Homological Sensor Networks

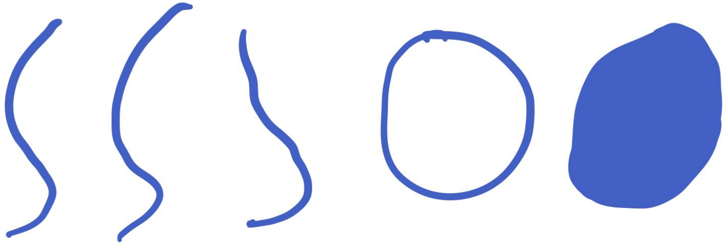
SIG Math Fall 2025

A brief note on topology

- Generally when you ask normies about topology, they generally have two intuitions:
    - Mobiüs strips
    - Lol mathematicians think a coffee cup and donut are the same lmao
- I think mobiüs strips are interesting and actually do provide some interesting stuff for analysis (they're an example of a quotient space).
- I hate the coffee cup and donut meme, it makes math feel stupid and it's a very bastardized version of what topology actually is. So I'd like to provide my own intuition for topology that I think more accurately gets at what is going on.

My intuition for topology

- Take these 5 shapes. Intuitively some feel more similar than others:



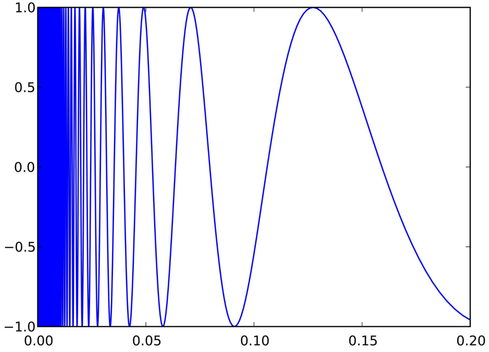
- The first two curves are almost identical. Obviously the pixels are slightly different, but you'd be forgiven for calling them equivalent.
- The third curve is slightly different, but still has the same vibe.
- The fourth shape is now a circle. In some sense it feels "different" from the first curves. There's also a "hole" in the middle that's not present in the last.
- The last shape is a filled circle. It feels very different, but in a different way.

My intuition for topology (cont.)

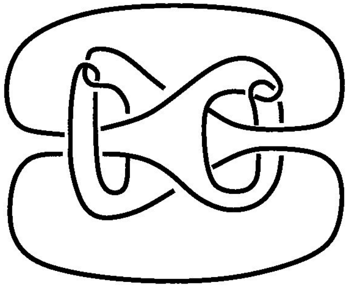
- Topology is **all about formalizing the geometric intuitions we have in everyday life**. Geometrically, all these shapes are completely different. Topologically, some of them are similar!
- The three curves on the left are all effectively the same (or *homotopy equivalent)*. You can deform one into another.
- The fourth curve is different from the first three: it's not simply connected, it has no boundary, it stays connected after removing a point.
- The fifth curve is even more different: it has higher dimension, has a boundary, stays connected after removing multiple points.
- Topology is all about making these intuitions rigorous!

Fields of topology

Point-set topology



Geometric topology



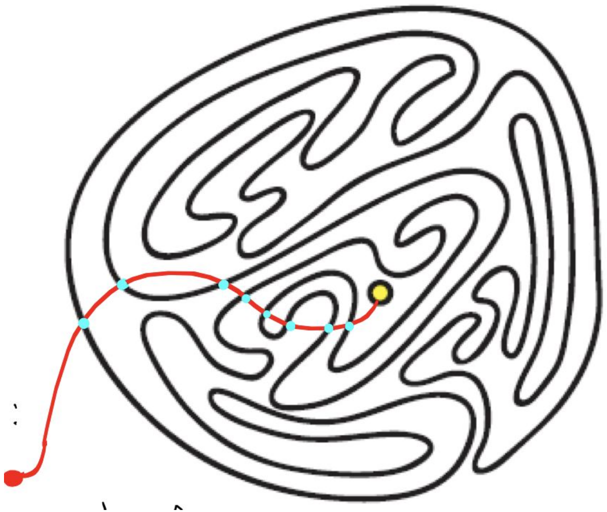
Differential topology

Algebraic topology

$$
\begin{tikzcd}
 & H_n(U_i, U_i - x_i) \arrow[r, "f_*"] \arrow[dl, "\approx"'] \arrow[d, "k_i"] & H_n(V, V - y) \arrow[d, "\approx"] \\
H_n(S^n, S^n - x_i) & H_n(S^n, S^n - f^{-1}(y)) \arrow[l, "p_i"'] \arrow[r, "f_*"] & H_n(S^n, S^n - y) \\
 & H_n(S^n) \arrow[r, "f_*"] \arrow[ul, "\approx"] \arrow[u, "j"] & H_n(S^n) \arrow[u, "\approx"']
\end{tikzcd}
$$

Inside and outside

- The core idea from algebraic topology that we will be talking about is **homology**. However, homology is a difficult topic, so we will build up to it using increasingly complex examples.
- Consider a simple closed curve in the plane and a point. How can we tell if the point is inside or outside the curve?
- One simple method is to draw a curve from the point to a point "far" from the curve (infinity if you like). Then count the crossings: if there's an even # it's outside, an odd # means inside.



Inside and outside (cont.)

- But what happens if the curve overlaps at a tangency?

