, since we are dealing with *stable* families, no Reidemeister move at all is possible, therefore one can push the cycles to a neighborhood of the geodesics by just performing isotopies. This leads to the following proposition.

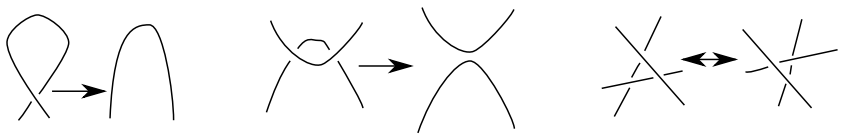


FIGURE 5.7: de Graaf and Schrijver show that one can push a family of cycles to their geodesics using only isotopies and Reidemeister moves which do not increase the number of crossings.

Proposition 5.3.5. *No cycle in Γ_1 and Γ_2 is null-homotopic. Furthermore, for $i = 1, 2$, for each $\varepsilon > 0$, up to replacing Γ_i with its image by an ambient isotopy of S, we may assume that each cycle $\gamma_{i,j}$ has a lift that belongs to an ε-neighborhood of a lift of g_j.*

Proof It follows from the aforementioned result by de Graaf and Schrijver [64, Proposition 13] that the cycles $\gamma_{i,j}$ in Γ_i can be moved through Reidemeister moves (not increasing the number of crossings) and ambient isotopies into cycles which have lifts that are ε-close to some lift of a geodesic α_j if $\gamma_{i,j}$ is not null-homotopic, and ε-close to a point of $\widetilde{S}$ otherwise. As Γ_i is stable, no Reidemeister move *at all* is possible, so the only possible moves are actually isotopies of the surface.

Now, assume that $\gamma_{i,j}$ is null-homotopic. For ε small enough, $\gamma_{i,j}$ is a contractible cycle in a disk; hence if it is simple, it forms a 0-gon; and if it is non-simple, by Corollary 5.3.3, it contains a k-gon for some $k \leq 3$. This contradicts the stability of Γ_1 and Γ_2, hence no cycle $\gamma_{i,j}$ is null-homotopic.

Now, since lifts of $\gamma_{i,j}$ and α_j are ε-close, they share the same endpoints and thus α_j is the unique geodesic homotopic to $\gamma_{i,j}$, which shows that $\alpha_j = g_j$ and concludes the proof. □

So henceforth we assume that Γ_1 and Γ_2 satisfy the conclusion of the above proposition.

The union of the geodesics g_j forms a graph, possibly with simple cycles without vertices, on the surface S; we denote by E and V its edges and vertices, see Figure 5.8. (Simple cycles without vertex are considered to be closed edges.) Each vertex has even degree, and each geodesic arriving at a vertex from an edge leaves it via the opposite edge.

Now, following ideas by de Graaf and Schrijver [64], we introduce a polygonal decomposition of an ε-neighborhood of the graph (V, E); see Figure 5.8. To each edge $e \in E$, we associate an *edge polygon* P_e (actually, a quadrilateral), and to each vertex $v \in V$ of degree $2d$, we associate a *vertex polygon* P_v with $2d$ sides, such that each edge $e = uv$ lies in the interior of $P_e \cup P_u \cup P_v$ and v lies in the interior of P_v, and such that the union of all the polygons forms a tubular neighborhood of the cycles g_i.[1] We can assume that all

1. In the case where a geodesic coincides with a boundary of the surface, one of the edges of each polygon of that geodesic actually lies on this boundary. Also, for simple cycles without vertex, we introduce an edge polygon with two opposite sides glued together.

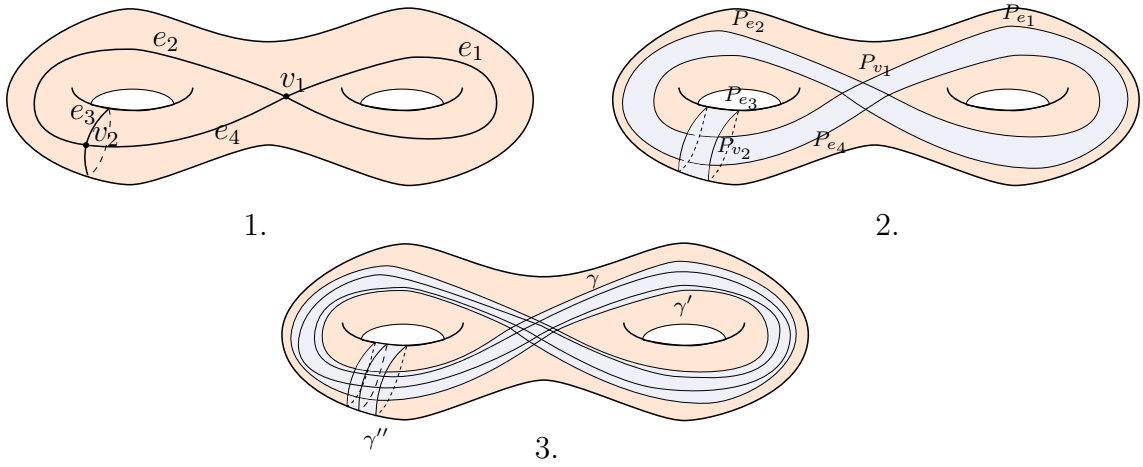


FIGURE 5.8: 1. Two geodesics on a double torus, and the associated graph. 2. The corridors associated to them. 3. Cycles in the corridors. In vertex polygons, they intersect in a grid-like pattern

the polygons are mutually disjoint, except for P_e and P_u for u a vertex incident to e, which share an edge.

Let us call a *corridor* the polygonal neighborhood of a single geodesic g_i, that is $\bigcup P_e \cup \bigcup P_u$ for all e and u in g_i. Every cycle in Γ_i belongs to a single corridor.

The following proposition is proved using Lemmas 5.3.2 and 5.3.4 and by standard flipping arguments.

Proposition 5.3.6. *For $i = 1, 2$, up to replacing Γ_i with its image under an ambient isotopy of S, we may assume that:*

- *each maximal piece of a cycle in Γ_i within a polygon is simple, and has its endpoints on opposite sides of the polygon;*
- *two such pieces cross at most once; moreover, if they cross, then the four endpoints of these two pieces are all in different sides of the polygon (in particular, the polygon is a vertex polygon).*

Proof If a maximal piece of Γ_i within a polygon is non-simple, it forms a 1-gon, which is impossible (Lemma 5.3.4). If two pieces cross twice, they form a 2-gon, which is impossible for the same reason.

If such a piece has its endpoints on two different sides of a polygon that are not opposite in that polygon, that polygon is a vertex polygon, and the corresponding cycle does not belong to a single corridor, which is impossible by construction.

If such a piece p has its endpoints on the same side s of a polygon, without loss of generality assume that the disk D bounded by s and p contains no other piece with both

endpoints on s. Thus the pieces inside D are simple, pairwise disjoint (otherwise these two pieces would form a 3-gon with p), and connect s to p; so with an isotopy of the surface, we can push p across s, decreasing the total number of intersections between the cycles and the sides of the polygon. After finitely many such operations, no such piece p exists.

There only remains to prove that there cannot be any (self-)intersection among cycles in the same corridor C. We distinguish two cases:

- If C contains at least one vertex polygon, some cycle crosses every cycle in C; if there were a (self-)intersection in C, there would be a 3-gon.
- On the other hand, if the corridor C contains no vertex polygon, then C is an annulus. Consider the arrangement of the cycles in Γ, and assume there is a crossing. One connected component of this arrangement is a graph where all vertices have degree four. Thus, by Corollary 5.3.3, Γ contains a k-gon for $k \leq 3$, contradicting Lemma 5.3.4. □

We can now assume that Γ_1 and Γ_2 satisfy the conclusion of Proposition 5.3.6. It follows that all the arcs of Γ_i in a given edge polygon P_e are simple, disjoint, and belong to different cycles. Moreover, each polygon P_v is actually a quadrilateral, because otherwise three arcs in P_v coming from six different sides would cross inside P_v, yielding a 3-gon because of the general position assumption, which is impossible (Lemma 5.3.4). Finally, within each polygon P_v, the arcs intersect in a grid-like fashion, as in Figure 5.8.

5.3.3 A technical result on corridors

We now show that the orders of the cycles in Γ_1 and Γ_2 are the same in the corridors.

Henceforth, let us choose an arbitrary orientation on S. Let C be a corridor, oriented in the direction of its geodesic g; let C_l and C_r be the left and right boundaries of C, respectively. We recall that h is the oriented homeomorphism specified in the hypotheses of Theorem 5.3.1.

Let Γ_1^C be the subfamily of cycles in Γ_1 that belong to C, and let Γ_2^C be its image by h. Recall that Γ_i^C has no crossing in an edge polygon P_e. For $i = 1, 2$, the *ordering* of Γ_i^C along C is defined as follows: Consider an arc in an edge polygon P_e, going from C_l to C_r, crossing each cycle in Γ_i^C exactly once, and record the index of the cycles in Γ_i encountered, in this order along the arc. By construction, this ordering does not depend on the choice of the polygon and arc.

Lemma 5.3.7. *The orderings of Γ_1^C and Γ_2^C are the same.*

Proof We first claim that the oriented homeomorphism $h : S \to S$ lifts to an oriented homeomorphism $\widetilde{h} : \widetilde{S} \to \widetilde{S}$. Indeed, if we denote by π the projection $\pi : \widetilde{S} \to S$

and apply the lifting theorem [1] to $h \circ \pi$, we get a continuous map $\widetilde{h} : \widetilde{S} \to \widetilde{S}$ satisfying $\pi \circ \widetilde{h} = h \circ \pi$. Let $x \in \widetilde{S}$ and $y = \widetilde{h}(x)$. Similarly, we lift h^{-1} to a continuous map $\widetilde{h^{-1}}$ such that $\widetilde{h^{-1}}(y) = x$. Then $\widetilde{h^{-1}}$ and $\widetilde{h}$ are inverse continuous maps on $\widetilde{S}$, so $\widetilde{h}$ is a homeomorphism. Furthermore, $\widetilde{h}$ is oriented because h is oriented.

Let $\widetilde{g}$ be the lift of a geodesic inside a lift $\widetilde{C}$ of C. The homeomorphism $\widetilde{h}$ maps $\widetilde{g}$ into a possibly different lift of C. However, up to composing $\widetilde{h}$ with a deck transformation of $\widetilde{S}$ [2], we may assume that $\widetilde{h}$ maps $\widetilde{g}$ into $\widetilde{C}$. Furthermore, $\widetilde{g}$ and its image by $\widetilde{h}$ have the same orientation in $\widetilde{C}$, because otherwise the endpoints of each lift of Γ_1^C in $\widetilde{C}$ would be exchanged under $\widetilde{h}$, which is not the case since h preserves the homotopy classes of the cycles in Γ_1^C. Therefore, $\widetilde{h}$ maps $\widetilde{g}$ to a path in $\widetilde{C}$ with the same source and target. Since $\widetilde{h}$ is oriented, the orderings of the cycles in $\widetilde{\Gamma}_1^C$ and $\widetilde{\Gamma}_2^C$, from left to right in $\widetilde{C}$, are the same. It follows that they are also the same in C. □

5.3.4 End of proof

Proof of Theorem 5.3.1 Let P_v and P_e be incident vertex and edge polygons, respectively, and let p be the path that is their common boundary. We first build an isotopy of the surface such that, when restricting to p, the image of each cycle in Γ_1 is the same as the corresponding cycle in Γ_2. For this purpose, note that the restriction of Γ_1 to p is a finite set of points, and similarly for Γ_2; furthermore, the numbers of points are the same (by Lemma 5.3.7). We can easily push the intersection points on p so that they coincide, by an isotopy of S that is the identity outside a neighborhood of p. Lemma 5.3.7 now implies that, after this isotopy, each arc in P_e corresponds to the same cycle in Γ_1 and Γ_2.

We can do this operation for every intersection p of a vertex and an edge polygon. Now, within each edge polygon, the arcs of Γ_1 are simple, pairwise disjoint and in the same order as the arcs of Γ_2; thus, there exists a homeomorphism from P_e to P_e that is the identity on its boundary and maps the image of Γ_1 inside P_e to the image of Γ_2 inside P_e. By Alexander's lemma, this homeomorphism is an ambient isotopy. Now, within each edge polygon P_e, the images of Γ_1 and Γ_2 are the same, and each arc corresponds to the same cycle in Γ_1 and Γ_2.

Now, within each vertex polygon P_v, the endpoints of the arcs of Γ_1 and Γ_2 coincide; moreover, they form combinatorially isomorphic arrangements of arcs (namely, grids) inside P_v. The same argument as above shows that an isotopy of P_v maps the arcs of Γ_1 to the arcs of Γ_2.

Finally, we have found an ambient isotopy i of S that maps each cycle $\gamma_{1,j}$ in Γ_1 to the

1. The *lifting theorem* is a classical result in algebraic topology giving conditions for a function to lift to one of its covers. See for example [129, Proposition 1.33] for a proof and discussion. In our case, the hypotheses are trivially fulfilled since the universal cover has trivial fundamental group.

2. Deck transformations are homeomorphisms of $\widetilde{S}$ which preserve the projection $\pi : \widetilde{S} \to S$. Therefore, composing with a deck transformation amounts to choosing the "correct" lift of h.

corresponding cycle $\gamma_{2,j}$ in Γ_2, as sets but not necessarily pointwise. Furthermore, since $i(\gamma_{1,j})$ is homotopic to $\gamma_{1,j}$, it is also homotopic to $\gamma_{2,j}$, so the ambient isotopy i preserves the orientations of the cycles. □

5.4 Isotopies of graph embeddings

In the last section, we showed that if two *stable* families of cycles pass the homotopy and the homeomorphism test, then they are isotopic, although not necessarily pointwise. The next step is thus to build two stable families of cycles Λ_1 and Λ_2 out of our input graphs G_1 and G_2, such that the isotopy of the families of cycles translates to an isotopy of the graphs. In doing this one also needs to ensure that the complexities of Λ_1 and Λ_2 is not too big compared to the ones of the input graphs. We will then apply the result of the previous section to obtain an isotopy between the stable families – however, this isotopy does not have to be pointwise, and it requires quite a bit of work to handle this difficulty.

The main result of this section is the following.

Theorem 5.4.1. *Let G_1 and G_2 be two graph embeddings of a graph G on an orientable surface S. Assume that there is an oriented homeomorphism h of S mapping G_1 to G_2. There exists a family Λ of cycles in G such that the following holds: If, for each cycle γ in Λ, the images of γ in G_1 and G_2 are homotopic, then there exists an ambient isotopy of S taking G_1 to G_2 pointwise.*

Furthermore, the cycles in Λ use each edge of G at most four times in total and, given only the combinatorial map of G_1 on S, one can compute the cycles of Λ in linear time in the complexity of that combinatorial map.

Note that, by the homeomorphism condition, the combinatorial maps of G_1 and G_2 on S have to be the same. We also emphasize that in contrast to Theorem 5.3.1, the ambient isotopy we obtain is pointwise.

Conversely, if G_1 and G_2 are isotopic (in particular, if there is an ambient isotopy taking G_1 to G_2), there must exist an oriented homeomorphism mapping one to the other, and the images of any cycle of G in G_1 and G_2 are homotopic. Therefore, Theorem 5.4.1 implies Ladegaillerie's result [162] stated in the introduction, and also:

Corollary 5.4.2. *Let G_1 and G_2 be two graph embeddings of a graph G in the interior of an orientable surface S. Assume that there exists an isotopy between G_1 and G_2. Then there exists an ambient isotopy of S between G_1 to G_2.*

In our proof of Theorem 5.4.1, if the input graph embeddings are piecewise-linear with respect to a fixed triangulation of S (which we can assume, after an ambient isotopy, by using techniques as in Epstein [81, Appendix]), our ambient isotopy can be chosen so as to be piecewise-linear. In particular:

Corollary 5.4.3. *Let G_1 and G_2 be two piecewise-linear graph embeddings of a graph G in the interior of an orientable surface S. Assume that there exists a (not necessarily piecewise-linear) isotopy between G_1 and G_2. Then there exists a piecewise-linear ambient isotopy of S between G_1 to G_2.*

For the proof of Theorem 5.4.1, one difficulty of the construction resides in the fact that the families Λ_1 and Λ_2 (the images of Λ in G_1 and G_2) must have small complexity. In a sense, Λ_1 forms a topological decomposition of the tubular neighborhood of G_1 using cycles. However, all known topological decompositions of surfaces made of cycles (like pants decompositions [57], octagonal decompositions [56], or systems of loops [91]) have worst-case complexity $\Omega(gn)$, where n is the complexity of the surface; our construction has linear size in the complexity of the object studied. We suspect that our construction can be useful for other purposes as well.

The proof of Theorem 5.4.1 starts by some preprocessing of the graphs, which reduces the problem to the case where either G_1 and G_2 are cut-graphs, or none of their faces are disks. Then we will compute stable families of cycles Γ_1 and Γ_2 in the tubular neighborhood of G_1 and G_2 that "surround" the graphs, so that isotoping them also naturally isotopes the graphs. However, this does not yield a *pointwise* isotopy. To reach that stronger conclusion, we will add a handful of cycles to the families Γ_1 and Γ_2, yielding Λ_1 and Λ_2. The somewhat intricate details of this construction are different depending on the output of the preprocessing. For the sake of readability, we only present the main part of our construction in this chapter and defer the technicalities related to the additional cycles to the Appendix A. This should help the reader to initially focus on obtaining isotopies as sets, and only look at the subtleties needed for pointwise isotopy in a second pass.

5.4.1 Preprocessing step

For the proof of Theorem 5.4.1, we assume for simplicity of exposition that G_1 and G_2 are known. It is immediate to check that, actually, only the combinatorial map of G_1 on S is needed in the constructions.

Proposition 5.4.4. *Without loss of generality, we may assume that (1) G_1 has a single face, or none of its faces is a disk, and (2) G_1 has no vertex of degree zero or one. Via the oriented homeomorphism h, the same holds for G_2.*

Intuitively, the proof is simple: Whenever e is an edge of G bounding two different faces, at least one of which is a disk, removing e in G_1 and G_2 does not change whether G_1 and G_2 are isotopic. Similarly, removing vertices of degree zero has no effect on the existence of an isotopy.

In more detail, we will need the following two lemmas.

Lemma 5.4.5. *Let e be an edge of G bounding two different faces, at least one of which is a disk, in the embedding G_1 (and thus also in G_2). Let G_1' and G_2' be the embedded graphs*

obtained after the removal of e. Then G_1 and G_2 are ambient isotopic if and only if G'_1 and G'_2 are ambient isotopic.

Proof The direct implication is obvious. Now, assume we have an isotopy i mapping G'_1 to G'_2; we want to deduce that there is an isotopy mapping e_1 to e_2 (the images of e in G_1 and G_2). By composition with i, we may assume that $G'_1 = G'_2$. By the existence of h, we know that e_1 and e_2 are arcs with the same endpoints in the same face of $G'_1 = G'_2$; furthermore, that face is split into two pieces, one of which is a disk, by e_1 (resp., e_2).

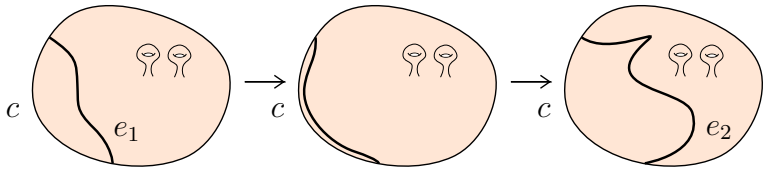


FIGURE 5.9: Illustration of the proof of Lemma 5.4.5.

The rest of the proof is illustrated in Figure 5.9. Since one of the faces bounded by e_1 is a disk, e_1 can be isotoped (with fixed extremities) to a neighborhood of the curve c closing this disk, and the same goes for e_2. After this isotopy, consider a disk neighborhood of c containing both e_1 and e_2. Since these two edges have the same endpoints in this disk, the Jordan–Schönflies theorem implies that there is a homeomorphism of the disk, fixed on the boundary, that maps one to another; then, by Alexander's lemma, that homeomorphism can be obtained by an isotopy of the disk. □

Lemma 5.4.6. *Let v be a vertex of G of degree one, and let e be its incident edge. Let G'_1 and G'_2 be the embedded graphs obtained after the removal of e and v. Then G_1 and G_2 are ambient isotopic if and only if G'_1 and G'_2 are ambient isotopic.*

Proof Again, one direction is trivial; the converse can be proved using similar ideas as the previous lemma. Here, the topological statement that is used is the following: Let p be a point on the boundary of a disk, and let e_1 and e_2 be two simple paths having p as an endpoint and intersecting the boundary of the disk exactly at p; then there is an ambient isotopy of the disk, fixed on its boundary, that maps e_1 to e_2. This again follows by an application of the Jordan–Schönflies theorem and Alexander's lemma (by first extending e_1 and e_2 to simple arcs with the same endpoints). □

Proof of Proposition 5.4.4 We show below how to build in linear time a subgraph G'' of G satisfying the desired properties and such that, if G''_1 (resp., G''_2) denotes the restriction of G_1 (resp., G_2) to G'', then G_1 and G_2 are ambient isotopic if and only if G''_1 and G''_2 are ambient isotopic. This is enough to prove the proposition.

Let $G = (V, E)$. We initially set $E' := E$, and, for each edge of E' in turn, we remove it from E' if and only if it is incident to two distinct faces of (V, E'), at least one of which

is a disk (in the embedding G_1 or G_2). This is easy to do in linear time, by initially labeling each face of (V, E') with its topology (genus and number of boundary components) and maintaining this labeling during the process.

Let G'_1 and G'_2 be the embeddings of (V, E') induced by G_1 and G_2, respectively. Lemma 5.4.5 implies that G_1 and G_2 are isotopic if and only if G'_1 and G'_2 are isotopic. Furthermore, if G'_1 has at least two faces, one of which is a disk, there exists an edge in G'_1 incident to a disk and to another face; such an edge would have been removed in the process, which is a contradiction. So the first condition is satisfied.

Moreover, we can, in linear time, iteratively remove all degree-one vertices with their incident edges, until no degree-one vertex remains. (Put all degree-one vertices in any list-type data structure; while the structure is non-empty, extract any vertex; if it still has degree one, remove it with its incident edge; if the opposite vertex on that edge has now degree one, add it to the structure; repeat.) Lemma 5.4.6 implies that G_1 and G_2 are isotopic if and only if these new graph embeddings, G''_1 and G''_2, are isotopic.

Finally, if G''_1 (and G''_2) have isolated vertices, we can safely remove them: Since there is a homeomorphism of S taking G_1 to G_2, the isolated vertices belong to the same faces in both embeddings. □

Proposition 5.4.4 leads us to distinguish two cases, leading to slightly different constructions depending on whether after the preprocessing, G_1 has a single face, or none of its faces are disks. We focus on the latter case, since it contains the most important ideas. In order to ease the reading of this section, we deferred some proofs to Appendix A. Firstly, the proof of Theorem 5.4.1 relies on Proposition 5.4.8, which will only be proved in the Appendix A.1.1. Secondly, the case when G_1 has a single face builds on the other case and is deferred to Appendix A.1.2.

5.4.2 Proof of Theorem 5.4.1 if no face is a disk

In this section, we present the main part of the proof of Theorem 5.4.1 in the special case where no face of G_1 (or, equivalently, G_2) is a disk. We can assume without loss of generality that G_1 satisfies the properties of Proposition 5.4.4.

5.4.2.1 Construction of the stable family Γ.

We first build a family Γ of cycles in G whose images in G_1 or G_2 are slight perturbations of stable families Γ_1 and Γ_2. If the images of each cycle in Γ in G_1 and G_2 are homotopic, then this almost implies that G_1 and G_2 are isotopic, which suffices to prove Theorem 5.4.1. Unfortunately, this is not entirely true, and we need to test a larger family $\Lambda \supset \Gamma$ for homotopy.

Proposition 5.4.7. *In linear time, we can construct a family of cycles Γ in G such that:*

- *each edge of G is used at most twice by all the cycles in Γ;*
- *there exists a* stable *family Γ_1 on S whose cycles are homotopic (by an arbitrarily small perturbation) to the cycles in the images of Γ in G_1;*
- *G_1 does not meet the interior of the faces of the arrangement of Γ_1 that are not disks.*

Proof It is actually simpler to explain the construction of Γ_1 first; see Figure 5.10 for an example. For simplicity of notation, we let $G_1 := (V, E)$. Recall that the *cyclomatic number* of a connected graph is the minimum number of edges one needs to delete to obtain a tree. Equivalently, it equals its number of edges minus its number of vertices plus one.

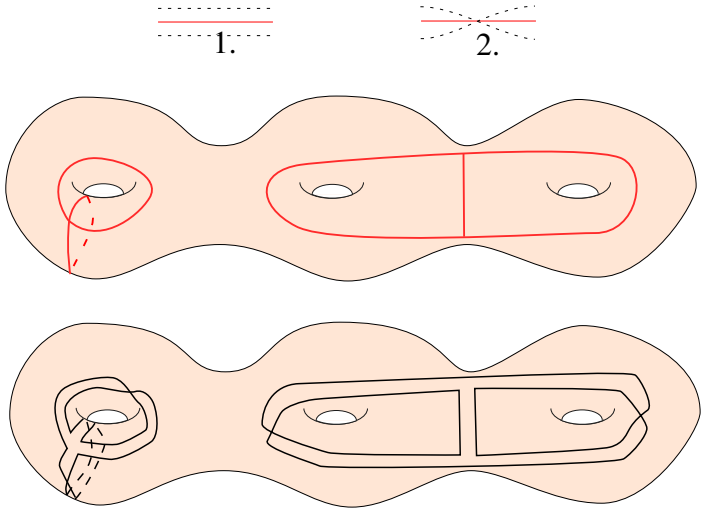


FIGURE 5.10: Top: A crossover along an edge of $E' \setminus E''$. Middle: An embedded graph G_1 on a genus 3 surface. Bottom: The corresponding family Γ_1

Let (V', E') be a connected component of (V, E). By Proposition 5.4.4, we can assume that each vertex has degree at least two; in particular, (V', E') has cyclomatic number at least one.

- If (V', E') has cyclomatic number one, then it must be a single cycle. In this case, we add that cycle to Γ_1.
- Otherwise, (V', E') has cyclomatic number at least two. We add to Γ_1 the cycles that are the boundaries of a tubular neighborhood of (V', E'). Let E'' be the edge set of a spanning tree of (V', E'). For each edge $e \in E' \setminus E''$, we introduce a "crossover" as in Figure 5.10 (top) on the two pieces of Γ_1 that run along edge e: Instead of locally having two pieces of cycles that run along edge e without touching it, we now have two pieces of cycles in the neighborhood of edge e that cross at a single interior point of e. Of course, this operation may change the number of cycles of Γ_1 and create self-intersections. See Figure 5.10.

We now prove that Γ_1 is a stable family. By construction, for each connected component, the cycles in Γ_1 do not intersect the chosen spanning tree of (V', E'). Let f be a face of the arrangement of the cycles in Γ_1. We have to prove that f is not a k-gon with $k \leq 3$. Following the definition of Γ_1, we observe that f is either an *inner disk*, namely, a disk containing entirely a spanning tree (V', E'') of some connected component of (V, E), or is contained entirely in a single face of (V, E).

- Assume first that f is an inner disk, containing the spanning tree (V', E''). By construction, (V', E') has cyclomatic number at least two, so $|E' \setminus E''| \geq 2$. Each edge $e \in E' \setminus E''$ corresponds to a single crossing between cycles of Γ_1, and this crossing appears twice along the boundary of f; so f has $2|E' \setminus E''| \geq 4$ crossings of Γ along its boundary.
- Otherwise, f has the same topology as a face of (V, E), and therefore cannot be a disk.

Hence Γ_1 is a stable family. It follows from the construction that the computation of Γ_1 takes linear time in the complexity of the combinatorial map of G_1. The cycles in Γ_1 have been constructed in a tubular neighborhood of the graph G_1; more precisely, by construction, they run along a side of the edges of G_1, swapping side whenever they run along an edge not in a spanning tree. Therefore (by retracting the tubular neighborhood) they naturally correspond to a family of cycles Γ in G, and deducing the family Γ from Γ_1 takes linear time. All these cycles use each edge of G at most twice. Furthermore, also by construction, G_1 is included in Γ_1 and in the faces of Γ_1 that are disks. □

Now, the basic idea of the proof of Theorem 5.4.1 is as follows. Assume that each cycle in Γ_1 is homotopic to the corresponding cycle in Γ_2. Theorem 5.3.1 implies that, after an ambient isotopy of S, we can assume that each cycle in Γ_1 coincides with the corresponding cycle in Γ_2 not necessarily pointwise, but with the same orientation. If this was the case pointwise, then, since G_i is "surrounded" by cycles in Γ_i, this would imply that G_1 and G_2 almost coincide and could be moved one into the other by another isotopy. However, the first isotopy does not necessarily map Γ_1 to Γ_2 pointwise, and we need to test that a few more pairs of cycles are homotopic to ensure that it is the case. This is summarized in the next proposition.

