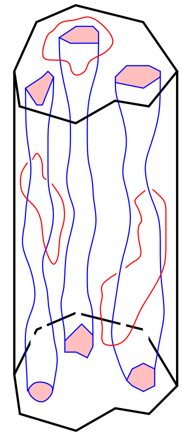
Math 529: Computational Topology (Spring 2026)

# Credits	3
Time	Tue+Thu 10:35–11:50 am
Location	WEBS B8 (Pullman), VECS 120 (Vancouver), Zoom
Instructor	Bala Krishnamoorthy
Office	VLIB 210P, Zoom
Check-in Hours	Tue, Wed 3–4 PM
Email	kbala@wsu.edu
Web page	bala-krishnamoorthy.github.io/Math529.html
Text	**Class notes and handouts**
References	Edelsbrunner and Harer: Computational Topology ISBN:0821-84925-5 (Preprint)

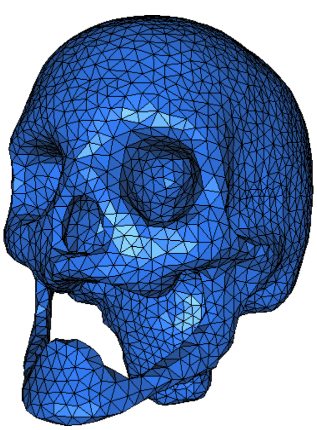


Required and Recommended Reading Materials

Required:	Class notes and handouts (posted on course web page)
Recommended:	Relevant sections from *Computational Topology: An Introduction* by Edelsbrunner and Harer.
Additional Materials/fees:	Not applicable.

Description of the Course

Topology studies how a shape or object is connected. In the past several years, there has been an increased interest in the development and use of topological methods for solving various problems in science and engineering. This new line of study is called **Computational Topology** or **Applied Algebraic Topology**. Computational topology combines topological results with efficient algorithms to analyze data and solve problems in *many* fields including biomedicine, phenomics, machine learning, computer graphics and image analysis, sensor networks, robotics, geography, and several others.



Prerequisites: This course will present an **introductory, self-contained overview** of computational topology. There are no prerequisites, but **mathematical sophistication at the senior undergraduate level** and some **familiarity with the use of computer packages** (e.g., Python, Matlab, etc.) are expected. We will cover basic concepts from a number of areas of mathematics, such as abstract algebra, algebraic topology, and optimization. We will also look at algorithms and data structures, and efficient software for analyzing the topology of point sets and shapes—termed *topological data analysis*, or TDA. The grade will be based on a several homework assignments and a project, which will involve either an implementation of a method on your own, or recreation of results from recent research paper(s) using existing software tools.

Individuals with backgrounds in mathematics, engineering, or life sciences, all with some computational background, will find this class of interest.

Expected Student Learning Outcomes (SLOs)

Quantitative skills: After completing this course, the student should be able to employ standard TDA computational tools (i.e., software packages) along with basic coding skills to perform persistent homology and mapper structural analysis of large scale data sets. They should also gain a working knowledge of basic concepts from computational algebraic topology—see the Tentative Schedule for a list of relevant concepts covered. They should demonstrate this knowledge through responses to problems—both proof-type as well as computational ones involving the use of software packages or coding—in the homework assignments as well as in the project.

Written communication skills: The student should be able to explain and communicate basic concepts of computational topology clearly in the homework assignments. They should be able to elucidate logical mathematical arguments as part of the proof-based exercises. They should also be able to write a cohesive yet brief (5–7 pages long) report for the project in the form of a short research paper.

Assessment: Required Assignments and Grading

Homework assignments: The course will have **six** homework assignments, which will include mathematical problems (including a few proof-based exercises) as well as ones involving use of software packages. The Tentative Schedule specifies the due dates for each homework assignment. The student will be scored based on their level of command over basic concepts demonstrated in proof-type problems as well as their command over the use of computational tools. These assignments will evaluate the student's **quantitative skills** and **written communication skills**. Grades for the homework assignments will be assigned using the following guidelines. *A*: logic is consistent, computing steps are accurate, work is explained in a clear manner, only minor mistakes such as typos; *B*: some logical inconsistency is present, or explanations are not clear and well justified, but underlying understanding of material as well as basic command over computing tools is demonstrated; *C*: a lack of understanding of some underlying conceptual material and/or difficulty in using computing tools is evident; and *D* or *F*: a lack of understanding of much of the underlying concepts and/or inability to use basic computing tools and/or a lack of meeting submission deadlines.

Project: Each student will also be required to work on a course project, which will involve either the implementation and testing of a particular computational topology method on a real life data set, or summarizing at a low level the results of 2–3 related papers. The student will be evaluated on their understanding of underlying concepts, command over use of computational tools (**quantitative skills**), as well as their ability to present their results concisely in a 5–7 page long project report (**written communication skills**). A sample project description is available here: https://bala-krishnamoorthy.github.io/FilesMath529/S24/Homeworks/Project.pdf.

Late submissions will receive zero credit. The student should contact the instructor well before the deadline to request an extension if valid and unavoidable circumstances arise.

