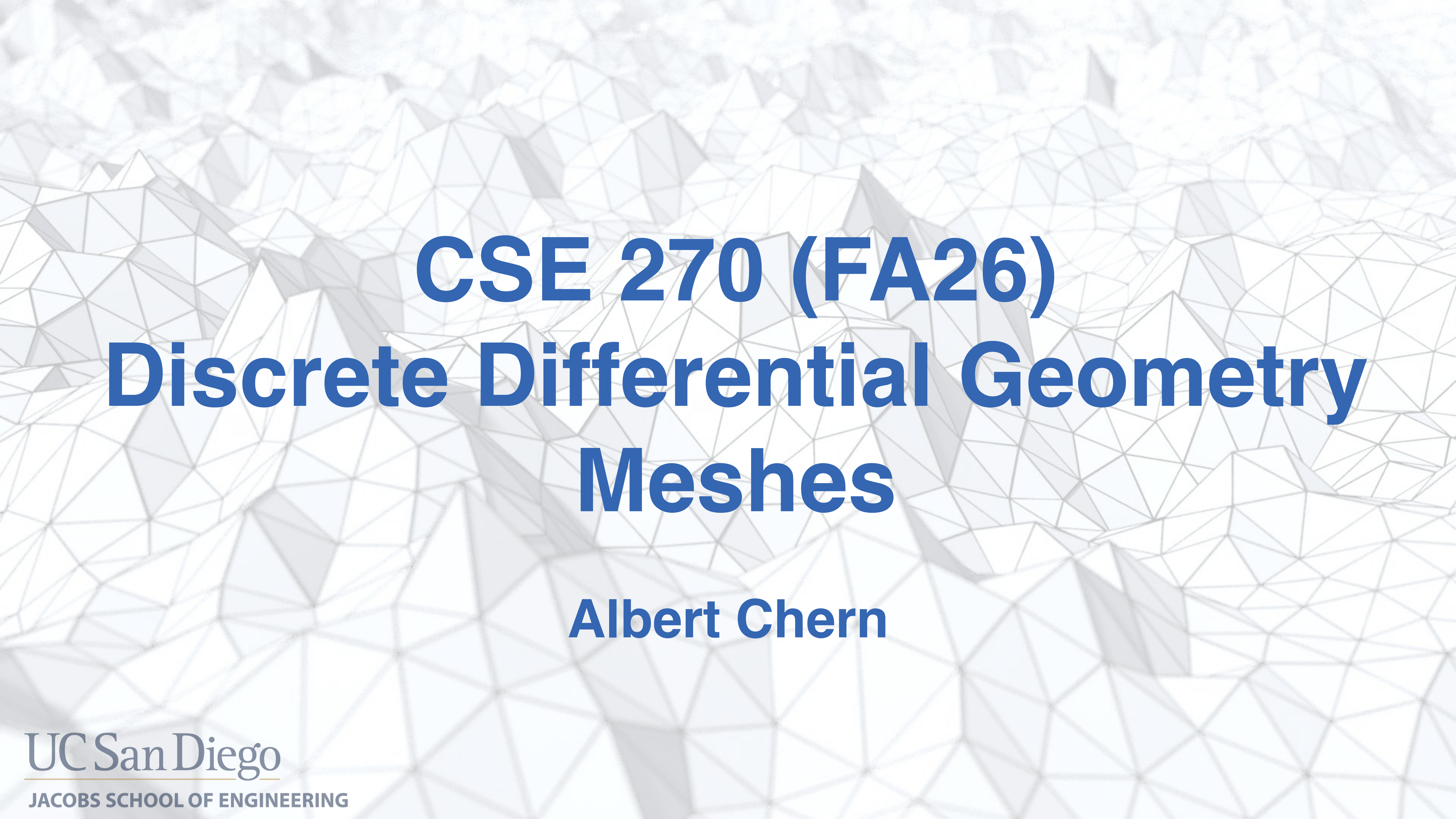




# **CSE 270 (FA26)**

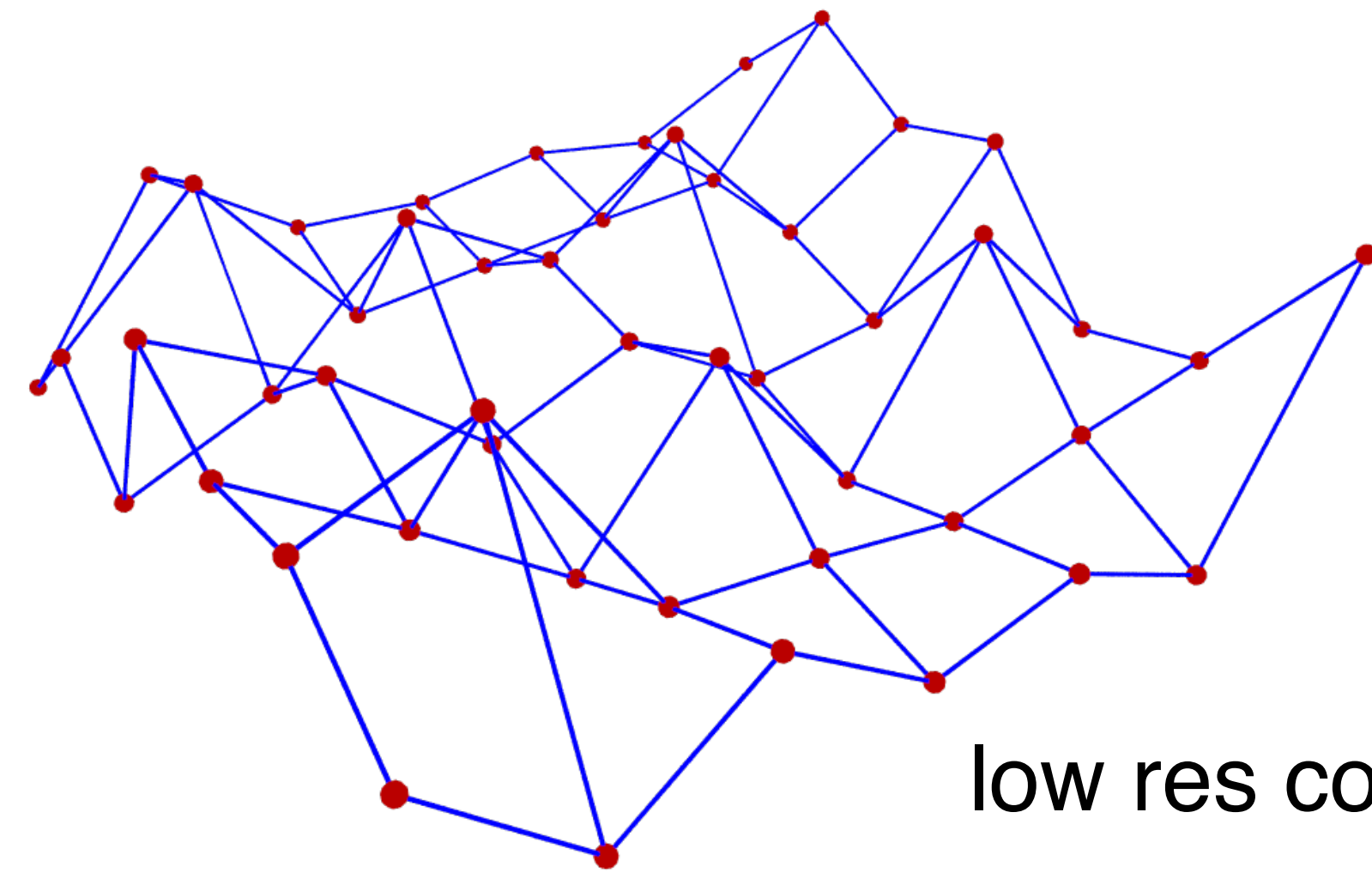
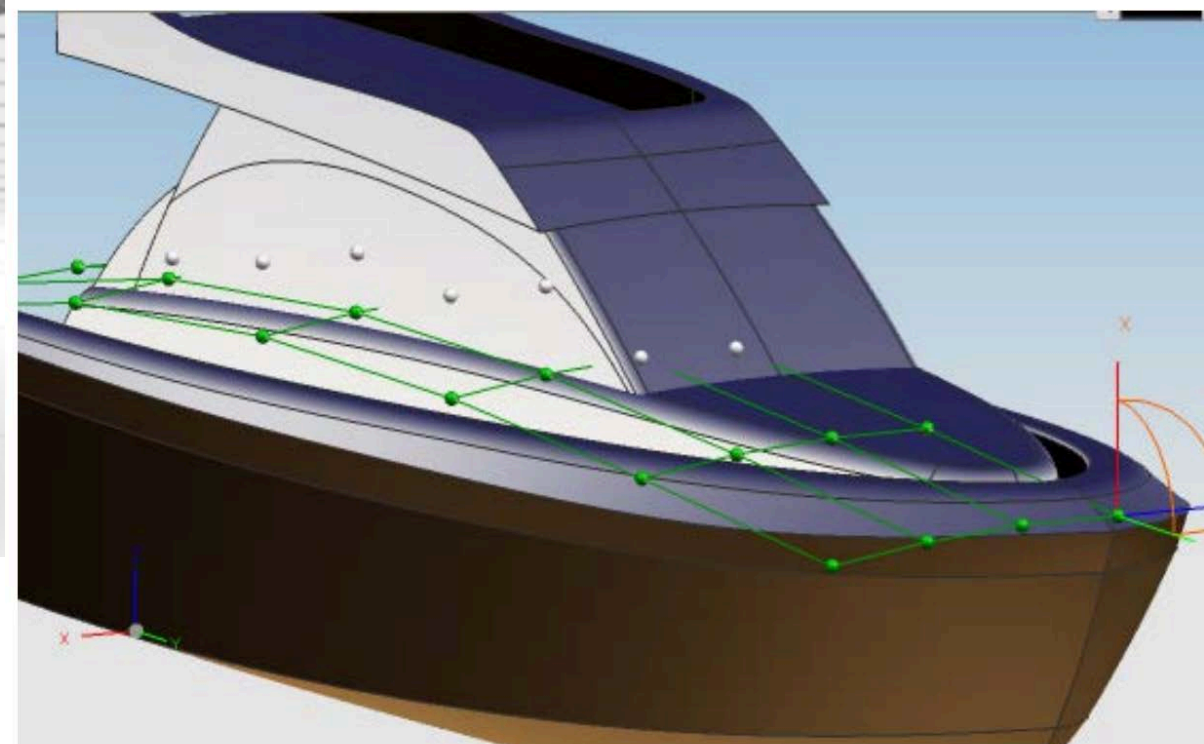
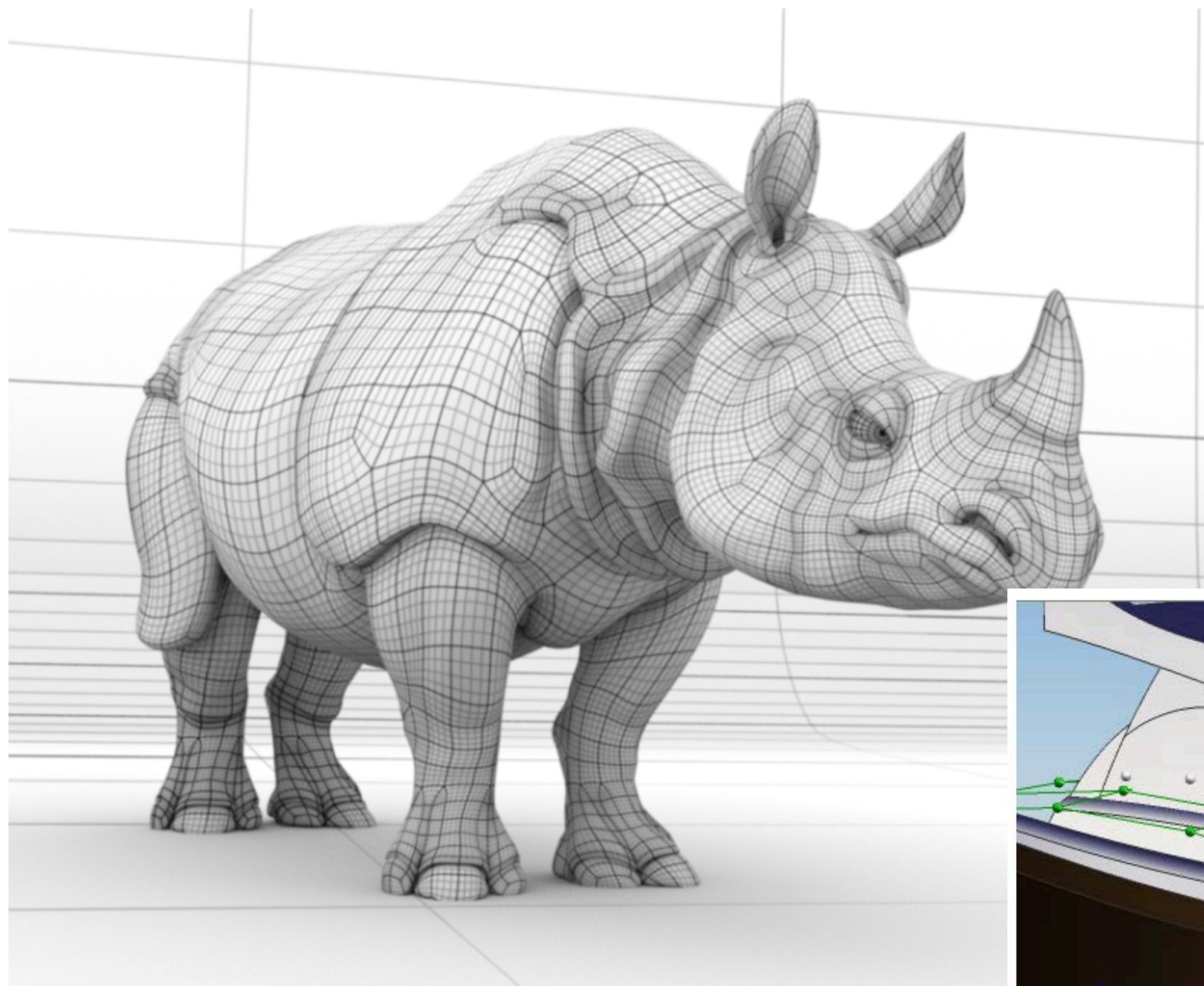
# **Discrete Differential Geometry**

