0 Introduction

The primary goal of the course is to present basic concepts from *topology* and *Morse theory* to enable a non-specialist to grasp and participate in current research in computational topology. As such, this course will *not* be a readings course in computational topology. Rather, I will present basic mathematical concepts from a computer scientist's point of view, focusing on computational challenges, and presenting algorithms and data-structures when appropriate. Near the end of the course, we will examine recent advances in the area by reading research papers from the area.

0.1 Why was this course organized?

Mathematics is written for mathematicians.
— Nicholas Copernicus (1473–1543)

The motive for organizing this course is that concepts in topology are useful in solving problems in computer science. These problems arise naturally in computational geometry, graphics, robotics, structural biology, and chemistry. Often, the questions themselves have been known and considered by topologists. Unfortunately, there are many barriers to interaction:

1. We do not know the language of topologists. Topology, unlike geometry, is not a required subject in high school mathematics, and almost never dealt with in undergraduate computer science. The problem is compounded by the axiomatic nature of topology, which generates a lot of cryptic terminology, making the field inaccessible to non-topologists.

2. Topology can be very unintuitive and therefore appear extremely complicated, often scaring away interested computer scientists.

3. Topology is a large field with many branches. We often need simple concepts from each branch. There are certainly a number of courses in topology offered by the Math department in which one may become acquainted with the material. However, the focus of these courses is theoretical, concerned with deep questions and existential results

Because of the relative dearth of interaction between topologists and computer scientists, there are many opportunities for research. Many topological questions have large complexity: the best known bound, if any bound is known, may be exponential. For example, I once heard a talk on an algorithm that ran in quadruply exponential time! Let me make this clear. It was

$$O(2^{2^{2^{2^x}}})!$$

And you may overhear topologists boasting that their software can now handle 14 tetrahedra, not just 13. However, better bounds may exist for questions that are not general, such as problems in low dimensions, where our interests chiefly lie. We need better algorithms, parallel algorithms, approximation schemes, data structures, and software to solve these problems within our life time (or the lifetime of the universe.)

The goal of this class is to make algorithmically minded individuals fluent in the language of topology. Currently, most researchers in computational topology have a mathematics background. My hope is to recruit more computer scientists into this emerging field.

0.2 What is Topology?

A topologist is a man who doesn't know the difference between a coffee cup and a donut.
— Unknown

Topology concerns itself with how things are connected, not how they look. Let's start with a few examples.

Example 0.1 (Loops of String) Imagine you're given two pieces of strings. We tie the ends of one of them, so it forms a loop. Are they connected the same way, or differently? One way to find out is to cut both, as shown in Figure 1. When we cut each string, we are obviously changing its connectivity. Since the result is different, they must have been connected differently to begin with.

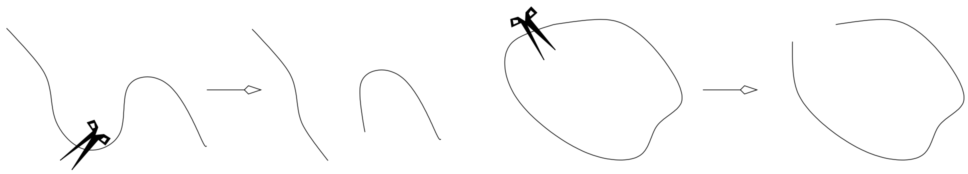
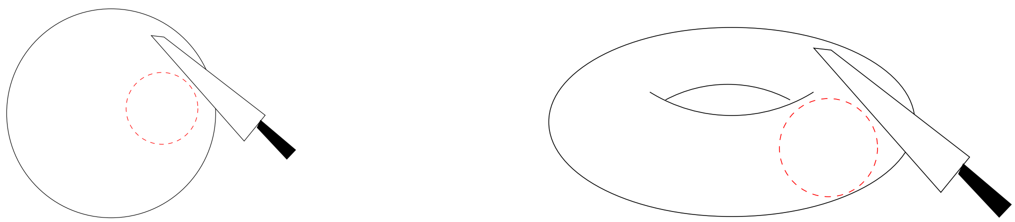


Figure 1. The string on the left is cut into two pieces. The loop string on the right is cut, but still is in one piece.