Proposition 5.4.8. *In linear time, we can construct a family of cycles Λ in G such that:*

- *each edge of G is used at most thrice by all the cycles in Λ.*
- *if we denote by Λ_1 and Λ_2 the images of Λ in G_1 and G_2, if every cycle in Λ_1 is homotopic to its counterpart in Λ_2, then an ambient isotopy of S maps Γ_1 to Γ_2 pointwise.*

For the sake of readability, the proof of Proposition 5.4.8 is deferred to Appendix A.1.1, as it gets technical and not very enlightening – though by no means easy.

5.4.2.2 End of proof of Theorem 5.4.1

We now conclude the proof of Theorem 5.4.1 if none of the faces of G_1 are disks.

According to the hypotheses, for all the cycles $\gamma \in \Lambda$, the images of γ in G_1 and G_2 are homotopic, which implies by Proposition 5.4.8 that we can assume that $\Gamma_1 = \Gamma_2$ pointwise. Then, each face of Γ_1 is mapped by h to itself, because h is an oriented homeomorphism.

In particular, $G_1 \cap \Gamma_1 = G_2 \cap \Gamma_2$. In every disk of $S \setminus \Gamma_1 = S \setminus \Gamma_2$, the oriented homeomorphism h is the identity on the boundary; therefore, by Alexander's lemma, it is an ambient isotopy relatively to the boundary. This gives us an isotopy between G_1 and G_2 on every such disk, relatively to $\Gamma_1 = \Gamma_2$. By gluing these isotopies together along their boundaries, we get an isotopy of S mapping G_1 to G_2, because G_1 and G_2 are included in the closures of the faces of $\Gamma_1 = \Gamma_2$ that are disks (Proposition 5.4.7).

Since the family Λ covers each edge of G at most thrice, it has linear complexity. As it can be computed in linear time, this concludes the proof of Theorem 5.4.1 if none of the faces of G_1 are disks.

5.4.3 Proof of Theorem 5.4.1 if the only face of G_1 is a disk

To lighten the reading of this chapter, this proof is deferred to Section A.1.2.

5.5 Algorithms

The previous sections showed that the problem of testing isotopy amounts to testing oriented homeomorphism and homotopy. Indeed, to prove Theorems 5.1.1 and 5.1.2, it suffices to be able to test the existence of an oriented homeomorphism between G_1 and G_2, and of homotopies between the cycles in Λ, as computed by Theorem 5.4.1, in the indicated amount of time. We prove Theorems 5.1.1 and 5.1.2 in Sections 5.5.1 and 5.5.2, respectively.

5.5.1 Surfaces

Recall that, in Theorem 5.1.1, the input of the algorithm consists of a fixed graph H cellularly embedded on a fixed surface S, and of embeddings G_1 and G_2 of a graph G. Furthermore, k_1 (resp., k_2) denotes the complexity of the combinatorial map of the arrangement of G_1 (resp., G_2) with H.

Homeomorphism test. For the case of graphs on surfaces, the existence of an oriented homeomorphism that maps G_1 to G_2 can be checked in $O(k_1 + k_2)$ time. Indeed, let us choose an arbitrary orientation on S; this induces an orientation of the combinatorial map of H, and hence an orientation of the combinatorial map of the arrangement of G_i and H, for $i = 1, 2$. Computing the number of boundary components of S in each face, as well as the genus of each face (using the Euler characteristic), and "erasing" the graph H in both

arrangements gives us oriented combinatorial maps for each G_i. They are isomorphic if and only if there exists an oriented isomorphism of S between G_1 and G_2 (Lemma 5.2.2), and this can be checked in linear time.

Homotopy tests. The homotopy tests can also be performed in $O(k_1 + k_2)$ time. Indeed, recall that the input to the algorithm consists of the combinatorial maps of the arrangement of G_1 and H on one hand, and of G_2 and H on the other hand, where H is a fixed cellular graph embedding; k_1 and k_2 denote the complexities of these two maps. In $O(k_1 + k_2)$ total time, we can compute the cyclically ordered list of edges of H crossed by each cycle of Λ in the embeddings G_1 and G_2. This gives us a set of pairs of cycles in the dual graph H^* of H that have to be tested for homotopy. The total complexity of these cycles is $O(k_1 + k_2)$, and H^* has complexity $O(k_1 + k_2)$ as well. Lazarus and Rivaud, and later Erickson and Whittlesey [92, 164] prove that, after a preprocessing linear in the complexity of the cellular graph H^*, one can test homotopy of cycles in H^* in time linear in the complexities of these cycles. (An earlier paper by Dey and Guha [71] claims a similar result, except for some low-genus surfaces, but Lazarus and Rivaud point out some problems in their proof.) These papers address only the case of surfaces without boundary, but the case of surfaces with boundary is easier, as the fundamental group is free; alternatively, homotopy tests for cycles on surfaces with boundary can be performed using an algorithm for surfaces without boundary by first attaching a handle to each boundary component, which does not change the outcomes of the homotopy tests.

This concludes the proof of Theorem 5.1.1.

5.5.2 Punctured plane

We now give our algorithm for the punctured plane model; so let G_1 and G_2 be two embeddings of a graph G in the punctured plane $\mathbb{R}^2 \setminus P$. Let $\mathcal{P}$ be a set of disjoint open polygons (for example squares), one around each point of P, that avoid G_1 and G_2; also, let B be a large closed square such that G_1, G_2, and the closure of $\mathcal{P}$, are in the interior of B. By compactness, any isotopy between G_1 and G_2, if it exists, must avoid neighborhoods of P and stay in a bounded area of the plane; therefore, such an isotopy exists if and only if such an isotopy exists in $B \setminus \mathcal{P}$. In other words, since $B \setminus \mathcal{P}$ is a surface with boundary, we are exactly in the topological setting of the previous sections, except that the input to the algorithm is given in a different form.

As above, our algorithm relies on two subroutines: a test for the existence of an oriented homeomorphism, and a test for homotopy between cycles. We actually give two algorithms for the latter problem, because, depending on the ratio between $k_1 + k_2$ and p, one is faster than the other.

Homeomorphism test. To test whether there exists an oriented homeomorphism of the plane that maps G_1 to G_2 in $\mathbb{R}^2 \setminus P$, we compute the oriented combinatorial map of G_1 in the punctured plane using a sweep-line algorithm for $G_1 \cup P$ in $O((k_1 + p) \log(k_1 + p))$

time [14]. Then we apply the same procedure with G_2 instead of G_1, and check that the two resulting oriented combinatorial maps are isomorphic.

First algorithm for homotopy tests. We transform the input into the surface model. For this purpose, we compute a triangulation T of B in $O(p \log p)$ time (for example, a Delaunay triangulation). We can then easily determine the arrangement of G_1 with T in $O(k_1 p)$ time, because each segment in G_1 has $O(p)$ crossings with T; that arrangement has complexity $O(k_1 p)$. We can apply the same procedure to G_2. After a slight modification of T that does not affect its complexity, we may assume that T is a triangulation of the bounding box minus a set of small square obstacles. We can then test homotopy of cycles in time linear in the number of their crossings with T, either by computing and comparing their cyclically reduced crossing words with T (since the fundamental group is a free group) or by applying the algorithm by Lazarus and Rivaud or the one by Erickson and Whittlesey. This takes $O((k_1 + k_2)p)$ time.

Second algorithm for homotopy tests. To get a subquadratic running time in the input size, we improve the homotopy test by adapting an algorithm by Cabello et al. [39, Section 4]. Their algorithm tests homotopy for paths in the punctured plane, not homotopy for cycles; however, it can be modified to handle this case also. More precisely, we show below that, after $O(p^{1+\varepsilon})$ preprocessing time (for any $\varepsilon > 0$), one can test homotopy of two (possibly non-simple) cycles γ_1 and γ_2 of complexities m_1 and m_2, respectively, in $O((m_1 + m_2)\sqrt{p} \log p)$ time.

The main idea is to replace the triangulation T in the first algorithm above with a cellular decomposition of B that has a nicer property: Each line in the plane crosses at most $O(\sqrt{p})$ segments of that decomposition. Cabello et al. [39] show how to compute such a cellular decomposition in $O(p^{1+\varepsilon})$ time. The two input cycles γ_1 and γ_2 cross this decomposition $O((m_1 + m_2)\sqrt{p})$ times. Then, computing the cyclically ordered lists of edges of the decomposition crossed by these two cycles takes $O((m_1 + m_2)\sqrt{p} \log p)$ time using ray shooting, as done also in the paper by Cabello et al. We conclude using the same method as in the first algorithm, with the cellular decomposition in place of the triangulation. This proves Theorem 5.1.2.

5.6 Graph isotopies with fixed vertices

In this section, we briefly indicate how the previous techniques extend to the graph isotopy problem where, in addition, we require the isotopy to fix some vertices. Formally, for G_1 and G_2 two embeddings of a graph $G = (V, E)$ on the interior of a surface S and a set $V_f \subseteq V$ such that the embeddings of V_f are the same in both graphs, we want to test whether there exists an isotopy h_t between G_1 and G_2 such that $h_t|_{V_f}$ is the identity for all $t \in [0, 1]$. We call this the *fixed-vertices graph isotopy problem*.

Our strategy is to reduce this problem to the usual graph isotopy problem by applying

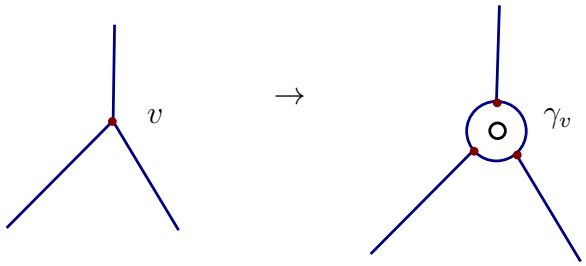


FIGURE 5.11: Gadget reducing the case of fixed vertices to the general case: A fixed vertex v of degree d is replaced with d vertices and edges forming a cycle γ_v around a new boundary component of the surface.

the gadget in Figure 5.11. It is very clear intuitively that the existence of an isotopy with fixed vertices is equivalent to the existence of a usual isotopy where every fixed vertex is replaced by the pictured gadget. It is however somewhat hard to prove under the minimal hypothesis of a continuous (i.e., not necessarily smooth nor piecewise-linear) isotopy. The key difficulty is that in this setting there are up to our knowledge no results showing that isotopies and ambient isotopies of graphs are equivalent notions. This led us to use an alternative approach in the journal article [A] corresponding to this chapter, basically following every step of the proof for the basic graph isotopy problem with slight modifications. But similarly to the situation already observed in Corollary 5.4.2, this approach actually proves the equivalence theorem that we need. For the sake of clarity, in this thesis we will take a shortcut and use this theoretical result as a black-box to prove the correctness of our reduction, and thus to give an easy solution to the fixed-vertices graph isotopy problem.

Theorem 5.6.1 ([A, Section 6])**.** *Let G_1 and G_2 be two graph embeddings of a graph G in the interior of an orientable surface S, with a specified set V of vertices in common. Assume that there exists an isotopy between G_1 and G_2 fixing V. Then there exists an ambient isotopy of S between G_1 and G_2 fixing V.*

Relying on this theorem, the proof of the correctness of the reduction easily boils down to Alexander's Lemma.

Proposition 5.6.2. *Let G_1 and G_2 be two graph embeddings of a graph G on a surface S, with a specified set V of vertices in common. Then G_1 and G_2 are isotopic with fixed vertices V if and only if G'_1 and G'_2 are isotopic, where G'_1 and G'_2 are the graphs obtained by replacing the fixed vertices with the gadget pictured in Figure 5.11.*

Proof of Proposition 5.6.2 If G_1 and G_2 are isotopic with fixed vertices V, then by Theorem 5.6.1, they are ambient isotopic, where the ambient isotopy i also fixes the vertices V. This ambient isotopy also maps G'_1 to G'_2 pointwise, except possibly for the edges of the circles added by the gadgets. But since these bound disks (on S), applying Alexander's

Lemma separately on every circle gives a pointwise isotopy between G'_1 and G'_2, as the isotopy interpolated by Alexander's Lemma fixes the new boundary component.

The reverse direction follows exactly the same proof, applying Alexander's Lemma in every disk to obtain the isotopy between G_1 and G_2, since the ambient isotopy obtained through Alexander's Lemma fixes the center of the disk if the target homeomorphism fixes it. □

Using this reduction, we obtain algorithms to test isotopy with fixed vertices, with the same guarantees on the run-time as the standard isotopy tests.

Although this reduction is conceptually simple, its correctness relies on Theorem 5.6.1, of which proof is rather long-winded. Although we did not find a rigorous proof of Theorem 5.6.1 in the literature apart from ours, the Appendix A in [166] discusses the proof of an isotopy extension theorem for graphs embedded cellularly on the sphere. It seems that the techniques used there, based on complex analysis, could be adapted to handle the general case of graphs embedded on surfaces and thus to provide a direct proof of Theorem 5.6.1.

CHAPTER 6

Discrete systolic inequalities and decompositions of triangulated surfaces

In this chapter, we investigate the following question: How much cutting is needed to simplify the topology of the surface? We provide new bounds for several instances of this question, answering an old conjecture of Przytycka and Przytycki from 1993, and we also provide a new algorithm to compute a short decomposition into *pair of pants*. These results rely on several constructions inspired by the continuous case, as well as a random model for discrete surfaces.

The results of this chapter have been obtained with Éric Colin de Verdière and Alfredo Hubard. A conference version will appear in the Proceedings of the Thirtieth Symposium on Computational Geometry [B], and it has been invited to a special issue of the journal Discrete and Computational Geometry. Compared to the conference article, this chapter is a more comprehensive account of our work, as it contains all the proofs that had been left out due to page limits.

6.1 Introduction

After studying deformations, and therefore classification, of embedded graphs in Chapter 5, we now shift our interest to a different side of discrete surfaces: the problems around decompositions. As we quickly surveyed in Section 4.1.2.3, this topic has been much studied in the recent computational topology literature; a lot of effort has been devoted towards efficient algorithms for computing shortest non-trivial curves, or shortest topological decompositions of surfaces [38, 40, 85–87, 91, 93, 160] (we also refer to the recent surveys [55, 84]). These objects provide "canonical" simplifications of surfaces, which

turn out to be crucial for algorithm design in the case of surface-embedded graphs, where making the graph planar is needed [37, 45, 47, 167]. These topological algorithms are also relevant in a number of applications that deal with surfaces with non-trivial topology, notably in computer graphics and mesh processing, to remove the topological noise on a surface [119, 253], for approximation [52] and compression [9] purposes, and to split a surface into planar pieces, for texture mapping [171, 197], surface correspondence [172], parameterization [118], and remeshing [8].

In this chapter, we study the worst-case length of shortest curves and graphs with prescribed topological properties on combinatorial surfaces. An important parameter in topological graph theory is the notion of *edge-width* of an (unweighted) graph embedded on a surface [38, 213], which is the length of the shortest closed walk in the graph that is non-contractible on the surface (cannot be deformed to a single point on the surface). The model question that we study is the following: What is the largest possible edge-width, over all triangulations with n triangles, of a closed orientable surface of genus g? It was known that an upper bound is $O(\sqrt{n/g}\log g)$ [141], and we prove that this bound is asymptotically tight, namely, that some combinatorial surfaces (of arbitrarily large genus) achieve this bound. We also study similar questions for other types of curves (non-separating cycles, null-homologous but non-contractible cycles) and for decompositions (pants decompositions, and cut-graphs with a prescribed combinatorial map), and give an algorithm to compute short pants decompositions.

We always assume that the surface has *no boundary*, that that the underlying graph of the combinatorial surface is a *triangulation*, and that its edges are *unweighted*; the curves and graphs we seek remain on the edges of the triangulation. Lifting any of these three restrictions transforms the upper bound above to a function with a linear dependency in n. In many natural situations, such requirements hold, such as in geometric modeling and computer graphics, where triangular meshes of closed surfaces are typical and, in many cases, the triangles have bounded aspect ratio (which immediately implies that our bounds apply, the constant in the $O(\cdot)$ notation depending on the aspect ratio).

Most of our results build upon or extend to a discrete setting some known theorems in *Riemannian systolic geometry*, the archetype of which is an upper bound on the systole (the length of shortest non-contractible cycles – a continuous version of the edge-width) in terms of the square root of the area of a closed Riemannian surface (or more generally the dth root of the volume of an essential Riemannian d-manifold). Riemannian systolic geometry [116, 147] was pioneered by Loewner and Pu [205], reaching its maturity with the fantastic work of Gromov [115].

After the preliminaries (Section 5.2), we prove three independent results (Sections 6.4–6.6), which are described and related to other works below. This chapter is organized so as to showcase the more conceptual results before the more technical ones. Indeed, the results of Section 6.4 exemplify the strength of the connection with Riemannian geometry, while the results in Sections 6.5 and 6.6 are perhaps a bit more specific, but feature deeper algorithmic and combinatorial tools.

6.2 Our results

Systolic inequalities for cycles on triangulations Our first result (Section 6.4) gives a systematic way of translating a systolic inequality in the Riemannian case to the case of triangulations, and vice-versa. This general result, combined with known results from systolic geometry, immediately implies bounds on the length of shortest curves with given topological properties: On a triangulation of genus g with n triangles, some non-contractible cycle has length $O(\sqrt{n/g}\log g)$, and, moreover, this bound is best possible. We also obtain the same bound for non-separating and null-homologous but non-contractible cycles.

These upper bounds are new, except for the non-contractible case, which was proved by Hutchinson [141] with a worse constant in the $O(\cdot)$ notation. The optimality of these inequalities is also new. Actually, Hutchinson [141] had conjectured that the correct upper bound was $O(\sqrt{n/g})$; Przytycka and Przytycki refuted her conjecture, building, in a series of papers [202–204], examples that show a lower bound of $\Omega(\sqrt{n\log g/g})$. They conjectured in 1993 [203] that the correct bound was $O(\sqrt{n/g}\log g)$; here, we confirm this conjecture.

Short pants decompositions A pants decomposition is a set of disjoint simple cycles that split the surface into *pairs of pants*, namely, spheres with three boundary components. In Section 6.5, we focus on the length of the shortest pants decomposition of a triangulation. As in all previous works, we allow several curves of the pants decomposition to run along a given edge of the triangulation (formally, we work in the cross-metric surface that is dual to the triangulation).

The problem of computing a shortest pants decomposition has been considered by several authors [80, 200], and has found satisfactory solutions (approximation algorithms) only in very special cases, such as the punctured Euclidean or hyperbolic plane [80]. Strikingly, no hardness result is known; the strong condition that curves have to be disjoint, and the lack of corresponding algebraic structure, makes the study of short pants decompositions hard [120, Introduction]. In light of this difficulty, it seems interesting to look for algorithms that compute short pants decompositions, even without guarantee compared the optimum solution.

Inspired by a result by Buser [35, Th. 5.1.4] on short pants decompositions of Riemannian surfaces, we prove that every triangulation of genus g with n triangles admits a pants decomposition of length $O(g^{3/2}n^{1/2})$, and we give an $O(gn)$-time algorithm to compute one. In other words, while pants decompositions of length $O(gn)$ can be computed for arbitrary combinatorial surfaces [58, Prop. 7.1], the assumption that the surface is unweighted and triangulated allows for a strictly better bound in the case where $g = o(n)$ (it is always true that $g = O(n)$). We note that the greedy approach coupled with Hutchinson's bound only gives a subexponential bound on the length of the pants decomposition, since after cutting along a short cycle, one needs to glue disks along new boundaries before finding

the new cycle to cut along [13, Introduction].

On the lower bound side, some surfaces have no pants decompositions with length $O(n^{7/6-\varepsilon})$, as proved recently by Guth, Parlier and Young [120] using the probabilistic method: They show that polyhedral surfaces obtained by gluing triangles randomly have this property.

Shortest embeddings of combinatorial maps Finally, in Section 6.6, we consider the problem of decomposing a surface using a short cut-graph with a prescribed combinatorial map. To build a homeomorphism between two surfaces, a natural approach is to cut both surfaces along a cut-graph, and put both disks in correspondence. For this approach to work, however, cut-graphs *with the same combinatorial map* are needed. In this direction, Lazarus, Pocchiola, Vegter and Verroust [163] proved that every surface has a *canonical systems of loops* (a specific combinatorial map of a cut-graph with one vertex) with length $O(gn)$, which is worst-case optimal, and gave an $O(gn)$-time algorithm to compute one.

There is, however, no strong reason to focus on canonical systems of loops: It is fairly natural to expect that other combinatorial maps will always have shorter embeddings (in particular, by allowing several vertices on the cut-graph instead of just one). However, we prove (essentially) that, for any choice of combinatorial map of a cut-graph, there exist triangulations with n triangles on which all embeddings of that combinatorial map have a *superlinear* length, actually $\Omega(n^{7/6-\varepsilon})$ (since n may be $O(g)$, there is no contradiction with the result by Lazarus et al. [163]). In particular, some edges of the triangulation are traversed $\Omega(n^{1/6-\varepsilon})$ times. This result translates to the case of polyhedral surfaces obtained by gluing together n equilateral triangles: In this model, some edges are intersected $\Omega(n^{1/6-\varepsilon})$ times. From the case of cut-graphs, we can also deduce the same results for all cellular graph embeddings with prescribed combinatorial maps.

Our proof uses the probabilistic method in the same spirit as the aforementioned article of Guth, Parlier and Young [120]: We show that combinatorial surfaces obtained by gluing triangles randomly satisfy this property asymptotically almost surely. This also sheds some light on the geometry of these "random surfaces", which have been heavily studied recently [106, 177] because of connections to quantum gravity [198] and Belyi surfaces [26]

Another view of our result is via the following problem: Given two graphs G_1 and G_2 cellularly embedded on a surface S, is there a homeomorphism $\varphi : S \to S$ such that G_1 does not cross the image of G_2 too many times? Our result essentially says that, if G_1 is fixed, for most choices of trivalent graphs G_2 with n vertices, for any φ, there will be $\Omega(n^{7/6-\varepsilon})$ crossings between G_1 and $\varphi(G_2)$. This is related to recent articles [107, 181], where upper bounds are proved for the number of crossings for the same problem, but with sets of disjoint curves instead of graphs. During their proof, Matoušek, Sedgwick, Tancer and Wagner [181] also encountered the following problem (rephrased here in the language of this chapter): For a given genus g, does there exist a *universal* combinatorial map cutting the surface of genus g into a genus zero surface (possibly with several boundaries), and with

a linear-length embedding on every such surface? We answer this question in the negative for cut-graphs.

6.3 Preliminaries

This chapter builds upon most of the concepts on the topology and geometry of surfaces that we introduced in Chapter 3. In this chapter, unless otherwise noted, all combinatorial surfaces are *triangulated* (each face is a disk with three sides) and *unweighted* (each edge has weight one). Dually, all cross-metric surfaces are *trivalent* (each vertex has degree three) and *unweighted* (each edge has crossing weight one). Furthermore, for the sake of clarity, we only consider orientable surfaces: although the results of Section 6.4 readily apply in the non-orientable case, this is not the case for our results on pants decompositions and cut-graphs. Let us also repeat that in contrast with the usual terminology used in graph theory, the cycles that we consider on embedded graphs may have repeated vertices or edges. We now introduce a few more notions specific to this work.

6.3.1 A few additional concepts

The focus on this work is on surface decompositions, and therefore it is natural to introduce these first. To that end, we will first make a detour through the 1-dimensional homology (over $\mathbb{Z}_2$). It is easy to define it for cycles embedded on combinatorial surfaces: a cycle γ embedded on S is *null-homologous* or *homologically trivial* if the faces of S can be partitioned into F_1 and F_2 such that every edge of γ is adjacent to a face of F_1 and a face of F_2. This definition extends naturally by duality to cycles embedded on cross-metric surfaces. However, we will also be dealing with cycles embedded on continuous surfaces, where defining homology gets more intricate. We refer to Hatcher [129] for the precise definitions and just provide the intuition that a cycle is null-homologous if it is the boundary of a subsurface of S.

If the cycle γ is simple, this is equivalent to being a *separating* cycle, i.e., cutting along γ gives two connected components. By a slight abuse of language, we will say *non-separating* instead of non-null-homologous, even for non-simple cycles.

We will consider three ways to decompose a triangulated surface. The first one is simply to cut it along a cycle with specific topological properties, and in particular we will be interested in *non-contractible*, *non-separating* and *null-homologous non-contractible* cycles. We will also consider cutting along a family of disjoint cycles: a *pants decomposition* of S is a family of disjoint simple closed curves Γ such that cutting S along all curves in Γ gives a disjoint union of pairs of pants. Every surface $S_{g,b}$ except the sphere, the disk, the annulus, and the torus admits a pants decomposition, with $3g+b-3$ cycles. Finally, cutting along a graph leads to the concept of cut-graphs, which we already discussed quickly in

Section 4.1.2.3: A graph G embedded on a surface S is a *cut-graph* if the surface obtained by cutting S along G is a disk.

It will be useful to estimate on the area of balls of radius r on arbitrary surfaces. In the usual Euclidean case, it is obviously πr^2, and when the metric is different, it is to be expected that the variation in area is controlled by the curvature for small values of r. This intuition was made into the following theorem, where K_p denotes the curvature at p:

Theorem 6.3.1 (Bertrand–Diquet-Puiseux [235, Chapter 3, Prop. 11])**.** *The area of the ball $B(p,r)$ of radius r centered at p equals $\pi r^2 - K_p \pi r^4 + o(r^4)$.*

Finally, if $P = \{p_1, \ldots, p_v\}$ is a set of points on a surface S, we define by

$$V_i := \{x \in (S,m) \mid \forall j \neq i, d(x,p_i) \leq d(x,p_j)\}$$

the *Voronoi regions* of the p_i, which partition S along the *Voronoi graph*. The *Delaunay graph* is the dual of the Voronoi graph, where the vertices are placed exactly at the points of P.

6.3.2 Systolic geometry

The field of *systolic geometry* revolves around finding bounds linking the length of the *systole* (or variants) of a Riemannian space, i.e., the shortest non-contractible cycle, to its volume – or area in the case of surfaces, which is the one we will be interested in. As a simple heuristic to see why such bounds ought to exist, consider a shortest non-contractible cycle γ of length ℓ, and grow cylinders of height $\ell/4$ on both sides of γ. The shortest non-contractible cycle on these cylinders will have length at least ℓ, and they will not meet as this would create a shorter non-contractible cycle as well [1]. Therefore, they are disjoint and have area at least $\ell^2/16$, see Figure 6.1. This explains why the systole should have length $O(\sqrt{A})$ where A is the area and the constant depends on g; the difficulty lying in a precise estimate of this constant.

We now collect the results from systolic geometry that we will use; for a general presentation of the field, see, e.g., Gromov [116] or Katz [147].

Theorem 6.3.2 ([36, 115, 116, 148, 221])**.** *There are constants $c, c', c'', c''' > 0$ such that, on any Riemannian surface with genus $g \neq 0$ and area A:*

1. *some non-contractible cycle has length at most $c\sqrt{A/g}\log g$;*
2. *some non-separating cycle has length at most $c'\sqrt{A/g}\log g$;*

1. We take $\ell/4$ instead of $\ell/2$ because we might need to follow a half of the systole to link the paths in both cylinders.

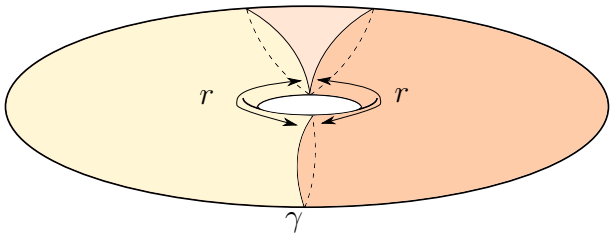


FIGURE 6.1: A systole of length ℓ and cylinders of height r grown from both sides. If $r \leq \ell/4$, the cylinders are disjoint, otherwise we could find a shorter non-contractible cycle.

3. *If $g \geq 2$, some null-homologous non-contractible cycle* [1] *has length at most $c'' \sqrt{A/g} \times \log g$.*

Furthermore,

4. *for an infinite number of values of g, there exist Riemannian surfaces of constant curvature -1 (hence area $A = 4\pi(g-1)$) and systole larger than $\frac{2}{3\sqrt{\pi}} \sqrt{A/g} \log g - c'''$. In particular, the three previous inequalities are tight up to constant factors.*

The bounds (1) and (2) are due to Gromov [115, 116], (3) is due to Sabourau [221], and (4) is due to Buser and Sarnak [36, p. 45]. Furthermore, Gromov's proof yields $c = 2/\sqrt{3}$ in (1), which has been improved asymptotically by Katz and Sabourau [148]: They show that for every $c > 1/\sqrt{\pi}$ there exists some integer g_c so that (1) is valid for every $g \geq g_c$.

6.4 A two-way street

In this section, we prove that any systolic inequality regarding closed curves in the continuous (Riemannian) setting can be converted to the discrete (triangulated) setting, and vice-versa.

6.4.1 From continuous to discrete systolic inequalities

Theorem 6.4.1. *Let (S, G) be a triangulated combinatorial surface of genus g, without boundary, with n triangles. Let $\delta > 0$ be arbitrarily small. There exists a Riemannian metric m on S with area n such that for every cycle γ in (S, m) there exists a homotopic closed curve γ' on (S, G) with $|\gamma'|_G \leq (1+\delta)\sqrt[4]{3}\,|\gamma|_m$.*

This theorem, combined with the aforementioned theorems from systolic geometry, immediately implies:

1. If $g = 1$, no non-contractible cycle is null-homologous, because the fundamental group and the 1-dimensional homology group over $\mathbb{Z}$ are the same.