# **Meshes**

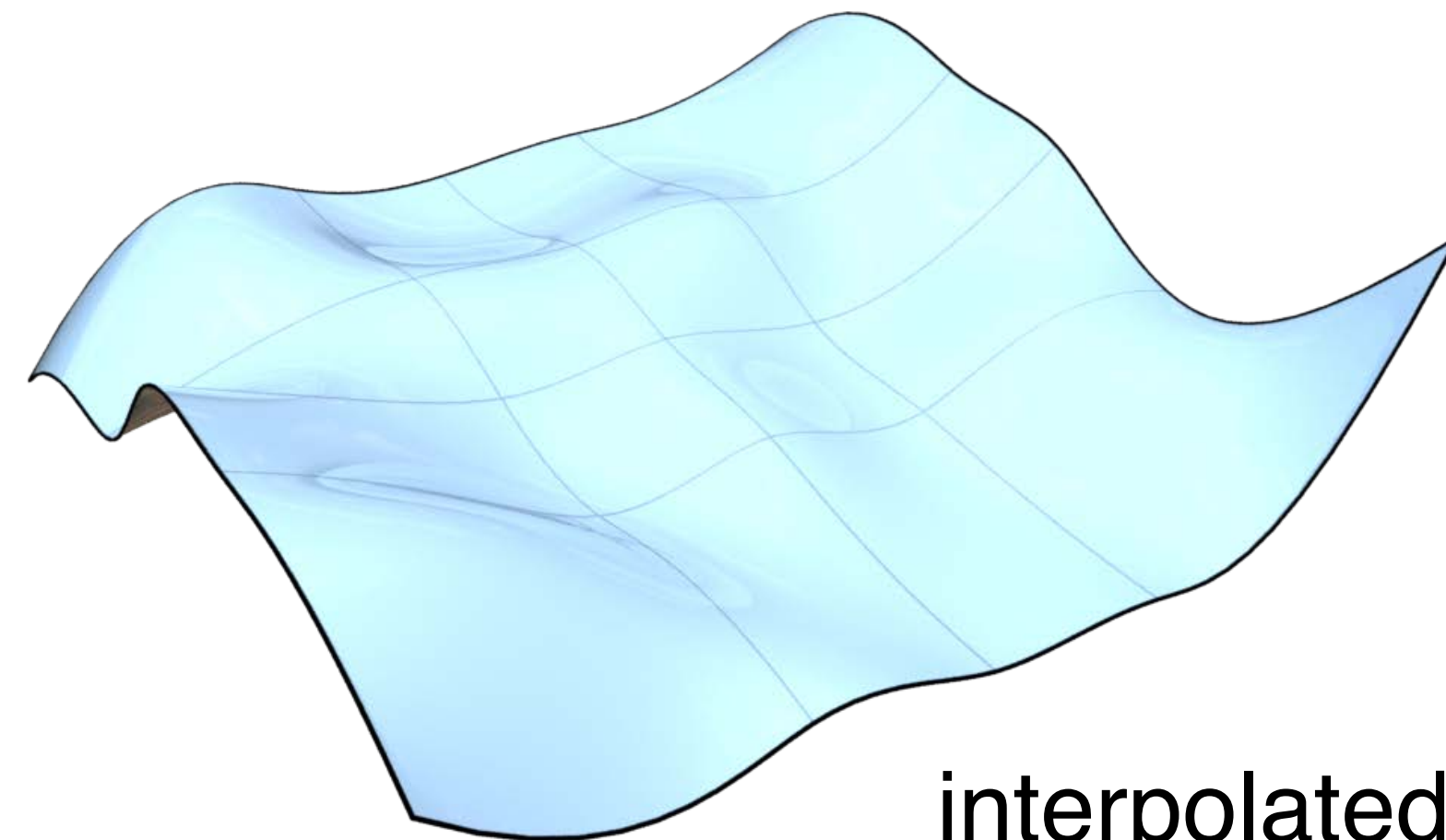
**Albert Chern**

# Discrete surface

- There are many digital surface representations
  - ▶ Spline / subdivision



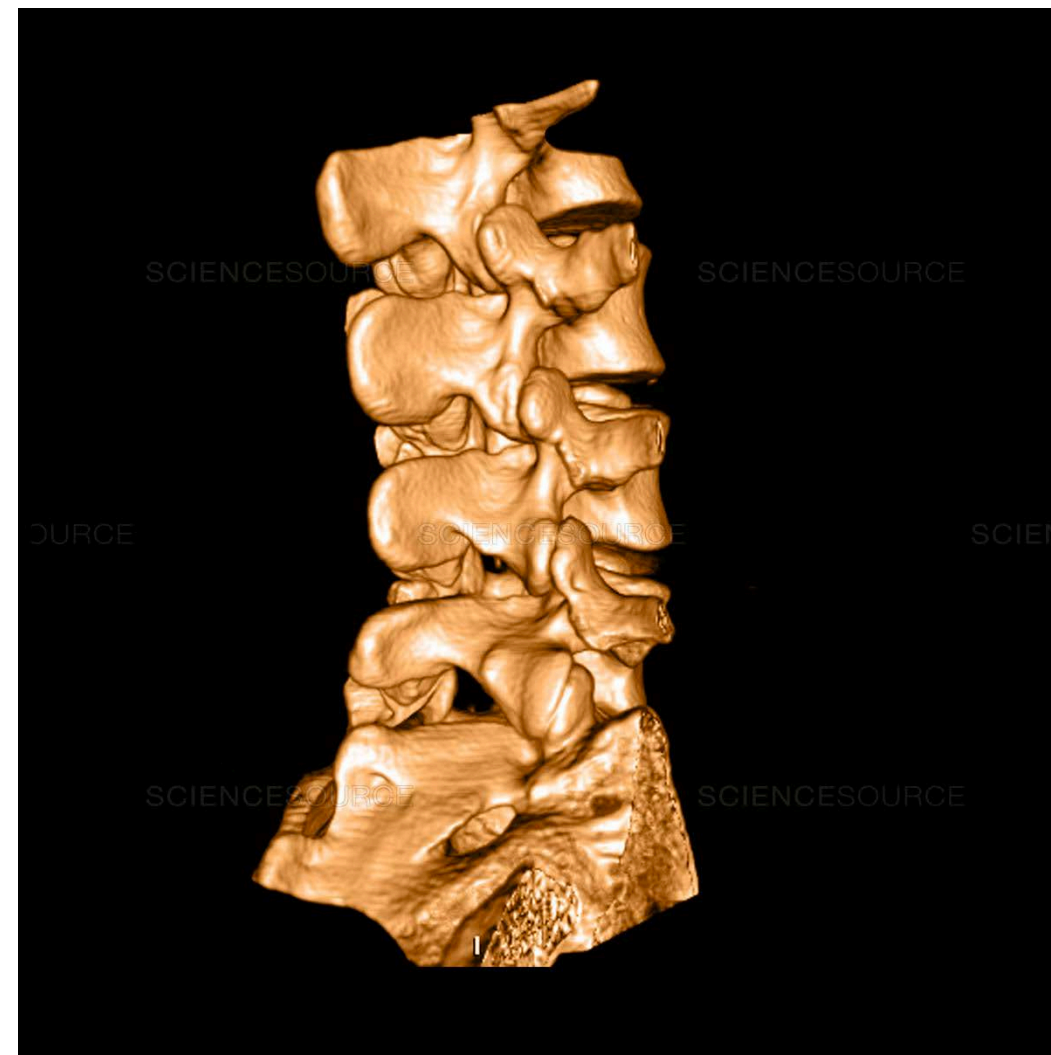
low res control polygon



interpolated smooth surface

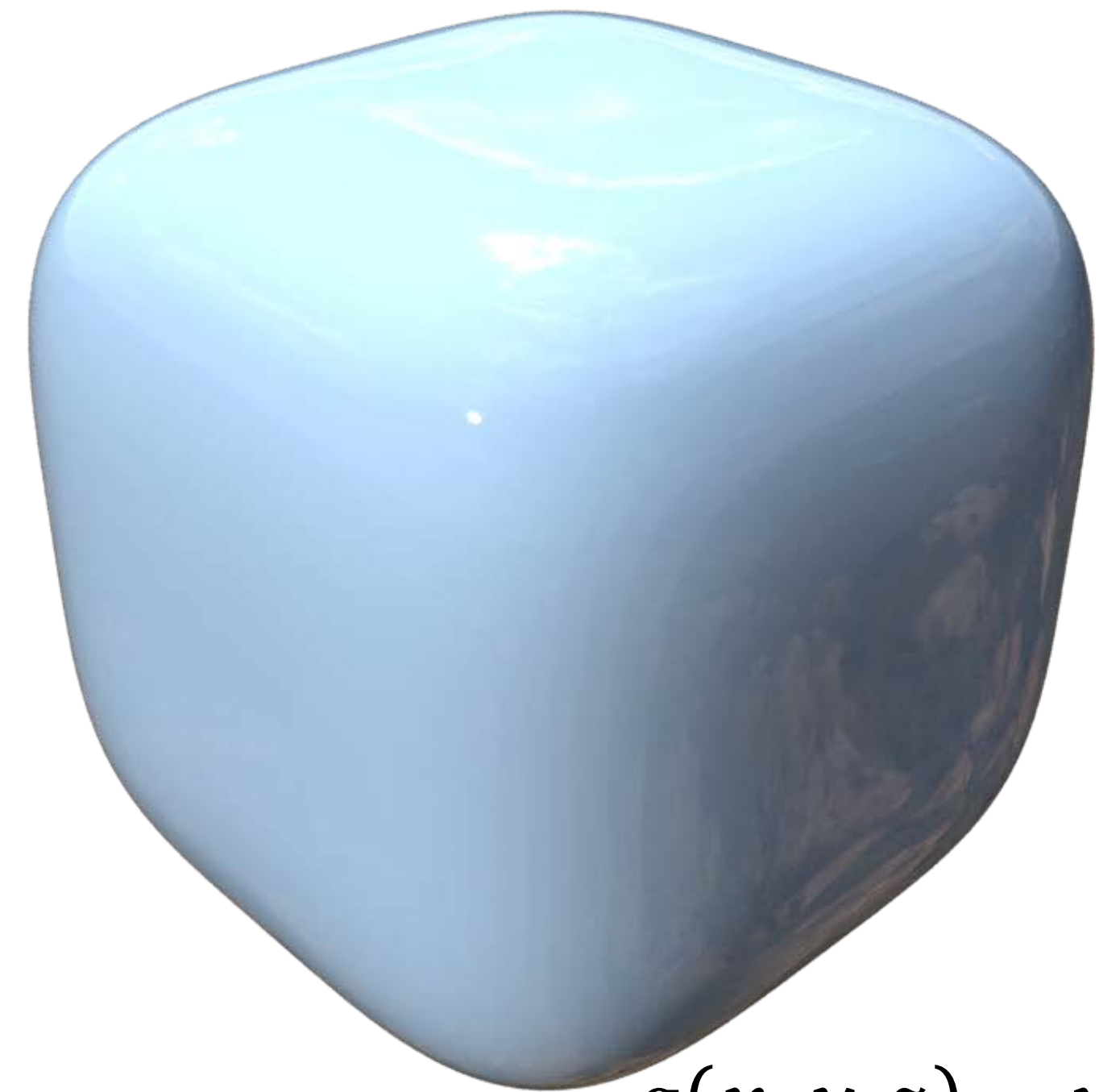
# Discrete surface

- There are many digital surface representations
  - ▶ Spline / subdivision
  - ▶ Level set



$g$  is obtained from a CT scan,  
stored on a volumetric grid

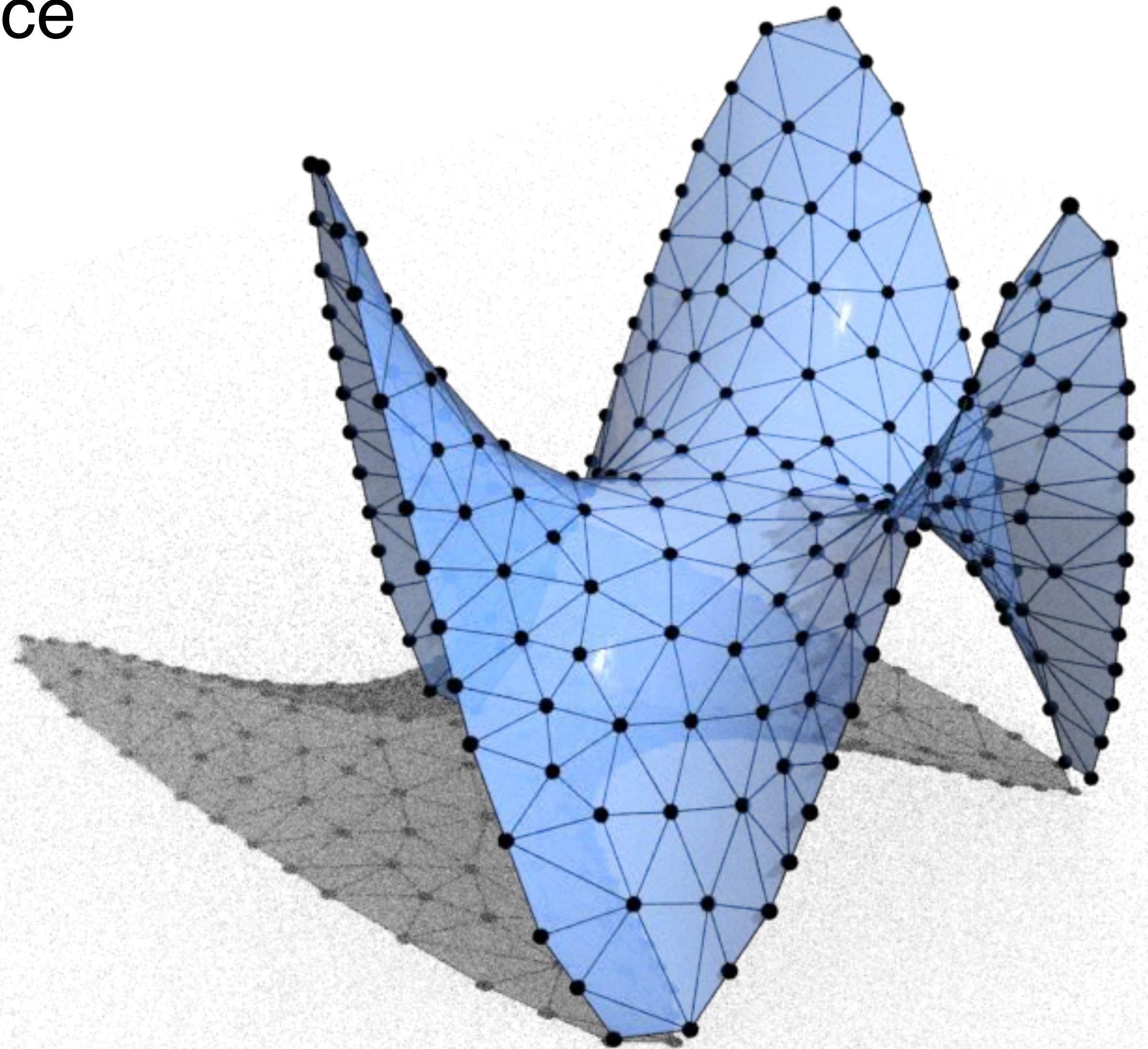
$$g(x, y, z) = 0$$



$$g(x, y, z) = x^4 + y^4 + z^4 - 1$$

# Discrete surface

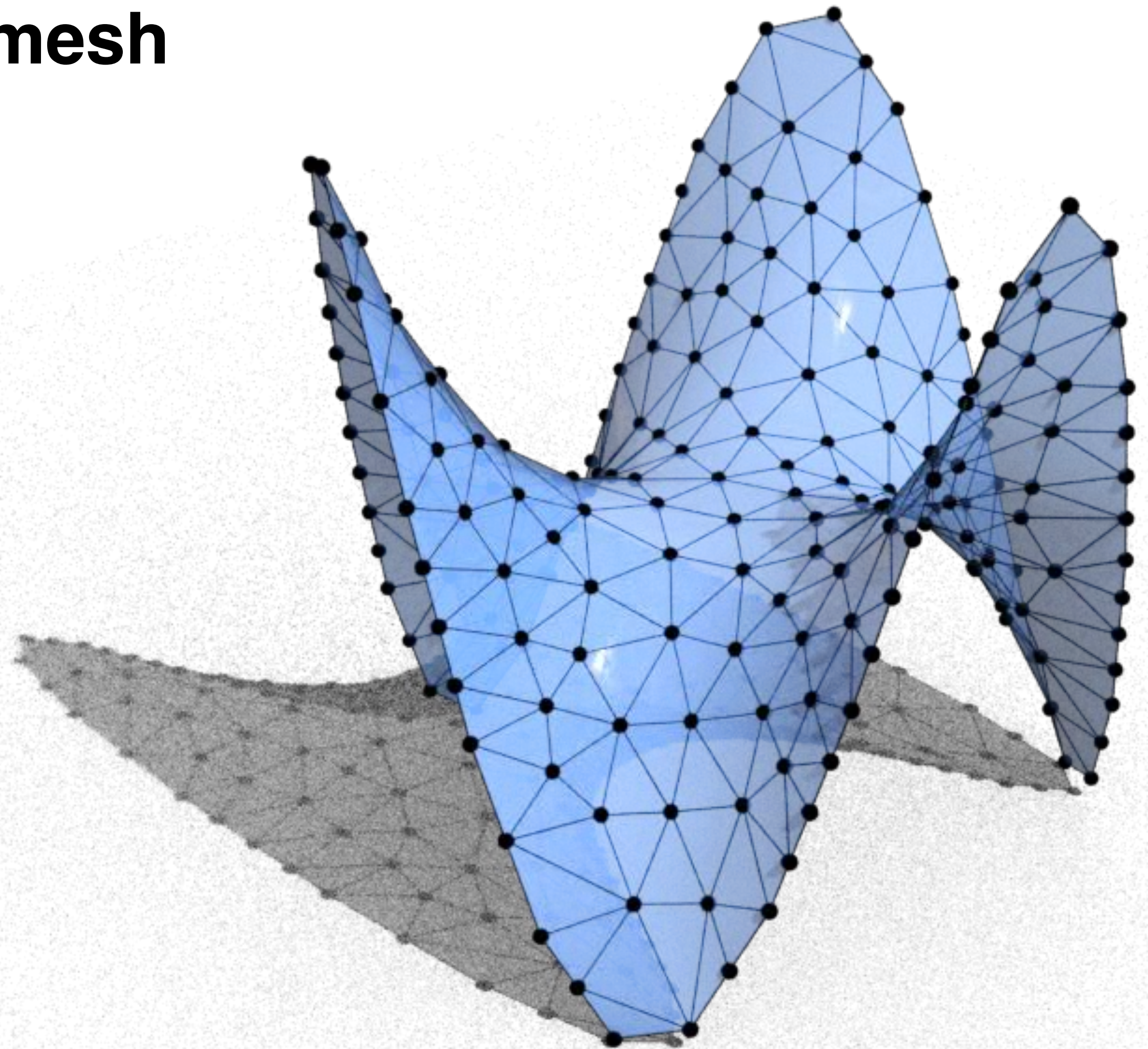
- There are many digital surface representations
  - ▶ Spline / subdivision
  - ▶ Level set
  - ▶ Triangle mesh
- We will primarily use triangle meshes
  - ▶ Flexible
  - ▶ Simple to describe



# Today

## Topology / connectivity of a mesh

- General simplicial complex
- Chain complex
- Manifold mesh
- Global topology
- Euler characteristics

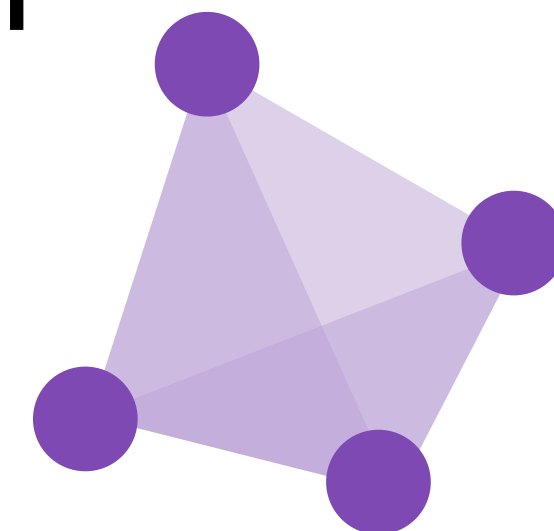
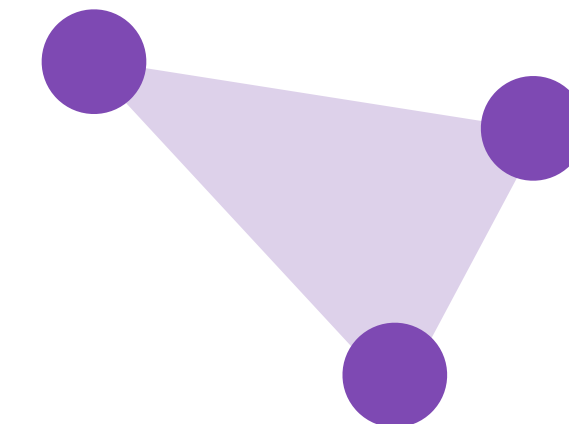
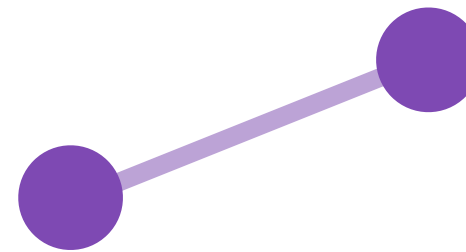


# Simplicial Complex

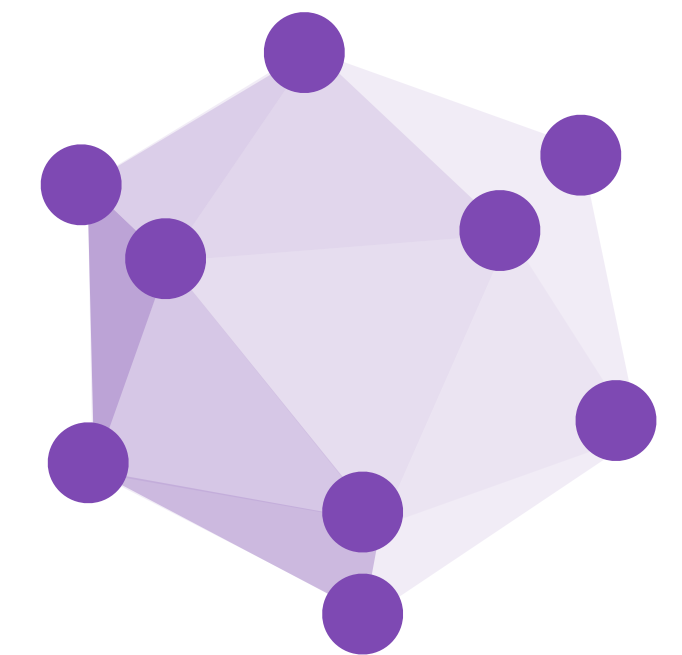
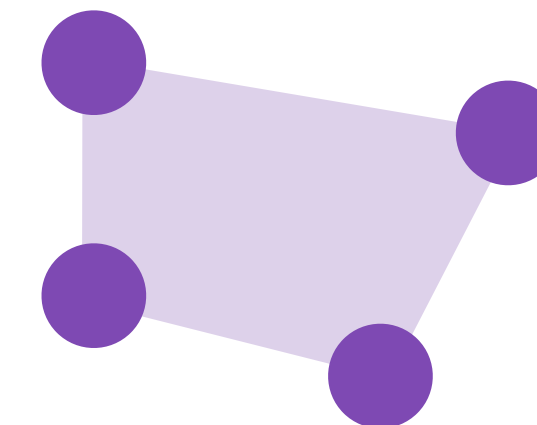
- Simplicial complex
- Chain complex
- Manifold surface mesh
- Surface topology
- Euler characteristics

# Simplex

- **Simplex** encompasses the idea of simplest polygonal/polyhedral object of a given dimension
- 0D simplex = point
- 1D simplex = edge
- 2D simplex = triangle
- 3D simplex = tetrahedron



non-simplex

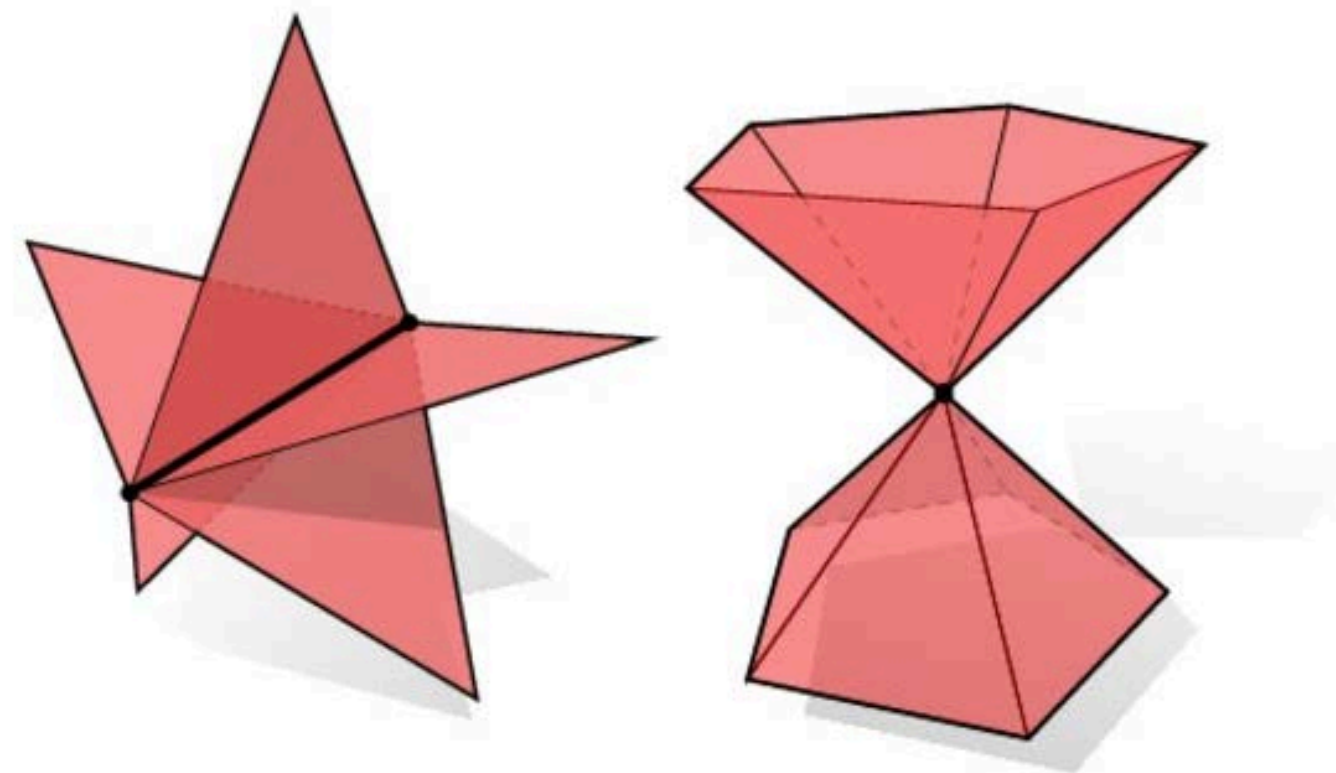
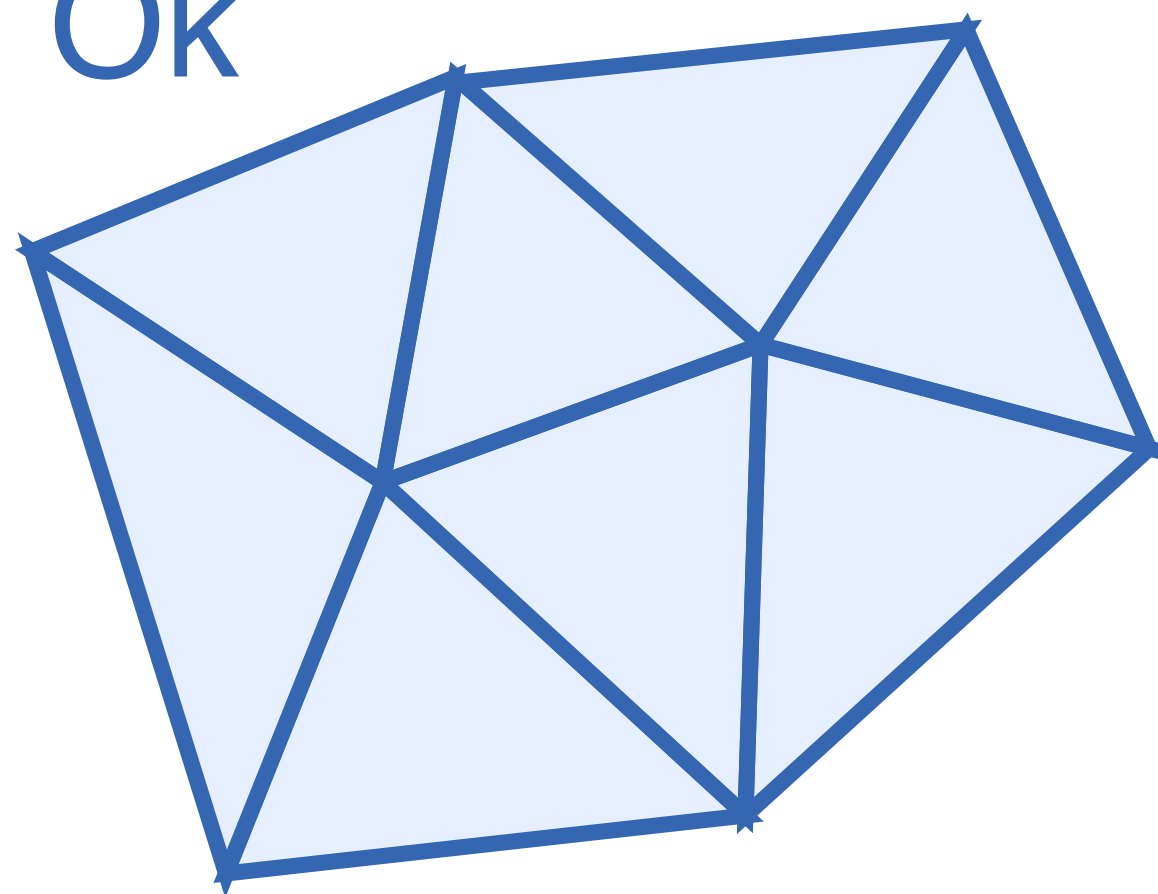


