Introduction to Computational Topology

COSC 249.09 (Fall 2021)

Hsien-Chih Chang
HSIEN-CHIH.CHANG@DARTMOUTH.EDU

Course Description

Topology is the art of studying shapes without precise measurements. It is not surprising then that topology has found many applications in computer science, both in theoretical and applied research including algorithms and complexity theory, data analysis, robotics, computer graphics, etc., where often the input data is geometrically constrained, or noisy due to measurement errors.

This graduate-level course serves as an introduction to the rapidly growing area(s) of computational topology, from a theoretical point of view. Naturally, due to the vast amount of work and literature in the area, the topics covered in this class will be biased towards the interest and expertise of the instructor.

Course Objectives

By the end of the course we expect the students to be able to read articles and follow the basic terminology required to conduct research in computational topology. Also, we hope that the students enjoy the beauty of topology and have fun with the puzzles offered.

Prerequisite

Students are assumed to have reasonable math maturity, in particular the ability to read and write proofs. A proof-based course like *COSC 30 Discrete Mathematics* or equivalent is required as prerequisite. Experience in the analysis of algorithms (say COSC 31: algorithms) is *strongly recommended*. Previous exposure in linear algebra (like MATH 24 or COSC 70) and graph theory (like MATH 38) will also help. We will hand out homework 0 in the first week and give the students an idea of the background knowledge expected. Background in general topology (the study of topological spaces) is not required.

Related Courses

- *MATH 32 The Shape of Space*: This is an undergrad-level course on geometry and topology (a descendant of the legendary course "Geometry and the Imagination" at Princeton by Thurston, Doyle, Conway, and Gilman), covering a subset of the topics we offered in this class. The course put emphasis on the *process* of thinking about mathematics, not in acquiring as much knowledge as possible. The students get to play with materials like Legos, mirrors, scissors, papers, and keeping a research journal while frequently discussing with other students.
- *MATH 74 Algebraic Topology*: This is a more rigorous treatment of the mathematical theory behind computational topology. We will focus more on the applications in computer science and not always give full proofs to the results. For students who took MATH 74 before or are familiar with the language of topology will benefit by seeing how the concepts are applied in computer science.

Textbook

There is no *required* textbook. We will use a mix of books and class notes (and sometimes papers) as references; none of them covers the exact set of topics we will talk about. The relevant reading materials will be emphasized on the course webpage each week for the interested students to read in their off-time.

Teaching Methods and Expectations

The class will mostly be ***lecture-based***. I firmly believe that the real learning only happens when the students are actually engaging the materials through a set of well-designed problems, and not through passive learning like sitting in lectures. Consequently, the students are not required to attend the lectures live. That being said, my hope is to light up the spark of interest within so people may raise questions and join the discussion; I will work hard to make sure the lectures are fun and inspirational to earn your attention. There is really no set topics that we have to go through; depending on the background, goal, and feedback from each student, we will adjust the topics covered.

There will be ***weekly assignments***. Each week we will assign just one or two questions, to make sure that the students are following the material. These problems are designed specifically at the right difficulty so that it is challenging (and even sometimes frustrating), but rewarding and aiming to expand our knowledge and skills after solving them (maybe with some help from your colleagues). Therefore it is okay if you find the problems hard; that is by design. The students are expected to spend a reasonable amount of time working on the assignments. In a sense, these are the real driver of the class. Discussions with other students and using online resources are not only *allowed*, but *encouraged*. Some homework problems might require additional readings; all the extra readings required for the homework will be made available.

In addition to the assignments, the students may opt in for an optional ***research project***. Students may choose a topic related to computational topology (broadly interpreted) of their interest, approved by the instructor; a list of possible topics will be provided. Choosing to work on research project replaces the later part of the homework sets on the more advanced topics. Students who share the same interest are encouraged to work together. At the end of the term, student groups who chose to work on the project has to give an *oral presentation* in class, and provide a *final report* summarizing the work, including (but not restricted to) a summary of the papers read, identifying research questions related to their own field, and the progress made. Students who are interested in implementation are also encouraged to demo the result.

Outside regular lecture hours the instructor will provide ***office hour*** on a weekly basis; all students are encouraged to attend this discussion-based session. There will be no ***exams***.

Grading

The final grade will be based on the homework grades and the optional *research project*. The tentative weights are homework (60%) and project (40%) for people who work on projects, and homework (100%) for people who work on homework sets alone. See the course webpage for more details on the rubrics, collaboration rules, submission policies, etc.

About the grading rubric: Because we have a small-size class, I would prefer to talk to each of you and set up concrete goals after reviewing your submission to homework 0.

Tentative Course Schedule

All materials are subject to change based on the actual class interaction and student feedback.

- Week 1: ***Curves***
  - logistics, introduction
  - planar curves, Jordan curve/polygon theorem, Monge property and applications
  - winding number, representations, homotopy, invariance
  - regular homotopy, rotation number, Whitney-Graustein Theorem; Gauss signing, smoothings, Seifert decomposition
- Week 2: ***Surfaces and Complexes***
  - surfaces, triangulations, polygonal schemata, homeomorphism, classification of surfaces
  - surface graphs, rotation system, dual graphs, tree-cotree decomposition and system of loops, Euler's formula
  - complexes, Čech complex, Vietoris-Rips complex, surface reconstruction through sampling
  - mesh generation, Delaunay triangulation, nerve theorem, alpha shape and alpha complex
- Week 3: ***Homotopy***
  - testing homotopy, crossing sequence, shortest homotopic path, funnel algorithm
  - covering spaces, fundamental groups, induced homomorphisms, Brouwer fixed-point theorem, homotopy equivalence
  - fundamental groups of surfaces, group presentations, configuration spaces and motion planing
  - tightening curves on surfaces, monotonicity
- Week 4: ***Homology***
  - chain complex, boundary maps, cycles and boundaries, simplicial homology
  - singular homology, homotopy invariance, winding number/Euler characteristics redux
  - Brouwer's fixed point theorem, Sperner's lemma
  - Borsuk-Ulam theorem, ham sandwich theorem
- Week 5: ***Linkage and Folding***
  - linkage reachability: Kempe universality, arm lemma, universality of planar linkages
  - rigidity and locking: flipping rigid polygon, carpenter's rule
  - folding and origami: Origamizer, map folding, straight-cut and disk packing
  - polyhedra unfolding: Cauchy theorem, Alexandrov's theorem
- Week 6: ***Meshes and Conformal Geometry***
  - Delaunay triangulations, edge flips, moving curvatures around
  - curvatures and Gauss-Bonnet theorem
  - uniformization theorem, hyperbolic geometry, circle packing theorem
  - discrete conformal geometry, intrinsic triangulations, triangulation refinement
- Week 7: ***Optimization***
  - minimum (s,t)-cut in planar graphs, minimum homotopic cycle, Reif's algorithm, speedup
  - multiple-source shortest paths: disk-tree lemma, parametric shortest paths, red-blue lemma
  - r-division: cycle separators, fundamental-cycle separator, BFS face levels, alternation
  - Monge property, distance emulator, fast approximate min-cut in planar graphs
- Week 8 and 9: ***Topological Data Analysis / Shape Analysis***
  - Persistent homology, barcodes

  - barcode computation, persistence diagram, stability, sketching persistence diagram
  - Reeb graphs, critical points, Morse functions, flow lines, Morse homology and inequalities
  - loop lemma, algorithm to compute Reeb graph
  - discrete Morse theory: discrete gradient field, simplicial collapse, discrete flow line
  - stable and unstable manifolds, graph reconstruction from 1-unstable manifold
- Week 10: ***Final project presentation***