Corollary 6.4.2. *Let (S, G) be a triangulated combinatorial surface with genus g and n triangles, without boundary. Then, for some absolute constants c, c', and c'':*

1. *some non-contractible cycle has length at most $c\sqrt{n/g}\log g$;*
2. *some non-separating cycle has length at most $c'\sqrt{n/g}\log g$;*
3. *If $g \geq 2$, some null-homologous non-contractible cycle has length at most $c''\sqrt{n/g} \times \log g$.*

Proof of Corollary 6.4.2 The proof consists in applying Theorem 6.4.1 to (S, G), obtaining a Riemannian metric m. For each of the different cases, the appropriate Riemannian systolic inequality is known, which means that a short curve γ of the given type exists on (S, m) (Theorem 6.3.2(1–3)); by Theorem 6.4.1, there exists a homotopic curve γ' in (S, G) such that $|\gamma'|_G \leq (1+\delta)\sqrt[4]{3}\,|\gamma|_m$, for any $\delta > 0$. □

Plugging in the best known constants for Theorem 6.3.2 (1) allows us to take $c = 2/\sqrt[4]{3}$, or any $c > \sqrt[4]{3/\pi^2}$ asymptotically using the refinement of Katz and Sabourau.

Furthermore, we note that, by Euler's formula and double-counting, we have $n = 2v + 4g - 4$, where v is the number of vertices of G. Thus, on a triangulated combinatorial surface with $v \geq g$ vertices, the length of a shortest non-contractible cycle is at most $2\sqrt{2}\sqrt[4]{3} \cdot \sqrt{v/g}\log g < 3.73\sqrt{v/g}\log g$. This reproves a theorem of Hutchinson [141], except that her proof technique leads to the weaker constant 25.27. This constant can be improved asymptotically to $\sqrt[4]{108/\pi^2} < 1.82$ with the aforementioned refinement.

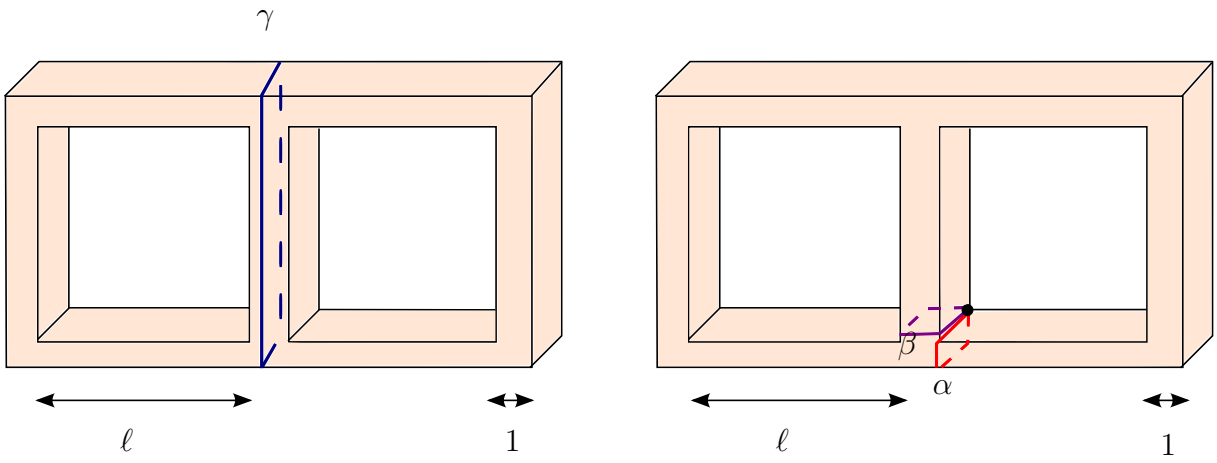


FIGURE 6.2: A piecewise linear double torus with area A such that the length of a shortest splitting cycle is $\Omega(A)$ (left), but the length of a shortest homologically trivial non-contractible curve, concatenation of $\alpha\beta\alpha^{-1}\beta^{-1}$, has length $\Theta(1)$.

We also remark that, in (3), we cannot obtain a similar bound if we require the curve to be simple (and therefore to be *splitting* [46]). Indeed, Figure 6.2 shows that the minimum length of a shortest homologically trivial, non-contractible cycles can become much larger if we additionally request the curve to be simple.

Proof of Theorem 6.4.1 The first part of the proof is similar to Guth, Parlier and Young [120, Lemma 5]. Define m_G to be the singular Riemannian metric given by endowing each triangle of G with the geometry of a Euclidean equilateral triangle of area 1 (and thus side length $2/\sqrt[4]{3}$): This is a genuine Riemannian metric except at a finite number of points, the set of vertices of G. The graph G is embedded on (S, m_G). Let γ be a cycle $\gamma\colon \mathbb{S}^1 \to S$. Up to making it longer by a factor at most $\sqrt{1+\delta}$, we may assume that γ is piecewise linear and transversal to G. Now, for each triangle T and for every maximal part p of γ that corresponds to a connected component of $\gamma^{-1}(T)$, we do the following. Let x_0 and x_1 be the endpoints of p on the boundary of T. (If γ does not cross any of the edges of G, then it is contractible and the statement of the theorem is trivial.) There are two paths on the boundary of T with endpoints x_0 and x_1; we replace p with the shorter of these two paths. Since T is Euclidean and equilateral, elementary geometry shows that these replacements at most doubled the lengths of the curve. Now, the new curve lies on the graph G. We transform it with a homotopy into a no longer curve that is an actual closed walk in G, by simplifying it each time it backtracks. Finally, from a closed curve γ, we obtained a homotopic curve γ' that is a walk in G, satisfying $|\gamma'|_G = \sqrt[4]{3}/2\,|\gamma'|_{m_G} \leq \sqrt{1+\delta}\sqrt[4]{3}\,|\gamma|_{m_G}$.

The metric m_G satisfies our conclusion, except that it has isolated singularities. However, we show in Lemma 6.4.3 that it is possible to smooth and scale m_G to obtain a metric m, also with area n, that multiplies the length of all curves by at least $1/\sqrt{1+\delta}$ compared to m_G. This metric satisfies the desired properties, and this concludes the proof. □

There remains to explain how to smooth the metric. There is a very classic way to do this using a *partition of unity*, which is a set R of topological functions from S to $[0,1]$ such that for every $x \in S$,

- there exists a neighborhood N of x such that only a finite number of functions of R are nonzero on N;
- $\sum_{\rho\in R} \rho(x) = 1$.

It is then a standard result [218, Theorem 2.13] that for any open cover $(U_i)_{i\in I}$ of S, there exists a partition of unity $R = (\rho_i)_{i\in I}$ such that for any i in I, the support of ρ_i is contained in U_i.

Lemma 6.4.3. *With the notations of the proof of Theorem 6.4.1, there exists a smooth Riemannian metric m on S, also with area n, such that any cycle γ in S satisfies $|\gamma|_m \geq |\gamma|_{m_G}/\sqrt{1+\delta}$.*

Proof The idea is to smooth out each vertex v of G to make m_G Riemannian, as follows.

On the ball $B(v, 2\varepsilon)$, consider a Riemannian metric m_v with area at most $\delta/3$ such that any path in that ball is longer under m_v than under m_G. This is certainly possible provided ε is small enough: For example, build a diffeomorphism from $B(v, 2\varepsilon)$ onto the unit disk in the plane in the natural way (v being mapped at the center of the disk, and the trace of the edges of G being mapped to line segments forming equal angles); endow the disk with a metric just large enough so that the corresponding metric on $B(v, 2\varepsilon)$ is larger than m_v. If ε is taken small enough, the area that is needed for the new metric can be made as small as we want.

We now use a partition of unity to define a smooth metric $\hat{m}$ that interpolates between m_G and the metrics m_v. By choosing an appropriate open cover, and therefore an appropriate partition of unity ρ, we obtain a metric $\hat{m} = \rho_G m_G + \sum_{v \in V} \rho_v m_v$ such that:

- outside the balls centered at a vertex v of radius 2ε, we have $\hat{m} = m_G$;
- inside a ball $B(v, \varepsilon)$, we have $\hat{m} = m_v$;
- in $B(v, 2\varepsilon) \setminus B(v, \varepsilon)$, the metric $\hat{m}$ is a convex combination of m_G and m_v.

The area of $\hat{m}$ is at most the sum of the areas of m_G and the m_v's, which is at most $n(1+\delta)$. Moreover, for any curve γ, we have $|\gamma|_{\hat{m}} \geq |\gamma|_{m_G}$.

Finally, we scale $\hat{m}$ to obtain the desired metric m with area n; for any curve γ, we indeed have $|\gamma|_m \geq |\gamma|_{\hat{m}}/\sqrt{1+\delta}$. □

6.4.2 From discrete to continuous systolic inequalities

Here we prove that, conversely, discrete systolic inequalities imply their Riemannian analogues. The idea is to approximate a Riemannian surface by the Delaunay triangulation of a dense set of points, and to use some recent results on intrinsic Voronoi diagrams on surfaces [74].

Theorem 6.4.4. *Let (S, m) be a Riemannian surface of genus g without boundary, of area A. Let $\delta > 0$. For infinitely many values of n, there exists a triangulated combinatorial surface (S, G) embedded on S with n triangles, such that every cycle γ in (S, G) satisfies $|\gamma|_m \leq (1+\delta)\sqrt{\frac{32}{\pi}}\sqrt{A/n}\,|\gamma|_G$.*

We have stated this result in terms of the number n of triangles; in fact, in the proof we will derive it from a version in terms of the number of vertices; Euler's formula and double counting imply that, for surfaces, the two versions are equivalent. Together with Hutchinson's theorem [141], this result immediately yields a new proof of Gromov's classical systolic inequality:

Corollary 6.4.5. *For every Riemannian surface (S, m) of genus g, without boundary, and area A, there exists a non-contractible curve with length at most $\frac{101.1}{\sqrt{\pi}}\sqrt{A/g}\log g$.*

Proof Let $\delta > 0$, and let (S, G) be the triangulated combinatorial surface implied by Theorem 6.4.4 with $n \geq 6g - 4$ triangles. Euler's formula implies that the number v of vertices of G is at least g, hence we can apply Hutchinson's result [141], which yields a non-contractible curve γ on G with $|\gamma|_G \leq 25.27\sqrt{(\frac{n}{2} + 2 - 2g)/g}\log g$. By Theorem 6.4.4, $|\gamma|_m \leq \frac{101.08(1+\delta)}{\sqrt{\pi}}\sqrt{A/g}\log g$. □

On the other hand, using this theorem in the contrapositive together with the Buser–Sarnak examples (Theorem 6.3.2(4)) confirms the conjecture by Przytycka and Przytycki [203, Introduction]:

Corollary 6.4.6. *For any $\varepsilon > 0$, there exist arbitrarily large g and v such that the following holds: There exists a triangulated combinatorial surface of genus g, without boundary, with v vertices, on which the length of every non-contractible cycle is at least $\frac{1-\varepsilon}{6}\sqrt{v/g}\log g$.*

Proof Let $\varepsilon > 0$, let (S, m) be a Buser–Sarnak surface from Theorem 6.3.2(4), and let G be the graph obtained from Theorem 6.4.4 from (S, m), for some $\delta > 0$ to be determined later. Combining these two theorems, we obtain that every non-contractible cycle γ in G satisfies

$$(1+\delta)\sqrt{\frac{32}{\pi}}\sqrt{\frac{A}{n}}\,|\gamma|_G \geq \frac{2}{3\sqrt{\pi}}\sqrt{\frac{A}{g}}\log g - c''',$$

where $A = 4\pi(g-1)$. If δ was chosen small enough (say, such that $1/(1+\delta) \geq 1 - \varepsilon/2$), and g was chosen large enough, we have $|\gamma|_G \geq \frac{1-\varepsilon}{3\sqrt{8}}\sqrt{\frac{n}{g}}\log g$. Finally, we have $n \geq 2v$ by Euler's formula. □

Before delving intro the proof of Theorem 6.4.4, we make a little detour to introduce a Riemannian notion that we will need. The *strong convexity radius* at a point in a Riemannian surface (S, m) is an invariant that refines the well-known injectivity radius. It is the supremum of the radius ρ_x such that for every $r < \rho_x$ the ball of radius r centered at x is strongly convex, that is, for any $p, q \in B(x, r)$ there is a unique shortest path in (S, m) connecting p and q, this shortest path lies entirely within $B(x, r)$, and moreover no other geodesic connecting p and q lies within $B(x, r)$, we refer to Klingenberg [155, Def. 1.9.9] for more details. The strong convexity radius is positive at every point, and its value on the surface is continuous (see also Dyer, Zhang and Möller [74, Sect. 3.2.1]). It follows that for every compact Riemannian surface (S, m), there exists a strictly positive lower bound on the strong convexity radius of every point. We will need the following lemma, which is a result of of Dyer, Zhang and Möller [74, Corollary 2].

Lemma 6.4.7. *Let (S, m) be a Riemannian surface, let $\rho > 0$ be smaller than the half of the strong convexity radius of any point in (S, m), and let P a point set of S in general position such that for every x on S, there exists a point p of P such that $d_m(x, p) \leq \rho$. Then the Delaunay graph of P is a triangulation of S.*

Proof of Theorem 6.4.4 Let η, $0 < \eta < 1/2$ be fixed, and $\varepsilon > 0$ to be defined later (depending on η). Let P be an ε-separated net on (S, m), that is, P is a point set such that any two points in P are at distance at least ε, and every point in (S, m) is at distance smaller than ε from a point in P. For example, if we let P be the centers of an inclusionwise maximal family of disjoint open balls of radius $\varepsilon/2$, then P is an ε-separated net. In the following we put P in general position by moving the points in P by at most $\eta\varepsilon$; in particular, no point in the surface is equidistant with more than three points in P.

Let $P = \{p_1, \ldots, p_v\}$, and denote by V_i be the Voronoi region of p_i. Since every point of (S, m) is at distance at most $(1 + \eta)\varepsilon$ from a point in P, each Voronoi region V_i is included in a ball of radius $(1 + \eta)\varepsilon$ centered at p_i. Note that if $V_i \cap V_j \neq \emptyset$, then the corresponding neighboring points of the Delaunay graph are at distance at most $2(1 + \eta)\varepsilon$.

It turns out that under these assumptions, and choosing ε smaller than $1/(1 + \eta)$ times the strong convexity radius of (S, m), the Delaunay graph, which we denote by G, can be embedded as a triangulation of S with shortest paths representing the edges; this follows from Lemma 6.4.7 with ε small enough so that $(1 + \eta)\varepsilon \leq \rho$.

Consider a cycle γ on G. Since neighboring points in G are at distance no greater than $2(1 + \eta)\varepsilon$ on (S, m), we have $|\gamma|_m \leq 2(1 + \eta)\varepsilon|\gamma|_G$. To obtain the claimed bound, there remains to estimate the number v of points in P. By compactness, the Gaussian curvature of (S, m) is bounded from above by a constant K. By the Bertrand–Diquet–Puiseux theorem, the area of each ball of radius $\frac{1-2\eta}{2}\varepsilon$ is at least $\pi(1 - 2\eta)^2\frac{\varepsilon^2}{4} - K\pi(1 - 2\eta)^4\frac{\varepsilon^4}{16} + o(\varepsilon^4) \geq \pi(1-2\eta)^3\frac{\varepsilon^2}{4}$ if $\varepsilon > 0$ is small enough. Since the balls of radius $(1-2\eta)\frac{\varepsilon}{2}$ centered at P are disjoint, their number v is at most $A/(\pi(1-2\eta)^3\frac{\varepsilon^2}{4})$. In other words, $\varepsilon \leq \frac{2}{\sqrt{\pi(1-2\eta)^3}}\sqrt{A/v}$. Putting together our estimates, we obtain that

$$|\gamma|_m \leq \frac{4(1+\eta)}{\sqrt{\pi(1-2\eta)^3}}\sqrt{\frac{A}{n/2-2g+2}}\,|\gamma|_G,$$

where n is the number of triangles of G. Thus, if $\varepsilon > 0$ is small enough, n can be made arbitrarily large, and the previous estimate implies, if η was chosen small enough (where the dependency is only on δ) that $|\gamma|_m \leq (1 + \delta)\sqrt{\frac{32}{\pi}}\sqrt{\frac{A}{n}}\,|\gamma|_G$. □

6.5 Computing short pants decompositions

Recall that the problem of computing a shortest pants decomposition for a given surface is open, even in very special cases. In this section, we describe an efficient algorithm that computes a short pants decomposition on a triangulation. Technically, we allow several curves to run along a given edge of the triangulation, which is best formalized in the dual cross-metric setting. If g is fixed, the length of the pants decomposition that we compute is of the order of the square root of the number of vertices:

Theorem 6.5.1. *Let (S, G^*) be an (unweighted, trivalent) cross-metric surface of genus $g \geq 2$, with n vertices, without boundary. In $O(gn)$ time, we can compute a pants decomposition $(\gamma_1, \dots, \gamma_{3g-3})$ of S such that, for each i, the length of γ_i is at most $C\sqrt{gn}$ (where C is some universal constant).*

Actually, we can obtain that the length of γ_i is at most $C\sqrt{in}$ with a little more effort, but for the sake of clarity we focus on the weaker bound.

The inspiration for this theorem is a result by Buser [35], stating that in the Riemannian case, there exists a pants decomposition with curves of length bounded by $3\sqrt{gA}$. The proof of Theorem 6.5.1 consists mostly of translating Buser's construction into the discrete setting and making it algorithmic. The key difference is that for the sake of efficiency, unlike Buser, we cannot afford to shorten the cycles in their homotopy classes, and we have to use contractibility tests in a careful manner.

Given cycles Γ in general position on a (possibly disconnected) cross-metric surface (S, G^*), cutting S along Γ, and/or restricting to some connected components, gives another surface S', and restricting G^* to S' naturally yields a cross-metric surface that we denote by $(S', G^*_{|S'})$. Also, to simplify notation, we denote by $|c|$ (instead of $|c|_{G^*}$) the length of a curve c on a cross-metric surface (S, G^*).

The main tool is to cut off a pair of pants of a surface with boundary, while controlling the length of the boundary of the new surface:

Proposition 6.5.2. *Let (S, G^*) be a possibly disconnected cross-metric surface, such that every connected component has non-empty boundary and admits a pants decomposition. Let n be the number of vertices of G^* in the interior of S. Assume moreover that $|\partial S| \leq \ell$, where ℓ is an arbitrary positive integer.*

We can compute a family Δ of disjoint simple cycles of (S, G^*) that splits S into one pair of pants, zero, one, or more annuli, and another possibly disconnected surface S' containing no disk, such that $|\partial S'| \leq \ell + 2n/\ell + 8$. The algorithm takes as input (S, G^*), outputs Δ and $(S', G^*_{|S'})$, and takes linear time in the complexity of (S, G^*).

We first show how Theorem 6.5.1 can be deduced from this proposition: It relies on computing a good approximation of the shortest non-contractible cycle, cutting along it, and applying Proposition 6.5.2 inductively.

Proof of Theorem 6.5.1 To prove Theorem 6.5.1, we consider our cross-metric surface without boundary (S, G^*), and we start by computing a simple non-contractible curve γ whose length is at most twice the length of the shortest non-contractible cycle. Such a curve can be computed in $O(gn)$ time [38, Prop. 9] (see also Erickson and Har-Peled [86, Corollary 5.8]) and has length at most $C\sqrt{n}$, where C is a universal constant, see Section 6.4. This gives a surface $S^{(1)}$ with two boundary components.

The end of the proof just consists of applying Proposition 6.5.2 inductively: We start with $S^{(1)}$, and applying it to $S^{(k)}$ gives another surface $S^{(k)\prime}$, in which we remove all the pair

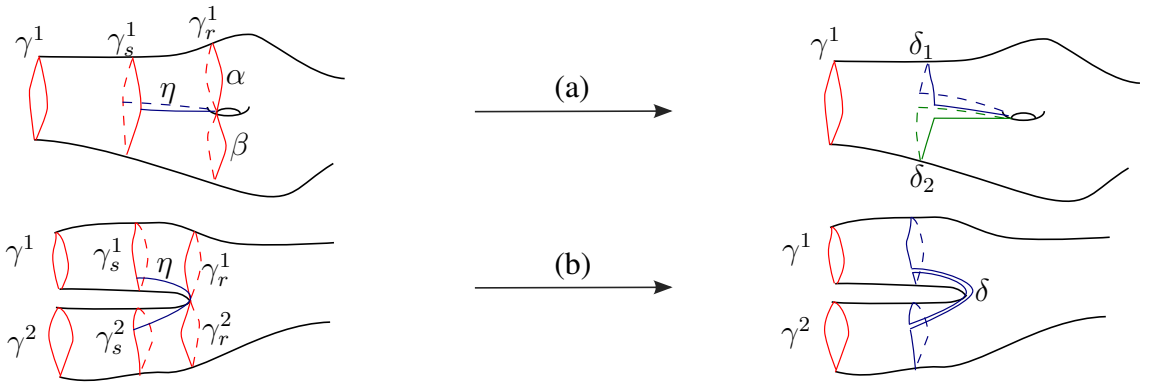


FIGURE 6.3: (a) Splitting phase. (b) Merging phase.

of pants. We denote the resulting surface by $S^{(k+1)}$ and apply Proposition 6.5.2 again. We apply this induction until we obtain a surface $S^{(m)}$ that is empty. Note that, for every k, $S^{(k)}$ contains no disk, annulus, or pair of pants, and that every application of Proposition 6.5.2 gives another pair of pants. Therefore, we obtain a pants decomposition of S by taking the initial curve γ together with all the curves in Δ in all the applications of Proposition 6.5.2 and, when there are homotopic curves, by removing all of them except the shortest one. Therefore, the number of applications of Proposition 6.5.2 is bounded by the maximum size of a pants decomposition of S, i.e., $3g-3$. The length of the pants decomposition is at most the sum, over k, of $\ell_k = |\partial S^{(k)}|$. The sequence ℓ_k satisfies the induction $\ell_{k+1} \leq \ell_k + 2n/\ell_k + 8$, with $\ell_1 \leq C\sqrt{n}$. A small computation gives that $\ell_k \leq C\sqrt{kn}$ for C larger than 16 and $k \leq 3n$, which proves the bound on the lengths since $k \leq 3g-3 \leq 3n$. The total complexity of this algorithm is $O(gn)$ since we applied $O(g)$ times Proposition 6.5.2, which takes linear complexity. □

Now, onwards to the proof of the main proposition.

Proof of Proposition 6.5.2 The idea is to *shift* the boundary components simultaneously until one boundary component *splits*, or two boundary components *merge*. This is analog to Morse theory on the surface with the function that is the distance to the boundary. However, in order to control the length of the decomposition, some backtracking is done before splitting or merging, as pictured in Figure 6.3.

Let $\Gamma = (\gamma_0^1, \ldots, \gamma_0^k)$ be (curves infinitesimally close to) the boundaries of S. Initially, let $\gamma^i = \gamma_0^i$. We orient each γ^i so that it has the surface to its right at the start. We will shift these curves to the right while preserving their simplicity and homotopy classes. We will only describe how Δ is computed, since one directly obtains S' by cutting along Δ and discarding the annuli and one pair of pants.

Shifting phase: We say that two simple cycles on (S, G^*) are *tangent* if they both have a subpath in a common face of G^*. When a single cycle has two subpaths in the same

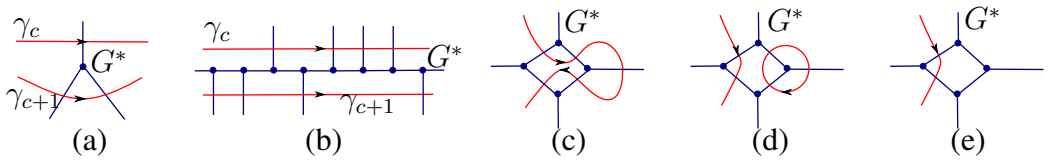


FIGURE 6.4: (a) Pushing a curve across a vertex. (b) The effect of a shifting step, if no self-tangency or tangency occurs. (c) A portion of a self-tangent curve. (d) The corresponding subcurves. (e) The curve after the removal of contractible subcurves.

face of G^*, it will be called a *self-tangent* cycle. The curves we handle in this phase are simple and homotopic to the γ^i. Since each such curve is separating, in a self-tangency, the two portions of a curve are oppositely oriented (Figure 6.4(c)). Therefore, "rewiring" such a curve at a self-tangency naturally splits it into two tangent cycles, which we call its *subcurves*, see Figure 6.4(d).

We define below how we shift a curve by one step to the right. The whole shifting phase consists of shifting the curves in a round robin way, i.e., we shift γ^1 by one step, then $\gamma^2, \ldots, \gamma^k$, and we reiterate. This phase is interrupted immediately whenever some tangency or self-tangency occurs, see below. To shift γ^i by one step, for every successive edge of G^* crossed by γ^i, in the order induced by γ^i, we push γ^i across the vertex adjacent to the edge (Figure 6.4(a)). The result of a shifting step is shown in Figure 6.4(b). Since G^* is trivalent, tangencies appear one at a time, determined by only two portions of curves. As soon as there is one (including before the very first step), we do the following:

- If γ^i is self-tangent, we test the two resulting subcurves for contractibility. If one of them is contractible, we discard it (Figure 6.4(e)) and continue the shifting process with the other one. Otherwise, both are non-contractible, and we go to the splitting phase below.
- If γ^i is tangent to γ^j for some $j \neq i$, we go to the merging phase below.

This finishes the description of the shifting phase. Let r be the integer such that each curve has been shifted between r and $r+1$ steps to the right. For each i, $1 \leq i \leq k$, and each c, $1 \leq c \leq r$, let γ^i_c be the curve γ^i shifted by c steps. At every step of the shifting phase, we also maintain the sum of the lengths of the current curves. Then, at the end we denote by s the largest $c \leq r$ such that $\sum_{i=1}^{k} |\gamma^i_c| \leq \ell$. (Remember that this is the case for $c = 0$ by hypothesis.)

Splitting phase: When a curve becomes self-tangent, we do a splitting, as is pictured on the top of Figure 6.3. For simplicity, let γ^1 denote the curve that became self-tangent during the shifting phase. First, for every $i \neq 1$, we add γ^i_s to the family Δ. During the shifting phase, the closed curve γ^1 split into two non-contractible cycles α and β. Let η be the shortest path with endpoints on γ^1_s that goes between α and β. This path can be

computed in linear time (in the complexity of the portion of the surface swept during the shifting phase) by shifting back, at the end of the shifting phase, γ^1 to γ^1_s, and adding pieces of η at every step. The path η cuts γ^1_s into two subpaths μ and ν, one of them being possibly empty. We denote by δ_1 the concatenation of μ and η, and by δ_2 the concatenation of ν and η. Then we add δ_1 and δ_2 to the family Δ and we are done.

Merging phase: When two shifted curves are tangent, we do a merging (Figure 6.3, bottom), by computing a curve δ homotopic to their concatenation. For simplicity, let us denote by γ^1 and γ^2 two curves that became tangent during the shifting phase. First, for every $i \neq 1, 2$, we add γ^i_s to the family Δ. Let η be the shortest path from γ^1_s and γ^2_s, which we can, similarly as above, compute in linear time. The curve δ is defined by the concatenation $\eta^{-1} \cdot \gamma^1_s \cdot \eta \cdot \gamma^2_s$. Now, we simply add δ to Δ and we are done.

Analysis: After joining or merging, we added curves to Δ that cut the surface into an additional pair of pants, (possibly) some annuli, and the remaining surface S'. We first observe that we did not add any contractible cycle to Δ; thus, S' has no connected component that is a disk.

After the joining or the merging phase, we added curves in Δ that cut the surface into a new pair of pants, some annuli, and a new subsurface S'. There remains to prove that the length of the boundary S' satisfies $|\partial S'| \leq \ell + 2n/\ell + 8$. We first explain the intuition behind the algorithm. The subtlety is the way the value of s was chosen: If s was equal to r (perhaps the most natural strategy), the boundary of S' would contain (at least) one curve γ^i_r, and we would have no control on its length. On the opposite, if we had chosen $s = 0$, we would have no control on the lengths of the arcs η involved in the merge or the split. The choice of s gives the right trade-off in-between: the lengths of the curves γ^s_i are controlled by this threshold, while the lengths of the arcs are controlled by the area of the annulus between γ^i_s and γ^i_r.

Lengths after the splitting phase After a splitting phase with the curve γ^1, the boundary $\partial S'$ of S' consists of all the other curves γ^i_s in Γ, and of the two new curves, whose sum of the lengths is bounded by $|\gamma^1_s| + 2|\eta|$. Hence $|\partial S'| \leq |\gamma^1_s| + 2|\eta| + \sum_{i=2}^{k} |\gamma^i_s|$, which is at most $\ell + 2|\eta|$ by the choice of s. Furthermore, by construction, $|\eta| \leq 2(r - s + 1)$.