- A better way is to assign an index to intersections between the loop and the curve, such that the **sum is invariant under deformations**. That is, if we shift the curve around the result stays the same.
- One way is to assign each intersection a value in Z mod 2, 1 if it's a crossover and 0 if it's a tangency. Then you sum the values to determine if the point is inside or outside the curve.
- Note that this method is **coordinate-free** and a **local computation.**

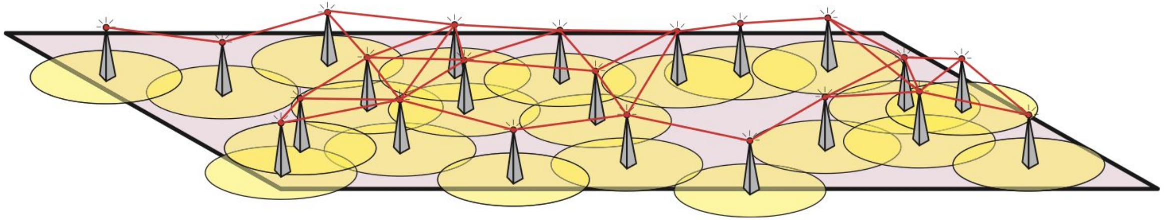
Our scenario

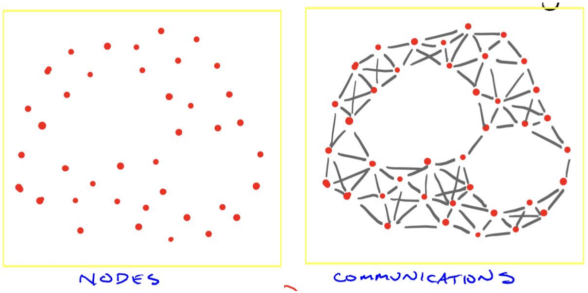
Sensor Networks

- There are tons of papers on networks – communications, sensors, WiFi, cameras, social, etc. For this presentation, we will **focus on networks that have an underlying geometry** (each node has an actual location), which is **strongly connected to communication between nodes**.
  - In short: nearby nodes can communicate, far away nodes cannot.
- In many situations, sensor nodes have known locations, either by fiat (video cameras are in a fixed location) or by GPS. In those situations, a geometric approach is fine and there are many efficient algorithms.
- However, this isn't always the case! Consider underwater, or inside a building, or in outer space. We will be looking at these **non-localized networks**.

Our situation

- We'll assume each sensor knows its identity and can broadcast it to nearby nodes.
- The boundary of the domain D is defined by special *fence nodes*.
- The result is we can make a **communications graph**: vertices are nodes, edges are communication links.





(it's clear there are holes...)

Coverage

- Coverage problems are ubiquitous when studying sensor networks. There are many types of coverage problems:
    - Does the sensor network cover the entire domain of interest?
    - Does the sensor network form a barrier – a separation between two areas of the domain?
    - If the nodes of the network move in time, do they cover all of the domain eventually?
    - What is the smallest number of nodes needed to cover the domain?
- We don’t need to specify what the sensors actually are – they could be radar, acoustic, optical, etc. Or they could be WiFi hotspots and you want to cover an entire building. Or they are lights/beacons and you want everything to be illuminated for robot navigation, etc…

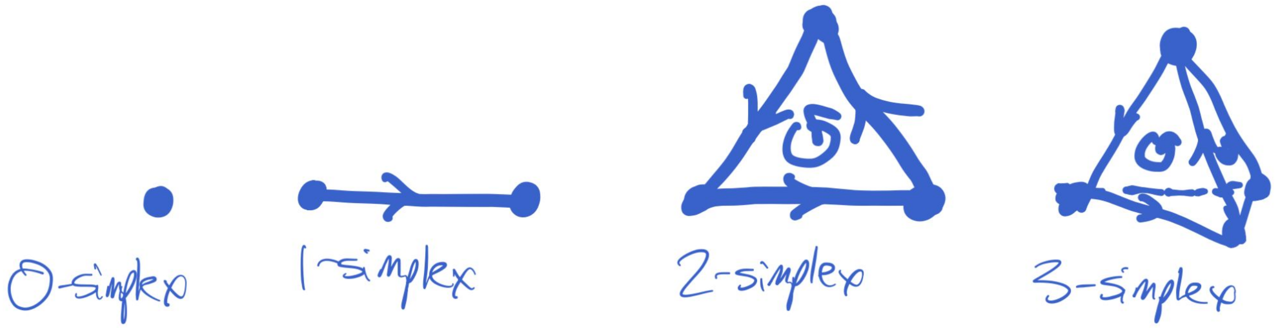
The main question we are trying to answer is: given an object, does it have holes?

But we first need to define what that object is.

Simplicial Complexes

Simplices

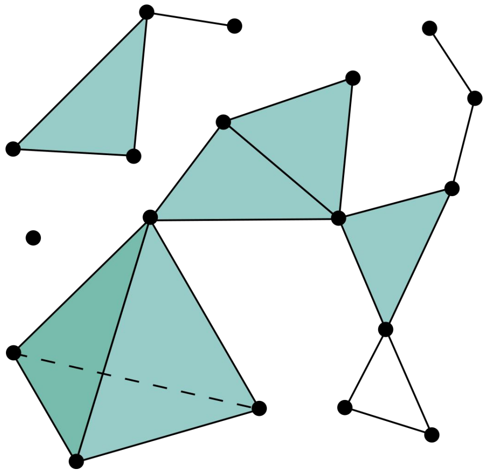
- The word simplices (singular simplex) sounds scary but it’s actually simple. It’s the generalization of a triangle or tetrahedron to arbitrary dimensions.



- These simplices also have a orientation. For a 1-simplex, the orientation is just the direction. For a 2-simplex, the orientation is clockwise or counterclockwise, and same for a 3-simplex and higher.

Simplicial complexes

- We can also add together simplices to make **simplicial complexes**.