ruled out for now

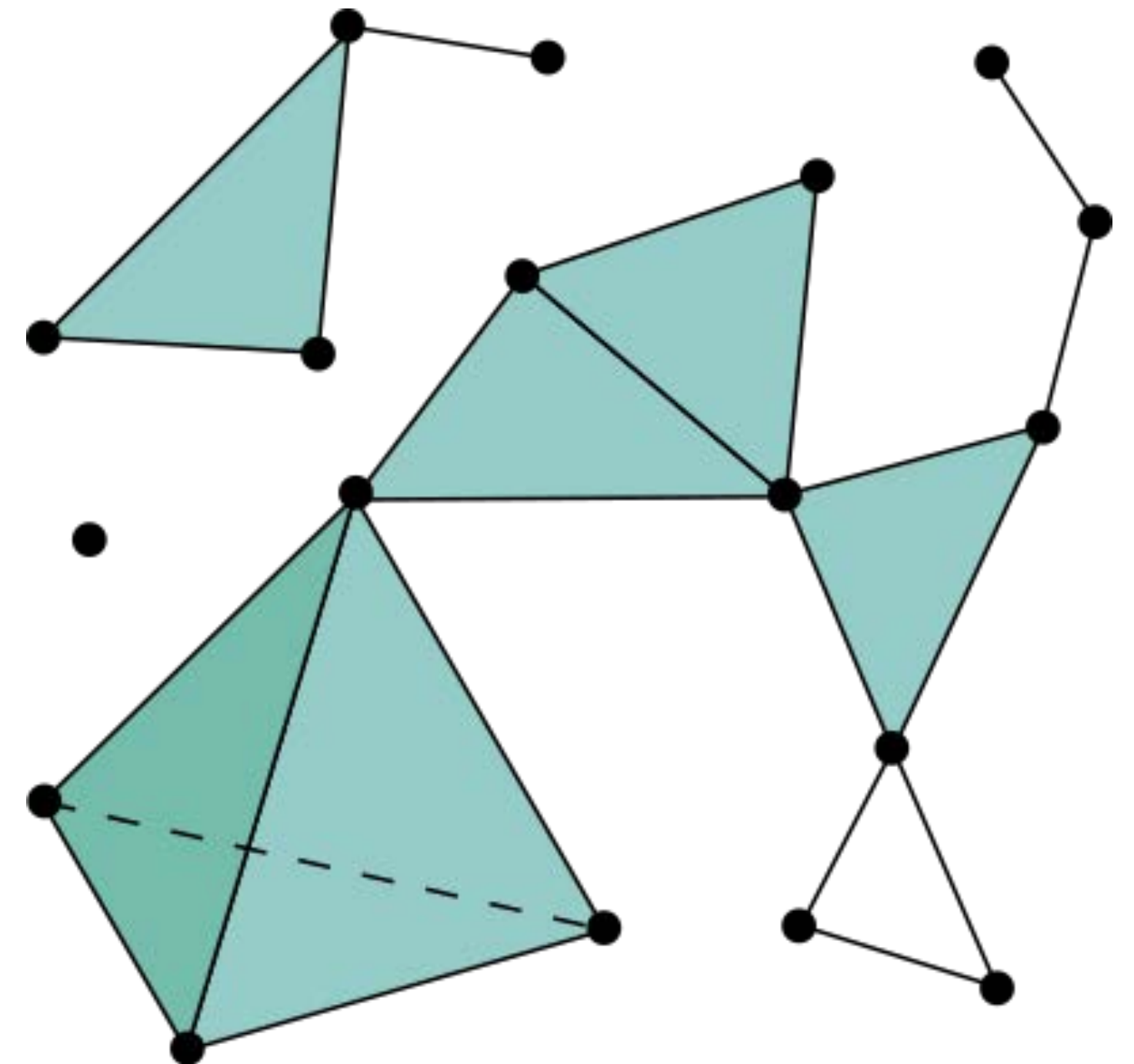
# Simplicial complex

- A **simplicial complex** is a collection of simplices of a variety of dimensions
- They need to satisfy certain rules, e.g. boundary of a triangle must also be collected in the complex
- A proper triangle mesh need to satisfy a bit more rule

Ok



Not ok



general simplicial complex

# Simplicial complex

- A **simplicial complex** is a collection of simplices of a variety of dimensions

$$M = (M_0, M_1, M_2, \dots, M_n)$$

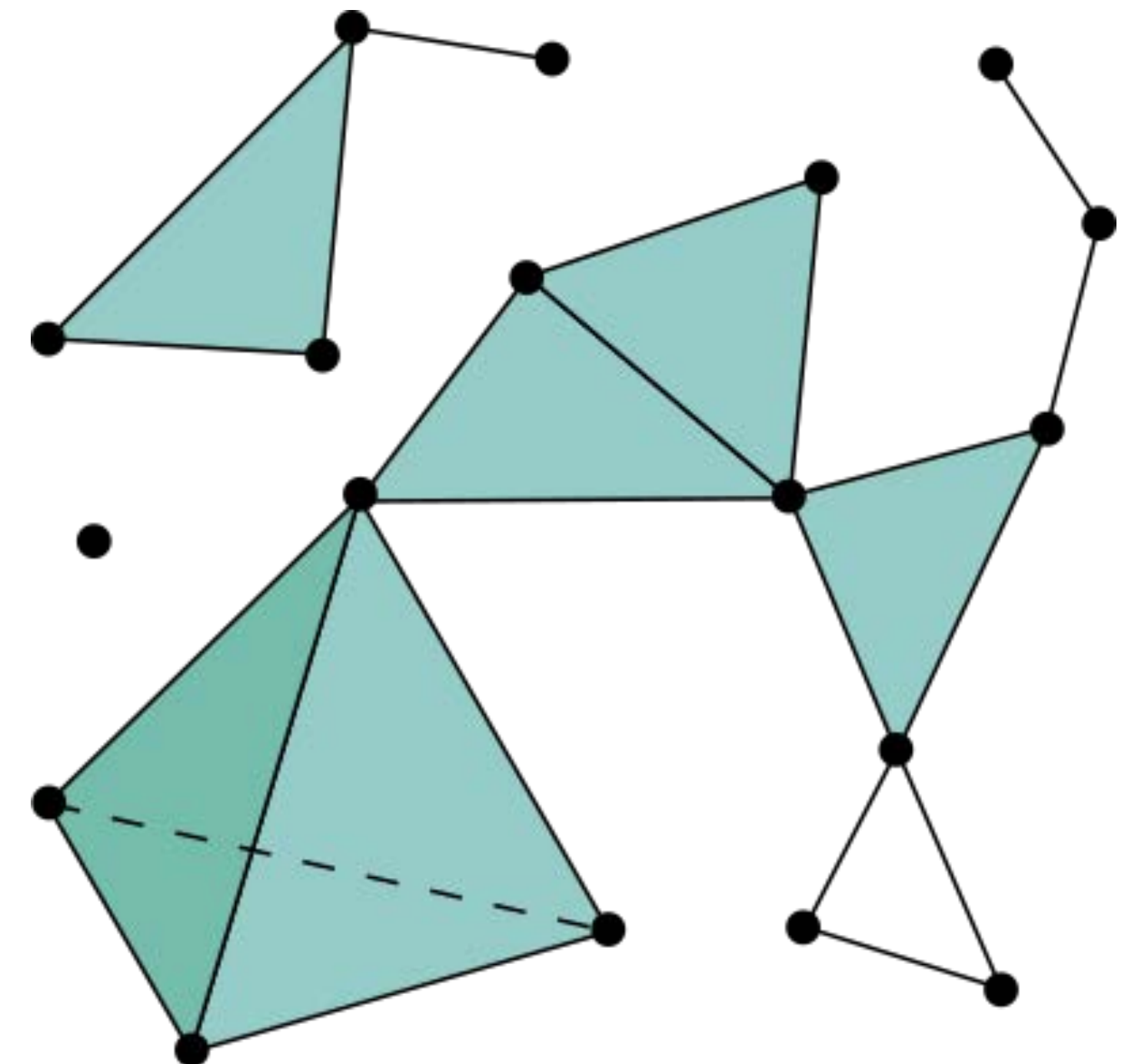
where  $M_k$  is the set of all k-D simplices

We also call  $M_k$  the k-skeleton

- For triangle mesh (maximal dimension  $n=2$ )

$$M = (M_0, M_1, M_2) = (V, E, F)$$

- ▶ V: vertices or “points” (in Houdini’s terminology)
- ▶ E: edges
- ▶ F: faces



general simplicial complex

# Simplicial complex

- The point set (0-skeleton)

$$M_0 = \{1, 2, \dots, i, \dots, |M_0|\}$$

is a finite set of indices, just as labeling

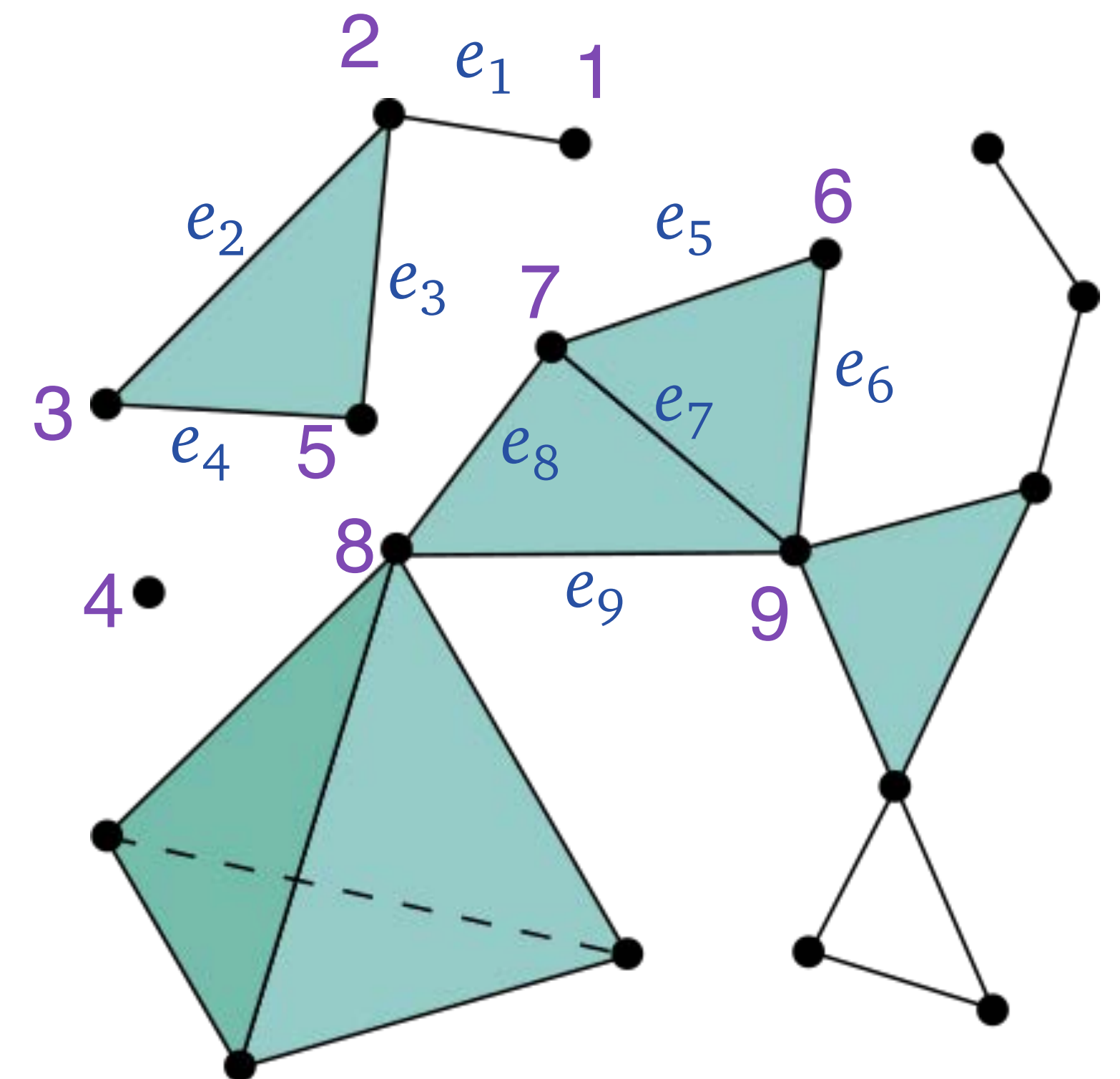
- The edge set (1-skeleton)

$$M_1 = \{e_1, e_2, \dots, e_{|M_1|}\}$$

where each edge is a pair of points

for example  $e_3 = (2, 5)$

- ▶ Unoriented edge: treat the pair like “set”  $e_3 = \{2, 5\} = \{5, 2\}$
- ▶ Oriented edge: treat the pair like “ordered list”  $e_3 = [2, 5] = -[5, 2]$



general simplicial complex

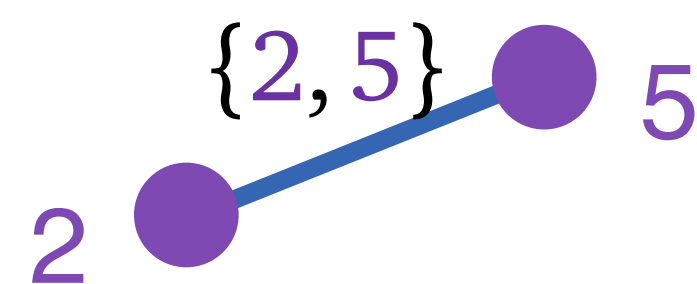
# Simplicial complex

- The edge set (1-skeleton)

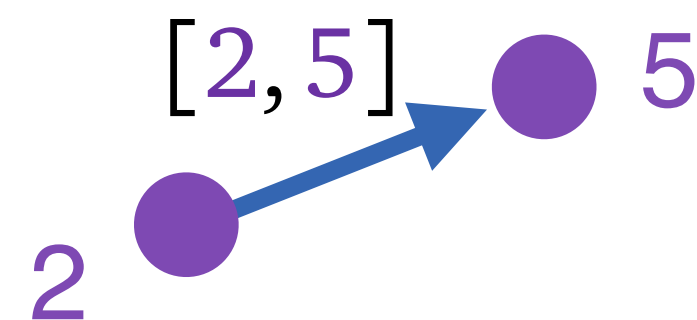
$$M_1 = \{e_1, e_2, \dots, e_{|M_1|}\}$$

where each edge is a pair of points  
for example  $e_3 = (2, 5)$

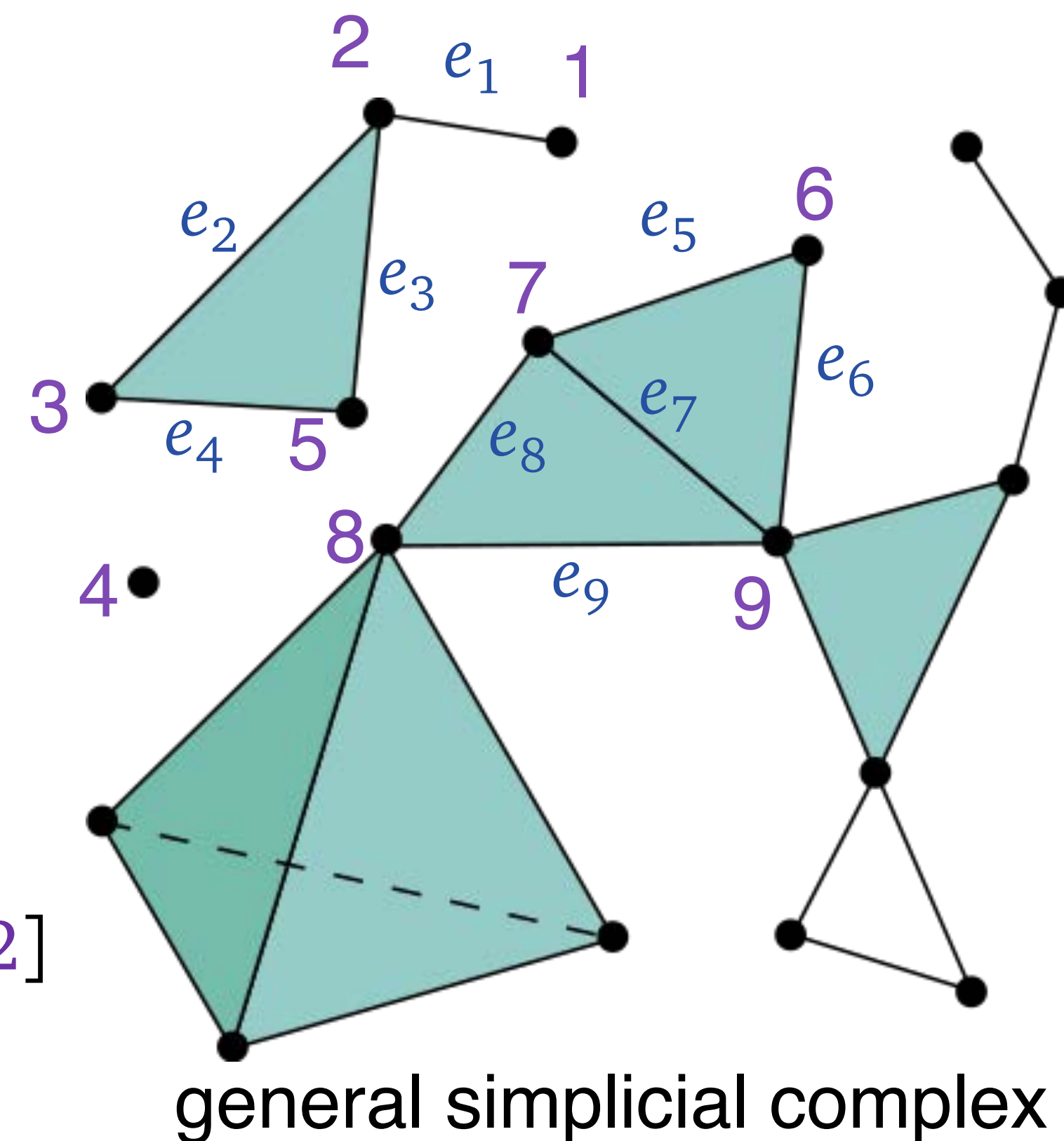
- Unoriented edge: treat the pair like “set”  $e_3 = \{2, 5\} = \{5, 2\}$