Lengths after the merging phase After a merging phase with the curves γ^1 and γ^2, the boundary $\partial S'$ of S' consists of all the other curves γ^i_s of Γ, and of the new cycle, whose length is bounded by $|\gamma^1_s| + |\gamma^2_s| + 2|\eta|$. Hence similarly, $|\partial S'| \leq \ell + 2|\eta|$. Furthermore, by construction, $|\eta| \leq 2(r - s + 1)$.

Final analysis Thus, after either the splitting or the merging phase, we proved that $|\partial S'| \leq \ell + 4(r - s + 1)$. To conclude the proof, there only remains to prove that $r - s \leq \frac{n}{2\ell} + 1$.

Let $c \in \{s, \ldots, r - 1\}$. The curves γ^i_c and γ^i_{c+1} bound an annulus K^i_c. The number $A(K^i_c)$ of vertices in the interior of this annulus, its *area*, is at least $|\gamma^i_c| + |\gamma^i_{c+1}|$ (see

Figure 6.4(b)—this is where we use, in a crucial way, the fact that G^* is trivalent), because we may only have added vertices in the annulus when we discarded contractible curves.

For $c \in \{s, \dots, r-1\}$ and $i \in \{1, \dots, k\}$, the annuli K_c^i have disjoint interiors, so the sum of their areas is at most n. By the above formula, this sum is at least $U_s + U_r + 2\sum_{c=s+1}^{r-1} U_c \geq 2\sum_{c=s+1}^{r-1} U_c$, where $U_c = \sum_{i=1}^{k} |\gamma_c^i|$. On the other hand, we have $U_c \geq \ell$ if $s+1 \leq c \leq r$, by definition of s. Putting all together, we obtain $n \geq 2(r-s-1)\ell$, so $r - s \leq \frac{n}{2\ell} + 1$.

Complexity: The complexity of the splitting phase or the merging phase is clearly linear in n. The complexity of outputting the new surface $(S', G^*_{|S'})$ is linear in the complexity $\partial S'$, which is, by construction, also linear in n. To conclude, it suffices to prove that the shifting phase takes linear time, and to do that it suffices to prove that the contractibility tests take linear time in total.

To perform a contractibility test on two subcurves α and β, we perform a tandem search on the surfaces bounded by α and β, and stop as soon as we find a disk. If we find one, the complexity in the tandem search is at most twice the complexity of this disk, which is immediately discarded and never visited again. If we do not, the complexity is linear in n, but the shifting phase is over. Therefore, the total complexity of the contractibility tests is linear in the number of vertices swept by the shifting phase or in the disks, until the very last contractibility test, which takes time linear in n. In the end, the shifting phase takes time linear in n, which concludes the complexity analysis. □

6.6 Lower bounds for the length of cellular graphs with prescribed combinatorial map

In this section, we essentially prove that, for any combinatorial map M of any cellular graph embedding (in particular, of any cut-graph) of genus g, there exists an (unweighted, trivalent) cross-metric surface S with n vertices such that any embedding of M on S has length $\Omega(n^{7/6})$. We are not able to get this result in full generality, but are able to prove that it holds for infinitely many values of g. On the other hand, the result is stronger since it holds "asymptotically almost surely" with respect to the uniform distribution on unweighted trivalent cross-metric surfaces with given genus and number of vertices.

Let (S, G^*) be a cross metric surface without boundary, and M a combinatorial map on S. The *M-systole* of (S, G^*) is the minimum among the lengths of all graphs embedded in (S, G^*) with combinatorial map M. Given g and n, we consider the set $\mathcal{S}(g, n)$ of trivalent unweighted cross-metric surfaces of genus g, without boundary, and with n vertices, where we regard two cross-metric surfaces as equal if some self-homeomorphism of the surface maps one to the other (note that vertices, edges, and faces are unlabeled). This refines the model introduced by Gamburd and Makover [106]. Here is our precise result:

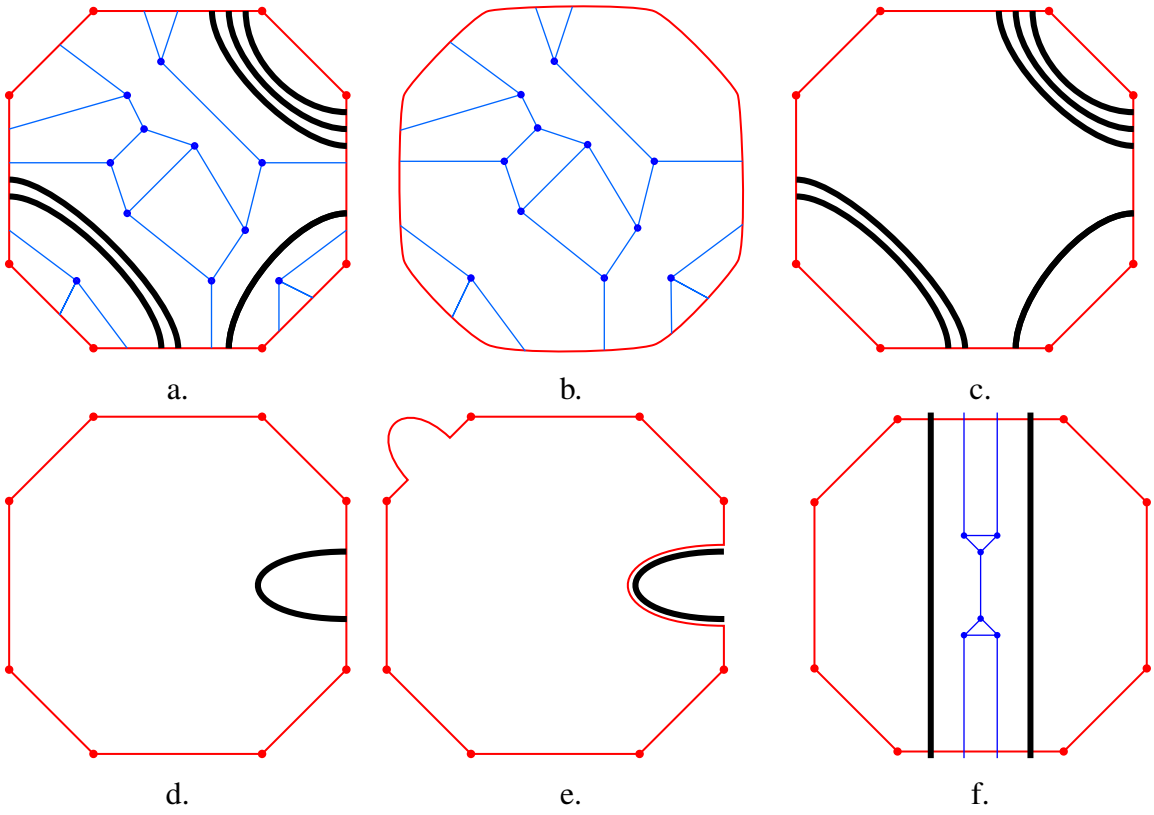


FIGURE 6.5: a. The graph H, obtained after cutting S open along C. The vertices in B (on the outer face) and the vertices of G^* (not on the outer face) are shown. The chords are in thick (black) lines. b. The graph H_1. c. The graph H_2. d., e.: The exchange argument to prove (i). f.: Two chords violating (ii).

Theorem 6.6.1. *Given strictly positive real numbers p and ε, and integers n_0 and g_0, there exist $n \geq n_0$ and $g \geq g_0$ such that, for any combinatorial map M of a cellular graph embedding with genus g, with probability at least $1 - p$, a cross-metric surface chosen uniformly at random from $\mathcal{S}(g, n)$ has M-systole at least $n^{7/6-\varepsilon}$.*

We remark that a tighter estimate on the number $h(g, n)$ of triangulations with n triangles of a surface of genus g could lead to the same result *for any large enough g*, instead of *for infinitely many values of g*. The general strategy is inspired by Guth, Parlier and Young [120], proving a related bound for pants decompositions, but the details of the method are rather different. The main tool is the following proposition.

Proposition 6.6.2. *Given integers g, n, and L, and a combinatorial map M of a graph embedding of genus g, at most*

$$f(g, n, L) = 2^{O(n)} L \left(L/g + 1\right)^{12g-9}$$

cross-metric surfaces in $\mathcal{S}(g,n)$ have M-systole at most L.

Proof First, note that it suffices to prove the result for cut-graphs with minimum degree at least three. Indeed, one can transform any cellular graph embedding into such a cut-graph by removing edges, removing degree-one vertices with their incident edges, and *dissolving* degree-two vertices, namely, removing them and replacing the two incident edges with a single one. So let M be the combinatorial map of such a cut-graph of genus g; let (S, G^*) be a cross-metric surface in $\mathcal{S}(g,n)$, and let C be an embedding of M of length at most L. Euler's formula and double-counting immediately imply that C has at most $4g-2$ vertices and $6g-3$ edges.

Let H' be the graph that is the overlay of G^* and C. Cutting S along C yields a topological disk D, and transforms H' into a connected graph H (Figure 6.5(a)) embedded in the plane, where the outer face corresponds to the copies of the vertices and edges of the cut graph C. The set B of vertices of degree two on the outer face of H exactly consists of the copies of the vertices of C; there are at most $12g-6$ of these. A *side* of H is a path on the boundary of D that joins two consecutive points in B.

Given the combinatorial map of H in the plane, we can (almost) recover the combinatorial maps corresponding to H' and to (S, G^*). Indeed, the set B of vertices of degree two on the outer face of H determines the sides of D. The correspondence between each side of D and each edge of the combinatorial map M is completely determined once we are given the correspondence between a single half-edge on the outer face of H and a half-edge of C; in turn, this determines the whole gluing of the sides of H and completely reconstructs H' with C distinguished. Finally, to obtain G^*, we just "erase" C. Therefore, one can reconstruct the combinatorial map corresponding to the overlay H' of G^* and C, just by distinguishing one of the $O(L)$ half-edges on the outer face of H.

A *chord* of H is an edge of H that is not incident to the outer face but connects to vertices incident to the outer face. Two chords are *parallel* if their endpoints lie on the same pair of sides of D. We claim that we can assume the following:

(i) no chord has its endpoints on the same side of H (Figure 6.5(d));

and that (at least) one of the two following conditions holds:

(ii) the subgraph of H between any two parallel chords only consists of other parallel chords (Figure 6.5(f) shows an example not satisfying this property), or

(ii') there are two parallel chords such that the subgraph of H between them contains all the interior vertices of H.

Indeed, without loss of generality, we can assume that our cut-graph C has minimum length among all cut-graphs of (S, G^*) with combinatorial map M. If a chord violates (i), one could shorten the cut-graph by sliding a part of the cut-graph over the chord (Figure 6.5(d–e)), which is a contradiction.

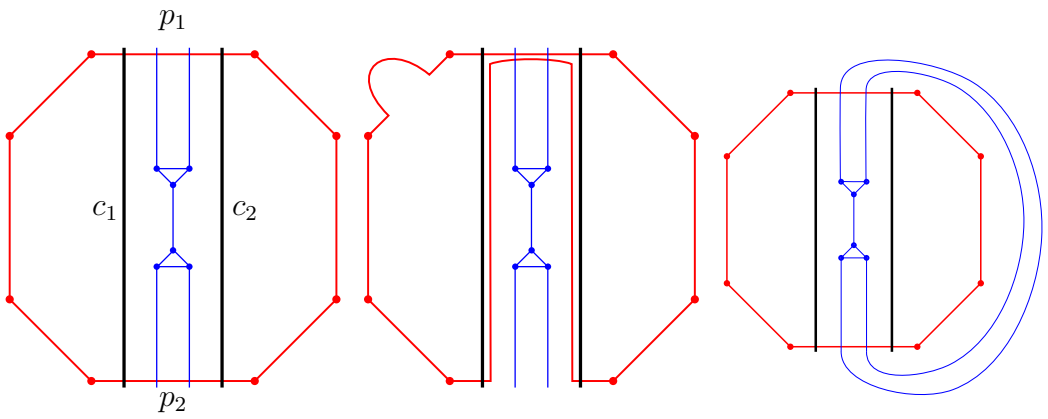


FIGURE 6.6: The exchange argument to prove (ii) or (ii'). Left: Two chords violating (ii). Middle: The exchange argument, in case p_1 and p_2 have different perturbed lengths. Right: A schematic view of the situation, in case p_1 and p_2 have the same perturbed length.

For (ii) and (ii'), the basic idea is to use a similar exchange argument as to prove (i), but we need a perturbation argument as well. Specifically, let us temporarily perturb the crossing weights of the edges of G^* as follows: The weight of each edge e of G^* becomes $1 + w_e$, where the w_e's are i.i.d. real numbers strictly between 0 and $1/L$. Let C be a shortest embedding of M under this perturbed metric.

It is easy to see that C is also a shortest embedding of M under the unweighted metric: Indeed, two cut-graphs C_1 and C_2 with respective (integer) lengths $\ell_1 < \ell_2 \leq L$ in the unweighted metric have respective lengths $\ell_1' < \ell_2'$ in the perturbed metric, since the perturbation increases the length of each edge by less than $1/L$.

We claim that either (ii) or (ii') holds for this choice of C. Assume that (ii) does not hold; we prove that (ii') holds. So the region R of D between two parallel chords c_1 and c_2 of D contains internal vertices; without loss of generality (by (i)), assume that the region R contains no other chord in its interior. Let p_1 and p_2 be the two subpaths of the cut-graph on the boundary of R. If p_1 and p_2 have different lengths under the perturbed metric, e.g., p_1 is shorter, then we can push the part of p_2 to let it run along p_1 and shorten the cut-graph, which is a contradiction. Therefore, p_1 and p_2 have the same length under the perturbed metric, which implies with probability one that they cross exactly the same set S of edges of G^*. (We exclude from S the edges on the endpoints of p_1 and p_2.) Since none of the edges in S are chords, all the endpoints of the edges in S belong to the region of D bounded by p_1, p_2, c_1, and c_2, which implies (ii'). This concludes the proof of the claim.

We now estimate the number of possible combinatorial maps for H, by "splitting" it into two connected plane graphs H_1 and H_2, estimating all possibilities of choosing each of these graphs, and estimating the number of ways to combine them.

Let H_1 be the graph (see Figure 6.5(b)) obtained from H by removing all chords and dissolving all degree-two vertices (which are either in B or endpoints of a chord). H_1

is connected, trivalent, and has at most n vertices not incident to the outer face, so $O(n)$ vertices in total. By a classic calculation (see for example [120, Lemma 4]), there are thus $2^{O(n)}$ possible choices for the combinatorial map of this planar trivalent graph H_1.

On the other hand, let H_2 be the graph (see Figure 6.5(c)) obtained from H by removing internal vertices together with their incident edges and dissolving all degree-two vertices not in B. Since the chords are non-crossing and connect distinct sides of D, the pairs of sides connected by at least one chord form a subset of a triangulation of the polygon having one vertex per side of D. To describe H_2, it therefore suffices to describe a triangulation of this polygon with at most $12g-6$ edges, which makes $2^{O(g)}$ possibilities, and to describe, for each of the $12g-9$ edges of the triangulation, the number of parallel chords connecting the corresponding pair of sides. Since there are at most L chords, the number of possibilities for the latter numbering is at most the area of the simplex $\{(x_1,\dots,x_{12g-9}) \mid x_i \geq 0, \sum_i x_i \leq L+12g-9\}$ (since this simplex contains all the copies of the unit cube translated by the non-negative integer points $(x_1,\dots,x_{12g-9})$ with total sum at most L), which is, using Stirling's formula,

$$\frac{1}{(12g-9)!}(L+12g-9)^{12g-9} \leq \left(\frac{e(L+12g-9)}{12g-9}\right)^{12g-9}.$$

Finally, in how many ways can we combine given H_1 and H_2 to form H? Let us first assume that (ii) holds; the parallel chords joining the same pair of sides are consecutive, so choosing the position of a single chord fixes the position of the other chords parallel to it. Therefore, given H_1, we need to count in how many ways we can insert the $O(g)$ vertices of B on H_2 into H_1, and similarly the $O(g)$ intervals where endpoints of chords can occur, respecting the cyclic ordering. After choosing the position of a distinguished vertex of H_2, we have to choose $O(g)$ positions on the edges of the boundary of H_1, possibly with repetitions, which leaves us with $\binom{O(n+g)}{O(g)} \leq 2^{O(n+g)} = 2^{O(n)}$ possibilities. In case (ii') holds, a very similar argument gives the same result. The claimed bound follows by multiplying the number of all possible choices above. □

Proof of Theorem 6.6.1 Let g_0, n_0, p, ε be as indicated. Euler's formula implies that a cross-metric surface with n vertices has genus $g \leq (n+2)/4$. We now show that, if n is large enough,

$$\sum_{g=g_0}^{(n+2)/4} f(g,n,n^{7/6-\varepsilon}) \leq n^{(1-\varepsilon)n/2}\ (*).$$

Indeed, we have

$$f(g,n,n^{7/6-\varepsilon}) \leq 2^{C_0 n}\left(n^{7/6-\varepsilon}/g+1\right)^{12g-9}$$

for some constant C_0. We need to sum up these terms from $g=g_0$ to $(n+2)/4$. For n large enough, the largest term in this sum is for $g=(n+2)/4$. Thus the desired sum is bounded from above by

$$n2^{C_0 n}\left(4n^{1/6-\varepsilon}+1\right)^{12(n+2)/4-9},$$

which is at most $2^{C_1 n} n^{(1/6-\varepsilon)3n}$ (for n large enough, for some constant C_1), which in turn is at most $n^{(1-\varepsilon)n/2}$ for n large enough.

Furthermore, let $h(g,n) = |\mathcal{S}(g,n)|$ be the number of (connected) cross-metric surfaces with genus g and n vertices. We have $\sum_{g=0}^{(n+2)/4} h(g,n) \geq e^{Cn} n^{n/2}$ if n is large enough and even, for some absolute constant C, the proof is deferred to Lemma 6.6.3. But, if g is fixed, $h(g,n) = O(e^{C'n})$ for some constant C' [120, Lemma 4]. Thus, since g_0 is fixed, there is a constant C'' such that, for n large enough and even, $\sum_{g=g_0}^{(n+2)/4} h(g,n) \geq e^{C''n} n^{n/2}$ (**).

Choose any (even) $n \geq n_0$ such that $n^{-\varepsilon n/2} e^{-C''n} \leq p$ and such that (*) and (**) hold. This implies that, for some $g \geq g_0$, we have

$$f(g,n,n^{7/6-\varepsilon})/h(g,n) \leq n^{(1-\varepsilon)n/2}/(e^{C''n} n^{n/2}) \leq p$$

and the denominator is non-zero. In other words, among all $h(g,n)$ cross-metric surfaces with genus g and n vertices, for any combinatorial map M of a cellular graph embedding of genus g, a fraction at most p of these surfaces have an embedding of M with length at most $n^{7/6-\varepsilon}$. □

To conclude the proof, there remains to prove the bound on the number of connected surfaces.

Lemma 6.6.3. *The number of (unweighted, trivalent)* connected *cross-metric surfaces with n vertices is, for n even large enough, at least $e^{Cn} n^{n/2}$ for some absolute constant C.*

Proof By duality, this is equivalent to counting triangulations with n triangles. Guth, Parlier and Young [120, Lemma 3] prove that, for $n \geq 2$ even and large enough, the number of *possibly disconnected* triangulations with n triangles is between $e^{Kn} n^{n/2}$ and $e^{K'n} n^{n/2}$, where K and K' are absolute constants. Like us, they actually need to prove such bounds for *connected* surfaces. We shall fill this gap here.

Every disconnected triangulation with n triangles can be expressed as the disjoint union of two (possibly disconnected) triangulations with k and $n-k$ triangles, respectively. Therefore, the number of disconnected triangulations with n triangles is bounded from above by

$$\sum_{\substack{2 \leq k \leq n/2 \\ k \text{ even}}} e^{K'n} k^{k/2} (n-k)^{(n-k)/2}.$$

This sum is dominated by its first term, so the number of disconnected triangulations with n triangles is

$$O\left(e^{K'n} (n-2)^{(n-2)/2}\right).$$

Therefore, the number of connected triangulations with n triangles is at least $e^{Kn} n^{n/2} - K'' e^{K'n} (n-2)^{(n-2)/2}$ for some constant K'', which is $\Omega\left(e^{Kn} n^{n/2}\right)$, as desired. □

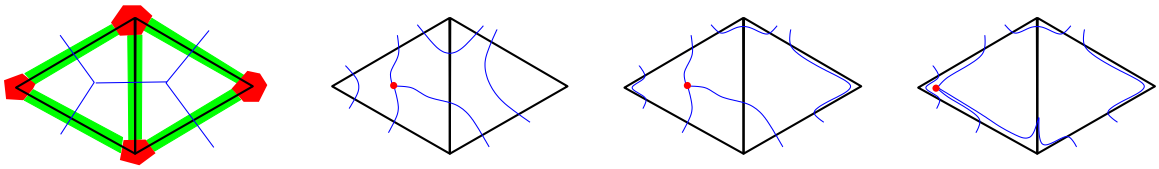


FIGURE 6.7: Illustration of the proof of Theorem 6.6.4. a.: Two triangles of the graph G, the corresponding part of the tubular neighborhood N, made of disks and strips, and the dual cross-metric graph G^*, whose traces on the strips constitute the paths P_s. b.: A part of C. c.: Pushing the pieces not incident to vertices of C into N. d.: Pushing the vertices of C.

We can obtain a similar result in the case of polyhedral triangulations, obtained by gluing n equilateral triangles with sides of unit length. We first note that an element of $\mathcal{S}(g,n)$ naturally corresponds to a polyhedral triangulation by gluing equilateral triangles of unit side length on the vertices. The notion of M-systole is defined similarly in this setting, and we now prove that Theorem 6.6.1 implies an analogous result for polyhedral triangulations:

Theorem 6.6.4. *Given strictly positive real numbers p and ε, and integers n_0 and g_0, there exist $n \geq n_0$ and $g \geq g_0$ such that, for any combinatorial map M of a cellular graph embedding with genus g, with probability at least $1-p$, a polyhedral triangulation chosen uniformly at random from $\mathcal{S}(g,n)$ has M-systole at least $n^{7/6-\varepsilon}$.*

Proof As in the proof of Theorem 6.6.1, it suffices to prove the result for maps M that are cut-graphs with minimum degree three, which have at most $4g-2$ vertices and $6g-3$ edges. Let G be the vertex-edge graph of a polyhedral triangulation on a surface S with genus g. Assume that M has an embedding C of length $O(n^{7/6-\varepsilon})$ on that polyhedral surface. We prove that M has an embedding of length $O(n^{7/6-\varepsilon})$ in the dual cross-metric surface (S, G^*). Since, by Theorem 6.6.1, the proportion of such surfaces is arbitrarily small, this implies the theorem.

Without loss of generality, we assume that C is piecewise-linear, and in general position with respect to G. We consider a tubular neighborhood N of G (Figure 6.7(a)), obtained by first building a small *disk* around each vertex of G, and then building a rectangular *strip* containing each part of edge not covered by a disk. The disks are pairwise disjoint, the strips are pairwise disjoint; each strip intersects only the disks corresponding to the incident vertices of the corresponding edge, along paths. We first push C into N as follows. First consider the maximal pieces of edges C that lie inside a triangle, but do not contain a vertex of C. It is easy, using elementary geometry in equilateral triangles, to prove that one can push, by an isotopy, all such pieces, without moving their endpoints, into C, while at most doubling their total length (Figure 6.7(b–c)). Finally, we push the $O(g)$ vertices of C into

the disks, thereby pushing also the incident pieces into N; this adds $O(g)$ to the length of C (Figure 6.7(d)).

For each strip s, draw a shortest path P_s with endpoints on its boundary, that separates the two sides touching disks. If a piece of C inside s crosses P_s, it forms a bigon with P_s; by flipping innermost bigons, without increasing the length of C, we can assume that each piece of C inside s crosses P_s at most once.

Now we extend the paths P_s to form the graph G^* (Figure 6.7(a)). By the paragraph above, each crossing of a path P_s corresponds to a piece of a path of C that crosses the strip containing P_s, and thus has length at least $1 - \delta$, for $\delta > 0$ arbitrarily close to zero. Therefore, the length of C on the cross-metric surface (S, G^*) is at most $(1 - \delta)$ times that of the length of C on the polyhedral triangulated surface. $\square$

CHAPTER 7

On the complexity of immersed normal surfaces

In this chapter, we investigate a natural variant of normal surfaces towards better algorithms for topological problems in 3 dimensions. This variant, called *immersed normal surfaces*, naturally leads to a problem about detecting singularities, which we show to be NP-hard. We also investigate variants, and give an algorithm to test for a local version of this problem.

The results of this chapter were obtained with Benjamin A. Burton and Éric Colin de Verdière. An extended abstract has been presented at the European Workshop on Computational Geometry [C], but it was very brief due to the four page limit. This is the full version of this work, including ampler details and context as well as full proofs. The algorithm we provide to test for local immersibility is also new. We advise the reader to be familiar with the material in Sections 3.1.3, 3.3.2 and 4.2.2 before delving into this chapter.

7.1 Introduction

In this chapter, we deal with topological problems in 3 dimensions. This story starts where Section 4.2.2 of the survey ended. As we saw there, the complexity of many problems in 3-dimensional topology, like unknot or 3-sphere recognition, is the object of much scrutiny, since it is not known to be polynomial, yet no hardness proof is known. The best known algorithms to solve these problems are exponential at best and heavily rely on *normal surfaces*.

These provide a compact and structured way to analyze and enumerate the most interesting surfaces embedded in a 3-manifold. Starting with a triangulation T of a 3-manifold M with t tetrahedra, a normal surface is a vector in $\mathbb{R}^{7t}$ describing one or multiple embedded surfaces in M. Many interesting surfaces, such as for example a Seifert disk for the unknot, are witnessed by a normal surface having coordinates at most exponential in t.

This is the starting point of many algorithms based on the enumeration of normal surfaces, which naturally have an exponential complexity.

In addition to providing a succinct representation of embedded surfaces, normal surfaces also possess an additional algebraic structure. Indeed, the natural addition and scalar multiplication of vectors translate to operations on normal surfaces, and the space of normal surfaces in $\mathbb{R}^{7t}$ is characterized by a set of equations: the *matching equations* and the *quadrilateral conditions*. The former are linear equations specifying the way to glue normal surfaces locally, while the latter are non-linear and ensure that the resulting surface is embedded. Spaces defined by linear constraints can be studied by the means of linear programming [180], which provides a very powerful framework to deal with decision and optimization problems. This motivates the study of a notion of relaxed normal surfaces, where we remove the quadrilateral conditions to obtain a polyhedral structure on the space of normal surfaces.

As we shall see later, removing the quadrilateral conditions amounts to removing the embeddedness of normal surfaces. Therefore, it amounts to dealing with *singular normal surfaces*. Among these, the *immersed normal surfaces* are well behaved, in the sense that while they can self intersect, they are still 2-manifolds locally. Moreover, their Euler characteristic depends linearly on their normal coordinates —this fact is crucial in algorithms that work with embedded normal surfaces, but does not hold in general for singular normal surfaces. By coupling singular normal surface theory with an algorithm that efficiently separates immersed normal surfaces from the others, we would have powerful tools at our disposal: this could lead to efficient algorithms to find immersed low genus surfaces in 3-manifolds. Furthermore, through classical topological results like Dehn's lemma or the loop theorem [130] we would obtain embedded surfaces, which are the key behind the unknot problem and many others.

In this chapter, we show some inherent limitations of this method by proving in Theorem 7.3.1 that it is NP-hard to detect whether a singular normal surface is immersed [1].

Immersed normal surfaces have been studied from a mathematical point of view by Letscher [170] and from a computational perspective by Aitchinson, Matsumoto and Rannard [4], Rannard [208] and Matsumoto and Rannard [183], in the particular case of the figure-eight knot complement. In the latter papers, the authors devise and implement an algorithm to decide whether a given singular normal surface is immersed. While the complexity of this algorithm is not explicitly computed, it is at least doubly exponential in the input size. Our main result shows that the problem is inherently hard and that no polynomial-time solution is to be expected.

The complexity reduction used in the proof of this theorem works by reducing the problem to a satisfiability problem, which may sound straightforward and lackluster. However, the flavor of this reduction is that in this problem, it turns out to be very hard to obtain more

1. To be accurate, our theorem is actually stated in terms of *normal coordinates* and *immersibility*, we refer to the preliminaries for more detail.

than two copies of every variable. Our proof thus relies on relatively intricate classification theorems on the complexity of Boolean constraint satisfaction problems where every variable occurs at most twice [61, 98]. This approach is thus, to some extent, original, and might prove useful to obtain hardness proofs that could be hard to achieve by other traditional means.

Hardness results are scarce in 3-dimensional computational topology, and to our knowledge all the other difficulty results are deduced from the Agol, Hass and Thurston construction [2], except for the recent hardness results on computing taut angle structures[30] and optimal Morse matchings [32]. Our result displays a different intractability aspect of this theory. In contrast to the aforementioned result by Agol, Hass and Thurston [2], we also prove that this problem is NP-hard even when the input triangulation is a sub-manifold of $\mathbb{R}^3$, which is for example the case for the very important class of knot complements.

