

# Panorama

Francis Lazarus and Arnaud de Mesmay

September 19, 2016

---

## Contents

0.1 Topology. . . . .	1
0.2 Computational Topology. . . . .	2

---

### 0.1 Topology.

*Topology* deals with the study of *spaces*. One of its goals is to answer the following broad class of questions:

*“Are these two spaces the same?”*

This naturally leads to the following subquestions:

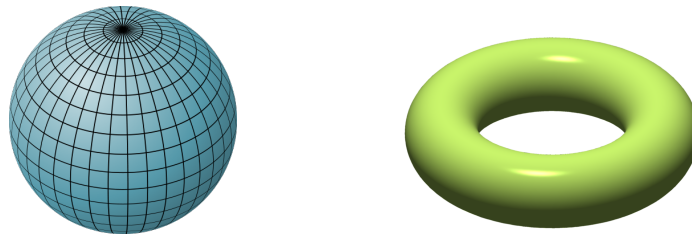
- **What is a space?** General topology typically defines topological spaces via open and closed sets. In order to avoid pathological examples, and with an eye towards applications, we will take a more concrete approach<sup>1</sup>: in this course, topological spaces will be obtained in the form of **complexes**, that is, by gluing together fundamental blocks. For example, gluing segments yields a graph, while by gluing together triangles one can obtain a surface (or something more complicated). The usual notions of distance on these fundamental blocks naturally induce a notion of proximity on such a complex, and therefore a topology whose properties are convenient to understand geometrically.
- **What is “the same” ?** It very much depends on the context. The most common equivalence relation is **homeomorphism**, which is a continuous map with a continuous inverse function. But in some contexts, when a space is **embedded** in another space, one will be interested in distinguishing between different embeddings. There, a convenient notion is **isotopy** : two embedded spaces will be considered the same if one can deform continuously one into the other one.

Let us look at examples.

---

<sup>1</sup>This is by no means original: see introductory textbooks on algebraic topology, for example Hatcher [Hat02] or Stillwell [Sti12].

**Example 1:** By gluing triangles or quadrilaterals, one can easily obtain a sphere (left figure), or a torus (right figure).



Are these two spaces homeomorphic? Obviously not: the torus has a hole. But what is a hole? Two naive answers will guide us to the two fundamental constructs of algebraic topology:

- **Homotopy:** On the sphere, every closed curve can be deformed into a single point. While on the torus, a curve going around the hole can not. Such a curve is not **homotopic** to a point.
- **Homology:** On the sphere, every closed curve separates the sphere into two regions. While on the torus, a curve going around the hole is not separating. Such a curve is not trivial in **homology**.

These intuitions can be formalized into algebraic objects which will constitute invariants (actually, **functors**) that one can use to distinguish topological spaces.

**Example 2:** By gluing segments in  $\mathbb{R}^3$ , one can obtain the following **knots**.



Are they homeomorphic? Certainly: they are both homeomorphic to the circle  $S^1$ . But are they **isotopic**: can one be deformed into the other without crossing itself? The answer is negative, but this is not that easy to prove. One way to see it is that the knot on the left bounds a disk, while the one on the right does not. Studying which surfaces one can find in a 3-dimensional space is the goal of **normal surface theory**.

## 0.2 Computational Topology.

**Computational topology** deals with effective computations on topological spaces. The main question now becomes:

*“How to compute whether these two spaces are the same?”*

Note that since we study spaces described by gluings of fundamental blocks, in most instances this can be made into a well-defined algorithmic problem, with a finite input. One can then wonder about the **complexity** of this problem, and aim to design the most efficient algorithm, or conversely prove hardness results. Throughout the course, we will investigate the complexity of various instances of this question, with practical algorithms computing homeomorphism, homotopy, homology or isotopy for example.

**Outline.** We will work by increasing progressively the dimension, and thus the complexity of the objects we consider.

1. We start with one of the simplest topological spaces : the **plane**  $\mathbb{R}^2$ . Describing it as a union of small blocks amounts to the study of **planar graphs**. This topological constraint on graphs has a strong impact on their combinatorics, which we will study through various angles.
2. Next come **surfaces**, which look locally like the plane. From a mathematical point of view, these are still fairly simple, as they can be easily classified. But once again, there is a very fruitful interplay between the topology of surfaces and the combinatorics of embedded graphs. Moreover, surfaces are a convenient and easy framework to introduce **homotopy** and **homology** and we will present efficient algorithms for the computation of these invariants.
3. In dimension 3, we will introduce **knots** and **3-manifolds**. Distinguishing various knots is hard: the whole field of **knot theory** is dedicated to this. We will see through various examples why this is hard, and will introduce **normal surface theory**, one of the main tools used for computational problems in 3 dimensions. As an application, we will use it to provide an algorithm to recognize trivial knots in **NP**.
4. As soon as we hit dimension 4, we start to hit the **limits** of computational topology: many problems are not only hard, they are **undecidable**. We will introduce **simplicial complexes**, which are the main model for high-dimensional topological spaces, and show that deciding homeomorphism of such complexes is already undecidable in dimension 4, as is testing the homotopy of curves in 2-dimensional complexes.
5. Although computing homotopy and homeomorphism quickly becomes intractable in high dimensions, homology does not: its simple algebraic structure allows for efficient computations that scale well with the dimension. This can be leveraged as a tool for big data: the techniques of **topological data analysis** aim at recognizing topological features in point clouds by computing their **persistent homology**. As we shall see, this is a surprisingly powerful way to infer information from a structured point cloud.

**Applications.** The approach in this course is to focus on the mathematical motivations to study topological objects and their computation. This does not mean that this

is all devoid of applications. Quite the contrary: topological spaces are ubiquitous in computer science, and the primitives we develop here have practical implications in computer graphics [LGQ09], mesh processing [GW01], robotics [Far08], combinatorial optimization (see references in [Eri12]), machine learning [ACC12], and many other fields. Believe it or not, they even revolutionized basketball [Bec12]!

## References

- [ACC12] Aaron Adcock, Erik Carlsson, and Gunnar Carlsson. The ring of algebraic functions on persistence bar codes. *Preprint available at <http://comptop.stanford.edu/u/preprints/multitwo.pdf>*, 2012.
- [Bec12] J. Beckham. Analytics reveal 13 new basketball positions. *Wired*, 2012.
- [Eri12] Jeff Erickson. Combinatorial optimization of cycles and bases. In Afra Zomorodian, editor, *Computational topology*, Proceedings of Symposia in Applied Mathematics. AMS, 2012.
- [Far08] Michael Farber. *Invitation to topological robotics*. European Mathematical Society, 2008.
- [GW01] Igor Guskov and Zoë J. Wood. Topological noise removal. In *Proceedings of Graphics Interface*, pages 19–26, 2001.
- [Hat02] Allen Hatcher. *Algebraic topology*. Cambridge University Press, 2002.
- [LGQ09] Xin Li, Xianfeng Gu, and Hong Qin. Surface mapping using consistent pants decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 15:558–571, 2009.
- [Sti12] John Stillwell. *Classical topology and combinatorial group theory*, volume 72. Springer Science & Business Media, 2012.