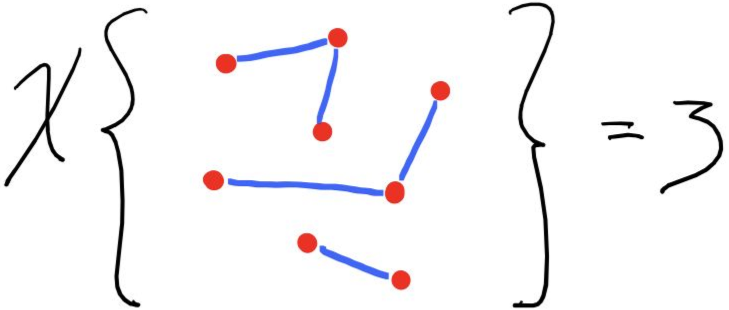


Euler characteristic

- One natural question when given a simplicial complex (or any object in general) is to ask how many components it has.
- We'll call this count the Euler characteristic, denoted as X.
- The Euler characteristic of a collection of vertices (0-simplices) is just the number of vertices:

Euler characteristic (cont.)

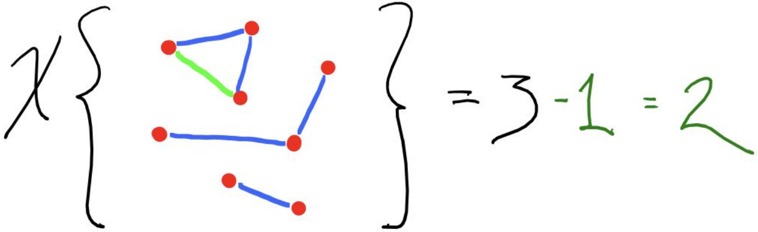
- What happens when we add edges? The Euler Characteristic decreases, as each edge connects a vertex to another vertex:



- We’ve added 5 edges (or 1-simplices), so there are only 3 connected components left.
- So X gets +1 from vertices and -1 from edges!

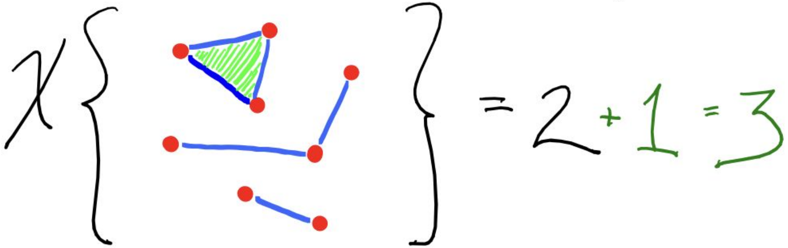
Euler characteristic (cont.)

- Adding another edge creates difficulties, as our current weighting scheme no longer has the correct number of components:



- Note that there's a "hole" in the graph, and it almost seems like the Euler characteristic (X) takes this into consideration.

- Let's fill in the hole with a "face" (a 2-simplex) and adjust X accordingly:



- We can proceed inductively from here to get a general formula for a n-simplical-complex.
- But up to dimension 2, our formula for X is V - E + F (this may look familiar…)

What just happened?

- It's easy to overlook, but what we just did is actually an instance of some pretty deep mathematics!
- This theorem, and other similar theorems that prove invariant properties of a space, all rely on **homology theory**. In general, homology has three ingredients:
  - Counting – there are some relevant objects in the space that we can count
  - Grading – the objects possess some relevant "size" or "dimension" that is useful for...
  - Cancelling – by using algebra, we can cancel out pairs of objects.
- Done properly, these ingredients form an algebraic device which is a global invariant.

Triangulations

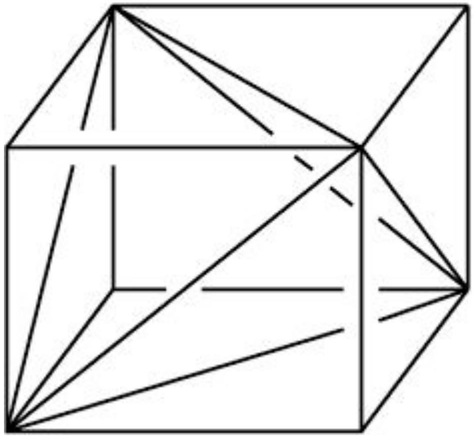
- The Euler characteristic was originally used to prove facts about the platonic solids. Note that the tetrahedron and icosahedron are already simplicial complexes, so the Euler characteristic works easily:

Name	Image	Vertices V	Edges E	Faces F	Euler characteristic: $\chi = V - E + F$
Tetrahedron		4	6	4	**2**
Icosahedron		12	30	20	**2**

- Applying the Euler characteristic to a cube is not so easy, since the cube's faces are squares and not triangles. However, there are ways to deal with this:

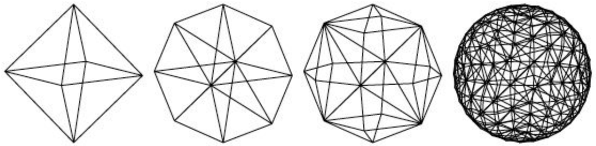
Euler characteristic for non-triangular shapes

- Although the cube looks like it's made of squares, we can **triangulate** the cube so it's made of triangular faces.
- Each square face is now made of two triangular faces next to each other.
- Even though this complex is in some sense different, it retains the same topological properties as the cube.
- Specifically, we can calculate its Euler characteristic using our formula (and it yields 2 just like the other platonic solids)