On the upper bound side, it is a very natural question to wonder whether this problem is fixed-parameter tractable with respect to the size of the triangulation. As a partial evidence for this, the aforementioned work of Matsumoto and Rannard [183] suggests that the problem may be solved in polynomial time in the very specific case of the figure-eight knot complement. Although we make no progress on this question in full generality, we show that if the triangulation just consists of tetrahedra all sharing a single edge, the problem has a polynomial-time solution as it can solved by computing a maximum flow. Another view on this algorithm is that it can certify *local* immersibility, where the locality means that it can only check whether there is an obstruction to being immersed around every edge, but it is not global since ensuring immersibility as some point might force a branch point at some other point for the triangulation. Another natural and connected question would be to study whether the problem is in NP, since the natural certificate (the global gluing, see Section 7.2) may have exponential complexity in the input.

This chapter is organized as follows. We start by explaining in detail the variant of normal surface theory that we will be investigating, as well as some background on Boolean constraint satisfaction problems in Section 7.2. In Section 7.3, we describe our reduction and prove the main theorem. Section 7.4 explores some variants of the immersibility problem that remain NP-hard. Finally, in Section 7.5, we provide an algorithm to test the immersibility in the very restricted case of tetrahedra glued around a single edge.

7.2 Preliminaries

In this section, we will introduce the new concepts specific to this chapter. For the needed background on triangulations of 3-manifolds and normal surfaces, we refer the reader to Chapter 3, especially Sections 3.1.3 and 3.3.2.

7.2.1 Singular and immersed normal surfaces

Consider a vector $v \in (\mathbb{Z}_+)^{7t}$ of normal coordinates satisfying the matching equations, but not necessarily the quadrilateral equations. For each $i \in \{1, \ldots, 7t\}$, build v_i normal disks of the corresponding type in the corresponding tetrahedron, in general position, in a way that, on each non-boundary face f of T, the images of the normal arcs arising from both sides of f agree. The matching equations imply that such a construction is always possible. Note that, with this gluing, we do not forbid intersections between normal disks in a single tetrahedron, and that such intersections are actually necessary if two different quadrilateral coordinates within the same tetrahedron are non-zero.

More precisely, consider a given normal arc type in a given non-boundary face f of T, corresponding (as above) to two normal coordinates $v_{t,1}$ and $v_{q,1}$ in a tetrahedron incident to f, and also to two normal coordinates $v_{t,2}$ and $v_{q,2}$ in the adjacent tetrahedron. Recall that the matching equations imply that $v_{t,1}+v_{q,1}$ and $v_{t,2}+v_{q,2}$ are equal. The data of a bijection between these $v_{t,1} + v_{q,1}$ normal disks in the first tetrahedron with these $v_{t,2} + v_{q,2}$ normal disks in the second tetrahedron is called the *local gluing* of that arc type. The aggregated information of all the local gluings is called the *global gluing*. Let us emphasize right away that the complexity of a global gluing can be exponential in the normal coordinates, since the latter ones are compressed by the bit-wise representation.

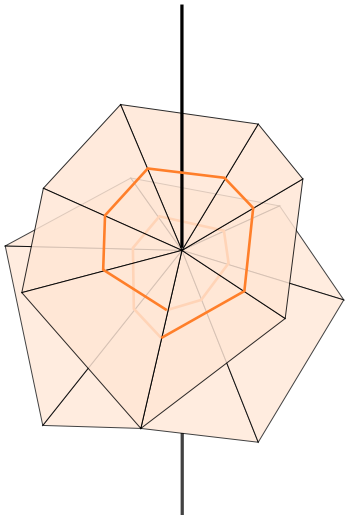


FIGURE 7.1: A branch point of a singular normal surface.

The union of these normal disks glued according to such rules is the image of a surface under a continuous map, since abstractly gluing triangles and quadrilaterals by pairwise identifications of edges always results in a surface (whose actual geometric realization in T may self-cross). This is called a *singular normal surface*. The continuous map may either be locally one-to-one, or have *branch points*, as pictured in Figure 7.1. Since normal disks

are embedded within each tetrahedron, any branch point of a singular surface is necessarily on an edge of the triangulation. A singular normal surface is an *immersed normal surface* if it is the image of a continuous map that is locally one-to-one.

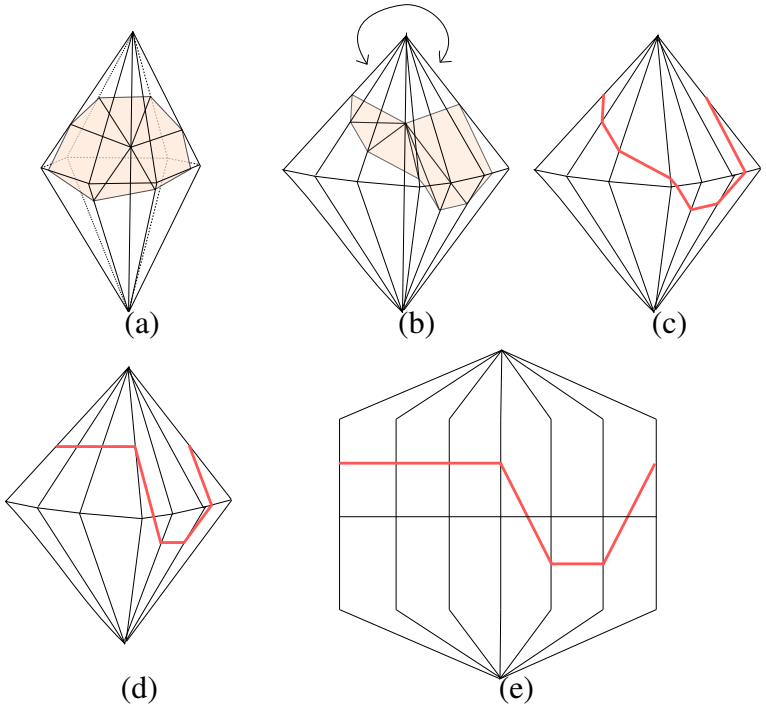


FIGURE 7.2: A *block*, namely, a set of tetrahedra sharing a common non-boundary edge, depicted with a part of a singular normal surface inside it, and its abstract representation.

To describe the singular normal surfaces more accurately, we now introduce a schematic representation of singular normal surfaces.

We will draw a family of tetrahedra that all have one edge in common, which we call a *block* like in Figure 7.2(a); we always assume that the edge is not on the boundary of T. Since we want to picture cleanly what happens on the back of this block, we will unfold it as in Figure 7.2(b), with the implicit convention that the rightmost face is glued to the leftmost face. Although normal disks can be drawn inside this block, the pictures easily become congested when there are several of them. Instead, we will forget the edge in common in the representation and represent the normal disks by their normal arcs, i.e., by their intersection with the front faces (Figure 7.2(c)). These normal arcs are glued together and form possibly self-intersecting closed curves, called *block curves*. Abstracting a bit more, horizontal lines will represent triangles, while diagonal ones will stand for quadrilaterals (Figure 7.2(d)). Finally, to make these pictures even more readable, we will draw the edges between the tetrahedra vertically, only linking them at the extreme top and bottom parts of the figures (Figure 7.2(e)).

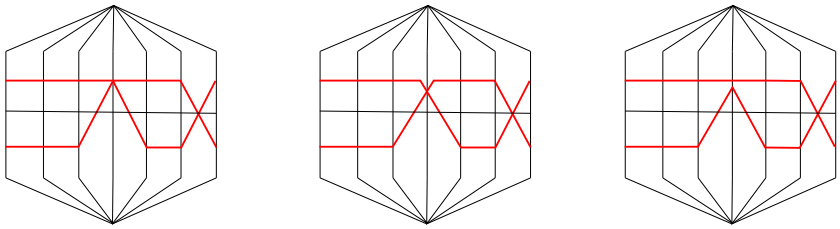


FIGURE 7.3: Normal coordinates drawn (a) without specifying gluings, (b) with a specified gluing, (c) with the opposite gluing from (b).

We will use the following convention in the figures: Whenever we want to represent normal coordinates, without a specific gluing, the normal arcs are drawn so that they connect the midpoints of the corresponding edges of the triangulation. Whenever we want to represent normal coordinates with a particular gluing, we perturb these normal arcs to emphasize the crossings, as pictured in Figure 7.3.

7.2.2 The immersibility problem

Recall that any branch point is necessarily on an edge of the triangulation. Let e be an edge of T, and consider the block around that edge. The singular normal surface has a branch point at e if and only if some block curve "winds more than once" around e or, equivalently, self-intersects, like in Figure 7.4. With this in mind, it is easy to see that a branch point can only occur on a *non-boundary* edge of T.

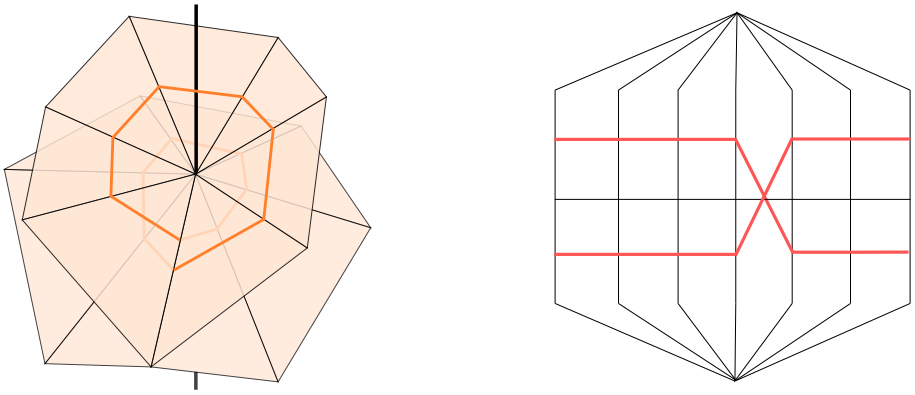


FIGURE 7.4: A branch point and its representation by a block curve winding twice around an edge.

Therefore, given the data of normal coordinates satisfying the matching equations, together with a global gluing, we can easily determine whether the corresponding singular normal surface is immersed or not, in time linear in the size of the input (namely, sum of

all the normal coordinates, of the complexity of the triangulation T and the complexity of the global gluing).

However, consider now only the data of normal coordinates satisfying the matching equations. Then, depending on the choice of the gluings, some of the resulting surfaces may be immersed while some other may have branch points. If there exists a global gluing whose corresponding singular surface is immersed, we say that the normal coordinates are *immersible*. In this chapter, we study the computational complexity of the following problem.

Problem 7.2.1 (IMMERSIBILITY)**.**
***Input**: A triangulation T and normal coordinates N.*
***Output**: Are the normal coordinates N immersible?*

Two difficulties lie at the heart of this problem: Not only do we need to guess a "good" gluing, but this gluing may have an exponential complexity in the input, since the normal coordinates are naturally compressed by the bit representation – this is similar to the issues we discussed in Section 4.2.3. Therefore, the naive algorithm (implemented by Matsumoto and Rannard [183]) is doubly exponential.

7.2.3 Boolean constraint satisfaction problems

In this section, we recall a few basic results about Boolean constraint satisfaction problems; our presentation is inspired from Dalmau and Ford [61]. For a more detailed account of this tremendous body of research, we refer to Creignou, Khanna and Sudan [60].

We start by introducing the *generalized satisfiability problem* SAT(R), which is a variant of the usual SAT problem. A r-ary relation R is any nonempty subset of $\{0,1\}^r$. A CNF(R)-formula is a finite conjunction of clauses $C_1 \wedge \ldots \wedge C_n$ such that each clause, C_i, is an *atomic formula* of the form $R(v_1, \ldots, v_r)$ where $v_1 \ldots v_r$ are Boolean variables. An atomic formula $R(v_1, \ldots v_r)$ is *satisfied* by a variable assignment $f : V \to \{0,1\}$ if and only if $(f(v_1) \ldots f(v_r)) \in R$, and a CNF($R$) formula is *satisfiable* if and only if there exists an assignment satisfying all its clauses simultaneously. Each relation R gives rise to the *generalized satisfiability problem* SAT(R): given a CNF(R)-formula, is it satisfiable?

It is sometimes convenient to assume that constants can appear in CNF(R)-formulas: Each clause is an atomic formula of the form $R(v_1, \ldots, v_r)$ where each v_i is a Boolean variable or a constant (0 or 1). We call any formula obtained this way a CNF$_C(R)$-formula. Similarly, the *generalized satisfiability problem with constants*, SAT$_C(R)$, is defined with CNF$_C(R)$-formula.

The computational complexity of the generalized satisfiability problem with constants has been completely classified by Schaefer in a celebrated paper [229]. We introduce the following definitions in order to state this classification. Here, the symbols $\wedge$, $\vee$, and $\oplus$: $\{0,1\}^r \times \{0,1\}^r \to \{0,1\}$ denote the usual logical operations AND, OR and XOR, applied bit-wise.

A relation R is

- *Horn* if $x, y \in R \to x \wedge y \in R$,
- *dual-Horn* if $x, y \in R \to x \vee y \in R$,
- *bijunctive* if $x, y, z \in R \to (x \wedge y) \vee (x \wedge z) \vee (y \wedge z) \in R$,
- *affine* if $x, y, z \in R \to x \oplus y \oplus z \in R$.

A relation R is *Schaefer* if it is Horn, dual-Horn, bijunctive or affine.

Theorem 7.2.2 ([229])**.** *Let R be a relation. If R is Schaefer, then SAT$_C(R)$ is in P, otherwise it is NP-complete.*

For our reduction, we will restrict ourselves to constraint satisfaction problems where the number of occurrences of every variable is at most 2. We denote by SAT$(2, R)$ the instances of SAT(R) in which every variable occurs at most twice. Similarly, we denote by SAT$_C(2, R)$ the instances of SAT$_C(R)$ in which every variable occurs at most twice. The following definition is key to the classification of these problems.

Let $R \subseteq \{0,1\}^r$ be a relation. Let $x, y, x' \in \{0,1\}^r$, then x' is a *step* from x to y if $d(x, x') = 1$ and $d(x, x') + d(x', y) = d(x, y)$, where d is the Hamming distance. R is a **Δ-*matroid*** (relation) if it satisfies the following *two-step axiom*:

> For all $x, y \in R$ and for all x' a step from x to y, either $x' \in R$ or there exists $x'' \in R$ which is a step from x' to y.

We now come to the classification theorem for SAT$_C(2, R)$:

Theorem 7.2.3 ([98])**.** *Let R be a relation that is not a Δ-matroid relation. Then SAT$_C(2, R)$ is polynomially equivalent to SAT$_C(R)$.*

Theorems 7.2.2 and 7.2.3 immediately imply the complexity result that we will use:

Corollary 7.2.4. *Let R be a relation that is not Schaefer (that is, not Horn, dual Horn, bijunctive, or affine) and not a Δ-matroid. Then SAT$_C(2, R)$ is NP-complete.*

7.3 NP-hardness of detecting immersibility

In this section, we prove the following theorem.

Theorem 7.3.1. *The problem* IMMERSIBILITY *is NP-hard.*

The proof of Theorem 7.3.1 will proceed by a reduction of SAT$_C(2, R)$ to the problem IMMERSIBILITY for a relation R that is neither Schaefer nor a Δ-matroid, which implies by Corollary 7.2.4 that SAT$_C(2, R)$ and hence IMMERSIBILITY are NP-hard.

7.3.1 Gadgets

We now show how to reduce $\mathrm{SAT}_C(2, R)$ to the problem IMMERSIBILITY. To this end, we use a 6-ary relation R which we will describe later. We start with a formula Φ that is, by definition, a conjunction of clauses of the form $R(x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4}, x_{i_5}, x_{i_6})$, where x_i is either a variable or a constant, and every variable appears at most twice in Φ.

The gadgets that we use for our reduction are of three types. Each clause is represented by a *clause gadget*, a block of six tetrahedra glued together around an edge. For each variable occurring exactly twice in Φ, we connect these two occurrences in the clauses using *tubes*, which are also blocks of six tetrahedra. Finally, the *constant gadgets* are used to represent the constants 0 or 1 appearing in the clauses. The idea for the proof is that a clause is satisfiable if and only if the normal coordinates in the clause gadget are immersible; the tubes then enforce consistency between the clauses. Therefore, the whole formula will be satisfiable if and only if the associated normal coordinates are immersible.

The clause gadget. Consider the gadget G pictured in Figure 7.5. It consists of six tetrahedra that all have an edge in common, and contain each three normal disks: two triangles and one quadrilateral. For every clause in Φ, we create a copy of the clause gadget G; these copies will be connected using tubes, described below.

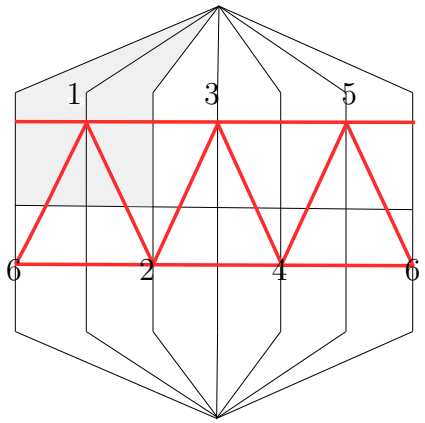
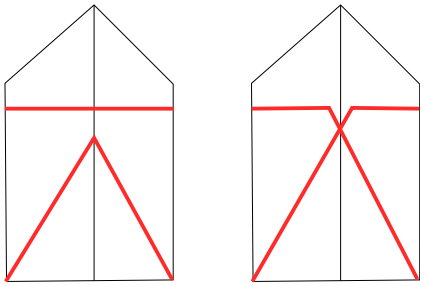


FIGURE 7.5: The clause gadget. The pair of faces corresponding to the variable x_1 is shaded.

The rationale behind this gadget is the following. At the interface of two adjacent tetrahedra, exactly two gluings can be done (see Figure 7.6). This choice of gluing can be described by a variable $x_i \in \{0, 1\}$, where 0 corresponds to the gluing (a) and 1 to (b). Equivalently, a value of 1 corresponds to the fact that the two block curves at the specified position cross. Therefore, each variable in a clause has a *pair* of associated faces on the boundary of the clause gadget; for example, the two shaded triangles in Figure 7.5 are the pair of faces associated to variable x_1.

This way, a global gluing of the singular normal surface in G is described by an element $x \in \{0,1\}^6$. The order on the six variables is pictured in Figure 7.5.



a. b.

FIGURE 7.6: a. Local gluing corresponding to $x_i = 0$. b. Local gluing corresponding to $x_i = 1$.

We define the following relation R on 6 variables:

$$\begin{aligned} R \;=\; \{\; & (0,0,0,0,0,0); (0,0,0,1,0,1); (0,0,1,0,1,0); (0,1,0,0,0,1); (0,1,0,1,0,0); \\ & (0,1,1,0,1,1); (1,0,0,0,1,0); (1,0,1,0,0,0); (1,0,1,1,0,1); (1,1,0,1,1,0); \\ & (1,1,1,1,1,1) \;\} \end{aligned}$$

This allows us to get to the following lemma.

Lemma 7.3.2. *The singular normal surface in the gadget G specified by the gluing $x \in \{0,1\}^6$ is immersed if and only if $x \in R$.*

Proof of Lemma 7.3.2 The proof is done by exhaustive checking, i.e., checking for every possible 6-tuple whether there is a branch point around the central edge or not. As an example, Figure 7.7 pictures the singular normal surfaces obtained with the global gluings (1,0,1,1,0,1) and (1,0,1,1,1,1), yielding in one case an immersed normal surface and in the other a singular normal surface with a branch point. □

The relation R. We now show that the relation R is neither Schaefer nor a Δ-matroid. The proofs are somewhat tedious but straightforward.

Proposition 7.3.3. *The relation R is not Schaefer, that is, neither (i) Horn, (ii) dual Horn, (iii) bijunctive, nor (iv) affine.*

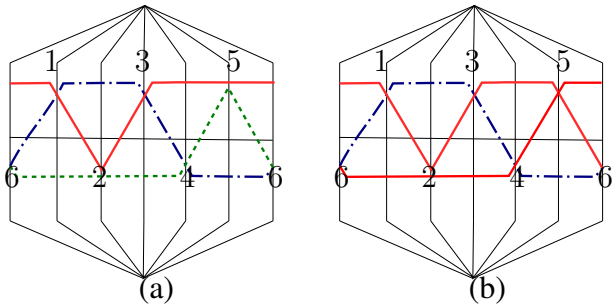


FIGURE 7.7: (a) Immersed normal surface corresponding to coordinates (1,0,1,1,0,1), yielding three block curves (b) Singular normal surface corresponding to coordinates (1,0,1,1,1,1), yielding two block curves. The one winding twice around the central edge witnesses a branch point.

Proof of Proposition 7.3.3

(i) $(1,0,1,0,0,0)$ and $(1,1,0,1,1,0)$ are in R, but their conjunction $(1,0,0,0,0,0)$ is not.

(ii) $(1,0,1,0,0,0)$ and $(1,1,0,1,1,0)$ are in R, but their disjunction $(1,1,1,1,1,0)$ is not.

(iii) If we take $x = (1,0,1,0,0,0)$, $y = (1,1,0,1,1,0)$, and $z = (0,0,0,0,0,0)$, $(x \wedge y) \vee (x \wedge z) \vee (y \wedge z) = (1,0,0,0,0,0)$, which is not in R.

(iv) If we take $x = (1,0,1,0,0,0)$, $y = (1,1,0,1,1,0)$, and $z = (0,0,0,0,0,0)$, we have $x \oplus y \oplus z = (0,1,1,1,1,0)$, which is not in R.

□

Proposition 7.3.4. *The relation R is not a Δ-matroid.*

Proof of Proposition 7.3.4 We take $x = (1,1,1,1,1,1)$, $y = (1,0,0,0,1,0)$, and $x' = (1,0,1,1,1,1)$, x and y are in R and x' is a step from x to y, but x' is not in R and there does not exist any $x'' \in R$ which is a step from x' to y. □

The tubes. A *tube* is the block pictured in Figure 7.8; it is comprised of six tetrahedra that all have one edge in common, and contain each two normal disks: either a pair of one triangle and one quadrilateral, or a pair of triangles of different types. As in Figure 7.8, we denote by A_1 and A_2 the two pairs of faces that are crossed by two adjacent quadrilaterals.

Similarly as for the clause gadget, the gluings in a tube gadget at A_1 and A_2 can be specified by two variables, again with the convention of Figure 7.6: 0 and 1 respectively for non-crossing and crossing block curves.

Lemma 7.3.5. *The singular normal surface in a tube specified by the gluing is immersed if and only if both variables of the tube are equal.*

Proof This is immediate by checking the four possible assignments of the variables. □

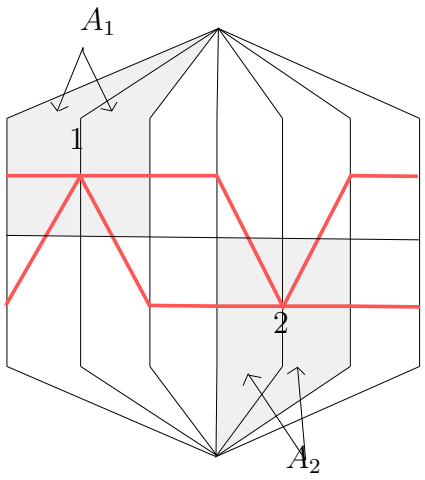


FIGURE 7.8: A tube, where the shaded regions denote the locations where the tube will be connected to clauses.

Consider a variable v appearing exactly twice in Φ, in clauses C_1 and C_2 (it may be that $C_1 = C_2$). Let G_1 and G_2 be the copies of the clause gadget corresponding to C_1 and C_2, respectively; let B_1 be the pair of faces in G_1 corresponding to the occurrence of the variable v in G_1, and similarly let B_2 be the other pair of faces in G_2 corresponding to the occurrence of the variable v in G_2. We create a tube T_v, and we glue the pair of faces A_1 to B_1 and A_2 to B_2.

The constants. A constant gadget is one of the blocks pictured in Figure 7.9 (since each gadget only consists of one tetrahedron, we did not adopt the schematic representation in this drawing). The gadget CG_0 for the constant 0 consists of one tetrahedron containing two triangles, while the gadget CG_1 for the constant 1 consists of one tetrahedron containing two crossing quadrilaterals. In both cases, the pair of faces A, at which the constant will be connected to a clause gadget, is made of the two front faces in Figure 7.9.

Whenever a constant 0 appears in Φ, we create a copy of the constant gadget CG_0, and attach the corresponding pair of faces in the clause gadget to the pair A in the constant gadget. The same holds for the constant 1, with CG_1 instead of CG_0.

7.3.2 Proof of the reduction

We now have all the tools to prove Theorem 7.3.1. Starting with a formula Φ, we build a triangulation T and normal coordinates N with the clause gadgets, the tubes, and the

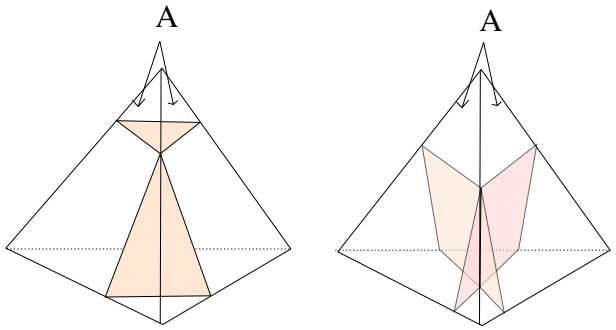


FIGURE 7.9: Left: The gadget CG_0. Right: The gadget CG_1. In both cases, the two front faces constitute the pair A. The trace of the singular normal surface on A is the same in both cases.

constant gadgets. We first prove that this triangulation forms a 3-manifold with boundary.

Proposition 7.3.6. *The triangulation T corresponding to a formula Φ is a 3-manifold with boundary.*

Proof of Proposition 7.3.6 First, we show that every vertex v of T has a neighborhood homeomorphic to the closed half-space. The vertex v is adjacent to a clause gadget C as well as between zero and three tubes or constant gadgets. We focus on the case with three tubes T_1, T_2 and T_3, the other ones being handled similarly. Since gadgets are glued along discs with disjoint interiors, and the tubes are locally disjoint except at their intersection with C, the neighborhood around v is the one pictured in Figure 7.10. One readily sees that this neighborhood is homeomorphic to a half-space.

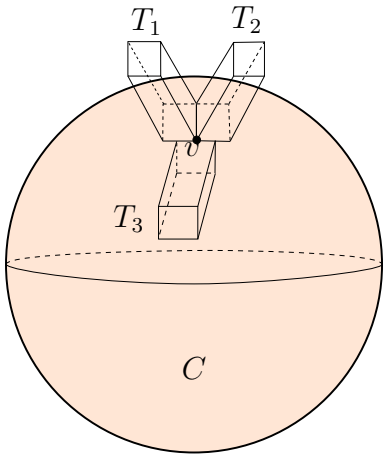


FIGURE 7.10: The local picture around a vertex v of the triangulation T.

Moreover, by construction, no edge is identified to itself in reverse. This shows that T is a 3-manifold with boundary. □

Proposition 7.3.7. *The normal coordinates N correspond to an immersible surface if and only if the formula Φ is satisfiable.*

Proof of Proposition 7.3.7 We will use the following observation. Consider an edge e between the two faces forming a pair of faces in a gadget. The block around e is made of three or four tetrahedra, depending on whether a constant gadget or a tube is glued to the pair on the clause gadget. Then the singular surface has no branch point on e if and only if the values of the variables on the two pair of faces are identical, and, in case one of the gadget is a constant gadget, the value takes the value specified by that gadget.

Assume first that the normal coordinates N correspond to an immersible surface. The gluings corresponding to variables appearing exactly once in Φ define a partial assignment of the variables. For the ones appearing twice, Lemma 7.3.5 implies that the two variables in the same tube are equal. Since they are glued to the pairs of faces representing their two occurrences in the clause gadgets, and by the observation, this defines a consistent assignment of the variables; moreover, again by the observation, the variables on the pair of faces on the clause gadgets corresponding to the constants take the appropriate values. Finally, since no branch point arises in a clause gadget, Lemma 7.3.2 implies that each clause is satisfied by the assignment of variables.

Conversely, assume that Φ is satisfiable. The satisfying assignment naturally defines the values of the variables on each pair of faces. We prove that these gluing rules do not create branch points. Branch points only occur on edges of the triangulation, and each edge is either (1) an edge in a clause gadget, (2) an edge in a tube gadget, (3) an edge at the interface between two pairs of faces, or (4) a boundary edge of T. No edge of type (1), (2), or (3) contains a branch point, by Lemmas 7.3.2 and 7.3.5 and the observation, respectively. Furthermore, no edge of type (4) can contain a branch point, since branch points occur on interior edges. This concludes the proof. □

7.4 Variants

We now establish a few corollaries, settling the complexity of a few variants of the IMMERSIBILITY problem.

The first one settles the complexity of the problem k-BOUNDED-IMMERSIBILITY, which is the one of testing immersibility when all the normal coordinates are bounded by $k \geq 1$.

Corollary 7.4.1. *The problem k-BOUNDED-IMMERSIBILITY is NP-complete.*

Proof of Corollary 7.4.1 The NP-hardness reduction for Theorem 7.3.1 only involves normal coordinates bounded by 1, so k-BOUNDED-IMMERSIBILITY is NP-hard as well.