- Oriented edge: treat the pair like “ordered list”  $e_3 = [2, 5] = -[5, 2]$



- To denote incidence relation, write  $5 \prec e_3$  “point 5 is incident to edge\_3”

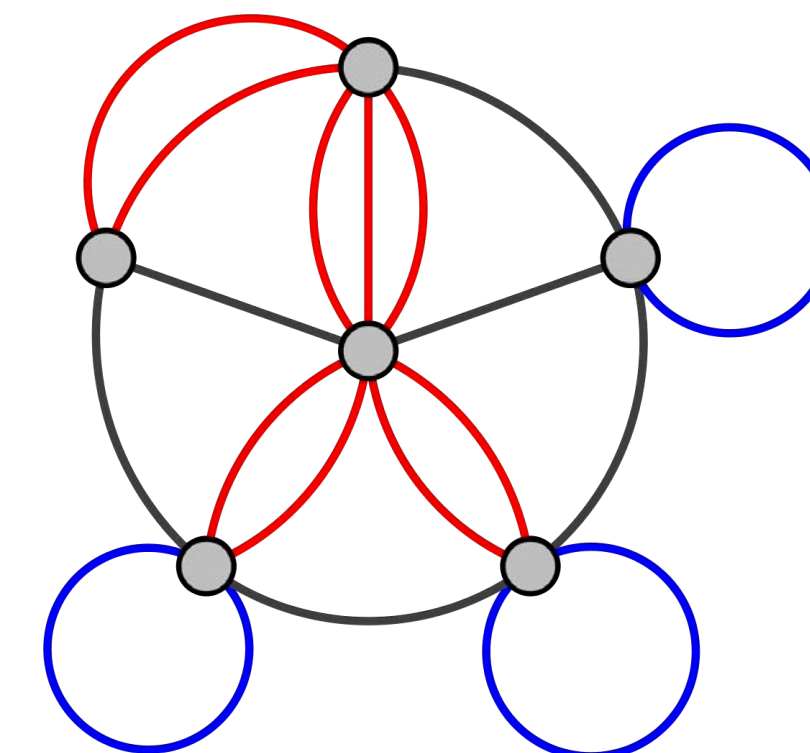
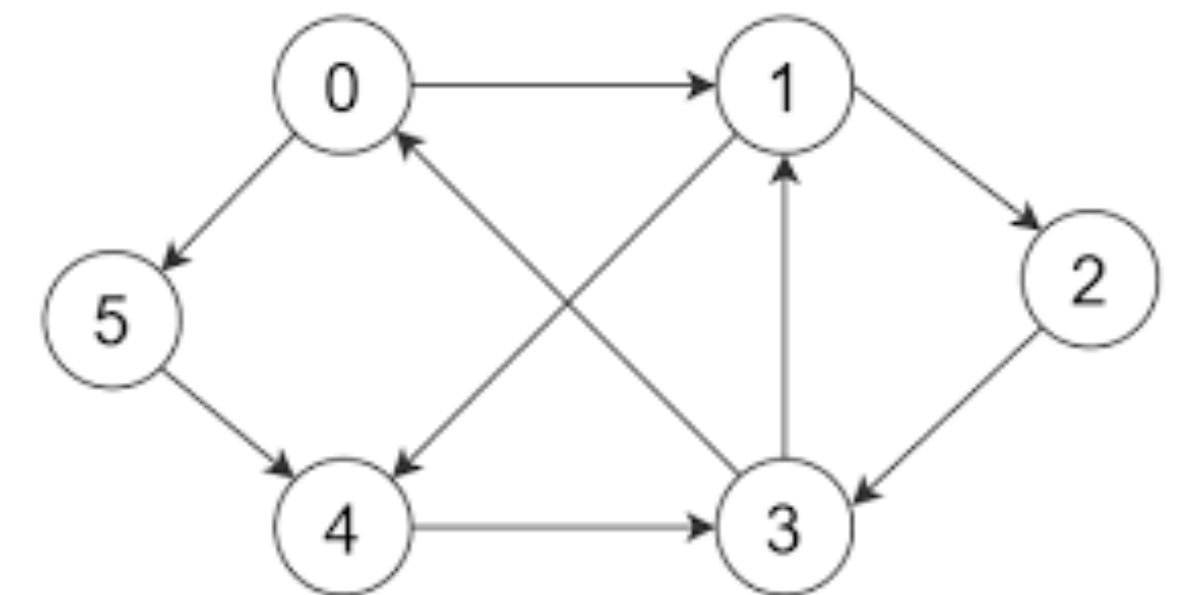
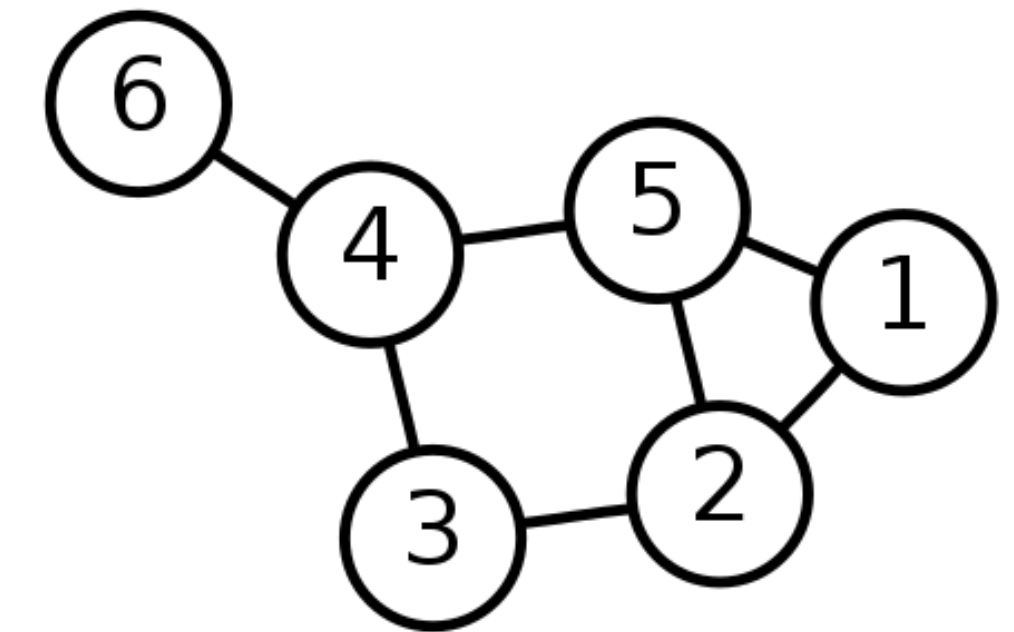


# Example: Graph

- A graph is a simplicial mesh of dimension 1

$$M = (V, E)$$

- If each edge is unordered pair, then we have an unoriented graph
- If each edge is ordered pair, then we have an oriented graph, or directed graph
- Side note: sometimes multigraph is considered in graph theory, so “edge = pair of points” is not enough to distinguish edges



# Simplicial complex

- The face set (2-skeleton)

$$M_2 = \{f_1, f_2, \dots, f_{|M_2|}\}$$

where each face is a triplet of points

**Un-oriented case:**

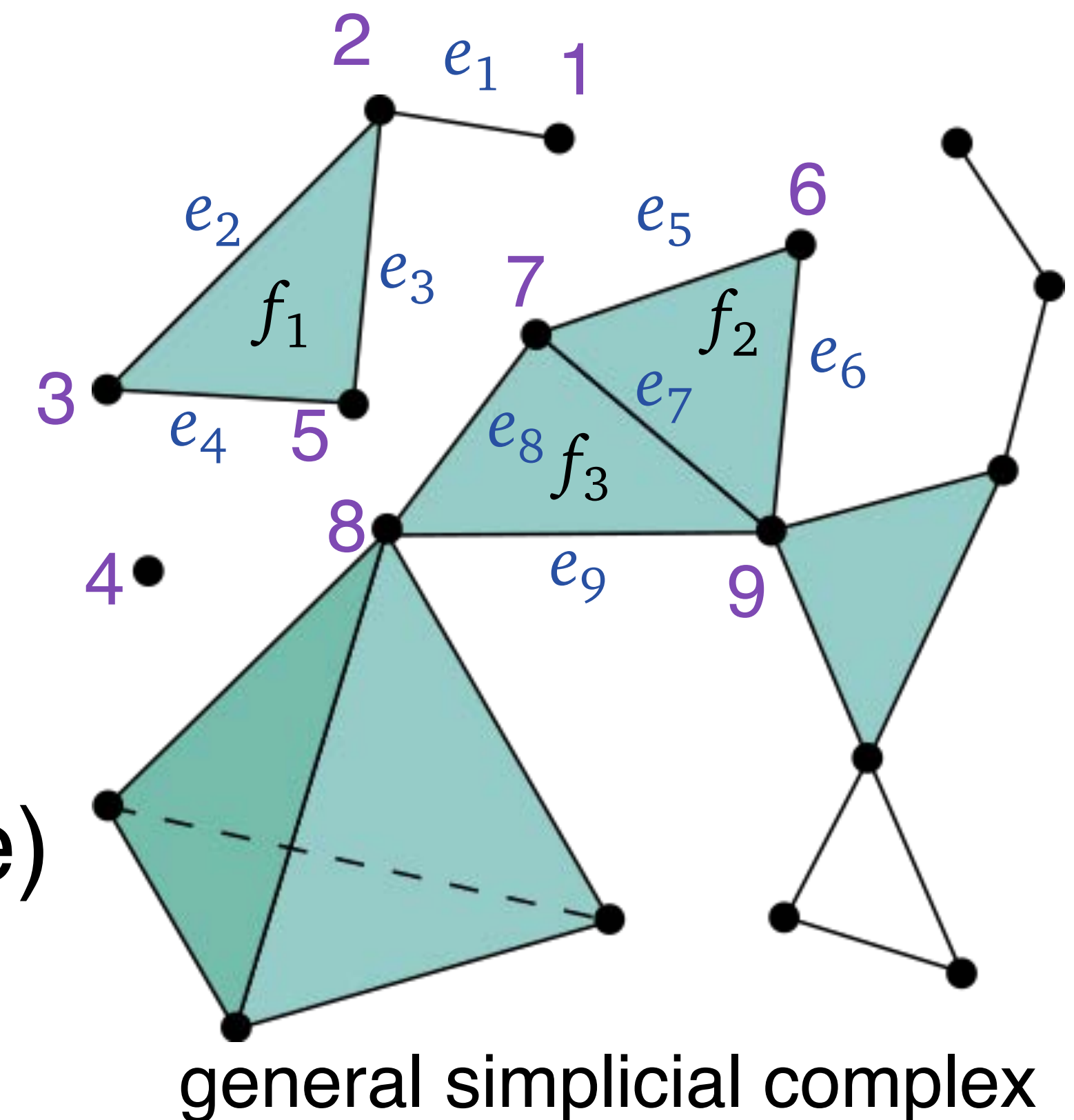
$$f_2 = \{6, 7, 9\} = \{7, 6, 9\} = \{7, 9, 6\} \\ = \dots$$

(All permuted versions represent the same face)

**Oriented case**

$$f_2 = [6, 7, 9] = -[7, 6, 9] = [7, 9, 6]$$

(Even permutation versions represent the same face)



# Simplicial complex

- The face set (2-skeleton)

$$M_2 = \{f_1, f_2, \dots, f_{|M_2|}\}$$

where each face is a triplet of points

**Un-oriented case:**

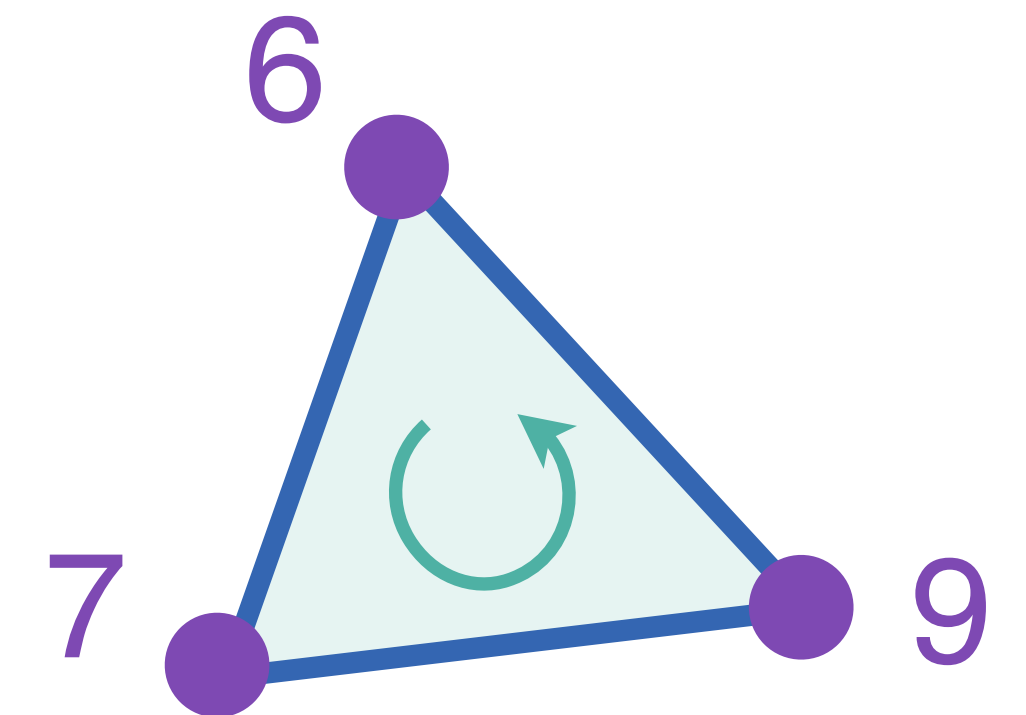
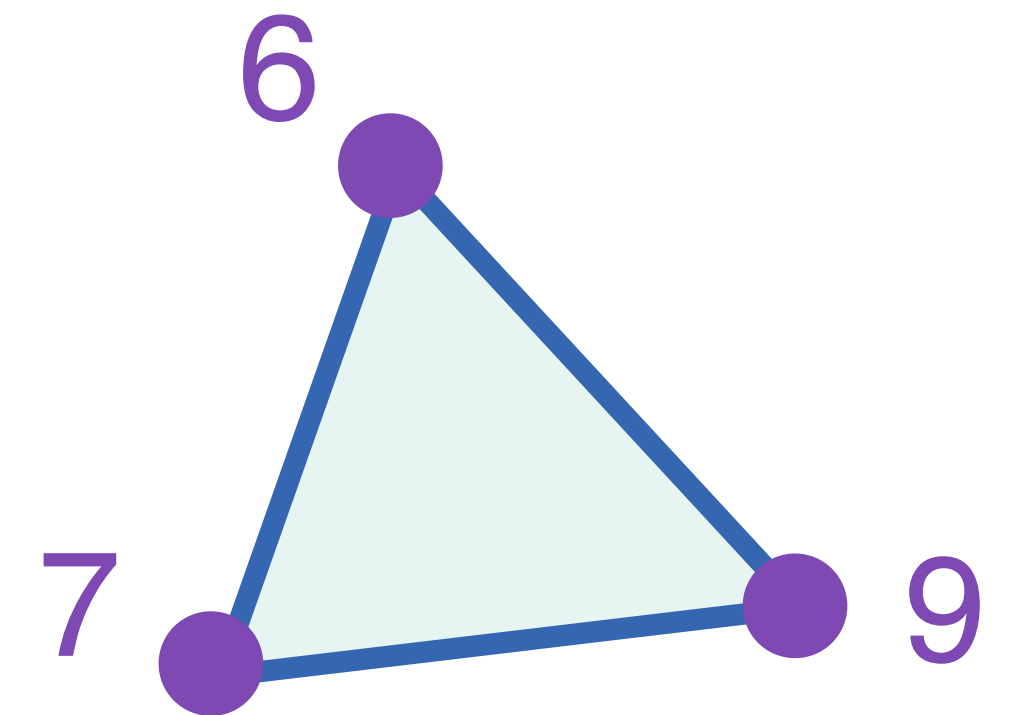
$$f_2 = \{6, 7, 9\} = \{7, 6, 9\} = \{7, 9, 6\} \\ = \dots$$

(All permuted versions represent the same face)

**Oriented case**

$$f_2 = [6, 7, 9] = -[7, 6, 9] = [7, 9, 6]$$

(Even permutation versions represent the same face)



# Simplicial complex

- The face set (2-skeleton)

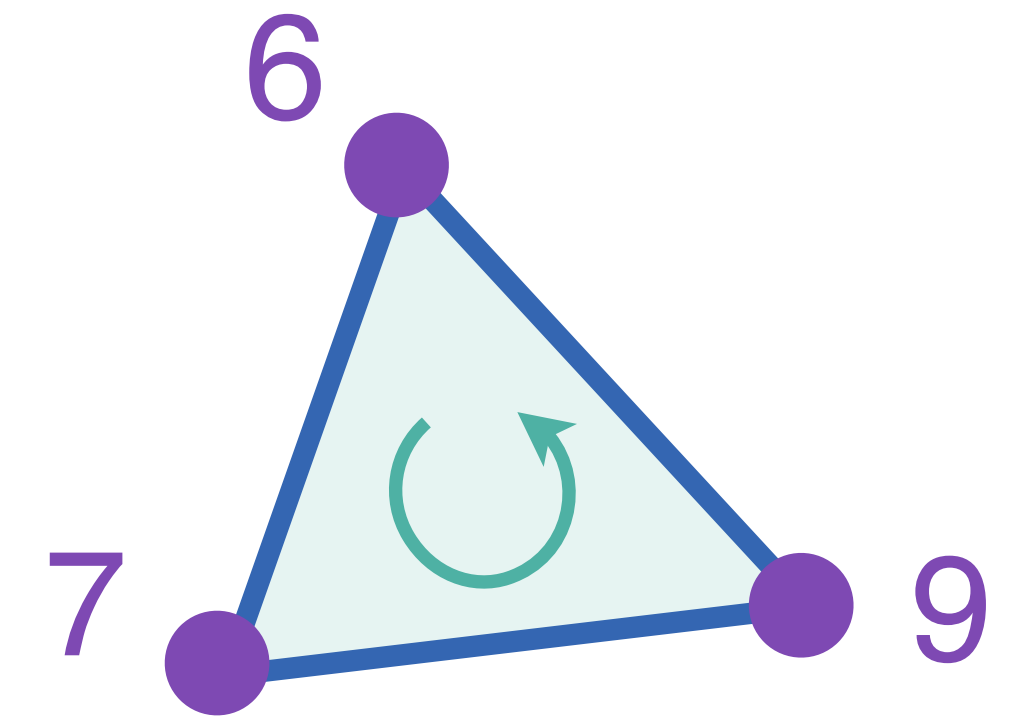
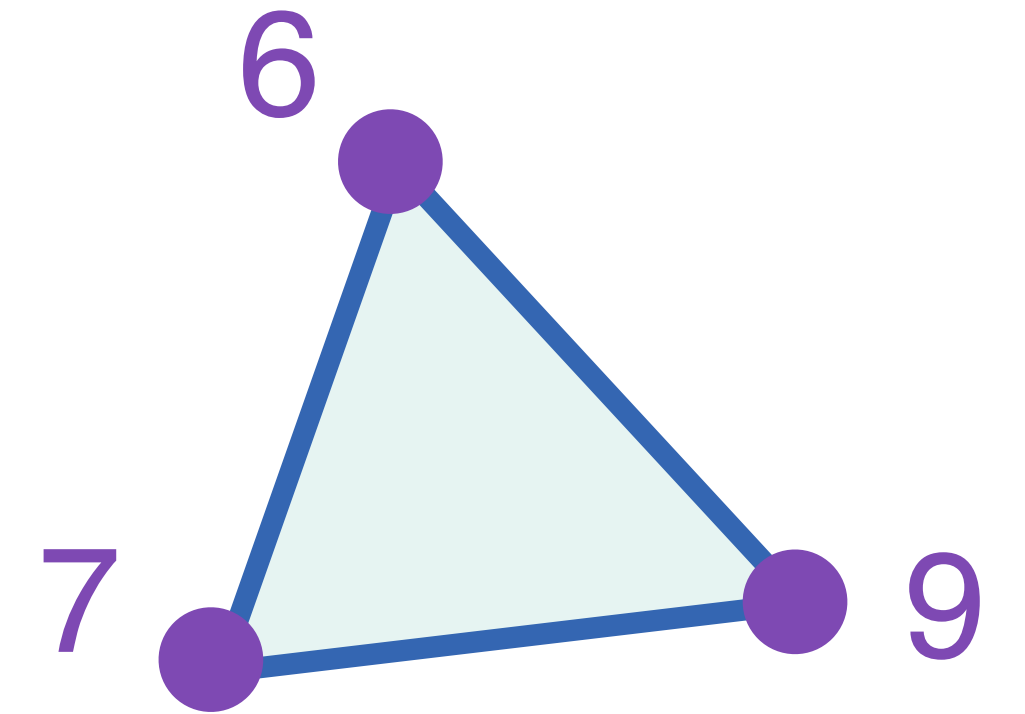
$$M_2 = \{f_1, f_2, \dots, f_{|M_2|}\}$$