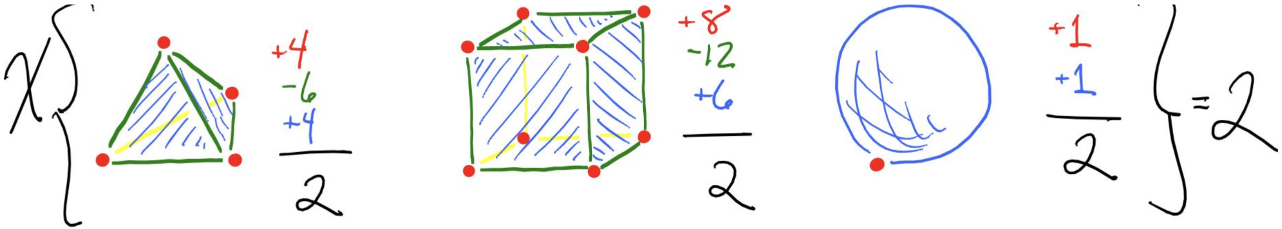
Euler characteristic for spheres

- It’s easy to see how to triangulate a cube since it’s relatively polygonal, but what about a sphere?
- The solution is to make up our own triangulation that is “relatively close” to the sphere.
  - When you do this rigorously, you’d be required to prove that the triangulation you made is topologically equivalent to the sphere, usually through geometric means.
- From this triangulation, we can calculate the Euler characteristic and other topological properties of the sphere.



Whitehead triangulation theorem

- You might have noticed that we gave many different ways to triangulate the sphere. But which one is the correct one?
- For a “triangulable” space, X is independent of the simplicial structure imposed. **Essentially, the specific arrangement of the triangles doesn’t matter, the value of X will stay the same.**
    - This is a very nontrivial result! There’s no reason the different triangulations should be topologically equivalent, but they are.



So to recap, the question we are trying to answer is: given a triangulated object, does it have holes?

Introducing homology

Intuition on holes

- We've talked a bit about holes, but what do we mean *precisely*?
  - We won't answer this yet, but we will lay the groundwork for making it precise.
- Let's say I draw a loop on the plane. It's easy to see that the loop is bounding a 2D region in the plane.

- Now let's fold up our plane into a torus. I can draw this loop so that it's bounding some space "between" the torus, but it's not bounding space that's physically on the torus.
- This means that there's a hole!

- So a hole is a **loop that does not bound any region in the space.**

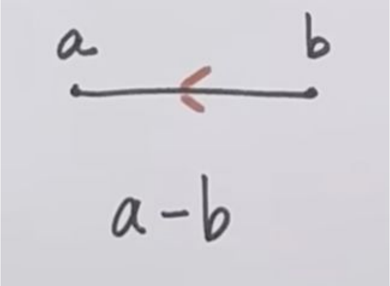
Making holes rigorous

- Of course, “loop that does not bound any region in the space” is not very rigorous. What is a loop? Or a boundary? Or a region?
- We’ve actually already answered one of these questions. Remember we can triangulate any object to turn it into a simplicial complex, so a region is just a simplicial complex!

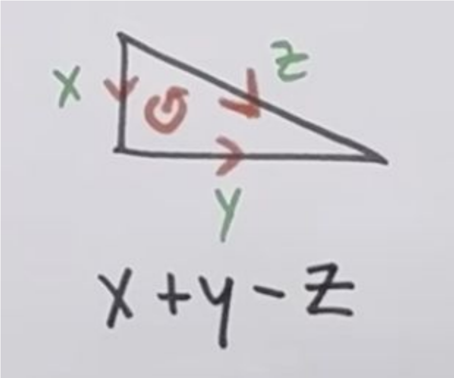
Boundaries

- Let's look at some examples of boundaries. The boundary (denoted del) of a 0-simplex is 0:

- The boundary of a 1-simplex is just the end point minus the start point.



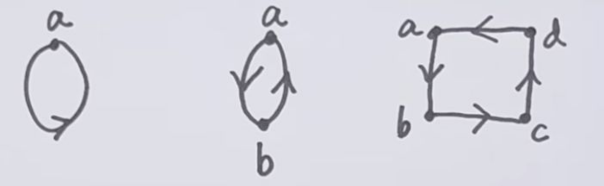
- For a 2-simplex, we travel along the direction of the orientation, and add up all the 1-simplex components.



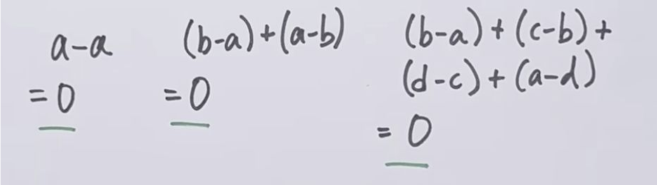
- It's very simple. But we can already prove an interesting theorem! The boundary of a boundary is always 0.

What is a loop?

- Here are some examples of loops:



- Let’s calculate their boundaries:



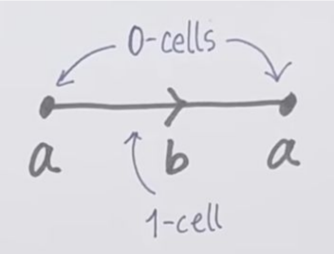
- So a loop is just a **collection of 1-simplices with boundary 0.**
  - This also elucidates upon the theorem from the last slide. Since the boundary of a boundary is zero, the boundary of anything is a loop!