Example 0.2 (Sphere and Torus) Suppose you have a hollow ball (a sphere) and the surface of a donut (a torus.) When you cut the sphere anywhere, you get two pieces: the cap, and the sphere with a hole, as shown in Figure 2 (a) But there are ways you can cut the torus so that you only get one piece. Somehow, the torus is acting like our string



(a) No matter where we cut the sphere, we get two pieces

(b) If we're careful, we can cut the torus and still leave it in one piece.

Figure 2. Two pieces or one piece?

loop and the sphere like the untied string.

Example 0.3 (Holding hands) Imagine you're walking down a crowded street, holding somebody's hand. When you reach a telephone pole and have to walk on opposite sides of the pole, you let go of the other person's hand. Why? [Hint: Think loops of string...]

Let's look back to the first example. Before we cut the string, the two points near the cut are near each other. We say that they are *neighbors* or in each other's *neighborhoods*. After the cut, the two points are no longer neighbors, and their neighborhood has changed. This is the critical difference between the untied string and the loop: the former has two ends. All the points in the loop have two neighbors, to their left and right. But the untied string has two points, each of whom has a single neighbor. This is why the two strings have different connectivity. Note that this connectivity does not change if we deform or stretch the strings (as if they are made of rubber.) As long as we don't cut them, the connectivity remains the same.

You may be rightly suspicious by now, as the toy problem we dealt with is not that complicated. Can we say anything for more complicated spaces? It turns out we can.

Intrinsic topology. Topology attempts to understand the global connectivity of an object by considering how the object is connected locally. This understanding is really as classifications: objects are grouped into classes with the same connectivity. Topology identifies *intrinsic* properties of objects by transforming a space in some fixed way, and observing properties that do not change. We call these properties the *invariants* of the space. (Felix Klein gave this unifying definition for geometry and topology in his *Erlanger Programm* address in 1872.) For example, *Euclidean geometry* refers to the study of invariants under rigid motion in $\mathbb{R}^d$, e.g. moving a cube in space does not change its geometry (thank god!) Topology, on the other hand, studies invariants under continuous, and continuously invertible, transformations. For example, we can mold and stretch a play-doh ball into a filled cube by such transformations, but not into a donut shape.

Extrinsic topology. Topology is concerned not only with how an object is connected (intrinsic topology), but how it is *placed* within another space (extrinsic topology.) For example, suppose we put a knot on a string, and then tie its ends together. Clearly, the string has the same connectivity as the loop we saw in Example 0.1. But no matter how we move the string around, we cannot get rid of the knot (in topology terms, we cannot unknot the knot into the *unknot*.) Or can we? Can we prove that we cannot?

Don't worry. We shall make all this excruciatingly formal later on.

0.3 But why are we interested?

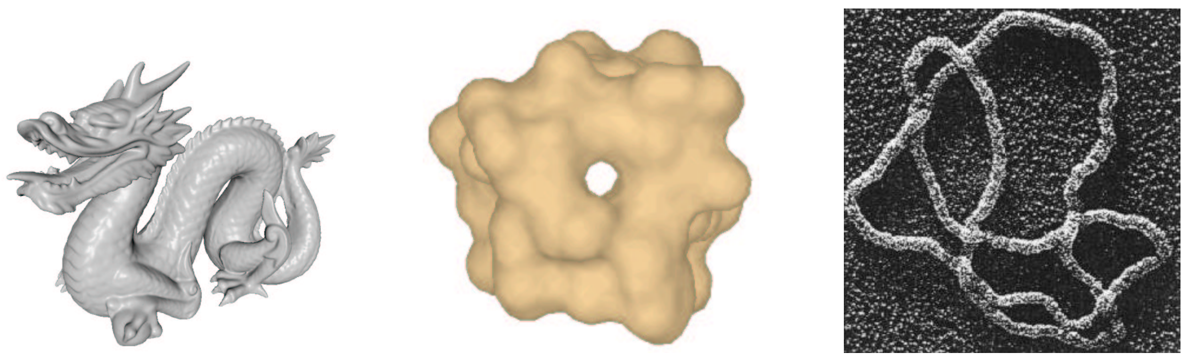
How can it be that mathematics, being after all a product of human thought independent of experience, is so admirably adapted to the objects of reality?