Grading Policy: The *total score* for the course will be computed using the following **weights:** homework: 65%, project: 35%. The total score will be *rounded up* at the first decimal place, e.g., $87.42 \rightarrow 87.5$, and $79.91 \rightarrow 80$. The **overall grade for the course** will be determined by the total score, based on the following scale: 93–100: A, 90–92.9: A−, 87–89.9: B+, 83–86.9: B, 80–82.9: B−, 77–79.9: C+, 73–76.9: C, 70–72.9: C−, 67–69.9: D+, 60–66.9: D, 0–59.9: F.

Attendance Policy: While student attendance will not be tracked on a regular basis, attending each lecture is critical to understand all concepts, and performing well in the course. The student is encouraged not to miss any lectures.

Software

We will introduce and use several packages for computational topology (e.g., Scikit-TDA, Giotto-TDA, GUDHI, Ripser, Kepler-Mapper, etc.). Python interfaces are available for many of them, while some of them come with fairly independent standalone implementations (i.e., one would not have to do much independent programming). The student will be expected to do a limited amount of basic scripting and/or programming (in Matlab, Python, C/C++, or another language/package).

Expectation for Student Effort

Course Expectation: Academic credit is a measure of the time commitment required of a typical student in a specific course. For the WSU semester system, one semester credit is assigned for a minimum of 45 hours. The anticipated time commitment for this course is 3 hours of work per week for each credit hour (a minimum of 9 hours per week). **For each hour of lecture equivalent, students should expect to have a minimum of two hours of work outside class.** A tentative weekly time commitment for the student in this class is listed below:

class time (lectures)	2.5 hours/week
reading class materials	1.5 hours/week
completion of assignments (homework, project)	5.0 hours/week
Total	9.0 hours/week

Academic Integrity: Discussion of homework problems with others is allowed, and is also encouraged. But each person should hand their own written solutions. **Students are expected to cite any external sources used in building their work for homework assignments and the project, including books, papers, web pages, online discussion forums, other individuals, or large language models (LLMs) such as ChatGPT, Gemini, Claude, etc.** Plagiarism or cheating will not be tolerated. Such behavior will result in a zero grade for a graded item and possibly a failing grade for the entire course.

Students are responsible for reading and understanding all university-wide policies and resources pertaining to all courses (for instance: accommodations, care resources, policies on discrimination or harassment), which can be found in the University Syllabus.

Tentative Schedule and Topics Covered

Week	Lec #	Date	Details
1	1	Tue, Jan 13	topology and connectivity, motivating applications
	2	Thu, Jan 15	definitions from topology, homeomorphism
2	3	Tue, Jan 20	manifolds, orientability
	4	Thu, Jan 22	simplices and simplicial complexes
3	5	Tue, Jan 27	abstract simplicial complexes, geometric realization **[HW 1 Due]**
	6	Thu, Jan 29	Euler characteristic, classification of manifolds
4	7	Tue, Feb 3	star and link, retractions, nerves
	8	Thu, Feb 5	nerve theorem, Čech and Vietoris-Rips complexes **[HW 2 Due]**
5	9	Tue, Feb 10	Voronoi diagram, Delaunay complex, filtration, alpha complex
	10	Thu, Feb 12	witness complexes, max-min selection
6	11	Tue, Feb 17	chains, boundary, chain complex
	12	Thu, Feb 19	fundamental lemma of homology, homology groups **[HW 3 Due]**
7	13	Tue, Feb 24	betti numbers (β_is), examples of homology groups
	14	Thu, Feb 26	Euler-Poincaré theorem, boundary matrix
8	15	Tue, Mar 3	Smith Normal Form (SNF), algo for SNF, reduced homology
	16	Thu, Mar 5	relative homology, topological persistence
9	17	Tue, Mar 10	incremental algo for β_is, pairing for persistence
	18	Thu, Mar 12	UNION-FIND data structure, persistence algorithm **[HW 4 Due]**
		Tue, Mar 17	*Spring break*
		Thu, Mar 19	*Spring break*
10	19	Tue, Mar 24	persistence diagram, fundamental lemma of persistent homology
	20	Thu, Mar 26	matrix algorithm for persistence
11	21	Tue, Mar 31	Reeb graph, pullback cover, mapper algorithm
	22	Thu, Apr 2	details of mapper: computational details **[HW 5 Due]**
12	23	Tue, Apr 7	homology over $\mathbb{Z}$, optimal homology problems
	24	Thu, Apr 9	boundary matrix $[\partial]$ and total unimodularity (TU)
13	25	Tue, Apr 14	linear and integer programming and TU
	26	Thu, Apr 16	conditions for $[\partial]$ being TU **[HW 6 Due]**
14	27	Tue, Apr 21	geometric measure theory (GMT) applications
	28	Thu, Apr 23	currents, flat norm, and TU of $[\partial]$
15	29	Tue, Apr 28	area minimizing surfaces, optimal bounding chains
	30	Thu, Apr 30	simplicial median shapes
16		Thu, May 7	**Project due by 5:00 PM**