Hole definition

We've now defined everything we need to define what a hole is. A hole is a **collection of 1-simplexes with boundary 0 that is not the boundary of a collection of 2-simplexes.**

Example: the circle

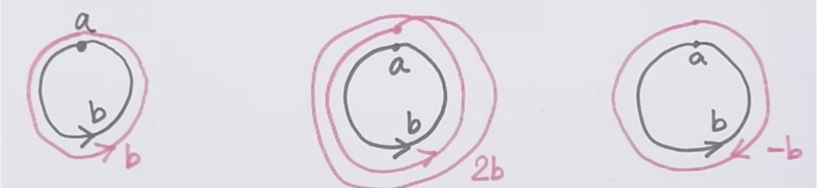
- This definition is very unintuitive and it's helpful to go through examples.
- Let's first make the simplicial complex for a circle:



- Obviously, there are no 2-simplices. So any collection of 1-simplices with boundary 0 implies a hole!
- Let's look at the 1-simplex b. What is its boundary? By our definition, it's the end point minus the start point, which is a - a or 0. So b is a loop, and therefore a hole!

Example: the circle (cont.)

- But let's be careful. What about the path 2b? Its boundary is also a - a = 0. So it's also a loop. We can do the same for -b, which is b in reverse.



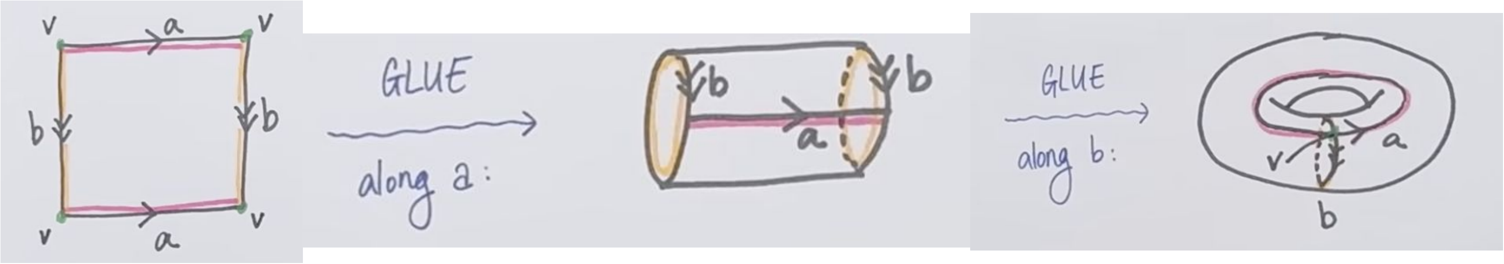
- So for the circle, the collection of 1-simplices with boundary 0 that are not the boundary of a collection of 2-simplices is the set of n * b, where n is an integer.
- This is a mouthful, so topologists call the holes the **first homology group**, or $H_1(X)$. Note that because $H_1(X)$ is an integer times b, it's effectively determined by 1 integer and we say that $H_1(X) = Z$.

Disclaimer on homology groups

- When we talk about homology groups, we are specifically referring to a mathematical structure known as a group.
- Group theory is a very deep and interesting field of mathematics. Algebraic topology specifically examines the intersection between it and topology.
- For the purposes of this presentation, we will elide the details of how groups operate.

Example: the torus

- Let's first construct the simplicial complex of a torus. This is the prototypical construction:



- We fold the red arrows onto each other, and then the yellow arrows. Note the four vertices go to 1 vertex. So it's really just one 0-simplex, two 1-simplices, and one 2-simplex.
  - If you're careful, you'll note there should actually be an extra 1-simplex and 2-simplex! It's necessary for the rigorous calculation, but for this presentation we'll skip it since the calculation is very convoluted.

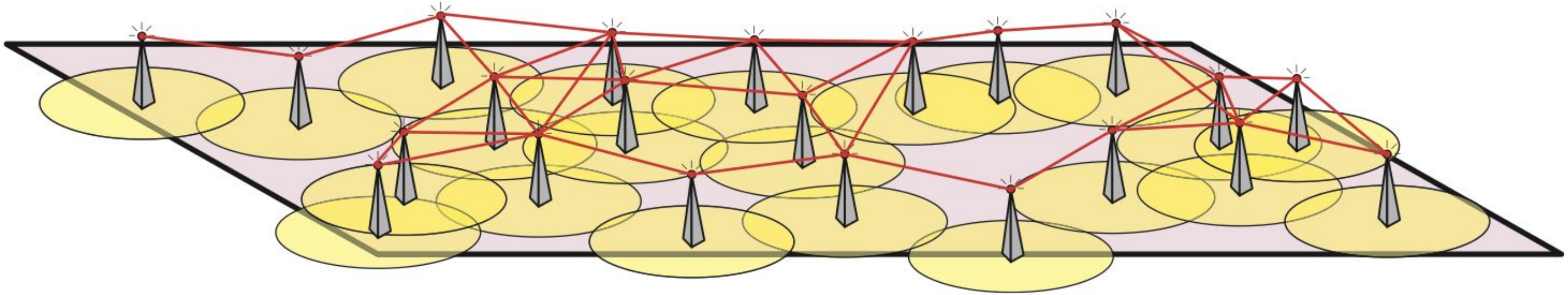
Example: the torus (cont.)

- To find the first homology group, we need to find the collection of 1-simplices that are not the boundary of a 2-simplex.
- There is only one 2-simplex in the figure. What is its boundary? We can find it by adding up all the sides, keeping orientation into account. Adding up the edges, we find the boundary is a + b - a - b = 0.
- There are two 1-simplices in the figure. Each one counts as Z, so the **first homology group is Z * Z**. Alternatively, there are **two 1-D holes**.
- What is the second homology group? There is one 2-simplex and no 3-simplices, so the **second homology group is Z**. Alternatively, there is **one 2-D hole**.
    - You can think of a 2-D hole as the inside of the torus being empty (like a basketball)

The Homological Criterion for Coverage

Scenario recap

- We've spent the past several slides discussing homological concepts that are not quite related to the situation at hand, so let's quickly recap:
- We have a set of nodes X. Each node can talk to other nodes in a radius R. The set of nodes is also bounded by a set of fence nodes F.
- Our question is: is the entire domain in the region bounded by F covered by the sensor network?