As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.

— Albert Einstein (1879–1955)

We are interested in topology because topological problems often arise in areas that we're *really* interested in. Not only that, whenever we encounter topology, it manifests itself in a pretty hairy fashion, so that we cannot do away with it by simple hacks. It keeps crashing our programs and making our lives miserable. Here are a few examples.

Example 0.4 (Graphics) Usually, a computer model is created by sampling the surface of an object and creating a point set. *Surface reconstruction*, a major area of research in computer graphics and computational geometry, refers to the recovery of the lost topology and geometry of a space. Often, we require a *watertight* surface, a surface with no holes or bad connections, as shown in Figure 3 (a). But this is a topological problem, as we are concerned with how



(a) The Stanford dragon, a surface represented by 3.2 million triangles. It has no holes, but 60 tunnels.

(b) Gramicidin A, a protein, with a tunnel for channeling ions across lipid membranes.

(c) A knotted DNA [1]

Figure 3. Some spaces with interesting topology

a surface is connected. Even if we get rid of all the holes, we are haunted by holes of another kind: tunnels. These tunnels, in turn, do not allow a full compression of the model. In topological terms, we need a 2-manifold with no handles.

Example 0.5 (Robotics) A robot must often plan a path in its world which contains many obstacles. We are interested in efficiently capturing and representing the *configuration space* in which a robot may travel. In other words, our representation of the configuration space must have the same connectivity as the space itself.

Example 0.6 (Geography) Planetary landscapes are modeled as elevations over grids, or triangulations in *geographic information systems*. Usually, there is noise inherent in the data, causing tiny mountains and lakes to arise. We'd like to remove this noise in a way that does not change important properties of the landscape, such as rain flow, or the watersheds of lakes and rivers.

Example 0.7 (Biology) A protein is a single chain of amino acids, which folds into a globular structure. The *Thermodynamics Hypothesis* states that a protein always folds into a state of minimum energy. To predict protein structure, we would like to model the folding of a protein computationally. As such, the *protein folding* problem becomes an optimization problem: we are looking for a path to the global minimum in a very high-dimensional energy landscape.

We are also interested in capturing the topology of proteins. The small protein in Figure 3 (b), for example, uses its tunnel to channel ions. Can we computationally find such features in proteins?

Example 0.8 (Chemistry) In the 1980's, it was shown that the DNA, the molecular structure of the genetic code of all living organisms, can become knotted during replication, as shown in Figure 3 (c). This finding initiated interest in knot theory among biologists and chemists for the detection, synthesis, and analysis of knotted molecules. One possibility is to build nano-scale chemical switches and logic gates with these structures. Eventually, chemical computer memory systems could be built from these building blocks.

Topology gives us tools and methodologies to tackle such problems. So, we become interested.

0.4 What next?

Young man, in mathematics you don't understand things, you just get used to them.
— John von Neumann (1903–1957)

We need to plow through literally hundreds of definitions from many different areas of mathematics. The right definition is often the most important step in solving a problem in topology. These definitions were refined in the last century to require the least initial assumptions, or *axioms*. The same refinement process, unfortunately, removed all intuition (impurities?) from the subject. This means that the first few weeks of the class will be especially dry and unmotivating, although I will try to provide some intuition when possible. We will not delve deeply into any one area of mathematics, but learn what we think is useful. Along the way, we will also discuss algorithms, existing software, and possible projects (things that I really need) for the class.

References

[1] WASSERMAN, S., DUNGAN, J., AND COZZARELLI, N. Discovery of a predicted DNA knot substantiates a model for site-specific recombination. *Science* **229** (1985), 171–174.