The certificate we use to prove the membership in NP is the global gluing for an immersed surface. At an interface, the local gluing can be described by a permutation in S_k,

which is of finite size since k is fixed. Since the number of interfaces is bounded by the size of the triangulation, the certificate has polynomial size. As mentioned in the preliminaries, when one is provided with normal coordinates and a global gluing, one can test in linear time whether the corresponding singular normal surface is immersed. □

The gadget we use to show the NP-hardness of IMMERSIBILITY only outputs a 3-manifold with boundary, but can be easily tweaked to handle the problem with boundaryless manifolds, which we name BOUNDARYLESS-IMMERSIBILITY.

Corollary 7.4.2. *The problem* BOUNDARYLESS-IMMERSIBILITY *is NP-hard.*

Proof of Corollary 7.4.2 The triangulation T_Φ with normal coordinates N obtained by the reduction in Theorem 7.3.1 can become boundaryless by *doubling* it, i.e., by taking $T_\Phi \bigsqcup T_\Phi$ and gluing one onto the other with the identity homeomorphism on their boundaries. It is straightforward to check that N is immersible in T_Φ if and only if $N \bigsqcup N$ is immersible in $T_\Phi \bigsqcup T_\Phi$. □

As we mentioned in the preliminaries, the triangulations we consider in this chapter may not be simplicial complexes, and indeed the gadgets we use in our reduction may display some slightly pathological behavior. For example, if a tube links the variables 1 and 5 of a clause gadget, the central edge of the tube has its endpoints identified. This can be avoided by replacing each tube gadget by two copies of itself, glued to each other along the pairs A_1 and A_2. These "double tubes" are glued to the clause gadgets as usual ones. Then it can be checked that the resulting triangulation is a simplicial complex, and this shows that IMMERSIBILITY is also NP-hard when the triangulation is a simplicial complex.

As a last remark, we note that since the triangulation obtained by the reduction in Theorem 7.3.1 consists of balls linked by tubes, it can be embedded in $\mathbb{R}^3$. This shows that IMMERSIBILITY is NP-hard even when restricted to submanifolds of $\mathbb{R}^3$, which is for example the case of knot complements.

7.5 Testing local immersibility

In this section, we provide a polynomial-time algorithm to test whether normal coordinates are immersible when the triangulation is just a block, i.e., a collection of tetrahedra glued around a single edge. When applied to every edge of a more complicated triangulation, this provides a *local* test for immersibility, in that it detects local obstructions but not more involved ones, as pictured in Figure 7.11.

For the rest of this section, we denote by T a triangulation comprised of a single block, and N the normal coordinates for this triangulation.

The algorithm works by reducing the problem to a flow computation on directed graphs. The construction of the directed graph G corresponding to a block and normal coordinates

FIGURE 7.11: When these two blocks are made into a single triangulation by gluing them along the shaded regions, the normal coordinates are locally immersible around every edge, but not globally immersible.

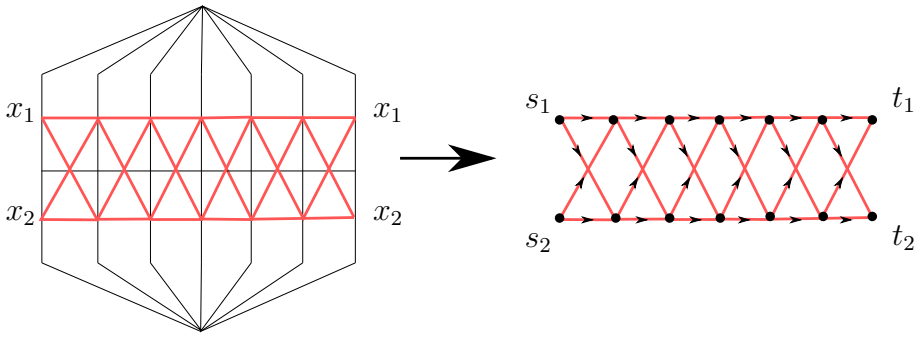


FIGURE 7.12: The directed weighted graph G is obtained by choosing an orientation around the block, adding a single directed edge for every normal disk type, and setting the capacity of an edge to be equal to the corresponding normal coordinate.

is pictured in Figure 7.12: every triangle type or quadrilateral type is associated to a directed edge, and the corresponding normal coordinate is carried over as the capacity of that edge. We denote by s_1 and s_2 the extremal vertices of this graph on one side of the block, and t_1 and t_2 the vertices on the other side. Note that in the triangulation, both sides of the block are identified, and therefore for $i = 1, 2$, s_i corresponds to the same vertex as t_i, which we denote by x_1 and x_2. The algorithm then simply computes the maximum flow between s_1 and t_1, and checks whether it is equal to the sum of the capacities of the edges going out of s_1.

The correctness of this algorithm is warranted by the following lemma.

Lemma 7.5.1. *The normal coordinates N are immersible if and only if the maximum flow between s_1 and t_1 equals the sum of the capacities of the edges going out of s_1.*

Proof of Lemma 7.5.1 For the first implication, if the normal coordinates are immersible, there exists a gluing inducing no branch point. The associated normal surface is therefore a union of disks winding exactly once around the central edge. Every such disk corresponds

in G to a weight 1 flow going from s_1 to t_1 or s_2 to t_2. Therefore, the maximum flow going from s_1 to t_1 is the sum of the capacities going out of s_1.

In the other direction, such a maximum flow naturally defines a gluing corresponding to an immersed surface. Indeed, every unit of flow corresponds to a normal disk, and the flow equations at every vertex allow to specify how to glue these normal disks together. Therefore, the maximum flow corresponds to a partial gluing of the normal disks, which yields a union of embedded disks passing through x_1. The gluing of the other normal disks will not be described by this flow, but it is easy to see that they will necessarily also form a union of embedded disks passing through x_2. All in all, we obtain an immersed surface, which completes the proof. □

It is an intriguing problem to extend this algorithm to deal with more complicated triangulations. With some work, we can for example obtain a polynomial-time algorithm to test immersibility for the triangulation pictured in Figure 7.11. Finding a general polynomial-time algorithm is ruled out by our main Theorem 7.3.1 though.

CHAPTER 8

Conclusions

In this thesis, we have studied topological problems on graphs, surfaces and 3-manifolds from a computational point of view, and we have provided new insights on their algorithmic properties, whether by designing algorithms, obtaining better combinatorial bounds or establishing hardness. We now investigate the work that lies ahead of us, first directly in the continuation of the three chapters, then on a broader scale.

8.1 Summary and continuations

Testing isotopy of graphs on surfaces. In Chapter 5, we have provided efficient algorithms to test whether two graphs embedded on a surface are isotopic. Our algorithms came in two flavors, depending on whether the surface is the plane and the input is described by coordinates, or the graphs are described by their arrangements on an arbitrary cross-metric surfaces. Our tools rely on previous algorithms to test homotopy, which we use as a black box, and a combinatorial characterization of isotopy which may be useful in its own right. Many interesting variants of this problem remain unsolved and may form the basis of future work.

For starters, our characterization heavily relies on testing the existence of an oriented homeomorphism; it is not clear how to adapt this test in the non-orientable case. We note that the research on mapping class groups, from which we drew parts our inspiration, is mostly focused on the orientable case, and a first step would be to identify which fragments of this theory still hold in the non-orientable world and whether they could be applied for our problem.

Regarding the case of the punctured plane, the most obvious open question is to improve the running time of the algorithm. Bespamyatnikh [16, Theorem 7] describes an $O(n^{4/3} \times \text{polylog}(n))$-time algorithm for testing path homotopy; however, it is not clear that this

algorithm extends to the homotopy test for cycles. One obstacle for this extension is that our cycles are not simple and may make turns always in the same direction: the algorithm by Bespamyatnikh [16, Section 5.5] considers maximal subpaths that always turn in the same direction, but in our case such maximal subpaths may be cycles without "starting" and "ending" points, for which the same approach does not seem to work.

Also, we only test the existence of a topological isotopy: The edges are allowed to bend during the deformation. It is easy to see that, in the presence of obstacles, the existence of a topological isotopy between two straight-line embeddings does not imply the existence of a straight-line isotopy, in contrast to the case without obstacles [18, 112]. Could it be that, in such a situation, there exists a straight-line isotopy after splitting each edge in two (or a constant number of) segments? Computing such an isotopy efficiently may not be an easy task, but related techniques [5, 10] might apply.

Finally, in both the surface model and the punctured plane model, computing shortest graph embeddings within a given isotopy class would be very interesting, and would generalize known results for computing shortest paths within a given homotopy class [15, 56–58, 77, 137], even though we expect the problem to be much harder.

Discrete systolic inequalities and decompositions of triangulated surfaces. Our contribution in Chapter 6 is threefold. Firstly, we improved the best bounds on the length of topologically meaningful cycles on combinatorial surfaces. Secondly, by adapting a construction from the Riemannian case, we improved the bound on the length of a shortest pants decomposition, and provided an algorithm to compute such a decomposition with guarantees on its length. Finally, a careful analysis of the case of random surfaces gave strong lower bounds for the length of cut-graphs with a prescribed topological map.

There are many avenues for pursuing this work. A first line of research involves further investigations on the problems around graph decomposition. The complexity of computing shortest pants decomposition is still hanging between polynomial and NP-complete, and similarly it is unknown whether computing a shortest cut-graph is fixed-parameter tractable in the genus of the surface, or whether it is approximable within a constant factor.

We believe that we can further our understanding of these problems in two directions. A first aspect is to get better algorithms, either from a complexity or an approximation point of view. To that end, the general framework of *brick decompositions* [23] allows to get efficient approximation schemes and fixed-parameter algorithms out of algorithms for surface-embedded graphs with bounded tree-width [1]. This can be attacked by the use of dynamic programming combined with tree decompositions with an added topological structure, such as *surface cut/split decompositions* [22, 219]. It is the subject of ongoing work to adapt these tools to obtain a fixed-parameter tractable approximation scheme to compute the optimal cut-graph of a surface.

1. Recall that the *tree-width* of a graph measures quantitatively how close it is to a tree, we surveyed some of its uses in Section 4.1.

Regarding lower bounds, our work and previous results [120] have shown that random surfaces display a pathological behavior both for cut-graphs and pants decompositions. It is therefore key to study this behavior in order to prove difficulty results. The tree-widths of cut-graphs are a natural bottleneck, and it is reasonable to expect that random surfaces lead to cut-graphs with high tree-width. Hence, investigating whether random surfaces could be used as hard instances for this problem looks like a promising perspective.

It is also worthwhile to study random surfaces for their own sake. As we mentioned, they were introduced because of strong connections with theoretical physics [198], and also as a discrete approximation to random Riemann surfaces [26] – getting a thorough understanding of their properties could lead to important insights, in the same vein as the impressive impact of random graph theory on combinatorics. In particular, there remains an important open question on the genus of these random surfaces: it is known that it grows like $\log 3n$ where n is the number of triangles, but the second order term is evasive, and it conjectured to be the Euler-Mascheroni constant γ [198].

Finally, we described how our bounds for cut-graphs translate by duality to a problem on crossing-numbers of graphs, which is about embeddings two graphs on a surface while minimizing the number of intersections. This is strongly connected with the theory of *simultaneous embeddings*, a topic in graph drawing concerned with finding the correct way to display multiple embedded graphs at the same time. A strong theory has emerged in the planar case [20], but it still remains unscratched in the case of surface-embedded graphs.

On the complexity of immersed normal surfaces. In this chapter, we have investigated how the theory of immersed normal surfaces could be used to improve the complexity of algorithms in 3-manifold topology. We showed that it naturally leads to the problem of detecting immersibility, which we showed to be NP-hard. Although this constitutes a serious roadblock to this approach, all hope is not lost. We provided an algorithm to solve this problem at a very local scale, and there is still room to study what can be done despite this hardness.

The main open question is whether this immersibility problem is fixed-parameter tractable with respect to the size of the triangulation. A promising approach is to generalize our approach with flows to certify local immersibility to handle the broader picture of global immersibility. It is the subject of ongoing work to see if techniques coming from integer programming in fixed dimensions could be adapted to handle this problem.

Another natural question is whether this approach can still work for restricted classes of triangulations. Once again, it turns out that the tree-width is a natural parameter to consider, and a lot of work has been devoted to studying the complexity of 3-manifold problems for spaces such that the dual of the triangulation has bounded tree-width [30–32, 34], because some "practical" spaces that arise tend to have this property. In our case, since the gadget we use certainly incurs unbounded tree-width, it is also a natural question to investigate.

8.2 Perspectives

As we surveyed in Section 4, many topics lie at the intersection of topology and theoretical computer science and beg to be explored. In the following we expose two broad research directions that fit well with our research themes, although not directly in line with the works we have presented in this thesis. Compared to the issues we discussed in the previous section, the perspectives we introduce here are more ambitious and include problems that have been open for quite some time, but we feel that the expertise we developed in this thesis gives us a solid basis, or dare we say an edge, to tackle them.

Combinatorics of 3-manifolds. There are many combinatorial open questions in 3-manifold theory. As we discussed in Section 4.2, in a recent breakthrough, Marc Lackenby [161] showed that when a knot is unknotted, i.e., can be continuously deformed to the usual circle, this unknotting procedure can be done in at most a polynomial number of Reidemeister moves. The natural way to improve on this result is to study whether a similar bound can be established for other classes of knots, for example starting with genus 1 knots. Looking at the proof, this would involve extending several techniques from the planar case to the case of surfaces, for which the tools around decomposition that we introduced in Chapter 6 might be handy.

In a different direction, when dealing with triangulated 3-manifolds, the *Pachner moves* are a natural analogue of the Reidemeister moves: they allow to transform a triangulation locally, and it is known that if two 3-manifolds are the homeomorphic, there exists a sequence of moves going from the first to the second. The bounds on the numbers of these moves are, when they exist, astronomical, and it is a daunting task to improve them, since it would imply improving the algorithm for 3-manifold recognition. While this last problem is notoriously difficult, simplifying 3-spheres may be more tractable: there is a doubly exponential bound on the number of moves to do so [184], and experimental evidence seems to suggest that the bounds should be much lower [29]. There is now a wide body of research on 3-sphere recognition, and it seems ripe to be combined with Lackenby's work – in particular, the now standard machinery of *0-efficient triangulations* [145] and *almost normal surfaces* could perhaps be adapted to this framework.

Random and extremal complexes. Triangulated surfaces are obtained by gluing triangles together such that every edge is adjacent to at most two triangles. Lifting this adjacency condition gives rise to 2-dimensional *simplicial complexes*. While the combinatorics of graphs embedded on surfaces are fairly well understood, since many properties are controlled by the Euler characteristic, there are many open questions about 2-dimensional simplicial complexes. Mimicking the tremendous success of the theory of random graphs, random (Linial-Meshulam) complexes have been the subject of active study recently, and their geometric and topological properties are now quite well understood [53, 59]. On the

other hand, the topological properties of *extremal complexes* are mostly open for investigation. In particular, the following extremal questions are wide open: What is the maximal number of 2-dimensional simplices in an n-vertex simplicial complex which does not contain a surface ? What about specifying a fixed surface ? In the case of the sphere, it is a classical result of Brown, Erdős and Sòs [27] that $O(n^{2.5})$ is the tight bound, but strikingly the value of the exponent is unknown for the remaining cases. On a wider combinatorial point of view, even in the 2-dimensional case, very little is known on the extremal properties of simplicial complexes – or hypergraphs – and studying their topological aspects might shed light on the main open questions on this area.

APPENDIX A

Appendices on graph isotopy testing

In this appendix, we present the proofs that have been stripped off from Chapter 5 to make it more streamlined. Section A.1.1 explains how to add additional cycles to the stable family computed in Section 5.4, to ensure that we obtain a pointwise isotopy, Section A.1.2 proves Theorem 5.4.1 when G_1 is a cut-graph and Section A.2 shows how to deal with non-hyperbolic surfaces, namely (punctured) spheres and tori.

A.1 Additional proofs for Theorem 5.4.1

A.1.1 Fixing the map automorphism

In this section, we prove the following proposition.

Proposition 5.4.8. *In linear time, we can construct a family of cycles Λ in G such that:*

- *each edge of G is used at most thrice by all the cycles in Λ.*
- *if we denote by Λ_1 and Λ_2 the images of Λ in G_1 and G_2, if every cycle in Λ_1 is homotopic to its counterpart in Λ_2, then an ambient isotopy of S maps Γ_1 to Γ_2 pointwise.*

We will need the following rather independent lemma in the course of the proof.

Lemma A.1.1. *Let C be a family of simple cycles on S, pairwise disjoint except at a single point p, where two cycles may or may not cross. Assume that no component of $S \backslash C$ is a disk bounded by one or two cycles. Then the cycles in C are pairwise (freely) non-homotopic.*

Proof We will use the fact that two simple homotopic cycles cross transversely an even number of times (because they form bigons [96, Proposition 1.7]).

First, no cycle in C is contractible; otherwise, it would bound a disk on the surface. The cycles inside that disk are all contractible, and therefore do not cross at p because of the aforementioned fact. Taking an innermost such cycle, we obtain a component of $S \setminus C$ bounded by one cycle, contradicting the assumption.

Assume now for the sake of a contradiction that two cycles c_1 and c_2 are homotopic. They meet at point p without crossing transversely. After a local perturbation, these cycles become disjoint, and therefore bound an annulus [81, Lemma 2.4]. Thus, the unperturbed cycles can be viewed as two loops ℓ_1 and ℓ_2 based at p that bound a disk. There may be other cycles inside that disk, but in all cases a face inside it is a disk bounded by one or two cycles, which is impossible. □

The proof of Proposition 5.4.8 revolves around finding additional cycles to add to the family Γ to ensure that the isotopy we get is pointwise. From the hypothesis that no face of the input graphs is a disk, one deduces that the inner disk in each connected component is mapped to itself by the homeomorphism between Γ_1 and Γ_2. Then, one just needs to ensure that this homeomorphism, restricted to each inner disk, is isotopic to the identity – to do so it is enough to add one well chosen cycle to Γ for each connected to component.

Proof of Proposition 5.4.8 The family Λ is the union of the stable family Γ defined in Proposition 5.4.7 and of the family Φ defined as follows. Recall that in the proof of Proposition 5.4.7, we considered each connected component (V', E') of the graph $G = (V, E)$ in turn. If (V', E') had cyclomatic number at least two, we considered the edge set E'' of a spanning tree of (V', E'). A *fundamental cycle* of (V', E') is a simple cycle in (V', E') containing exactly one edge in $E' \setminus E''$. We put in Φ an arbitrary fundamental cycle for each connected component (V', E') of cyclomatic number at least two. The fundamental cycles of a connected component (V', E') can be extended towards an arbitrary root p of the spanning tree (V', E'') and then slightly perturbed on S so that they become simple and pairwise disjoint except at p, where they may or may not cross. The faces of this new family C of perturbed cycles correspond to the faces of (V', E'). Moreover, C satisfies the hypotheses of Lemma A.1.1: Indeed, if there is a disk in $S \setminus C$ bounded by one or two cycles, there must be at least one connected component of G_1 inside it because no face of G_1 is a disk; but then this connected component is contractible, which is absurd since the preprocessing removed all the contractible components of G_1. Hence, the fundamental cycles of any given connected component (V', E') are pairwise non-homotopic.

Clearly the family $\Lambda = \Gamma \cup \Phi$ can be computed in linear time and uses each edge of G at most thrice. Assume that, for each cycle λ in Λ, the images of λ in G_1 and G_2 are homotopic. There remains to prove that some isotopy of S maps Γ_1 to Γ_2 pointwise.

By Proposition 5.4.7, Γ_1 is a stable family, and of course $\Gamma_2 := h(\Gamma_1)$ as well. Since each cycle in Γ_1 is homotopic to the corresponding cycle in Γ_2, Theorem 5.3.1 implies that some isotopy of S takes Γ_1 to Γ_2, not necessarily pointwise, but preserving the orientations

of the cycles. [1] Therefore, up to composing with this isotopy, we can assume that each cycle in Γ_1 coincides, as a set, with the corresponding cycle in Γ_2, and with the same orientation. Hence, this isotopy induces an orientation-preserving map isomorphism i between Γ_1 and Γ_2, and since Γ_1 and Γ_2 have the same extended combinatorial maps, i can be viewed as an orientation-preserving map automorphism of Γ_1. As each cycle in Γ_1 is isotoped to the corresponding cycle in Γ_2, i maps each connected component of Γ_1 to itself, and it maps each crossing of Γ_1 to a crossing of Γ_1.

We now want to ensure that i is the identity map automorphism, which would imply the existence of a pointwise ambient isotopy between Γ_1 and Γ_2. However, this is not necessarily the case, as was pictured in Figure 5.6.

Let (V', E') be a connected component of G; let Γ'_1 be the arrangement of the cycles of Γ_1 corresponding to that connected component. If (V', E') has cyclomatic number one, by construction, Γ'_1 is just a cycle, which i maps to itself, preserving its orientation; so i is the identity map automorphism on Γ'_1.

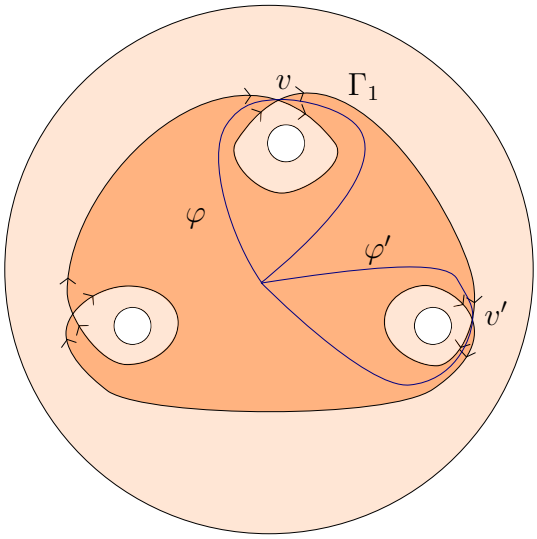


FIGURE A.1: A graph Γ_1 drawn on a sphere with four holes. The inner disk is the darker part, and the orientations at each vertex are pictured according to the orientation of the single cycle. If $i(v) = v'$, φ is sent to φ' which is not homotopic to it.

Otherwise, (V', E') has cyclomatic number at least two. We first note that Γ'_1 is connected; indeed, the inner disk D of (V', E') is bounded by all cycles in Γ'_1. Moreover, the faces of Γ_1 that are disks are exactly the inner disks; so i maps inner disks to inner disks, and therefore maps D to itself.

Let v be the vertex of Γ'_1 corresponding to the cycle φ in $\Phi = \Lambda \setminus \Gamma$ (see Figure A.1). If i maps v to another vertex v' of Γ'_1, as the inner disk is mapped to itself, i necessarily maps

1. Actually, if S has nonnegative Euler characteristic, the results in Section A.2 show that the isotopy can be chosen so as to be pointwise, which concludes the proof of this proposition.

φ to a cycle in the inner disk crossing v' once, i.e. another fundamental cycle in (V', E'), which is, as shown above, not homotopic to φ; this is a contradiction. So i maps v to itself. Furthermore, if we orient the four edges incident to v with the orientation of the corresponding cycles, v has two outgoing edges, consecutive in the cyclic order around v, and two incoming edges, also consecutive. Since i maps each edge to another edge with the same orientation, it maps v to v, and it is an orientation-preserving map automorphism, it must thus map each edge incident to v to itself, with the same orientation. Since Γ'_1 is connected, by propagation we deduce that i is the identity map automorphism on Γ'_1, which concludes the proof. □

A.1.2 Proof of Theorem 5.4.1 if the only face of G_1 is a disk

By Proposition 5.4.4, either (1) G_1 has no face that is a disk, or (2) G_1 has a single face, and that face is a disk. We proved Theorem 5.4.1 in case (1) in the previous section, and shall now deal with case (2). In other words, we assume that G_1 is a *cut-graph*.

In that case, the above construction does not seem to work: Since inner disks are not the only faces of Γ_1 that are disks, there is no guarantee that an inner disk is mapped to itself in the proof of Proposition 5.4.8. To circumvent this issue, the high-level idea is the following: We remove one cycle from G_1 so that the only face of G_1 is not a disk anymore but a cylinder, in which case the results from the previous section apply. We then check that the remaining cycle and its counterpart in G_2 are homotopic, and prove that this guarantees the existence of an isotopy between G_1 and G_2.

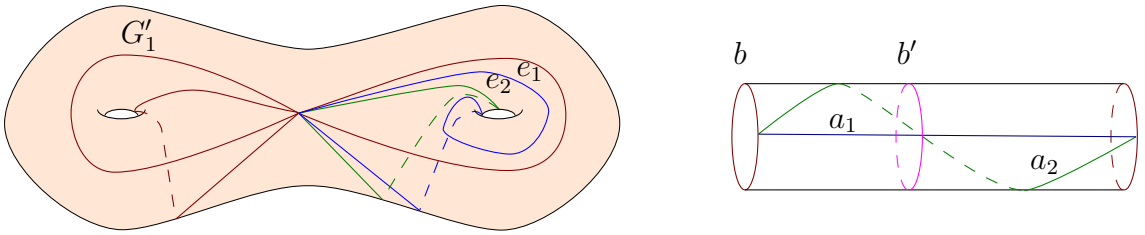
Since G_1 is a cut-graph, S has no boundary. Moreover, G is connected, and G_1 is made of a spanning tree $T = (V', E')$ (as in section 5.4.2.1) and $2g$ additional edges. Let e be one of these edges (chosen arbitrarily); let γ_e be the fundamental cycle with respect to T corresponding to edge e. Let G' be the graph G with edge e removed, and let G'_1 and G'_2 be the restrictions of G_1 and G_2 to G'. Note that G'_1 has a single face, which is a cylinder.

We can now apply the result of the previous section to G'_1 and G'_2: We obtain a family Λ' of cycles in G' with the property that, if their images in G'_1 and G'_2 are homotopic, then G'_1 and G'_2 are isotopic. Furthermore, Λ' can be computed in linear time, and uses each edge of G' at most thrice.

To prove Theorem 5.4.1 for our graph G, we take $\Lambda := \Lambda' \cup \{\gamma_e\}$. Obviously, Λ can be computed in linear time and uses each edge of G at most four times in total. Assume now that the images of each cycle of Λ in G_1 and G_2 are homotopic. It suffices to prove that, under this condition, some ambient isotopy of S takes G_1 to G_2. Since the images of each cycle in Λ' in G'_1 and G'_2 are homotopic, we may assume that $G'_1 = G'_2$. There remains to prove that an isotopy of the surface allows to push the image e_1 of e in G_1 to the image e_2 of e in G_2.

The surface obtained after cutting S along $G'_1 = G'_2$ is a cylinder C, and the images of e on C become arcs a_1 and a_2 with the same endpoints, one on each boundary, as shown on Figure A.2. (Indeed, if both endpoints were on the same boundary, a_1 would bound two

faces on C which would correspond to two faces on S, reaching a contradiction.) Now, to conclude, we only need to show that a_1 and a_2 are isotopic relatively to the boundary of this cylinder.



1. 2.

FIGURE A.2: 1. The graph $G_1' = G_2'$ and the edges e_1 and e_2. 2. The cylinder C obtained after cutting along G_1', with the arcs a_1 and a_2 corresponding to e_1 and e_2, and the cycles b and b'. If a_1 and a_2 are not isotopic, the images of γ_e in G_1 and G_2 cannot be (freely) homotopic

The end of the proof uses some elementary notions of homology, which we did not introduce in this thesis. It is a coarser notion of similarity for cycles than homotopy – we refer to [129] for the definitions. Assume, for the sake of a contradiction, that a_1 and a_2 are non-isotopic arcs relatively to the boundary of C. This implies that they are non-homotopic on C [81, Theorem 3.1]. Hence there exists an integer $n \neq 0$ such that a_2 is homotopic to $a_1 \cdot b^n$, where b is a loop that is a boundary of the cylinder C. Since the images of γ_e in G_1 and G_2 are (freely) homotopic on S, they are $\mathbb{Z}$-homologous. This implies that b^n, and thus b, has zero $\mathbb{Z}$-homology. By translating along the cylinder, b is homotopic on S to a simple cycle b' that crosses e_1 exactly once and crosses G_1 nowhere else, as pictured on Figure A.2. Hence b' is a simple cycle on S that crosses the image of γ_e in G_1 exactly once; thus b' is non-separating, and therefore cannot have zero $\mathbb{Z}$-homology. This contradiction completes the proof of Theorem 5.4.1.

A.2 Exceptional surfaces

In this section, we prove that Theorem 5.3.1 also holds for surface of nonnegative Euler characteristic. We note that in these cases we obtain a stronger theorem that in the general case, as the ambient isotopy we obtain maps the families of cycles pointwise. We split the proof in two parts depending on whether the surface is a plane with boundaries in Section A.2.1 or a torus in Section A.2.2. As a foreword, we note that the results of Lemmas 5.3.2, Corollary 5.3.3, and Lemma 5.3.4 and the first result of Proposition 5.3.5 (no cycle in a stable family is null-homotopic) also hold in the nonnegative Euler characteristic case.