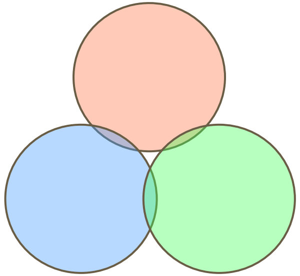


Creating a simplicial complex

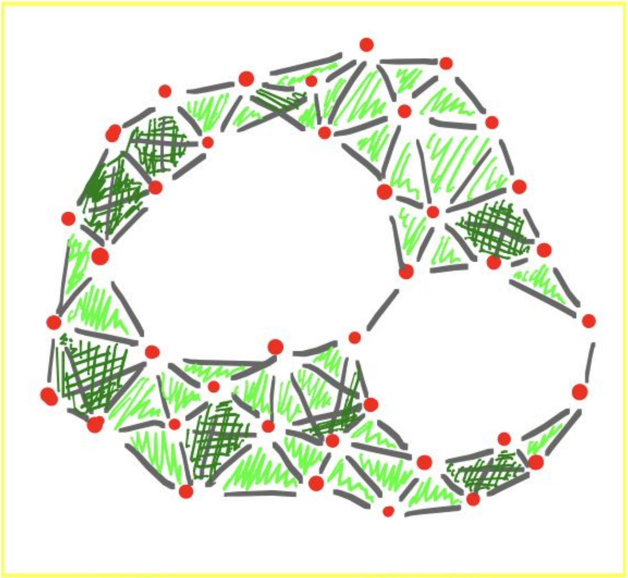
- The primary objects we've been working with in this presentation are simplicial complexes. So, it's natural to try to turn our network into a simplicial complex!
- There are many ways to do this, some being harder than others. We'd like a way that's easily computable while maintaining topological data!
- Our construction is called the flag complex:
    - Draw an edge (1-simplex) between two nodes if they can talk to each other
    - If three nodes can all talk to each other, draw a 2-simplex inside them
    - If four nodes can all talk to each other, draw a 3-simplex inside them, etc.

Creating a simplicial complex (cont.)

- This construction isn't perfect and can be slightly inaccurate in certain situations.



- Despite its inaccuracies, it's clear that the homology groups of the flag complex relate to how well the sensor network covers the domain.



FLAG COMPLEX

The basic homological criterion for coverage

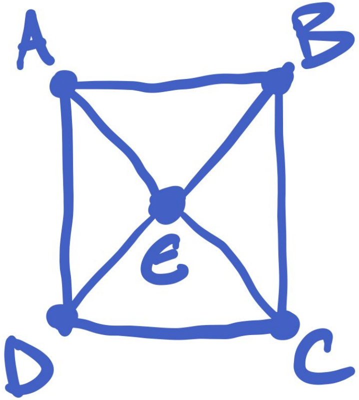
- A lot went on in the past several slides, but it's worth reflecting to see how far we've come:
    - We can turn a set of sensors into a simplicial complex, which is easy to work with
    - From our simplicial complex, we can calculate the first homology group, which detects the presence of 1-d holes
    - If the first homology group is nonzero, then there must be a hole in our simplicial complex, so the sensors do not cover the space!
- This gives us a homological criterion to detect if the space is covered by our sensor network. **Given a space X, if $H_1(X) = 0$, the space is fully covered by the sensor network!**

The actual homological criterion for coverage

- The basic criterion is not perfect. The actual homological criterion is: **the cover U contains D if there exists [α] ∈ H2(R, F) such that ∂α != 0.**
- The math for this is quite complicated and uses a concept called relative homology. But it's possible to understand this intuitively!
- The criterion is effectively asking: **given a collection of 2-simplices α, is their boundary equal to the fence?**
    - If there was a hole in the domain, the boundary of a would include the boundary of the hole! Thus it wouldn't be equal to the fence.
- H1(X) tells us whether or not there's a hole and nothing else. By computing H2(X, F), we get specific information about the cover.

Specific example

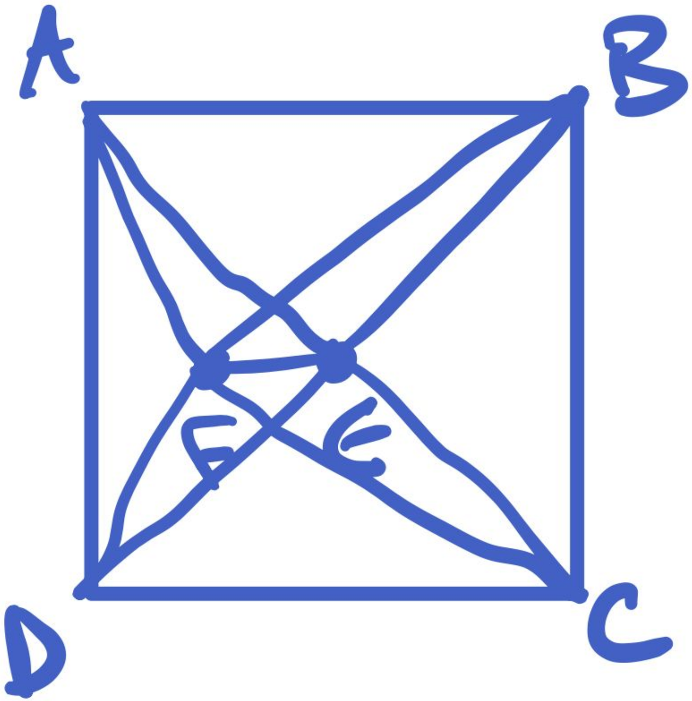
- This graph has 5 sensors, with sensors that can communicate given an edge.
- From there, we can fill in 2-simplices, but there are no 3-simplices (no instance of 4 nodes all being able to communicate)