where each face is a triplet of points

## Incidence

$$6 \prec (6, 7, 9)$$

$$(7, 6) \prec (6, 7, 9)$$

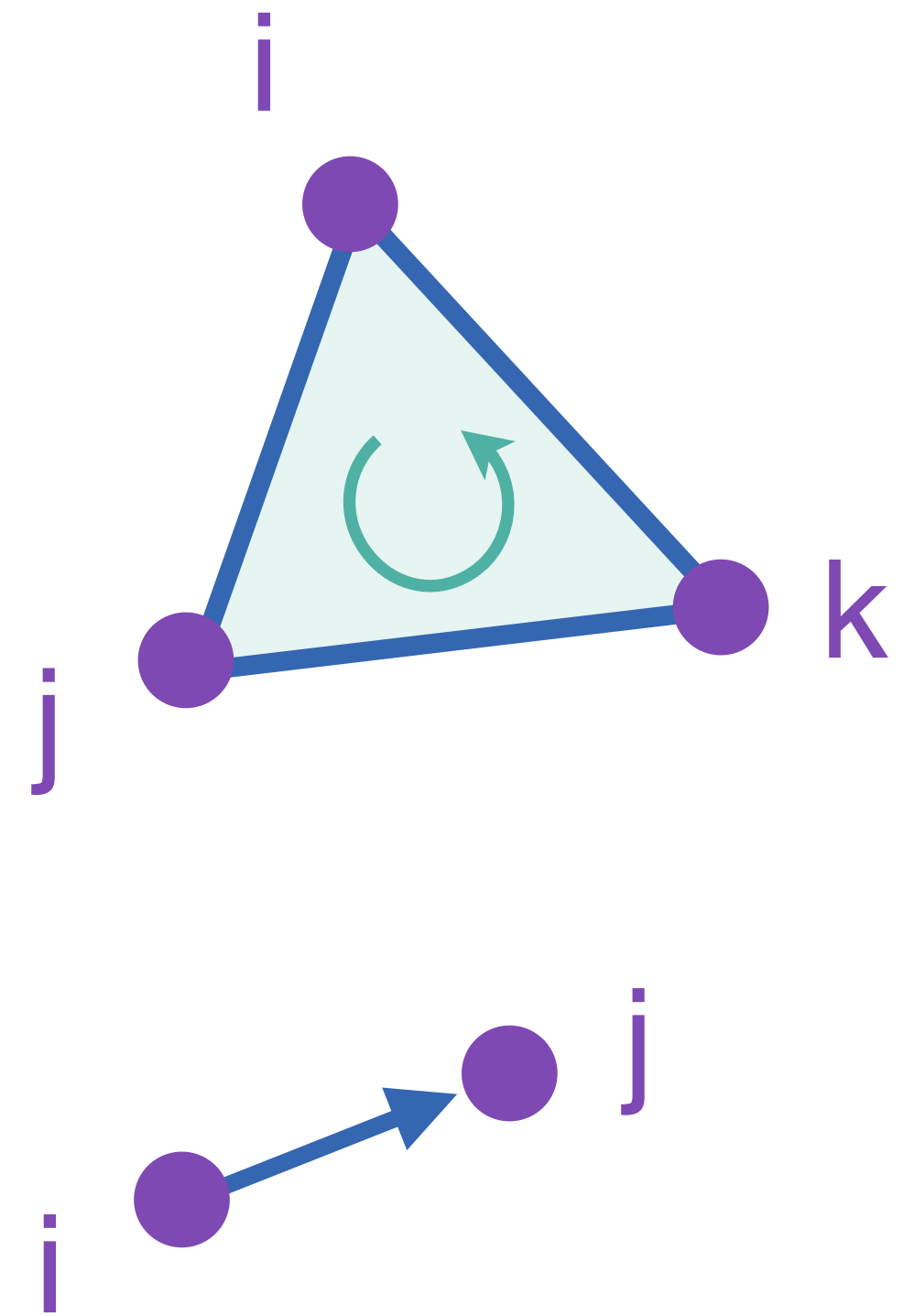


# Boundary

- What is the boundary of an (oriented) simplex?

$$\partial[i, j, k] = [i, j] + [j, k] + [k, i]$$

$$\partial[i, j] = -[i] + [j]$$

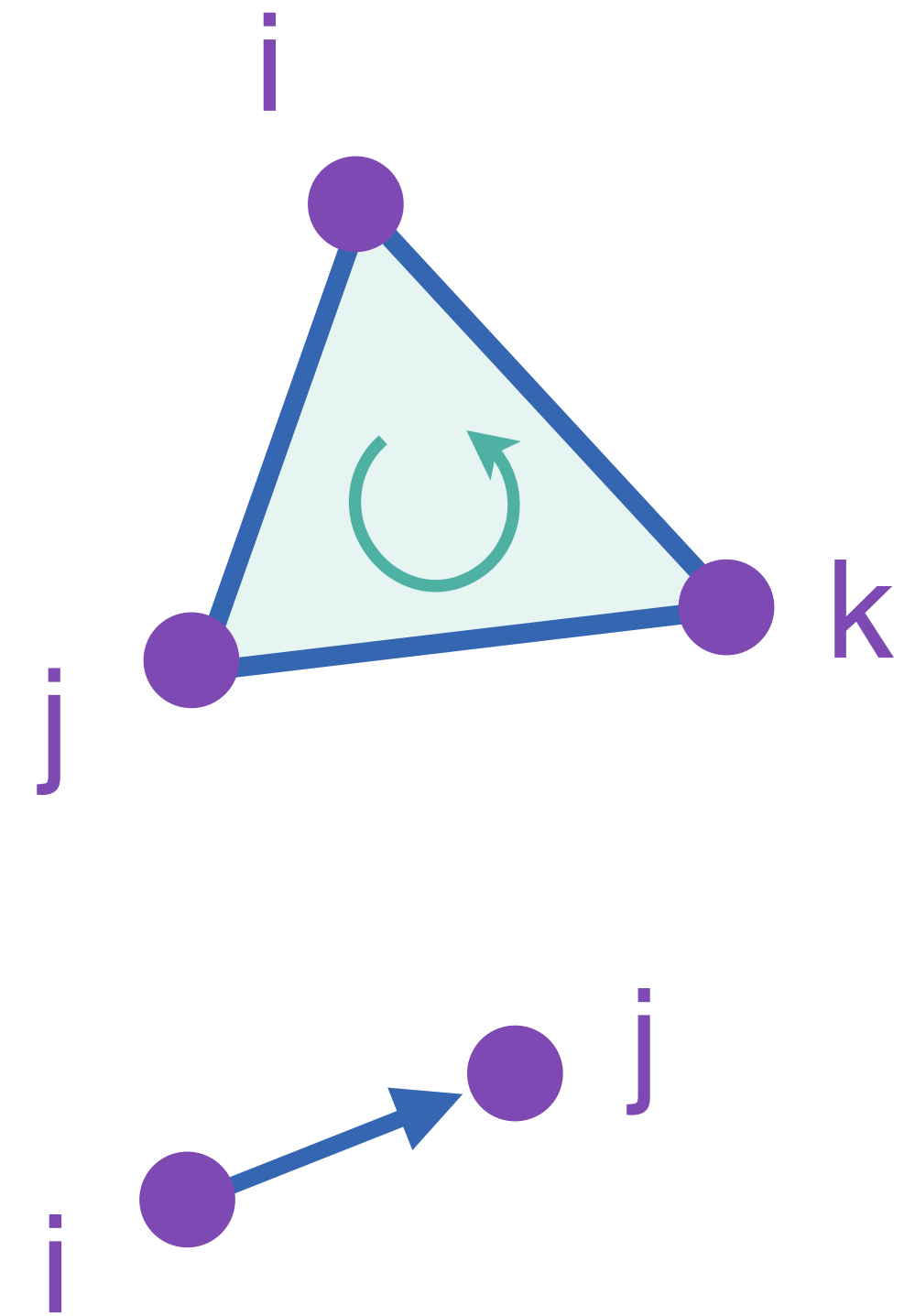


# Boundary

- What is the boundary of an (oriented) simplex?

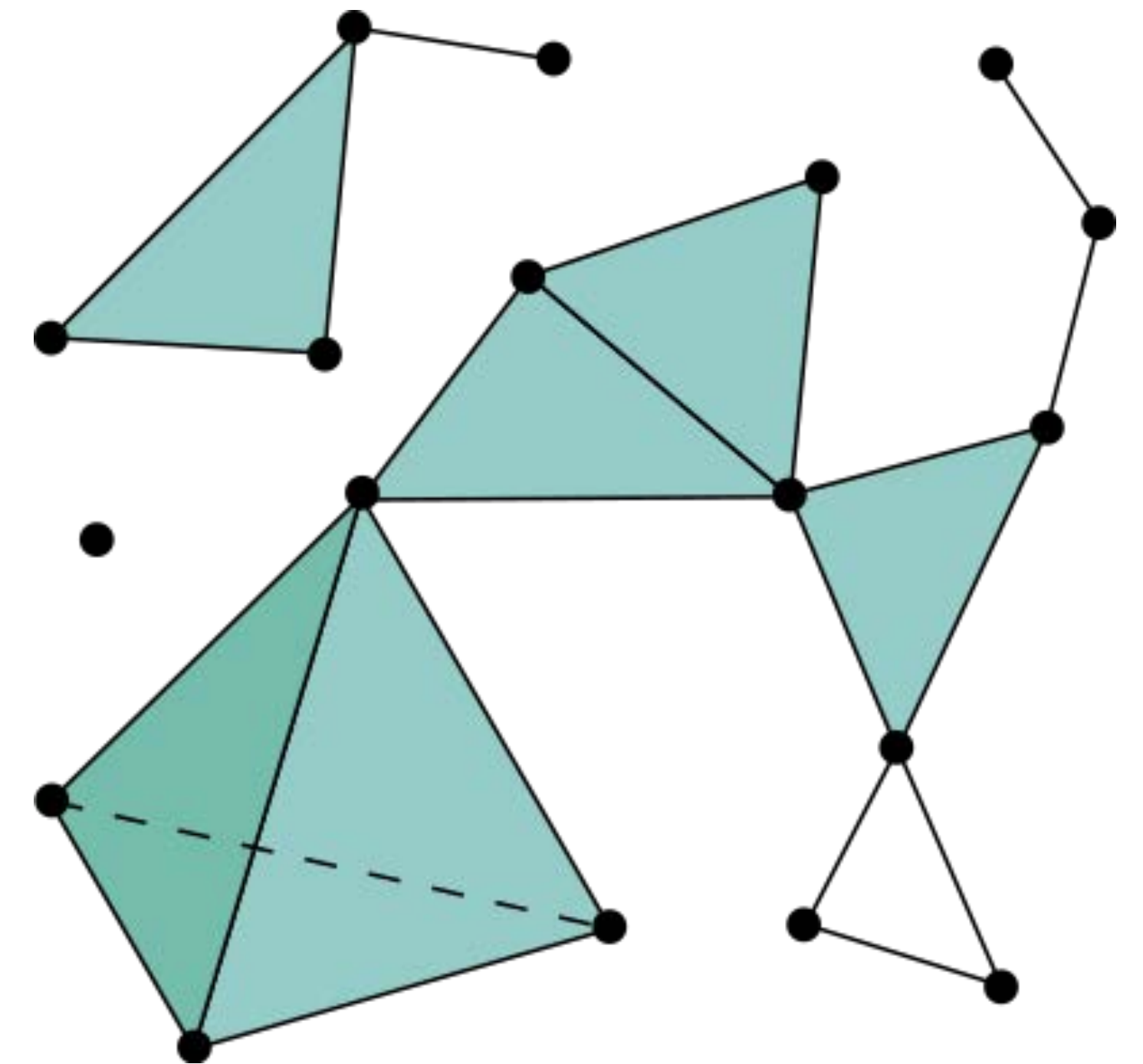
$$\partial[i_0, \dots, i_k] = \sum_{\ell=0}^k (-1)^\ell [i_0, \dots, \hat{i}_\ell, \dots, i_k]$$

the hat ^ means removing that term



# Simplicial complex (summary)

- A **simplicial complex** is a collection of points, edges, triangles, and potentially higher dimensional simplices
- Boundary of each simplex should be collected in the complex
- Oriented or non-oriented
- Incidence notation  $i \prec (i, j)$
- It's a generalization of graph
- Triangle mesh is a special case



# Chain Complex

- Simplicial complex
- Chain complex
- Manifold surface mesh
- Surface topology
- Euler characteristics

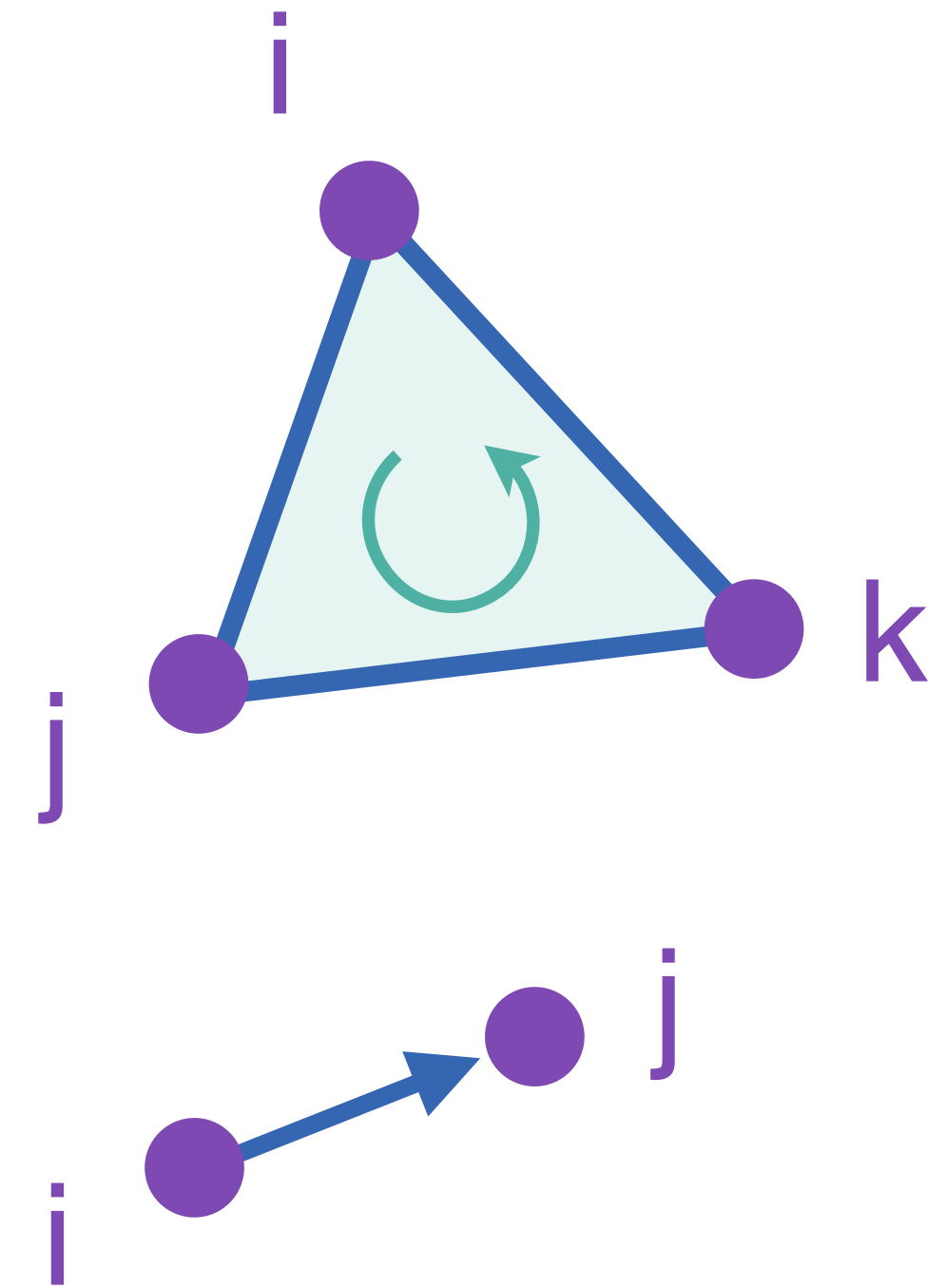
# Chains

- Recall:

$$\partial[i, j, k] = [i, j] + [j, k] + [k, i]$$

$$\partial[i, j] = -[i] + [j]$$

$$\partial[i_0, \dots, i_k] = \sum_{\ell=0}^k (-1)^\ell [i_0, \dots, \widehat{i}_\ell, \dots, i_k]$$



- As we can see, it is convenient to write “sums of simplices” to express the boundary operation
- We call “linear combinations of simplices” expressions **chains**.

# Chains

- Let  $M = (M_0, M_1, M_2, \dots, M_n)$  be a (simplicial) complex.
- $k$ -skeleton:  $M_k = \{\sigma_1, \dots, \sigma_{|M_k|}\}$  each  $\sigma_j$  is a  $k$ -dim simplex.
- The **space of  $k$ -chains** is defined by

$$\begin{aligned} C_k &:= \left\{ a_1 \sigma_1 + \dots + a_{|M_k|} \sigma_{|M_k|} \mid a_j \in R \right\} & R = \mathbb{Z} \text{ or } \mathbb{R} \\ &= \text{span}_R \{\sigma_1, \dots, \sigma_{|M_k|}\} \\ &= \text{span}_R M_k \end{aligned}$$

# Boundary operator is a linear map

Boundary  
operator  
for  $k$ -simplices

$k$ -simplex

$(k-1)$ -chain

$$\partial_k [i_0, \dots, i_k] = \sum_{\ell=0}^k (-1)^\ell [i_0, \dots, \widehat{i_\ell}, \dots, i_k]$$

- Extend this boundary operator to a map from chains to chains.