A.2.1 Sphere, disk, and annulus

Since the Euler characteristic of a surface is $\chi(S) = 2 - 2g - b$, the only cases where the surface is a plane with boundaries and has nonnegative Euler characteristic are the sphere, the disk, and the annulus. In these cases, the proof of Theorem 5.3.1 is a simple corollary of Lemmas 5.3.2 and 5.3.4. We even obtain a slightly stronger statement, because we can take the isotopy to map *pointwise* Γ_1 to Γ_2:

Theorem A.2.1. *Let S be a sphere, a disk or an annulus and let $\Gamma_1 = (\gamma_{1,1}, \ldots, \gamma_{1,n})$ and $\Gamma_2 = (\gamma_{2,1}, \ldots \gamma_{2,n})$ be two* stable *families of cycles on S in general position such that:*

1. *there exists an oriented homeomorphism h of S mapping each cycle $\gamma_{1,j}$ of Γ_1 to the corresponding cycle $\gamma_{2,j}$ of Γ_2 not necessarily pointwise, but preserving the orientations of the cycles, and*

2. *each cycle of Γ_1 is homotopic to the corresponding cycle of Γ_2.*

Then there is an ambient isotopy of S mapping each cycle of Γ_1 to the corresponding cycle of Γ_2 pointwise.

Proof If the surface S is a sphere or a disk, all the cycles in Γ_1 and Γ_2 are null-homotopic, which is impossible as noted above. Hence these families are empty and the theorem is trivial.

If the surface is an annulus, for $i = 1, 2$, we claim that there are no crossing points in Γ_i, i.e., all the cycles are simple and two distinct cycles do not intersect each other. Indeed, if there were a crossing point, the connected component of Γ_i containing it would form a planar graph such that every vertex has degree four. Hence, by Corollary 5.3.3, there would be at least a k-gon with $k \leq 3$, contradicting, with Lemma 5.3.4, the stability of the family Γ_i.

Thus, Γ_1 is a family of disjoint simple cycles homotopic to the boundaries of the annulus, and the same holds for Γ_2. There is an isotopy of S mapping one family into the other if and only if they have the same ordering, as defined in the proof for the hyperbolic case. But this is exactly what the oriented homeomorphism between them ensures. This concludes the proof. □

A.2.2 Torus

The proof in the case of the torus is slightly more involved. Let us introduce a few definitions before delving into it. We choose a Euclidean metric on the torus, which induces one on its universal cover $\mathbb{R}^2$. This allows to define *translations* on the torus, which are projections of the usual translations of $\mathbb{R}^2$. Geodesics on the torus lift to straight lines in the plane, and two geodesics are homotopic if and only if these lines have the same slope, as a slope $s = \frac{m}{n}$ determines a unique element (m, n) with $m \wedge n = 1$ of the fundamental

group of the torus. When we mention the *slope* of a geodesic on the torus, we refer to the slope of one of its lifts in the universal cover. Note that as a translation is an isometry, it maps a geodesic to another geodesic.

Theorem A.2.2. *Let S be a torus and let $\Gamma_1 = (\gamma_{1,1}, \dots, \gamma_{1,n})$ and $\Gamma_2 = (\gamma_{2,1}, \dots \gamma_{2,n})$ be two* stable *families of cycles on S in general position such that:*

1. *there exists an oriented homeomorphism h of S mapping each cycle $\gamma_{1,j}$ of Γ_1 to the corresponding cycle $\gamma_{2,j}$ of Γ_2 not necessarily pointwise, but preserving the orientations of the cycles, and*

2. *each cycle of Γ_1 is homotopic to the corresponding cycle of Γ_2.*

Then there is an ambient isotopy of S mapping each cycle of Γ_1 to the corresponding cycle of Γ_2 pointwise.

Proof For all the cycles γ in Γ_1 or Γ_2, we start by applying de Graaf and Schrijver [64, Proposition 13] as in the proof of Proposition 5.3.5: Up to applying an isotopy of S, we can assume that γ is contained in the ε-neighborhood of one of its corresponding geodesics (or of a point, if γ is contractible). As in the proof of Proposition 5.3.5, we infer that γ is not contractible. Since in a torus, geodesic cycles are either simple or multiple concatenations of the same simple cycle, their ε-neighborhoods are annuli. Hence, every cycle in Γ_1 and Γ_2 can be assumed to lie in an annulus.

If one of these cycles $\gamma \in \Gamma_i$ is non-simple, it forms a graph embedded on an annulus such that every vertex has degree four. By Corollary 5.3.3, one of the faces of this graph is a disk with degree lower than four, which with Lemma 5.3.4 contradicts the stability of Γ_i. Thus all the cycles in Γ_1 and Γ_2 are simple. By the same argument, for $i = 1, 2$, two homotopic cycles in Γ_i do not cross each other.

The isotopy can then be found as follows. We split the proof in two cases.

Case 1: If all the cycles in Γ_1 are homotopic or inverse homotopic [1], we just pick an arbitrary one, say $\gamma_{1,1}$, and apply the pointwise isotopy mapping it to $\gamma_{2,1}$, which exists because they are homotopic [2]. Cutting the surface along $\gamma_{1,1} = \gamma_{2,1}$ gives an annulus. In this annulus, since all the other cycles in Γ_1 and Γ_2 are disjoint from these, the existence of an isotopy between them follows from the case of the annulus in Section A.2.1. After gluing back the boundaries together, this gives the desired isotopy of the torus.

Case 2: If there are at least two homotopy classes (modulo inversion) in Γ_1, we pick two representatives, say $\gamma_{1,1}$ and $\gamma_{1,2}$. As two couple of lines with the same slopes pairwise can be moved one to the other with a translation, by doing a translation of the torus, we can assume that $\gamma_{1,1}$ and $\gamma_{2,1}$ lie in the neighborhood of the same geodesic, as well as $\gamma_{1,2}$ and

1. We say that γ_1 and γ_2 are inverse homotopic if γ_1 is homotopic to γ_2^{-1}.

2. Simple and noncontractible cycles which are homotopic are also isotopic, as proved by Eppstein [81, Theorem 2.1].

$\gamma_{2,2}$, and furthermore that the crossing points between $\gamma_{1,1}$ and $\gamma_{1,2}$ lie in a ε-neighborhood of the corresponding crossing points between $\gamma_{2,1}$ and $\gamma_{2,2}$ [1].

We are now in the same situation as in Section 5.3.2: Since both couples of cycles $\gamma_{1,1}$ and $\gamma_{2,1}$ and $\gamma_{1,2}$ and $\gamma_{2,2}$ lie in the ε-neighborhood of the same geodesic, if we take as stable families $\Gamma'_1 = \gamma_{1,1} \cup \gamma_{1,2}$ and $\Gamma'_2 = \gamma_{2,1} \cup \gamma_{2,2}$, we can similarly define corridors, as well as edge and vertex polygons. Then Proposition 5.3.6 holds with exactly the same proof. Since there is only one cycle of Γ'_1 in each corridor, Lemma 5.3.7 also holds trivially. Hence by applying the same techniques, we can conclude that there exists an ambient isotopy mapping Γ'_1 to Γ'_2. Note that here, since the crossing points of the cycles in Γ'_1 have been matched, the isotopy we obtain is also pointwise.

Finally, cutting along these cycles cuts the surface into one or more disks, and the isotopy between Γ_1 and Γ_2 is obtained by applying Alexander's lemma separately on each of these disks. □

1. This is not necessarily the case a priori, since a given crossing point c between $\gamma_{1,1}$ and $\gamma_{2,1}$ can be matched to another crossing point than $h(c)$. Note that this is why the result in the torus case in stronger than in the general case, in which the crossing points can *not* necessarily be matched.

List of publications

[A] É. COLIN DE VERDIÈRE AND A. DE MESMAY, *Testing graph isotopy on surfaces*, Discrete and Computational Geometry, 51 (2014), pp. 171–206. Preliminary version in Proceedings of the Thirtieth Symposium on Computational Geometry, 2012. Cited on page 7, 14, 55, 82

[B] É. COLIN DE VERDIÈRE, A. HUBARD, AND A. DE MESMAY, *Discrete systolic inequalities and decompositions of triangulated surfaces*. Proceedings of the Twenty-Eighth Symposium on Computational Geometry, 2014, to appear. Cited on page 7, 14, 85

[C] B. BURTON, É. COLIN DE VERDIÈRE, AND A. DE MESMAY, *On the complexity of immersed normal surfaces*. Manuscript. An extended abstract was presented at the European Workshop on Computational geometry in 2014. Cited on page 7, 14, 109

[D] J. R. LEE, A. DE MESMAY, AND M. MOHARRAMI, *Dimension reduction for finite trees in* ℓ_1, Discrete & Computational Geometry, 50 (2013), pp. 977–1032. Preliminary version in Proceedings of the Twenty-Third ACM–SIAM Symposium on Discrete Algorithms, 2012.

Bibliography

[1] I. AGOL, *Problems known to be in both NP and coNP, but not known to be in P*. `http://mathoverflow.net/questions/31821`, 2010. Comment on one of the answers. Cited on page 47

[2] I. AGOL, J. HASS, AND W. THURSTON, *The computational complexity of knot genus and spanning area*, Transactions of the American Mathematical Society, 358 (2006), pp. 3821–3850. Cited on page 50, 111

[3] A. V. AHO, J. E. HOPCROFT, AND J. D. ULLMAN, *The design and analysis of computer programs*, Addison-Wesley, 1974. Cited on page 23

[4] I. R. AITCHISON, S. MATSUMOTO, AND J. H. RUBINSTEIN, *Surfaces in the figure-8 knot complement*, Journal of Knot Theory and Its Ramifications, 7 (1998), pp. 1005–1025. Cited on page 110

[5] S. ALAMDARI, F. FRATI, P. ANGELINI, A. LUBIW, T. M. CHAN, M. PATRIGNANI, G. DI BATTISTA, V. ROSELLI, S. SINGLA, AND B. T. WILKINSON, *Morphing planar graph drawings with a polynomial number of steps*, in Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2013, pp. 1656–1667. Cited on page 128

[6] L. C. ALEARDI, O. DEVILLERS, AND G. SCHAEFFER, *Optimal succinct representations of planar maps*, in Proceedings of the 22nd ACM Symposium on Computational Geometry., 2006, pp. 309–318. Cited on page 39

[7] L. C. ALEARDI, É. FUSY, AND T. LEWINER, *Schnyder woods for higher genus triangulated surfaces*, in Proceedings of the 24th ACM Symposium on Computational Geometry., 2008, pp. 311–319. Cited on page 39

[8] P. ALLIEZ, U. GIULANA, AND M. ATTENE, *Recent advances in remeshing of surfaces*, in Shape analysis and structuring, L. De Floriani and M. Spagnuolo, eds., Springer-Verlag, 2007. Cited on page 86

[9] P. ALLIEZ AND C. GOTSMAN, *Recent advances in compression of 3D meshes*, in Advances in multiresolution for geometric modelling, N. A. Dodgson, M. S. Floater, and M. A. Sabin, eds., Springer-Verlag, 2005, pp. 3–26. Cited on page 86

[10] P. ANGELINI, G. DA LOZZO, G. DI BATTISTA, F. FABRIZIO, M. PATRIGNANI, AND V. ROSELLI, *Morphing planar graph drawings optimally*. arXiv: 1402.4364, 2014. Cited on page 128

[11] K. APPEL AND W. HAKEN, *Every planar map is four-colorable*, AMS, Providence, Rhode Island, 1989. Cited on page 34

[12] H. ATTIYA, A. CASTAÑEDA, M. HERLIHY, AND A. PAZ, *Upper bound on the complexity of solving hard renaming*, in Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing, PODC '13, New York, NY, USA, 2013, ACM, pp. 190–199. Cited on page 43

[13] F. BALACHEFF, H. PARLIER, AND S. SABOURAU, *Short loop decompositions of surfaces and the geometry of Jacobians*, Geometric and Functional Analysis, 22 (2012), pp. 37–73. Cited on page 88

[14] J. L. BENTLEY AND TH. OTTMANN, *Algorithms for reporting and counting geometric intersections*, IEEE Transactions on Computing, 28 (1979), pp. 643–647. Cited on page 81

[15] S. BESPAMYATNIKH, *Computing homotopic shortest paths in the plane*, Journal of Algorithms, 49 (2003), pp. 284–303. Cited on page 57, 128

[16] ——, *Encoding homotopy of paths in the plane*, in LATIN 2004: Theoretical Informatics, 6th Latin American Symposium, vol. 2976 of Lecture Notes in Computer Science, Springer-Verlag, 2004, pp. 329–338. Cited on page 127, 128

[17] L. BESSIÈRES, G. BESSON, S. MAILLOT, M. BOILEAU, AND J. PORTI, *Geometrisation of 3-manifolds*, vol. 13 of EMS Tracts in Mathematics, European Mathematical Society (EMS), Zürich, 2010. Cited on page 49

[18] R. H. BING AND M. STARBIRD, *Linear isotopies in E^2*, Transactions of the American Mathematical Society, 237 (1978), pp. 205–222. Cited on page 128

[19] P. BLAGOJEVIĆ AND G. ZIEGLER, *Convex equipartitions using equivariant obstruction theory*, Israel Journal of Mathematics, (2014), pp. 1–29. Cited on page 40

[20] T. BLASIUS, S. KOBOUROV, AND I. RUTTER, *Simultaneous embedding of planar graphs*, in Handbook of Graph Drawing and Visualization, R. Tamassia, ed., CRC Press, 2012, pp. 349–381. Cited on page 129

[21] H. BODLAENDER, *Treewidth: Characterizations, applications and computations*, in Graph-Theoretic Concepts in Computer Science. 32nd International Workshop, WG 2006, Bergen, Norway, F. Fomin, ed., Lecture Notes in Computer Science, Springer, 2006. Cited on page 34

[22] P. BONSMA, *Surface split decompositions and subgraph isomorphism in graphs on surfaces*, in Proceedings of the 29th Annual Symposium on Theoretical Aspects of Computer Science (STACS), 2012, pp. 531–542. Cited on page 128

[23] G. BORRADAILE, E. D. DEMAINE, AND S. TAZARI, *Polynomial-time approximation schemes for subset-connectivity problems in bounded-genus graphs*, in Proceedings of the 26th Annual Symposium on Theoretical Aspects of Computer Science (STACS), 2009, pp. 171–182. Cited on page 37, 128

[24] G. BORRADAILE, J. R. LEE, AND A. SIDIROPOULOS, *Randomly removing g handles at once*, Computational Geometry: Theory and Applications, 43 (2010), pp. 655–662. Cited on page 39

[25] M. BOUSQUET-MÉLOU, *Counting planar maps, coloured or uncoloured*, Surveys in Combinatorics, (2011). Cited on page 39

[26] R. BROOKS AND E. MAKOVER, *Random construction of Riemann surfaces*, J. Differential Geom., 68 (2004), pp. 121–157. Cited on page 88, 129

[27] W. G. BROWN, P. ERDŐS, AND V. T. SÓS, *On the existence of triangulated spheres in 3-graphs, and related problems*, Period. Math. Hungar., 3 (1973), pp. 221–228. Cited on page 131

[28] G. BURDE AND H. ZIESCHANG, *Knots*, De Gruyter, 1985. Cited on page 21, 47

[29] B. BURTON, *Simplification paths in the pachner graphs of closed orientable 3-manifold triangulations*, in Proceedings of the 27th Annual Symposium on Computational geometry (SOCG), 2011, pp. 153–162. Cited on page 130

[30] B. BURTON AND J. SPREER, *The complexity of detecting taut angle structures on triangulations*, in Proceedings of the Twenty-Fourth Annual Symposium on Discrete Algorithms (SODA), 2013, pp. 168–183. Cited on page 111, 129

[31] B. A. BURTON AND R. G. DOWNEY, *Courcelle's theorem for triangulations*. arXiv: 1403.2926, 2014. Cited on page

[32] B. A. BURTON, T. LEWINER, J. A. PAIXÃO, AND J. SPREER, *Parameterized complexity of discrete morse theory*, in Proceedings of the Twenty-ninth Annual Symposium on Computational Geometry, SoCG '13, ACM, 2013, pp. 127–136. Cited on page 111, 129

[33] B. A. BURTON AND M. OZLEN, *A fast branching algorithm for unknot recognition with experimental polynomial-time behaviour*. arXiv:1211.1079, 2012. Cited on page 47

[34] B. A. BURTON AND W. PETTERSSON, *Fixed parameter tractable algorithms in combinatorial topology*. arXiv: 1402.3876, 2014. Cited on page 129

[35] P. BUSER, *Geometry and spectra of compact Riemann surfaces*, vol. 106 of Progress in Mathematics, Birkhäuser, 1992. Cited on page 63, 87, 97

[36] P. BUSER AND P. SARNAK, *On the period matrix of a Riemann surface of large genus (with an appendix by J.H. Conway and N.J.A. Sloane)*, Inventiones Mathematicae, 117 (1994), pp. 27–56. Cited on page 90, 91

[37] S. CABELLO, E. W. CHAMBERS, AND J. ERICKSON, *Multiple-source shortest paths in embedded graphs*, SIAM Journal on Computing, 42 (2013), pp. 1542–1571. Cited on page 86

[38] S. CABELLO, É. COLIN DE VERDIÈRE, AND F. LAZARUS, *Algorithms for the edge-width of an embedded graph*, Computational Geometry: Theory and Applications, 45 (2012), pp. 215–224. Cited on page 38, 85, 86, 97

[39] S. CABELLO, Y. LIU, A. MANTLER, AND J. SNOEYINK, *Testing homotopy for paths in the plane*, Discrete & Computational Geometry, 31 (2004), pp. 61–81. Cited on page 56, 57, 81

[40] S. CABELLO AND B. MOHAR, *Finding shortest non-separating and non-contractible cycles for topologically embedded graphs*, Discrete & Computational Geometry, 37 (2007), pp. 213–235. Cited on page 85

[41] ——, *Adding one edge to planar graphs makes crossing number hard*, in Proceedings of the 26th Annual Symposium on Computational Geometry (SOCG), ACM, 2010, pp. 68–76. Cited on page 35

[42] J. W. CANNON, W. J. FLOYD, R. KENYON, AND W. R. PARRY, *Hyperbolic geometry*, in Flavors of geometry, S. Levi, ed., Cambridge University Press, 1997. Cited on page 20

[43] G. CARLSSON, T. ISHKHANOV, V. DE SILVA, AND A. ZOMORODIAN, *On the local behavior of spaces of natural images*, Internation Journal of Computer Vision, 76 (2008), pp. 1–12. Cited on page 40

[44] A. CASTAÑEDA AND S. RAJSBAUM, *New combinatorial topology bounds for renaming: The upper bound*, J. ACM, 59 (2012), pp. 3:1–3:49. Cited on page 43

[45] E. CHAMBERS, J. ERICKSON, AND A. NAYYERI, *Minimum cuts and shortest homologous cycles*, in Proceedings of the 25th Annual Symposium on Computational Geometry (SOCG), ACM, 2009, pp. 377–385. Cited on page 86

[46] E. W. CHAMBERS, É. COLIN DE VERDIÈRE, J. ERICKSON, F. LAZARUS, AND K. WHITTLESEY, *Splitting (complicated) surfaces is hard*, Computational Geometry: Theory and Applications, 41 (2008), pp. 94–110. Cited on page 38, 93

[47] E. W. CHAMBERS, J. ERICKSON, AND A. NAYYERI, *Homology flows, cohomology cuts*, in Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC), 2009, pp. 273–282. Cited on page 86

[48] ——, *Homology flows, cohomology cuts*, SIAM Journal on Computing, 41 (2012), pp. 1605–1634. Cited on page 39

[49] F. CHAZAL, D. COHEN-STEINER, AND Q. MÉRIGOT, *Geometric inference for probability measures*, Journal on Foundations of computational Mathematics, 11 (2011). Cited on page 40

[50] C. CHEKURI AND A. SIDIROPOULOS, *Approximation algorithms for Euler genus and related problems*, in Proceedings of the 54th IEEE Annual Symposium on Foundations of Computer Science (FOCS), 2013, pp. 167–176. Cited on page 35

[51] J. CHEN, S. P. KANCHI, AND A. KANEVSKY, *A note on approximating graph genus*, Information Processing Letters, 61 (1997), pp. 317–322. Cited on page 35

[52] S.-W. CHENG, T. K. DEY, AND S.-H. POON, *Hierarchy of surface models and irreducible triangulations*, Computational Geometry: Theory and Applications, 27 (2004), pp. 135–150. Cited on page 86

[53] D. COHEN, A. COSTA, M. FARBER, AND T. KAPPELER, *Topology of random 2-complexes*, Discrete Comput. Geom., 47 (2012), pp. 117–149. Cited on page 130

[54] É. COLIN DE VERDIÈRE, *Shortest cut graph of a surface with prescribed vertex set*, in Proceedings of the 18th European Symposium on Algorithms (ESA), part 2, no. 6347 in Lecture Notes in Computer Science, 2010, pp. 100–111. Cited on page 37

[55] ——, *Topological algorithms for graphs on surfaces*, PhD thesis, École normale supérieure, 2012. Habilitation thesis, available at `http://www.di.ens.fr/~colin/`. Cited on page 31, 38, 85

[56] É. COLIN DE VERDIÈRE AND J. ERICKSON, *Tightening nonsimple paths and cycles on surfaces*, SIAM Journal on Computing, 39 (2010), pp. 3784–3813. Cited on page 26, 38, 57, 74, 128

[57] É. COLIN DE VERDIÈRE AND F. LAZARUS, *Optimal system of loops on an orientable surface*, Discrete & Computational Geometry, 33 (2005), pp. 507–534. Cited on page 57, 74

[58] ——, *Optimal pants decompositions and shortest homotopic cycles on an orientable surface*, Journal of the ACM, 54 (2007), p. Article 18. Cited on page 57, 87, 128

[59] A. COSTA AND M. FARBER, *The geometry and topology of random 2-complexes*. arXiv:1307.3614, 2013. Cited on page 130

[60] N. CREIGNOU, S. KHANNA, AND M. SUDAN, *Complexity classifications of Boolean constraint satisfaction problems*, SIAM Monographs on Discrete Mathematics and Applications, SIAM, 2001. Cited on page 115

[61] V. DALMAU AND D. FORD, *Generalized satisfiability with limited occurrences per variable: a study through delta-matroid parity*, in Mathematical Foundations of Computer Science (MFCS), vol. 2747 of Lecture Notes in Computer Science, 2003, pp. 358–367. Cited on page 111, 115

[62] S. DATTA, A. GOPALAN, R. KULKARNI, AND R. TEWARI, *Improved bounds for bipartite matching on surfaces*, in Proceedings of the 29th Annual Symposium on Theoretical Aspects of Computer Science (STACS), 2012, pp. 254–265. Cited on page 37

[63] H. DAVENPORT, *Multiplicative number theory*, vol. 74 of Graduate Texts in Mathematics, Springer-Verlag, New York, third ed., 2000. Revised and with a preface by Hugh L. Montgomery. Cited on page 47

[64] M. DE GRAAF AND A. SCHRIJVER, *Making curves minimally crossing by Reidemeister moves*, Journal of Combinatorial Theory, Series B, 70 (1997), pp. 134–156. Cited on page 66, 68, 69, 139

[65] M. DEHN, *Über unendliche diskontinuierliche Gruppen*, Math. Ann., 71 (1911), pp. 116–144. Cited on page 43

[66] M. DEHN, *Transformation der Kurven auf zweiseitigen Flächen*, Mathematische Annalen, 72 (1912), pp. 413–421. Cited on page 43

[67] E. D. DEMAINE, F. V. FOMIN, M. HAJIAGHAYI, AND D. M. THILIKOS, *Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs*, Journal of the ACM, 52 (2005), pp. 866–893. Cited on page 36, 39

[68] E. D. DEMAINE AND M. HAJIAGHAYI, *The bidimensionality theory and its algorithmic applications*, The Computer Journal, 51 (2008), pp. 292–302. Cited on page 39

[69] M. DESBRUN, M. MEYER, AND P. ALLIEZ, *Intrinsic parameterizations of surface meshes*, in Eurographics conference proceedings, 2002, pp. 209–218. Cited on page 37

[70] T. K. DEY, H. EDELSBRUNNER, AND S. GUHA, *Computational topology*, in Advances in Discrete and Computational Geometry – Proc. 1996 AMS-IMS-SIAM Joint Summer Research Conf. Discrete and Computational Geometry: Ten Years Later, B. Chazelle, J. E. Goodman, and R. Pollack, eds., no. 223 in Contemporary Mathematics, AMS, 1999, pp. 109–143. Cited on page 31

[71] T. K. DEY AND S. GUHA, *Transforming curves on surfaces*, Journal of Computer and System Sciences, 58 (1999), pp. 297–325. Cited on page 45, 57, 80

[72] R. DIESTEL, *Graph theory*, Springer-Verlag, 2000. Available at `http://diestel-graph-theory.com/`. Cited on page 22, 36

[73] M. P. DO CARMO, *Riemannian geometry*, Birkhäuser, 1992. Cited on page 19

[74] R. DYER, H. ZHANG, AND T. MÖLLER, *Surface sampling and the intrinsic Voronoi diagram*, Computer Graphics Forum, 27 (2008), pp. 1393–1402. Cited on page 94, 95

[75] H. EDELSBRUNNER AND J. HARER, *Persistent homology—a survey*, in Surveys in discrete and computational geometry, vol. 453 of Contemporary Mathematics, AMS, 2008, pp. 257–282. Cited on page 40

[76] ——, *Computational topology. An introduction*, AMS, Providence, Rhode Island, 2009. Cited on page 31

[77] A. EFRAT, S. G. KOBOUROV, AND A. LUBIW, *Computing homotopic shortest paths efficiently*, Computational Geometry: Theory and Applications, 35 (2006), pp. 162–172. Cited on page 57, 128

[78] J. A. ELLIS-MONAGHAN AND I. MOFFATT, *Graphs on surfaces*, Springer Briefs in Mathematics, Springer, New York, 2013. Dualities, polynomials, and knots. Cited on page 23

[79] D. EPPSTEIN, *Dynamic generators of topologically embedded graphs*, in Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2003, pp. 599–608. Cited on page 23, 37

[80] ——, *Squarepants in a tree: sum of subtree clustering and hyperbolic pants decomposition*, ACM Transactions on Algorithms, 5 (2009). Cited on page 87

[81] D. B. A. EPSTEIN, *Curves on 2-manifolds and isotopies*, Acta Mathematica, 115 (1966), pp. 83–107. Cited on page 17, 61, 73, 134, 137, 139

[82] D. B. A. EPSTEIN, J. W. CANNON, D. F. HOLT, S. V. F. LEVY, M. S. PATERSON, AND W. P. THURSTON, *Word processing in groups*, Jones and Bartlett Publishers, Boston, MA, 1992. Cited on page 44, 45

[83] J. ERICKSON, *Computational topology*, 2009. Course notes available at `http://compgeom.cs.uiuc.edu/~jeffe/teaching/comptop/`. Cited on page 31

[84] ——, *Combinatorial optimization of cycles and bases*, in Computational topology, A. Zomorodian, ed., Proceedings of Symposia in Applied Mathematics, AMS, 2012. Cited on page 85

[85] J. ERICKSON, K. FOX, AND A. NAYYERI, *Global minimum cuts in surface embedded graphs*, in Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2012, pp. 1309–1318. Cited on page 39, 85

[86] J. ERICKSON AND S. HAR-PELED, *Optimally cutting a surface into a disk*, Discrete & Computational Geometry, 31 (2004), pp. 37–59. Cited on page 37, 38, 97

[87] J. ERICKSON AND A. NAYYERI, *Computing replacement paths in surface-embedded graphs*, in Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2011, pp. 1347–1354. Cited on page 85

[88] ——, *Minimum cuts and shortest non-separating cycles via homology covers*, in Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2011, pp. 1166–1176. Cited on page 39

[89] ——, *Tracing compressed curves in triangulated surfaces*, Discrete & Computational Geometry, (2013). To appear. Cited on page 52

[90] J. ERICKSON AND A. SIDIROPOULOS, *A near-optimal approximation algorithm for Asymmetric TSP on embedded graphs*. arXiv:1304.1810, 2013. Cited on page 36

[91] J. ERICKSON AND K. WHITTLESEY, *Greedy optimal homotopy and homology generators*, in Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2005, pp. 1038–1046. Cited on page 74, 85

[92] ——, *Transforming curves on surfaces redux*, in Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2013, pp. 1646–1655. Cited on page 45, 57, 80

[93] J. ERICKSON AND P. WORAH, *Computing the shortest essential cycle*, Discrete & Computational Geometry, 44 (2010), pp. 912–930. Cited on page 85

[94] L. FAJSTRUP, M. RAUSSEN, AND É. GOUBAULT, *Algebraic topology and concurrency*, Theoretical Computer Science, 357 (2006), pp. 241–278. Cited on page 40

[95] B. FARB, *Automatic groups: a guided tour*, Enseign. Math. (2), 38 (1992), pp. 291–313. Cited on page 45

[96] B. FARB AND D. MARGALIT, *A primer on mapping class groups*, Princeton University Press, 2011. Cited on page 17, 25, 57, 62, 63, 133

[97] A. FATHI, F. LAUDENBACH, AND V. POÉNARU, eds., *Travaux de Thurston sur les surfaces*, Société Mathématique de France, 1991. Séminaire Orsay, Reprint of the 1979 edition, Astérisque No. 66-67. Cited on page 53

[98] T. FEDER, *Fanout limitations on constraint systems*, Theoretical Computer Science, 255 (2001), pp. 281–293. Cited on page 111, 116