- The fence cycle is the sum of 1-simplices AB + BC + CD + DA.
- Let's form a collection of 2-simplices based on the four triangles, so alpha = ABE + BCE + CDE + DAE.
- Now let's take the boundary of alpha. Note that all the "interior edges" that touch E cancel out, so we are left with the outer loop:
  da = AB + BC + CD + DA
- This boundary is precisely equal to the fence cycle! So this is a valid cover of the space.

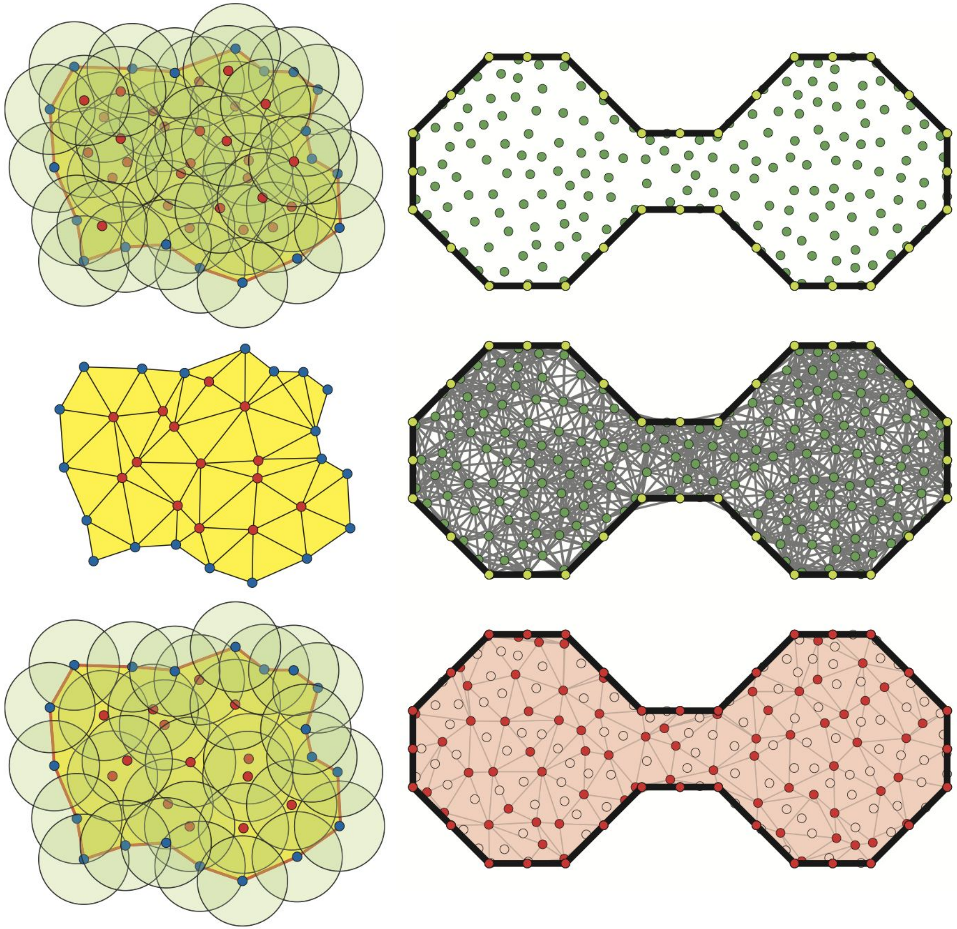
Specific example (cont.)

- Let's add in another point F, next to E.
  - Now there are a ton of 2-simplices and even some 3-simplices!
- There are actually multiple collections of 2-simplices that satisfy da = f.
  - Using E: a_e = ABE + BCE + CDE + DAE
  - Using F: a_f = ABF + BCF + CDF + DAF
- Both of these chains have boundary f. But notice that the boundary of ae - af = 0. So both fillings are actually equivalent.
- This also implies that e and f are redundant, which makes sense because our graph was still covered without f!



Power conservation

- The basic criterion works fine for detecting coverage, but the primary benefit of the actual criterion is to save power.
- If there are redundant sensors that don't actually need to be on, you can turn them off!
- **Homology theory gives us computational ways** to find the **minimal set of nodes required to cover a space**.

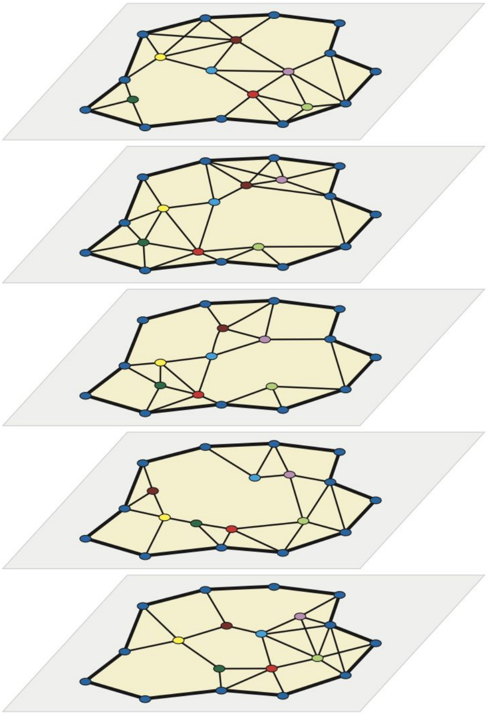


So why the homological approach?