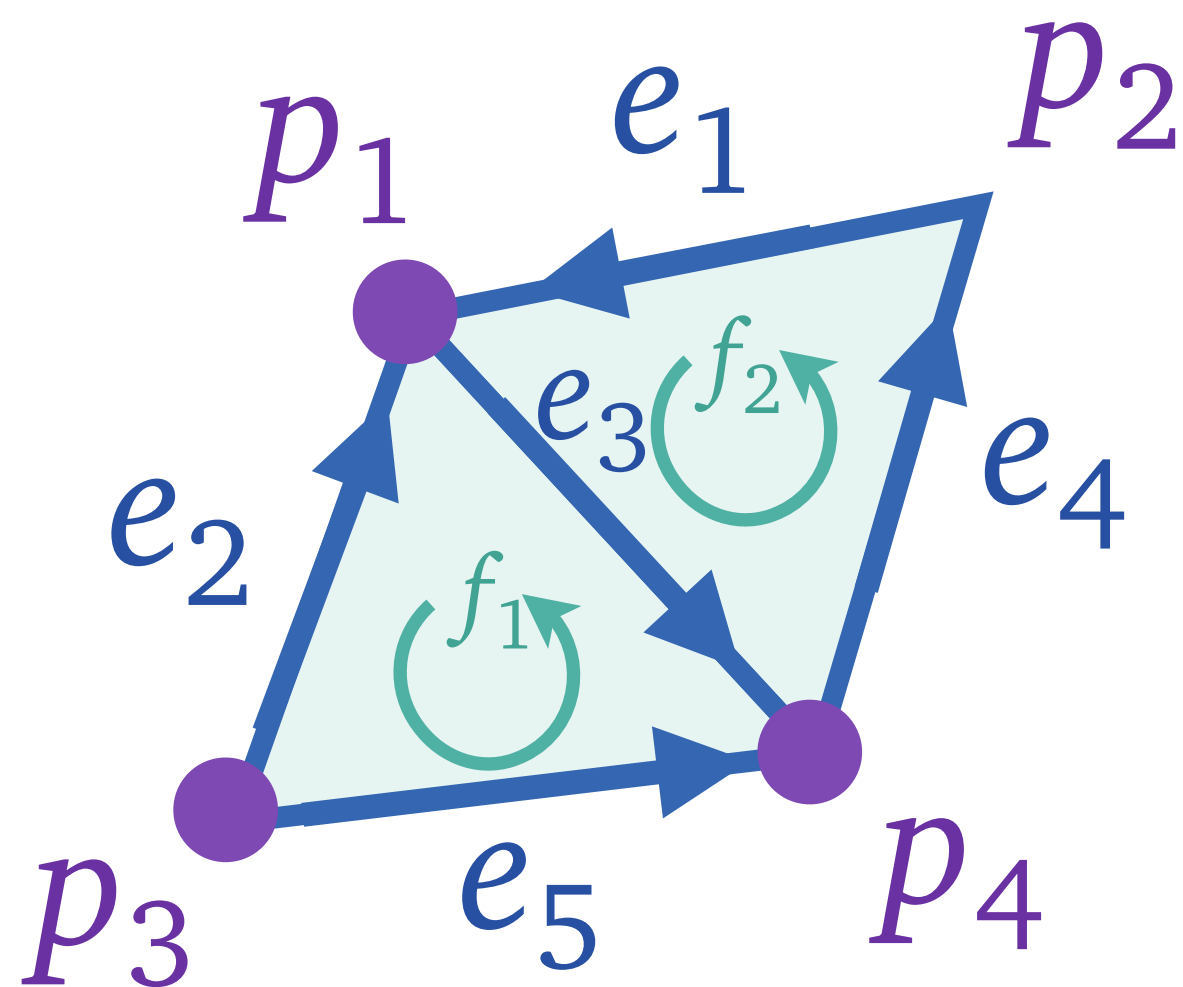
$$\partial_k : C_k \xrightarrow{\text{linear}} C_{k-1}$$

$$\partial_k (a_1 \sigma_1 + \dots + a_m \sigma_m) := a_1 \partial_1 \sigma_1 + \dots + a_m \partial_m \sigma_m$$

# Boundary matrix

- Express the boundary linear maps as matrices

$$[\partial_1] = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 1 \end{bmatrix} \quad [\partial_2] = \begin{bmatrix} 0 & 1 \\ -1 & 0 \\ -1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$



$$\partial_1(e_3 + e_4) = -\mathbf{p}_1 + \mathbf{p}_2$$

$$[\partial_1] \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

# Boundary matrix

- Express the boundary linear maps as matrices

$$[\partial_1] = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 1 \end{bmatrix} \quad [\partial_2] = \begin{bmatrix} 0 & 1 \\ -1 & 0 \\ -1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

- $[\partial_1][\partial_2] = [\mathbf{0}]$

# Chain complex

- Boundary operators are linear maps connecting the sequence of chain spaces

$$\cdots \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} \cdots$$

with the property that consecutive maps compose into the zero map

$$\partial_k \partial_{k+1} = 0$$

- In general, such a sequence of linear spaces and linear maps is a category called a **chain complex**.
- In a chain complex, we can define notions of “closedness” and “exactness.”

# Chain complex: Closed v.s. Exact

- Consider a chain complex

$$\cdots \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} \cdots \quad \partial_k \partial_{k+1} = 0$$

- A chain  $\gamma \in C_k$  is called **closed** if it has vanishing boundary  $\partial \gamma = 0$
- A chain  $\gamma \in C_k$  is called **exact** if it is the boundary of some chain:

$$\gamma = \partial \beta \quad \beta \in C_{k+1}$$

- Exact chains are closed  $\text{im}(\partial_{k+1}) \subset \ker(\partial_k)$
- Closed chains may not be exact.

# Chain complex (summary)

- A  $k$ -chain is a linear combination of  $k$ -dimensional cells

$$C_k = \text{span } M_k$$

- Boundary operators are linear maps between chains

$$\cdots \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} \cdots \quad \partial_k \partial_{k+1} = 0$$

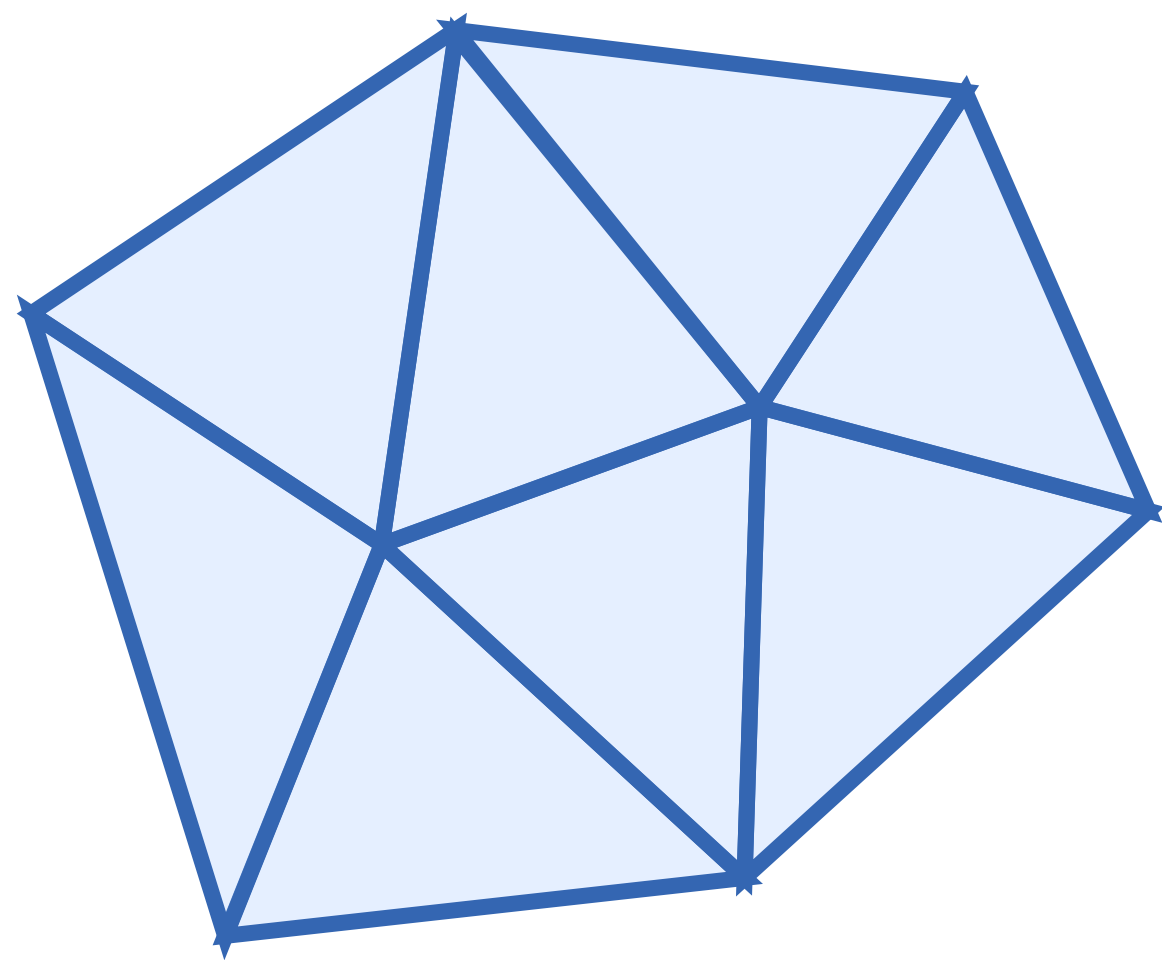
- Such a sequence of linear maps with the zero-consecutive-composition property is called a chain complex

- Closed chains  $\ker(\partial_k)$
- Exact chains  $\text{im}(\partial_{k+1})$
- $\text{im}(\partial_{k+1}) \subset \ker(\partial_k)$

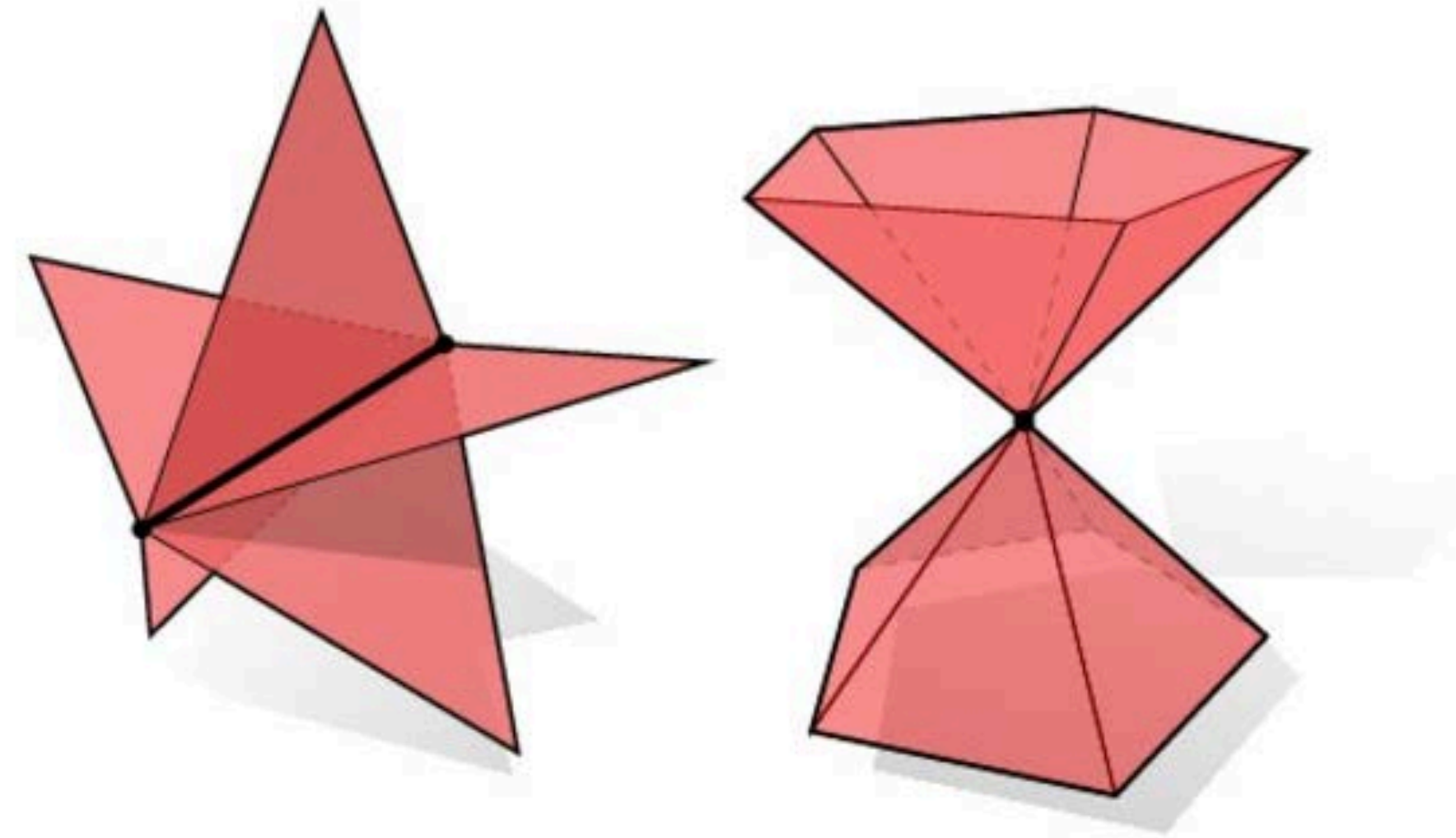
# Manifold Surface Mesh

- Simplicial complex
- Chain complex
- **Manifold surface mesh**
- Surface topology
- Euler characteristics

# Manifold mesh



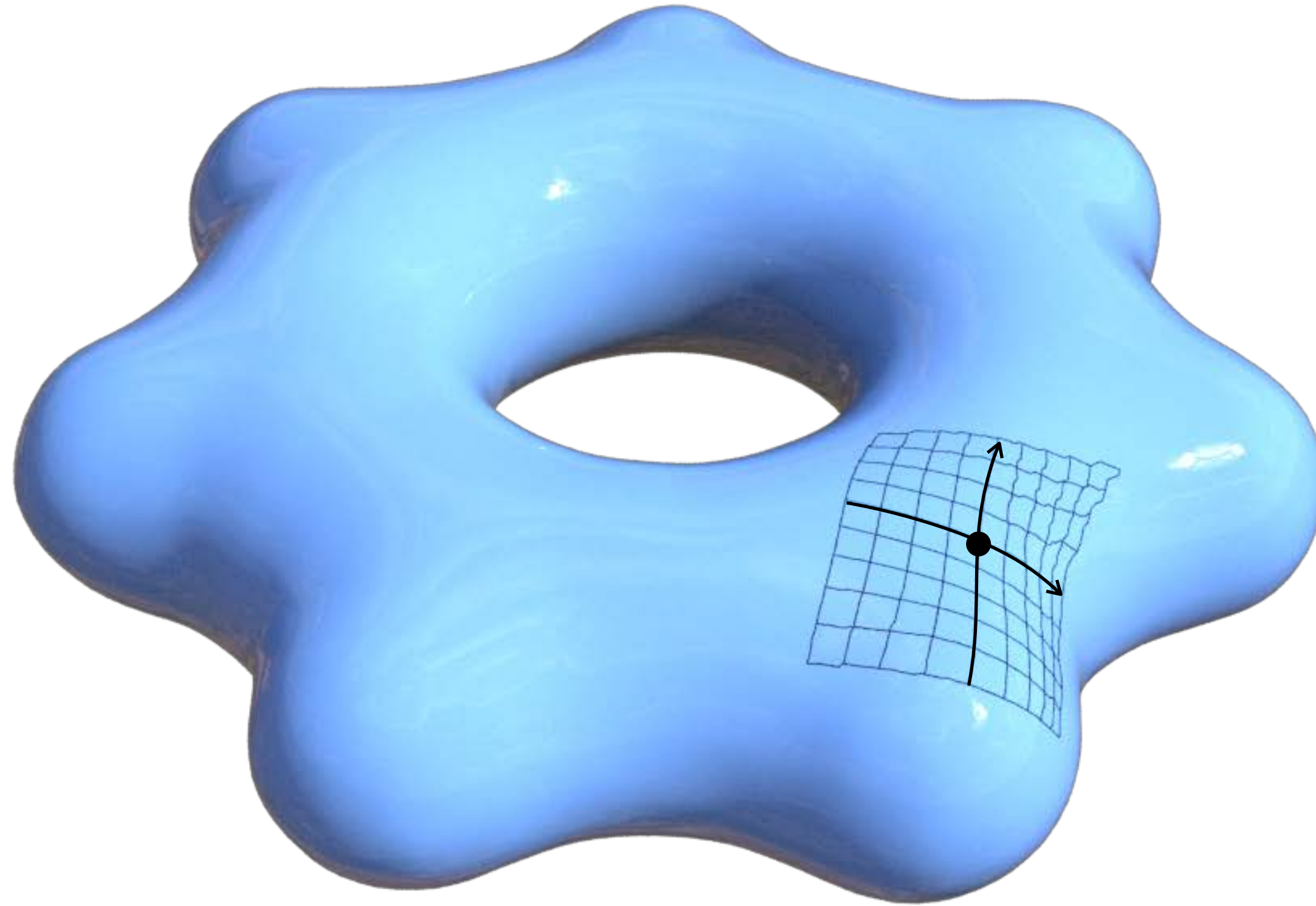
YES



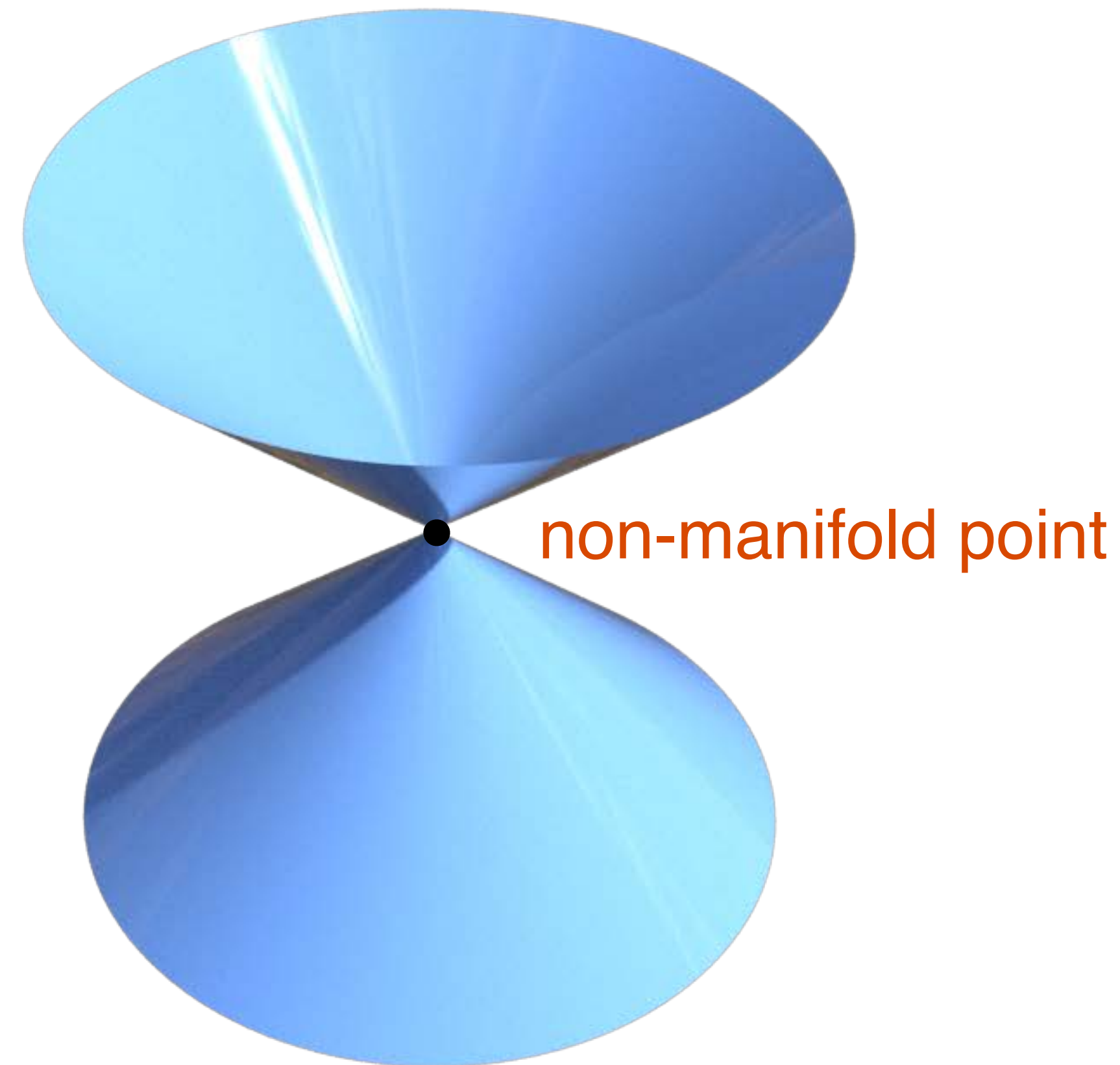
NO

# Manifold mesh

- Mathematically, an  $n$ -dimensional **manifold** is a space that can be continuously parametrized by  $n$  parameters around every point.