[99] M. J. FISCHER, N. A. LYNCH, AND M. S. PATERSON, *Impossibility of distributed consensus with one faulty process*, Journal of the ACM, 32 (1985), pp. 374–382. Cited on page 42

[100] W. FLOYD AND U. OERTEL, *Incompressible surfaces via branched surfaces*, Topology, 23 (1984), pp. 117–125. Cited on page 53

[101] F. V. FOMIN AND D. M. THILIKOS, *Fast parameterized algorithms for graphs on surfaces: linear kernel and exponential speed-up*, in Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP), 2004, pp. 581–592. Cited on page 39

[102] L. FORD AND D. FULKERSON, *Maximum flow through a network*, Canadian Journal of Mathematics, 8 (1956), pp. 399–404. Cited on page 39

[103] PH. FRANKLIN, *The four-color problem*, American Journal of Mathematics, 44 (1922), pp. 225–236. Cited on page 34

[104] G. N. FREDERICKSON, *Fast algorithms for shortest paths in planar graphs, with applications*, SIAM Journal on Computing, 16 (1987), pp. 1004–1022. Cited on page 34

[105] E. GAFNI AND E. KOUTSOUPIAS, *3-processor tasks are indecidable*, in Proceedings of the 14th annual ACM symposium on Principle of distributed computing, 95, p. 271. Cited on page 42

[106] A. GAMBURD AND E. MAKOVER, *On the genus of a random riemann surface*, in Complex Manifolds and Hyperbolic Geometry, no. 311 in Contemporary Mathematics, AMS, 2002, pp. 133–140. Cited on page 88, 101

[107] J. GEELEN, T. HUYNH, AND R. B. RICHTER, *Explicit bounds for graph minors*. arXiv:1305.1451, 2013. Cited on page 88

[108] S. M. GERSTEN AND H. B. SHORT, *Small cancellation theory and automatic groups*, Invent. Math., 102 (1990), pp. 305–334. Cited on page 45

[109] R. GHRIST, *Barcodes: the persistent topology of data*, Bulletin of the AMS, 45 (2008), pp. 61–75. Cited on page 40

[110] J. R. GILBERT, J. P. HUTCHINSON, AND R. E. TARJAN, *A separator theorem for graphs of bounded genus*, Journal of Algorithms, 5 (1984), pp. 391–407. Cited on page 37

[111] C. GORDON AND J. LUECKE, *Knots are determined by their complements*, J. Amer. Math. Soc., 2 (1989), pp. 371–415. Cited on page 22

[112] C. GOTSMAN AND V. SURAZHSKY, *Guaranteed intersection-free polygon morphing*, Computers and Graphics, 25 (2001), pp. 67–75. Cited on page 128

[113] M. GREENDLINGER, *Dehn's algorithm for the word problem*, Comm. Pure Appl. Math., 13 (1960), pp. 67–83. Cited on page 44

[114] ——, *On Dehn's algorithms for the conjugacy and word problems, with applications*, Comm. Pure Appl. Math., 13 (1960), pp. 641–677. Cited on page 44

[115] M. GROMOV, *Filling Riemannian manifolds*, Journal of Differential Geometry, 18 (1983), pp. 1–147. Cited on page 86, 90, 91

[116] ——, *Systoles and intersystolic inequalities*, in Actes de la table ronde de géométrie différentielle, 1992, pp. 291–362. Cited on page 86, 90, 91

[117] J. L. GROSS AND T. W. TUCKER, *Topological graph theory*, Wiley, 1987. Cited on page 23, 39

[118] X. GU AND S.-T. YAU, *Global conformal surface parameterization*, in Proceedings of the Eurographics/ACM Symposium on Geometry Processing, 2003, pp. 127–137. Cited on page 86

[119] I. GUSKOV AND Z. J. WOOD, *Topological noise removal*, in Proceedings of Graphics Interface, 2001, pp. 19–26. Cited on page 86

[120] L. GUTH, H. PARLIER, AND R. YOUNG, *Pants decompositions of random surfaces*, Geometric and Functional Analysis, 21 (2011), pp. 1069–1090. Cited on page 87, 88, 93, 102, 105, 106, 129

[121] F. HADLOCK, *Finding a maximum cut of a planar graph in polynomial time*, SIAM Journal on Computing, 4 (1975), pp. 221–225. Cited on page 35

[122] W. HAKEN, *Theorie der Normalflachen, ein Isotopiekriterium für den Kreisnoten*, Acta Mathematica, 105 (1961), pp. 245–375. Cited on page 46

[123] T. C. HALES, *The jordan curve theorem, formally and informally*, The American Mathematical Monthly, 114 (2007), pp. 882–894. Cited on page 32

[124] H. HAMIDI-TEHRANI, *On complexity of the word problem in braid groups and mapping class groups*, Topology and its Applications, 105 (2000), pp. 237–259. Cited on page 57

[125] Q. HAN AND J.-X. HONG, *Isometric embedding of Riemannian manifolds in Euclidean spaces*, vol. 130 of Mathematical Surveys and Monographs, American Mathematical Society, Providence, RI, 2006. Cited on page 19

[126] J. HASS, *What is an almost normal surface*. arXiv:1208.0568v1, 2012. Cited on page 48

[127] J. HASS AND G. KUPERBERG, *New results on the complexity of recognizing the 3-sphere*, in Oberwolfach Reports, vol. 9, 2012, pp. 1425–1426. Cited on page 48

[128] J. HASS, J. C. LAGARIAS, AND N. PIPPENGER, *The computational complexity of knot and link problems*, Journal of the ACM, 46 (1999), pp. 185–211. Cited on page 46

[129] A. HATCHER, *Algebraic topology*, Cambridge University Press, 2002. Available at `http://www.math.cornell.edu/~hatcher/`. Cited on page 15, 72, 89, 137

[130] ——, *Notes on basic 3-manifold topology*. Notes available on the author's webpage, 2007. Cited on page 110

[131] J. HEMPEL, *3-manifolds*, AMS Chelsea Publishing, Providence, RI, 2004. Reprint of the 1976 original. Cited on page 5, 12, 56

[132] M. R. HENZINGER, PH. KLEIN, S. RAO, AND S. SUBRAMANIAN, *Faster shortest-path algorithms for planar graphs*, Journal of Computer and System Sciences, 55 (1997), pp. 3–23. Cited on page 34

[133] M. HERLIHY, D. KOZLOV, AND S. RAJSBAUM, *Distributed Computing through Combinatorial Topology*, Elsevier, Morgan Kaufmann, 2014. Cited on page 41

[134] M. HERLIHY AND S. RAJSBAUM, *The decidability of distributed decision tasks (extended abstract)*, in Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, 1997, pp. 589–598. Cited on page 43

[135] M. HERLIHY AND N. SHAVIT, *The topological structure of asynchronous computability*, Journal of the ACM, 46 (1999), pp. 858–923. Cited on page 41, 42

[136] M. P. HERLIHY AND S. RAJSBAUM, *Set consensus using arbitrary objects*, in Proceedings of the 13th Annual Symposium on Principles of Distributed Computing, 1994. Cited on page 42

[137] J. HERSHBERGER AND J. SNOEYINK, *Computing minimum length paths of a given homotopy class*, Computational Geometry: Theory and Applications, 4 (1994), pp. 63–98. Cited on page 57, 128

[138] M. W. HIRSCH, *Differential topology*, vol. 33 of Graduate Texts in Mathematics, Springer-Verlag, 1994. Corrected reprint of the 1976 original. Cited on page 17

[139] J. HOPCROFT AND R. TARJAN, *Efficient planarity testing*, Journal of the ACM, 21 (1974), pp. 549–568. Cited on page 34

[140] J. E. HOPCROFT AND J. K. KONG, *Linear time algorithm for isomorphism of planar graphs*, in Proceedings of the 6th Annual ACM Symposium on Theory of Computing (STOC), 1974, pp. 172–184. Cited on page 35

[141] J. P. HUTCHINSON, *On short noncontractible cycles in embedded graphs*, SIAM Journal on Discrete Mathematics, 1 (1988), pp. 185–192. Cited on page 86, 87, 92, 94, 95

[142] P. INDYK AND A. SIDIROPOULOS, *Probabilistic embeddings of bounded genus graphs into planar graphs*, in Proceedings of the 23rd Annual Symposium on Computational Geometry (SOCG), ACM, 2007, pp. 204–209. Cited on page 39

[143] G. F. ITALIANO, Y. NUSSBAUM, P. SANKOWSKI, AND CH. WULFF-NILSEN, *Improved algorithms for Min Cut and Max Flow in undirected planar graphs*, in Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC), 2011, pp. 313–322. Cited on page 34

[144] W. JACO, *Peking summer school*, 2005. Lecture Notes available at `https://www.math.oakstate.edu/~jaco/pekinglectures.htm`. Cited on page 49

[145] W. JACO AND J. H. RUBINSTEIN, *0-efficient triangulations of 3-manifolds*, Journal of Differential Geometry, 65 (2003), pp. 61–168. Cited on page 130

[146] R. KARASEV, A. HUBARD, AND B. ARONOV, *Convex equipartitions: the spicy chicken theorem*, Geometriae Dedicata, (2013). Cited on page 40

[147] M. KATZ, *Systolic geometry and topology*, vol. 137 of Mathematical Surveys and Monographs, AMS, 2007. With an appendix by J. Solomon. Cited on page 86, 90

[148] M. G. KATZ AND S. SABOURAU, *Entropy of systolically extremal surfaces and asymptotic bounds*. arXiv:math/041031, 2004. Cited on page 90, 91

[149] K.-I. KAWARABAYASHI, P. N. KLEIN, AND C. SOMMER, *Linear-space approximate distance oracles for planar, bounded-genus and minor-free graphs*, in Proceedings of the 38th International Colloquium on Automata, Languages and Programming (ICALP), part 1, 2011, pp. 135–146. Cited on page 37

[150] K.-I. KAWARABAYASHI AND B. MOHAR, *Some recent progress and applications in graph minor theory*, Graphs and combinatorics, 23 (2007), pp. 1–46. Cited on page 36

[151] K.-I. KAWARABAYASHI, B. MOHAR, AND B. REED, *A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width*, in Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2008, pp. 771–780. Cited on page 35

[152] K.-I. KAWARABAYASHI AND B. REED, *A separator theorem in minor-closed classes*, in Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2010, pp. 153–162. Cited on page 36

[153] L. KETTNER, *Using generic programming for designing a data structure for polyhedral surfaces*, Computational Geometry: Theory and Applications, 13 (1999), pp. 65–90. Cited on page 23

[154] P. KLEIN AND S. MOZES, *Optimization algorithms for planar graphs*, 2014. Book draft available at `http://planarity.org`. Cited on page 35

[155] W. KLINGENBERG, *Riemannian Geometry*, Philosophie und Wissenschaft, de Gruyter, 1995. Cited on page 95

[156] K. H. KO, S. J. LEE, J. H. CHEON, J. W. HAN, J.-S. KANG, AND C. PARK, *New public-key cryptosystem using braid groups*, in Advances in Cryptology – CRYPTO 2000, vol. 1880 of Lecture Notes in Computer Science, Springer, 2000, pp. 166–183. Cited on page 57

[157] P. KOIRAN, *Hilbert's Nullstellensatz is in the polynomial hierarchy*, J. Complexity, 12 (1996), pp. 273–286. Special issue for the Foundations of Computational Mathematics Conference (Rio de Janeiro, 1997). Cited on page 47

[158] P. B. KRONHEIMER AND T. S. MROWKA, *Dehn surgery, the fundamental group and $SU(2)$*, Math. Res. Lett., 11 (2004), pp. 741–754. Cited on page 47

[159] G. KUPERBERG, *Knottedness is in NP, modulo GRH*, Adv. Math., (2014). To appear. Cited on page 47

[160] M. KUTZ, *Computing shortest non-trivial cycles on orientable surfaces of bounded genus in almost linear time*, in Proceedings of the 22nd Annual Symposium on Computational Geometry (SOCG), ACM, 2006, pp. 430–438. Cited on page 85

[161] M. LACKENBY, *A polynomial upper bound on Reidemeister moves*. arXiv:1302.0180, 2013. Cited on page 47, 53, 130

[162] Y. LADEGAILLERIE, *Classes d'isotopie de plongements de* 1*-complexes dans les surfaces*, Topology, 23 (1984), pp. 303–311. Cited on page 59, 73

[163] 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 Symposium on Computational Geometry (SOCG), ACM, 2001, pp. 80–89. Cited on page 88

[164] F. LAZARUS AND J. RIVAUD, *On the homotopy test on surfaces*, in Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2012, pp. 440–449. Cited on page 45, 57, 80

[165] J.-F. LE GALL, *Large random planar maps and their scaling limits*, in Proceedings of the 5th European Congress of Mathematics, 2010. Cited on page 39

[166] F. LE ROUX, *Étude topologique de l'espace des homéomorphismes de Brouwer*, PhD thesis, Institut Fourier, 1997. Cited on page 83

[167] J. R. LEE AND A. SIDIROPOULOS, *Genus and the geometry of the cut graph*, in Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2010, pp. 193–201. Cited on page 40, 86

[168] A. LEMPEL AND J. ZIV, *On the complexity of finite sequences*, IEEE Trans. Information Theory, IT-22 (1976), pp. 75–81. Cited on page 51

[169] D. LETSCHER, *On persistent homotopy, knotted complexes and the alexander module*, in Proceedings of the 3rd Innovations in Theoretical computer Science Conference, 2012, pp. 428–441. Cited on page 40

[170] D. M. LETSCHER, *Immersed normal surfaces and decision problems for 3-manifolds*, PhD thesis, University of Michigan, 1997. Cited on page 110

[171] B. LÉVY AND J.-L. MALLET, *Non-distorted texture mapping for sheared triangulated meshes*, in Proceedings of the 25th Annual Conference on Computer Graphics (SIGGRAPH), 1998, pp. 343–352. Cited on page 86

[172] X. LI, X. GU, AND H. QIN, *Surface mapping using consistent pants decomposition*, IEEE Transactions on Visualization and Computer Graphics, 15 (2009), pp. 558–571. Cited on page 86

[173] S. LINS, *Graph-encoded maps*, Journal of Combinatorial Theory, Series B, 32 (1982), pp. 171–181. Cited on page 23

[174] R. J. LIPTON AND R. E. TARJAN, *A separator theorem for planar graphs*, SIAM Journal on Applied Mathematics, 36 (1979), pp. 177–189. Cited on page 33

[175] L. LOVÀSZ, *Kneser's conjecture, chromatic number and homotopy*, J. Combinatorial Theory, Ser. A, 25 (1978), pp. 319–325. Cited on page 40

[176] R. C. LYNDON AND P. E. SCHUPP, *Combinatorial group theory*, vol. 89 of A Series of Modern Surveys in Mathematics, Springer-Verlag, 1977. Cited on page 44

[177] E. MAKOVER AND J. MCGOWAN, *The length of closed geodesics on random Riemann surfaces*, Geom. Dedicata, 151 (2011), pp. 207–220. Cited on page 88

[178] A. A. MARKOV, *Insolubility of the problem of homeomorphy*, 1958. Translated from German by Afra Zomorodian. Cited on page 46

[179] J. MATOUŠEK, *Using the Borsuk-Ulam theorem*, Universitext, Springer-Verlag, 2003. Cited on page 40

[180] J. MATOUŠEK AND B. GÄRTNER, *Understanding and using linear programming*, Springer-Verlag, 2007. Cited on page 110

[181] J. MATOUŠEK, E. SEDGWICK, M. TANCER, AND U. WAGNER, *Untangling two systems of noncrossing curves*, in Graph Drawing, S. Wismath and A. Wolff, eds., vol. 8242 of Lecture Notes in Computer Science, Springer International Publishing, 2013, pp. 472–483. Cited on page 88

[182] T. MATSUI, *The minimum spanning tree problem on a planar graph*, Discrete Applied Mathematics, 58 (1995), pp. 91–94. Cited on page 34

[183] S. MATSUMOTO AND R. RANNARD, *The regular projective solution space of the figure-eight knot complement*, Experimental Mathematics, 9 (2000), pp. 221–234. Cited on page 110, 111, 115

[184] A. MIJATOVIĆ, *Simplifying triangulations of S^3*, Pacific J. Math., 208 (2003), pp. 291–324. Cited on page 130

[185] G. L. MILLER, *Finding small simple cycle separators for 2-connected planar graphs*, in STOC84, 1984, pp. 376–382. Cited on page 33

[186] G. L. MILLER, S.-H. TENG, W. THURSTON, AND S. A. VAVASIS, *Separators for sphere-packings and nearest neighbor graphs*, Journal of the ACM, 44 (1997), pp. 1–29. Cited on page 33

[187] ——, *Geometric separators for finite-element meshes*, SIAM Journal on Scientific Computing, 19 (1998), pp. 364–386. Cited on page 33

[188] B. MOHAR AND C. THOMASSEN, *Graphs on surfaces*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 2001. Cited on page 22, 23, 39

[189] E. E. MOISE, *Affine structures in 3-manifolds. V. The triangulation theorem and Hauptvermutung*, Ann. of Math. (2), 56 (1952), pp. 96–114. Cited on page 21

[190] L. MOSHER, *Mapping class groups are automatic*, Annals of Mathematics, Second Series, 142 (1995), pp. 303–384. Cited on page 45, 57

[191] ——, *What is a train track?*, Notices of the AMS, 50 (2003), pp. 354–356. Cited on page 52

[192] P. S. NOVIKOV, *Ob algoritmičeskoĭ nerazrešimosti problemy tožděstva slov v teorii grupp*, Trudy Mat. Inst. im. Steklov. no. 44, Izdat. Akad. Nauk SSSR, Moscow, 1955. Cited on page 45

[193] V. PATEL, *Determining edge expansion and other connectivity measures of graphs of bounded genus*, in Proceedings of the 18th European Symposium on Algorithms (ESA), no. 6346 in Lecture Notes in Computer Science, 2010, pp. 561–572. Cited on page 37

[194] R. C. PENNER, *Combinatorics of train tracks*, Princeton University Press, 1992. Cited on page 52

[195] G. PERELMAN, *The entropy formula for the Ricci flow and its geometric application*. arXiv:math/0211159, 2002. Cited on page 2, 10

[196] ——, *Finite extinction time for the solutions to the Ricci flow on certain three-manifolds*. arXiv:math/0307245, 2003. Cited on page 2, 10

[197] D. PIPONI AND G. BORSHUKOV, *Seamless texture mapping of subdivision surfaces by model pelting and texture blending*, in Proceedings of the 27th Annual Conference on Computer Graphics (SIGGRAPH), 2000, pp. 471–478. Cited on page 86

[198] N. PIPPENGER AND K. SCHLEICH, *Topological characteristics of random triangulated surfaces*, Random Structures Algorithms, 28 (2006), pp. 247–288. Cited on page 88, 129

[199] H. POINCARÉ, *Analysis situs*, Journal de l'École Polytechnique, 1 (1895), pp. 1–121. Reprinted in Œuvres VI:193–288. Cited on page 38

[200] S.-H. POON AND S. THITE, *Pants decomposition of the punctured plane*, in Proceedings of the 22nd European Workshop on Computational Geometry (EWCG), 2006, pp. 99–102. Cited on page 87

[201] J.-PH. PRÉAUX, *Conjugacy problem in groups of oriented geometrizable 3-manifolds*, Topology, 45 (2006), pp. 171–208. Cited on page 49

[202] T. M. PRZYTYCKA AND J. H. PRZYTYCKI, *On a lower bound for short noncontractible cycles in embedded graphs*, SIAM Journal on Discrete Mathematics, 3 (1990), pp. 281–293. Cited on page 87

[203] ——, *Surface triangulations without short noncontractible cycles*, in Graph structure theory, N. Robertson and P. Seymour, eds., no. 147 in Contemporary Mathematics, AMS, 1993, pp. 303–340. Cited on page 87, 95

[204] ——, *A simple construction of high representativity triangulations*, Discrete Mathematics, 173 (1997), pp. 209–228. Cited on page 87

[205] P. M. PU, *Some inequalities in certain nonorientable riemannian manifolds*, Pacific Journal of Mathematics, 2 (1952), pp. 55–71. Cited on page 86

[206] T. RADO, *Über den Begriff der Riemannschen Fläche*, Acta scientiarum mathematicarum (Szeged), 2 (1924), pp. 101–121. Cited on page 22

[207] A. A. RANICKI, *On the Hauptvermutung*, in The Hauptvermutung book, vol. 1 of K-Monogr. Math., Kluwer Acad. Publ., Dordrecht, 1996, pp. 3–31. Cited on page 23

[208] R. RANNARD, *Computing immersed normal surfaces in the figure-eight knot complement*, Experimental Mathematics, 8 (1999), pp. 73–84. Cited on page 110

[209] J. H. REIF, *Minimum $s-t$ cut of a planar undirected network in $O(n\log^2(n))$ time*, SIAM Journal on Computing, 12 (1983), pp. 71–81. Cited on page 35

[210] G. RINGEL, *Teilungen der Ebene durch Geraden oder topologische Geraden*, Mathematische Zeitschrift, 64 (1955), pp. 79–102 (1956). Cited on page 68

[211] ——, *Map color theorem*, Springer-Verlag, 1974. Cited on page 39

[212] R. L. RIVEST AND J. VUILLEMIN, *On recognizing graph properties from adjacency matrices*, Theoretical Computer Science, 3 (1976), pp. 371–384. Cited on page 40

[213] N. ROBERTSON AND P. D. SEYMOUR, *Graph minors. VII. Disjoint paths on a surface*, Journal of Combinatorial Theory, Series B, 45 (1988), pp. 212–254. Cited on page 86

[214] N. ROBERTSON AND P. D. SEYMOUR, *Graph minors. XVII. Taming a vortex*, J. Combin. Theory Ser. B, 77 (1999), pp. 162–210. Cited on page 36

[215] N. ROBERTSON AND P. D. SEYMOUR, *Graph minors. XX. Wagner's conjecture*, Journal of Combinatorial Theory, Series B, 92 (2004), pp. 325–357. Cited on page 34, 36

[216] N. ROBERTSON AND R. P. VITRAY, *Representativity of surface embeddings*, in Paths, flows, and VLSI-layout, B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, eds., Springer-Verlag, Berlin, 1990, pp. 293–328. Cited on page 39

[217] J. H. RUBINSTEIN, *An algorithm to recognize the* 3*-sphere*, in Proceedings of the International Congress of Mathematicians, Vol. 1, 2 (Zürich, 1994), Basel, 1995, Birkhäuser, pp. 601–611. Cited on page 48

[218] W. RUDIN, *Real and complex analysis*, McGraw-Hill Book Co., New York, third ed., 1987. Cited on page 93

[219] J. RUÉ, I. SAU, AND D. M. THILIKOS, *Dynamic programming for graphs on surfaces*, in Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP), vol. 6198 of Lecture Notes in Computer Science, 2010, pp. 372–383. Cited on page 128

[220] ——, *Dynamic programming for minor-free graphs*, in Computing and Combinatorics, Proceedings of the 18th Annual International Conference, COCOON, 2012. Cited on page 36

[221] S. SABOURAU, *Asymptotic bounds for separating systoles on surfaces*, Commentarii Mathematici Helvetici, 83 (2008), pp. 35–54. Cited on page 90, 91

[222] M. SCHAEFER, *The graph crossing number and its variants: a survey*, The electronic journal of combinatorics, (2013). Dynamic Survey. Cited on page 35

[223] M. SCHAEFER, E. SEDGWICK, AND D. ŠTEFANKOVIČ, *Spiraling and folding: the topological view*, in Proceedings of the Nineteenth Canadian Conference on Computational Geometry, 2007, pp. 73–76. Cited on page 52

[224] M. SCHAEFER, E. SEDGWICK, AND D. ŠTEFANKOVIČ, *Computing Dehn twists and geometric intersection numbers in polynomial time*, in Proceedings of the 20th Canadian Conference on Computational Geometry (CCCG), 2008, pp. 111–114. Cited on page 51

[225] M. SCHAEFER, E. SEDGWICK, AND D. ŠTEFANKOVIČ, *Spiraling and folding: the word view*, Algorithmica, 60 (2011), pp. 609–626. Cited on page 52

[226] M. SCHAEFER, E. SEDGWICK, AND D. ŠTEFANKOVIČ, *Algorithms for normal curves and surfaces*, in Proceedings of the 8th International Conference on Computing and Combinatorics (COCOON), 2002, pp. 370–380. Cited on page 51, 52

[227] ——, *Recognizing strip graphs in NP*, Journal of Computer and System Sciences, 67 (2003), pp. 365–380. Cited on page 51, 52

[228] M. SCHAEFER AND D. ŠTEFANKOVIČ, *Decidability of string graphs*, Journal of Computer and System Sciences, 68 (2004), pp. 319–334. Cited on page 52

[229] T. J. SCHAEFER, *The complexity of satisfiability problems*, in Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC), 1978, pp. 216–226. Cited on page 115, 116

[230] S. SCHLEIMER, *Sphere recognition lies in NP*. Manuscript available on the author's webpage, 2011. Cited on page 48

[231] P. SCOTT AND H. SHORT, *The homeomorphism problem for closed 3-manifolds*. To appear in Algebraic and Geometric Topology, 2014. Cited on page 49

[232] H. SEIFERT, *Über das Geschlecht von Knoten*, Math. Ann., 110 (1935), pp. 571–592. Cited on page 21

[233] P. D. SEYMOUR AND R. THOMAS, *Call routing and the ratcatcher*, Combinatorica, 14 (1994), pp. 217–241. Cited on page 35

[234] A. SIDIROPOULOS, *Optimal stochastic planarization*, in Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2010, pp. 163–170. Cited on page 39

[235] M. SPIVAK, *A comprehensive introduction to differential geometry*, vol. II, Publish or Perish Press, 1999. Cited on page 90

[236] D. ŠTEFANKOVIČ, *Algorithms for Simple Curves on Surfaces, String Graphs and Crossing Numbers*, PhD thesis, University of Chicago, Illinois, june 2005. Cited on page 51

[237] K. STEPHENSON, *Introduction to circle packing, the theory of discrete analytic functions*, Cambridge University Press, Moscow, 2005. Cited on page 33

[238] J. STILLWELL, *Classical topology and combinatorial group theory*, Springer-Verlag, New York, 1980. Cited on page 15

[239] ——, *Classical topology and combinatorial group theory*, Springer-Verlag, New York, second ed., 1993. Cited on page 46

[240] D. W. SUMNERS, *Untangling DNA*, Math. Intelligencer, 12 (1990), pp. 71–80. Cited on page 2, 9

[241] C. THOMASSEN, *The graph genus problem is NP-complete*, Journal of Algorithms, 10 (1989), pp. 568–576. Cited on page 35

[242] ——, *Embeddings of graphs with no short noncontractible cycles*, Journal of Combinatorial Theory, Series B, 48 (1990), pp. 155–177. Cited on page 39

[243] ——, *Five-coloring maps on surfaces*, Journal of Combinatorial Theory, Series B, 59 (1993), pp. 89–105. Cited on page 39

[244] A. THOMPSON, *Thin position and the recognition problem for S^3*, Mathematical Research Letters, 1 (1994), pp. 613–630. Cited on page 48

[245] M. THORUP, *Compact oracles for reachability and approximate distances in planar digraphs*, Journal of the ACM, 51 (2004), pp. 993–1024. Cited on page 34

[246] W. THURSTON, *Three-dimensional geometry and topology*, Princeton University Press, 1997. Edited by Silvio Levy. Cited on page 20

[247] A. TURING, *Solvable and unsolvable problems*, Science News, 31 (1954), pp. 7–23. Cited on page 46

[248] W. T. TUTTE, *A census of planar maps*, Canadian Journal of Mathematics, 15 (1963), pp. 249–271. Cited on page 39

[249] T. UNIVALENT FOUNDATIONS PROGRAM, *Homotopy Type Theory: Univalent Foundations of Mathematics*, `http://homotopytypetheory.org/book`, Institute for Advanced Study, 2013. Cited on page 40

[250] K. WAGNER, *Über eine eigenschaft der ebenen komplexe*, Math. Ann., 114 (1937), pp. 570–590. Cited on page 34

[251] D. M. WALBA, *Topological stereochemistry*, Tetrahedron, 41 (1985), pp. 3161 – 3212. Cited on page 2, 10

[252] F. WALDHAUSEN, *The word problem in fundamental groups of sufficiently large irreducible* 3*-manifolds*, Ann. of Math. (2), 88 (1968), pp. 272–280. Cited on page 49

[253] Z. WOOD, H. HOPPE, M. DESBRUN, AND P. SCHRÖDER, *Removing excess topology from isosurfaces*, ACM Transactions on Graphics, 23 (2004), pp. 190–208. Cited on page 86

[254] E. ZHANG, K. MISCHAIKOW, AND G. TURK, *Feature-based surface parameterization and texture mapping*, ACM Transactions on Graphics, 24 (2005), pp. 1–27. Cited on page 37

[255] A. ZOMORODIAN, *Computational topology*, in Algorithms and Theory of Computation Handbook, M. J. Atallah, ed., Chapman & Hall, 2009. Cited on page 31

[256] A. ZOMORODIAN AND G. CARLSSON, *Computing persistent homology*, Discrete & Computational Geometry, 33 (2005), pp. 249–274. Cited on page 40

Index