- We've found a method to determine if our sensors cover the region *even if we don't know where the sensors are*. This approach being purely topological means **we don't have to worry about knowing specific coordinates or geometry.**
- This is also a way to *prove* that the domain has no holes. The typical method for ensuring robustness is sampling, which only shows the region has a high likelihood of having no holes.
- Computing it also tells us the minimal set of nodes to cover the space.
- This approach **can be extended to harder problems** like tracking holes over time, or finding minimal covers, which are highly non trivial problems when approached geometrically.

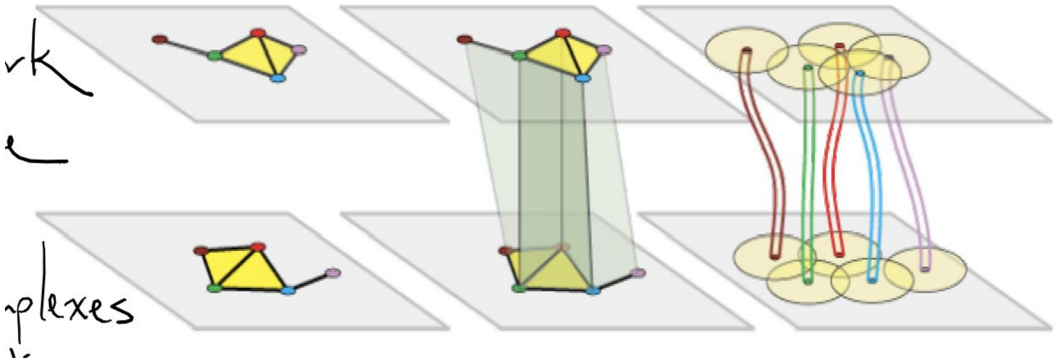
Pursuit and evasion

- Consider when nodes change position as a function of time. For example, when detecting a forest fire, one might set up a ring of fence nodes outside the forest and equip animals with sensors (that may move!)
- It might be the case that we don't have enough sensors to cover the entire domain at once. However, as the sensors change location, holes may open and close in a complex fashion.
- The evasion problem in this scenario is **whether an unknown evader could navigate through holes in the sensor cover without being detected**.



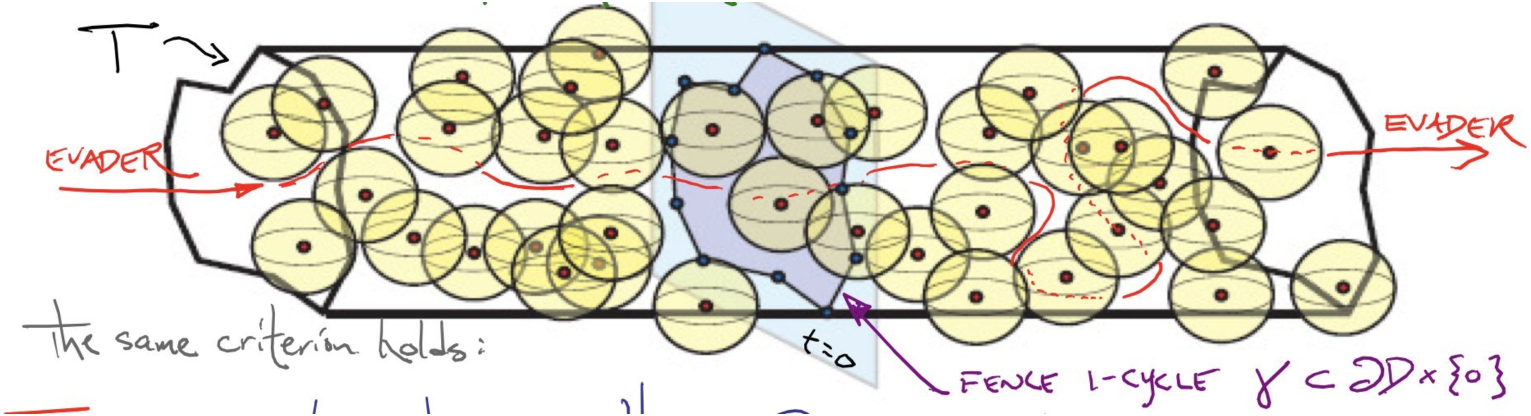
Pursuit and evasion (cont.)

- The idea is to sample the network at a series of small time steps, then create a 3D amalgamated complex.
- The homological criterion is actually the exact same!
- This is a problem that's very difficult geometrically, but when analyzed through homology becomes much more solvable.



Blocking

- Another question we might ask is, instead of asking if the sensors cover the space, do they determine a barrier such that a bad guy cannot cross the space without being detected by a sensor?
- The theorem is very similar to the homological criterion for coverage with some small modifications.



Conclusion

- This was a very long presentation and it's a feat in itself to sit through it!
- I don't expect anyone to remember all the details about homology or simplicial complexes or boundaries. But there are a few key conceptual ideas that I would like to reinforce:
    - When dealing with problems where you convert local information into global information, topology can be a key tool
    - Geometry is important but it's not everything. Sometimes you can find non-geometric approaches to seemingly geometric problems.
    - Finding invariants can be very helpful in making life easier for yourself.
- If you have any further questions about algebraic topology, feel free to ask me! I will also post extra resources about this topic if you are interested.

Thank you for listening!