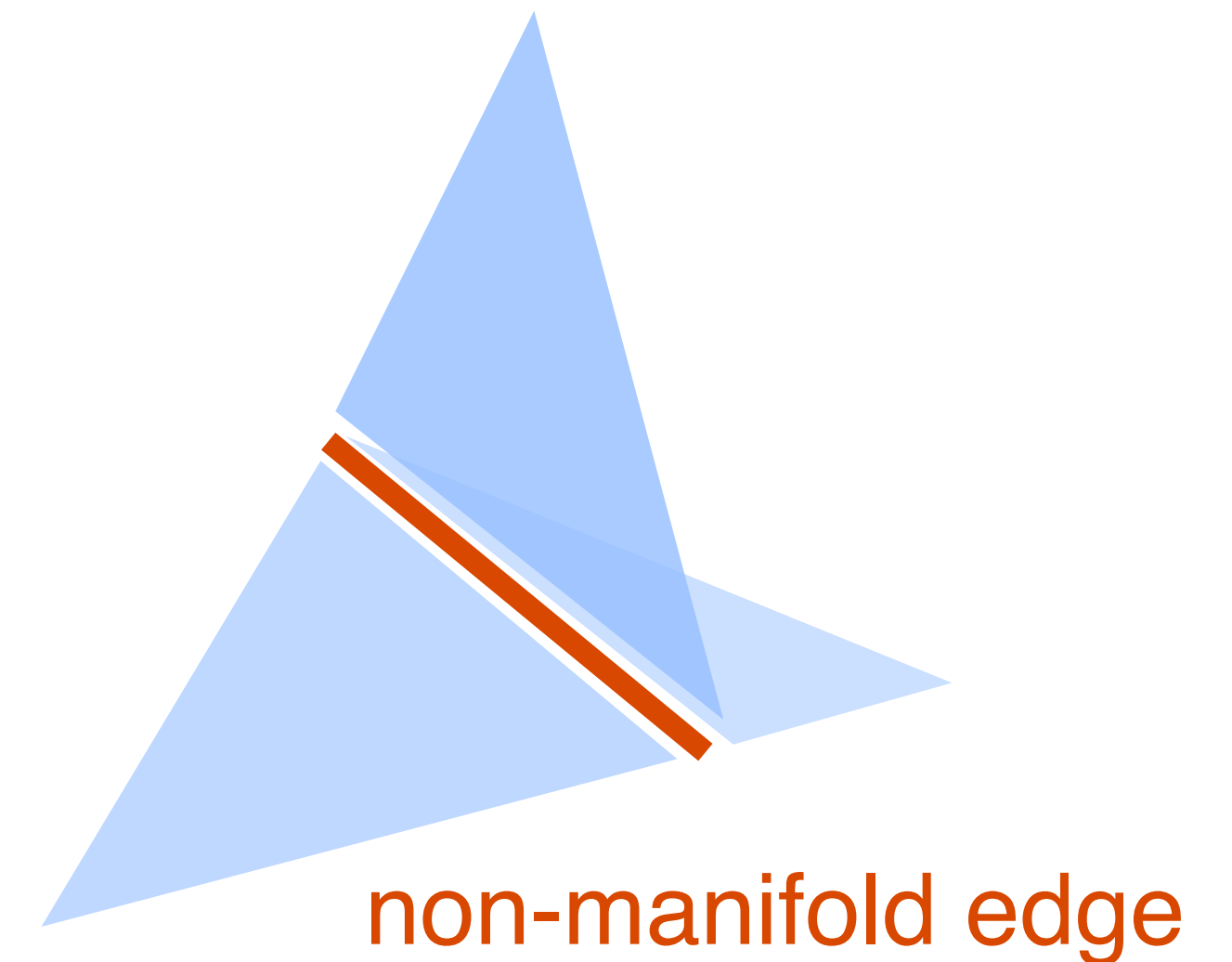
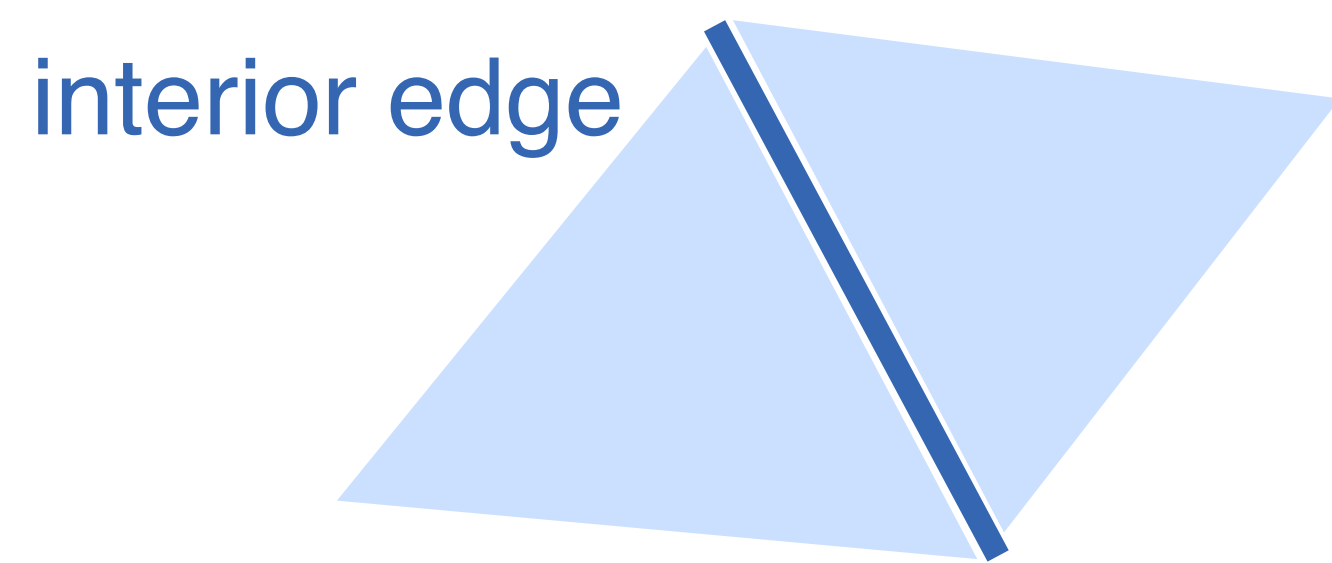
Locally parametrizable by 2 parameters



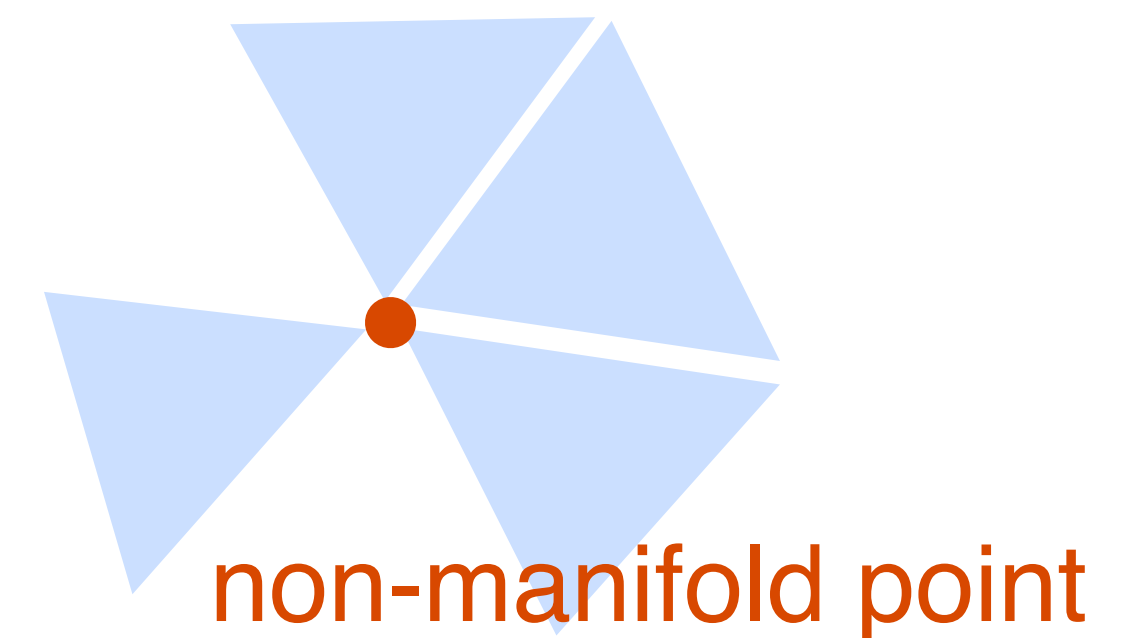
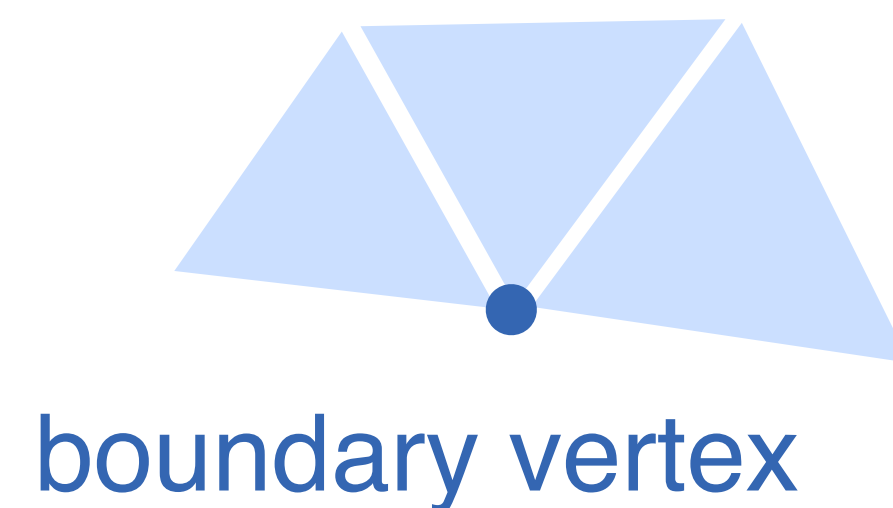
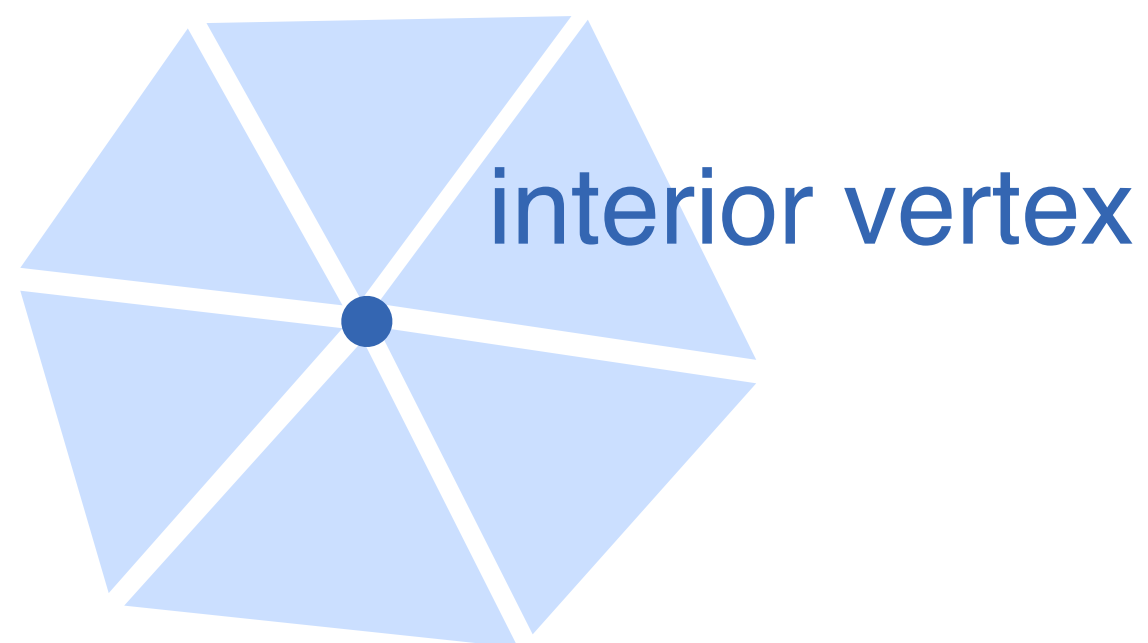
# Manifold mesh

- A triangle mesh  $M = (V, E, F)$  is a **manifold mesh** if

- ▶ Every edge is incident to 2 or 1 faces

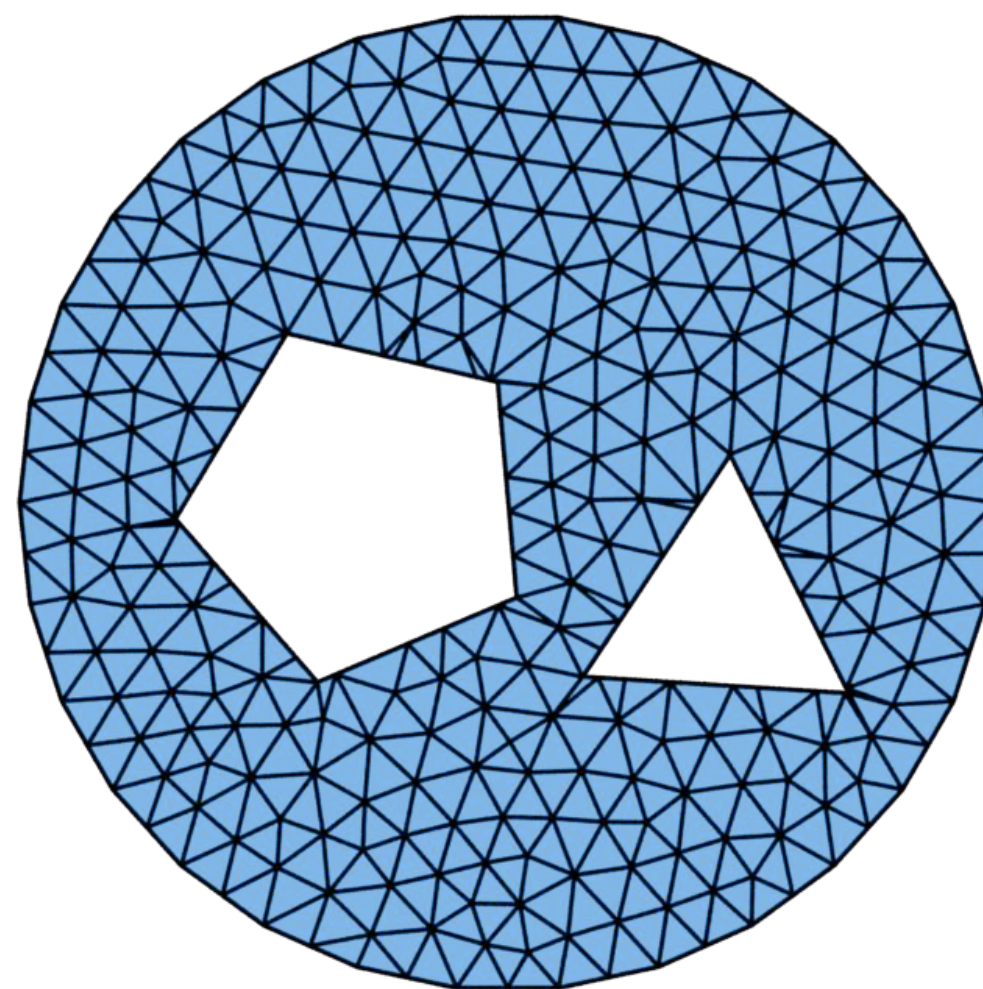


- ▶ Every vertex has incident faces forming a closed fan or an open fan

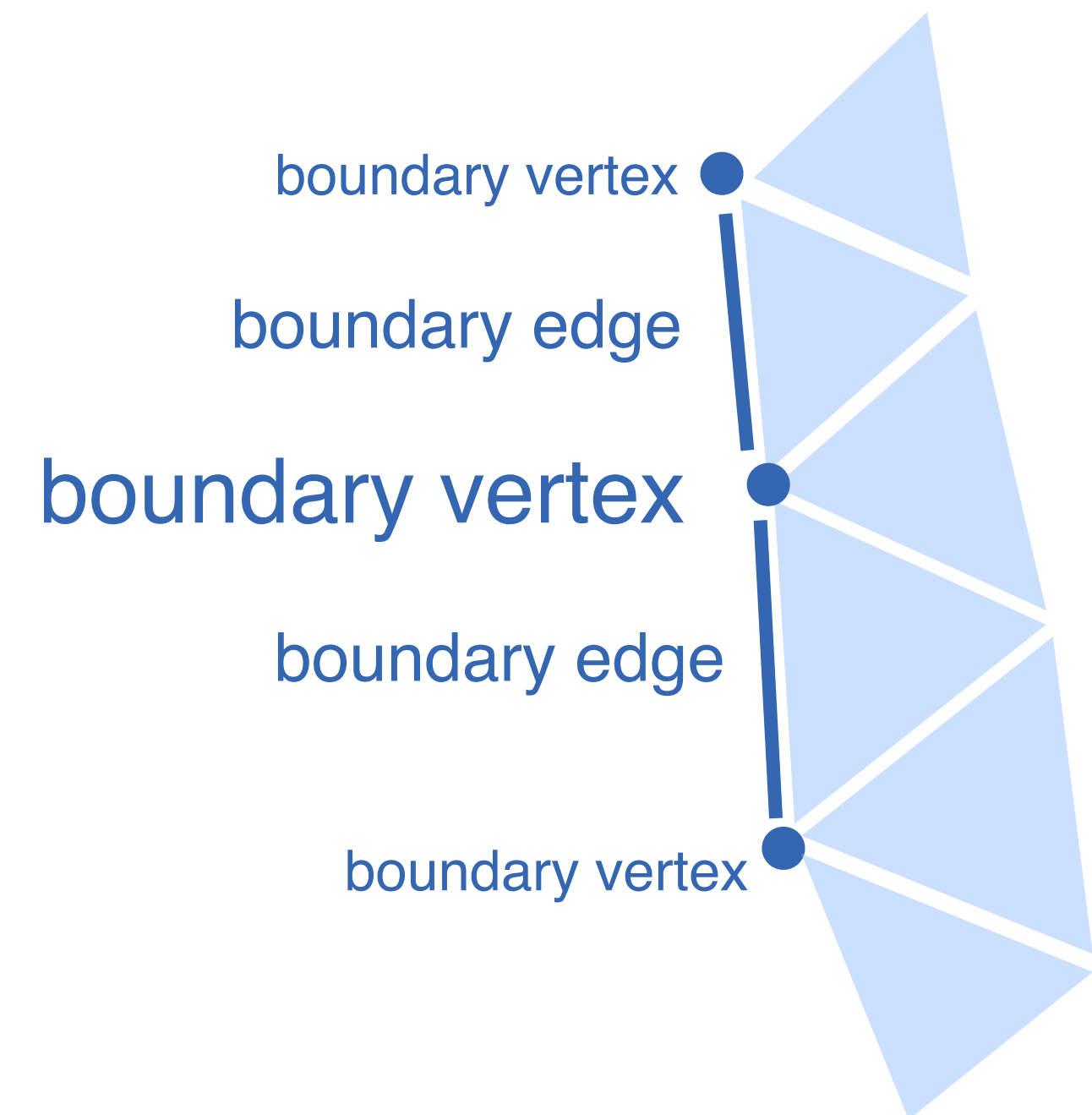


# About boundary

- Every boundary vertex is incident to two boundary edges
- Every boundary edge connects two boundary vertices
- Boundary vertices and edges form “doubly linked list(s)”
- A boundary component is a cycle of this list

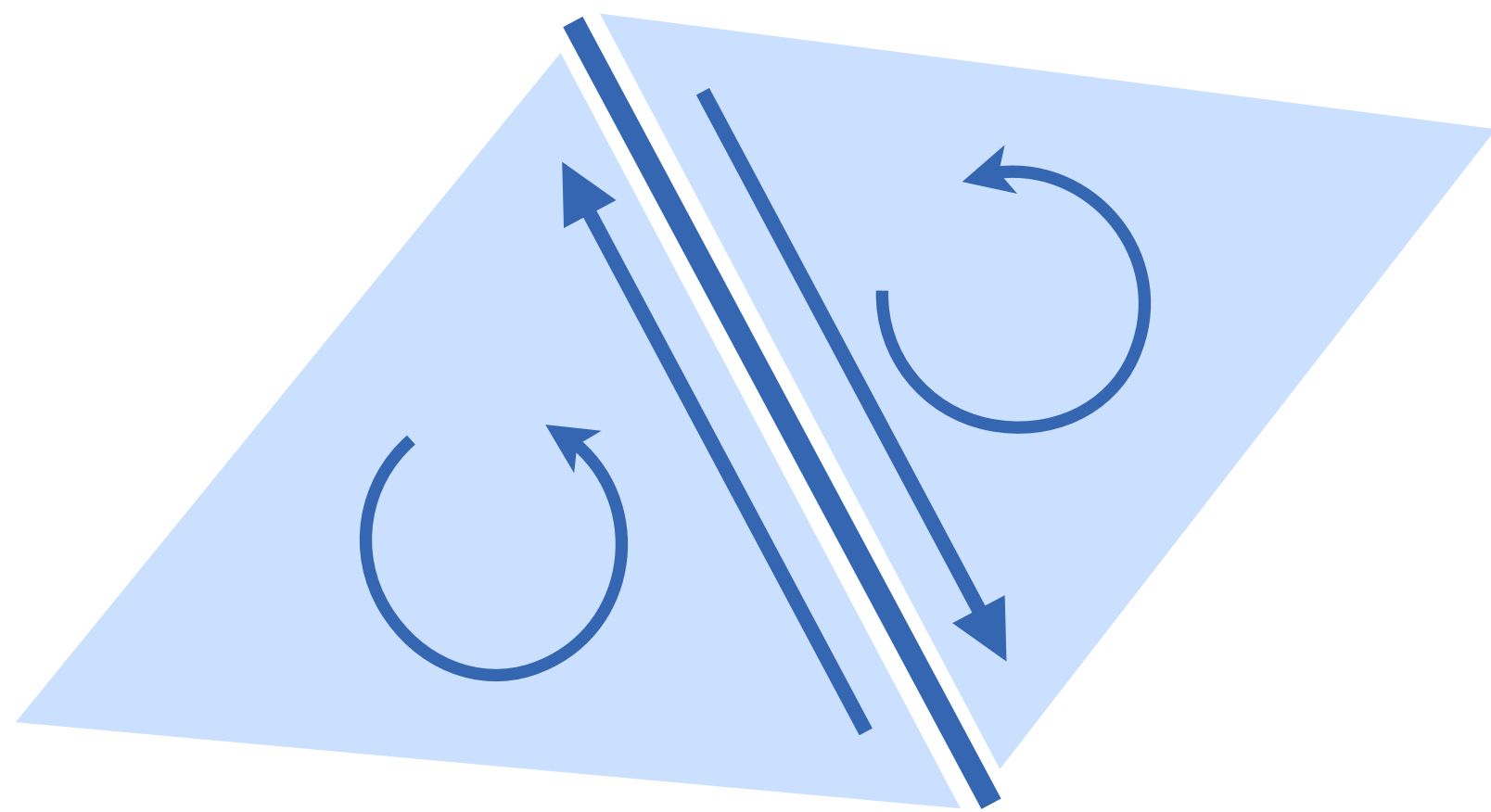


3 boundary components

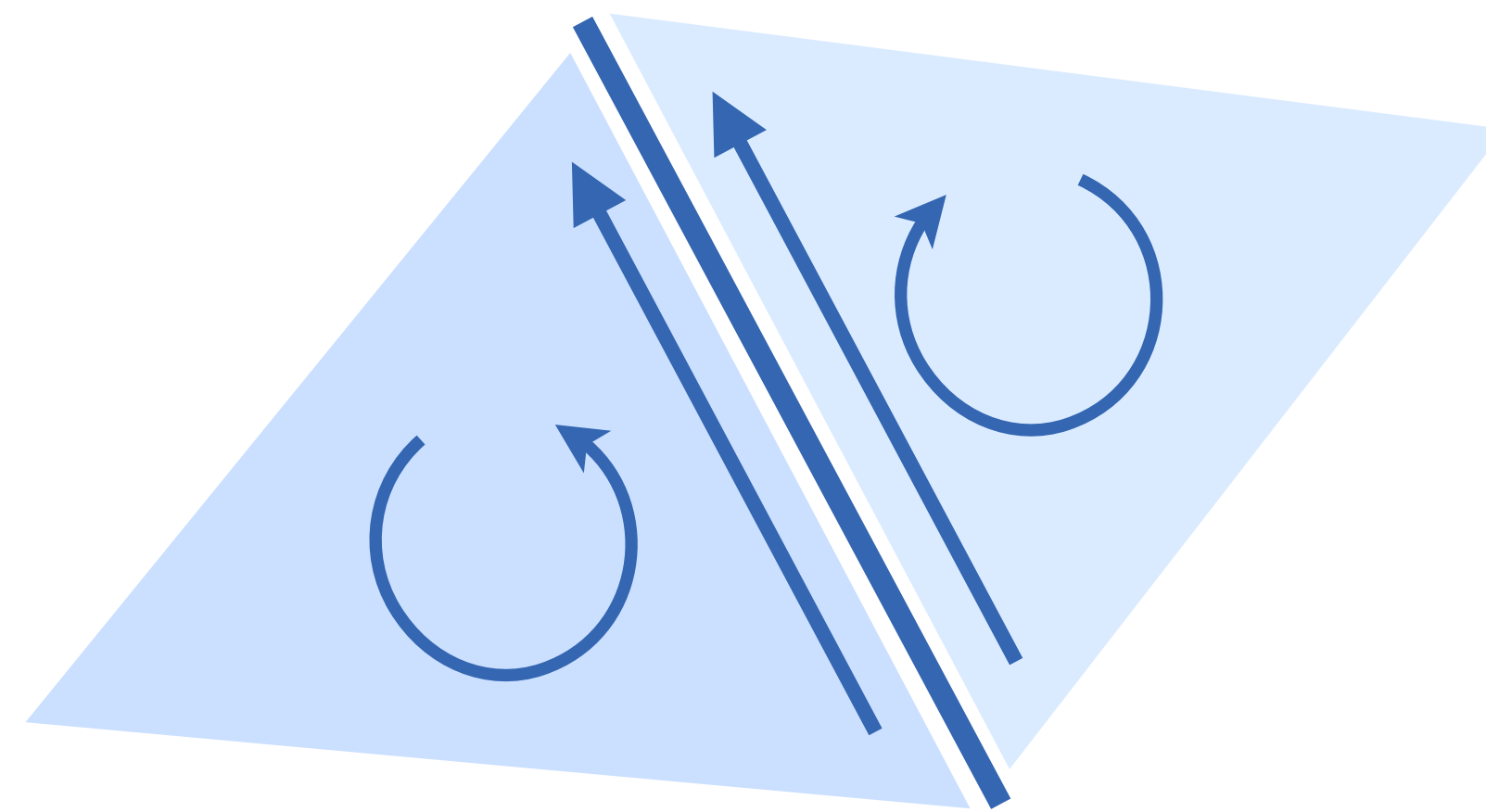


# About orientability

- Every manifold triangle mesh is either **orientable** or **non-orientable**
- An oriented mesh is to have all triangles (top-dimensional simplices) oriented so that the oriented boundary cancels out on shared interior edges



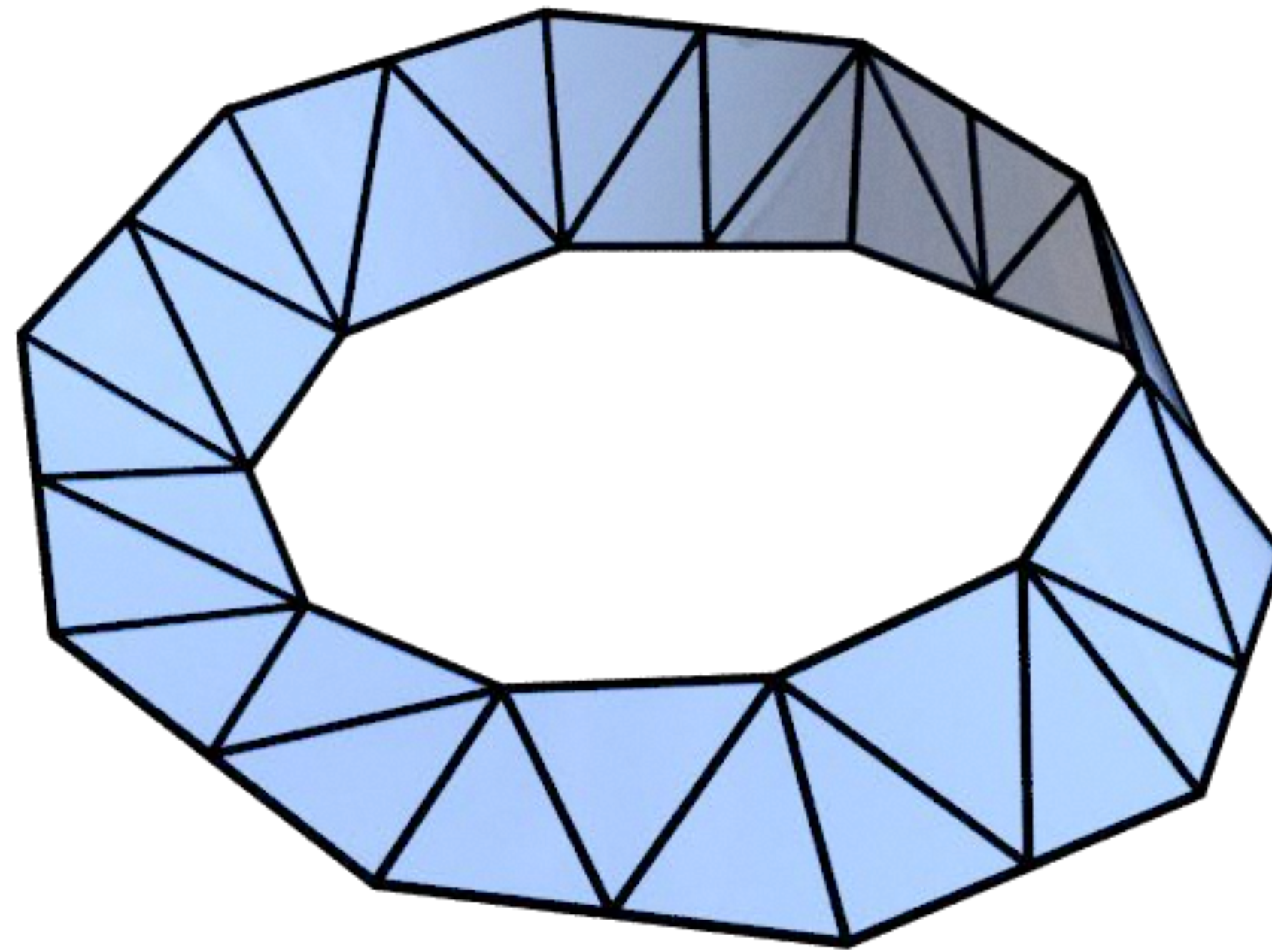
consistently oriented



not consistently oriented

# About orientability

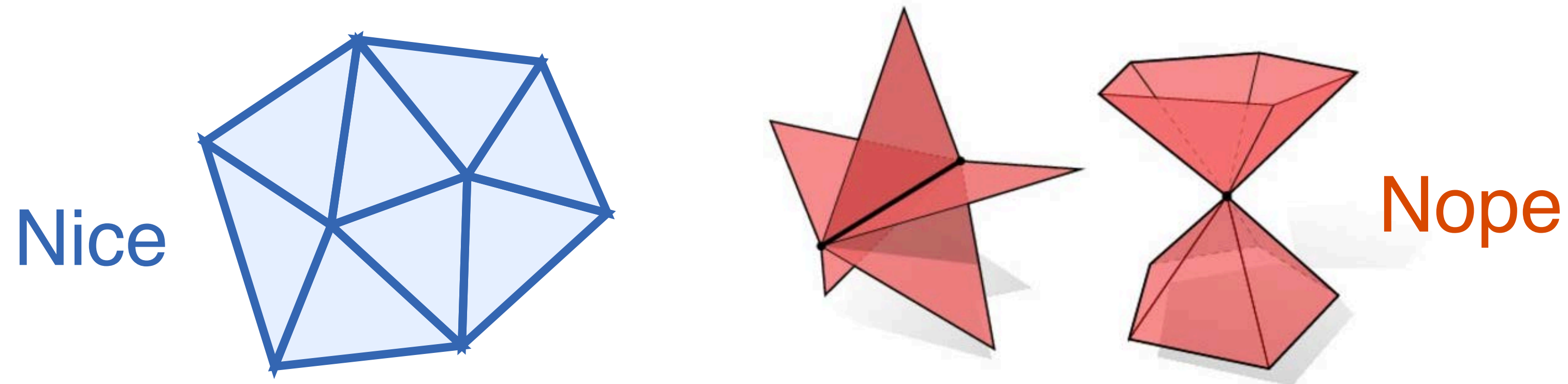
- Every manifold triangle mesh is either **orientable** or **non-orientable**



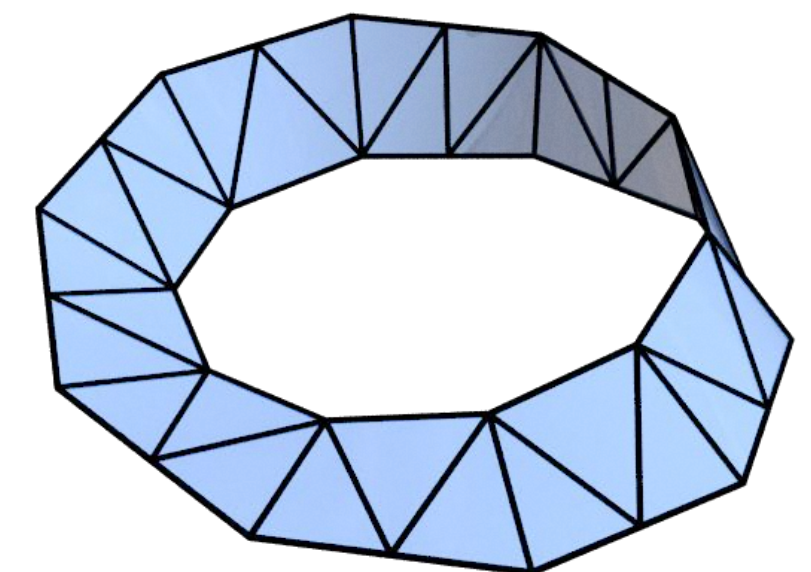
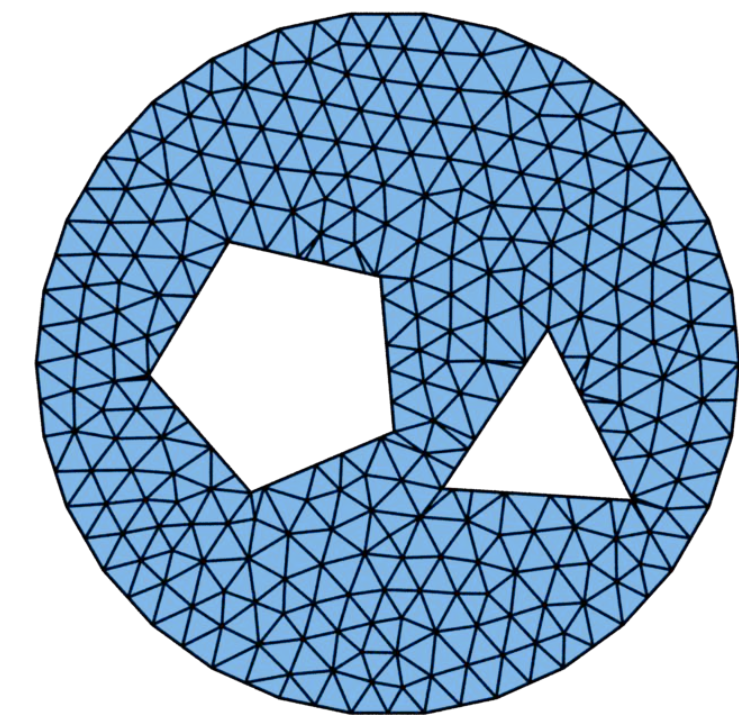
non-orientable

# Manifold mesh (summary)

- Discrete surfaces are manifold triangle meshes



- A manifold mesh may have a few boundaries and we can count the number of boundary components
- A manifold mesh may be oriented or non-orientable. We'll consider oriented surface most of the time.

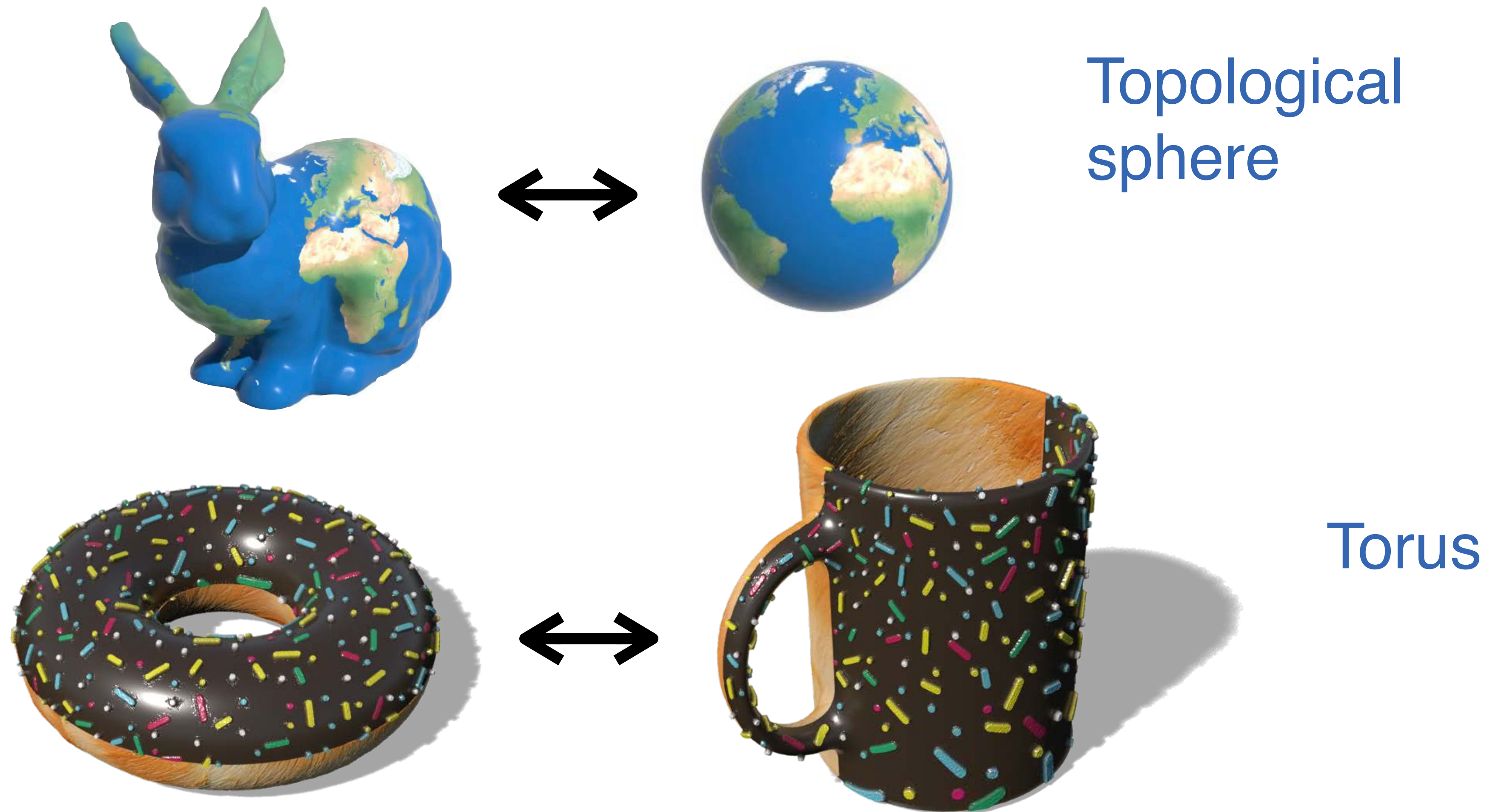


# Surface Topology

- Simplicial complex
- Chain complex
- Manifold surface mesh
- Surface topology
- Euler characteristics

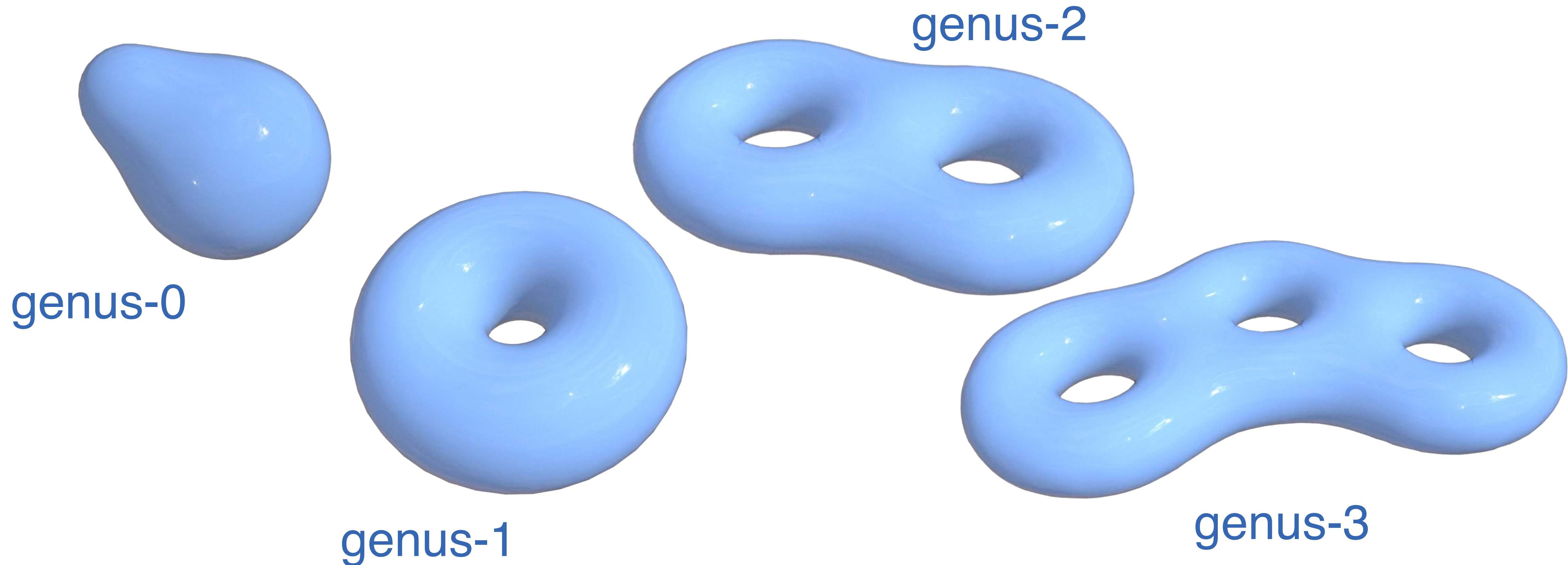
# Topology of surfaces

- Two surfaces are said to have the “same topology” (*homeomorphic*) if there exists continuous map between them (in both directions)



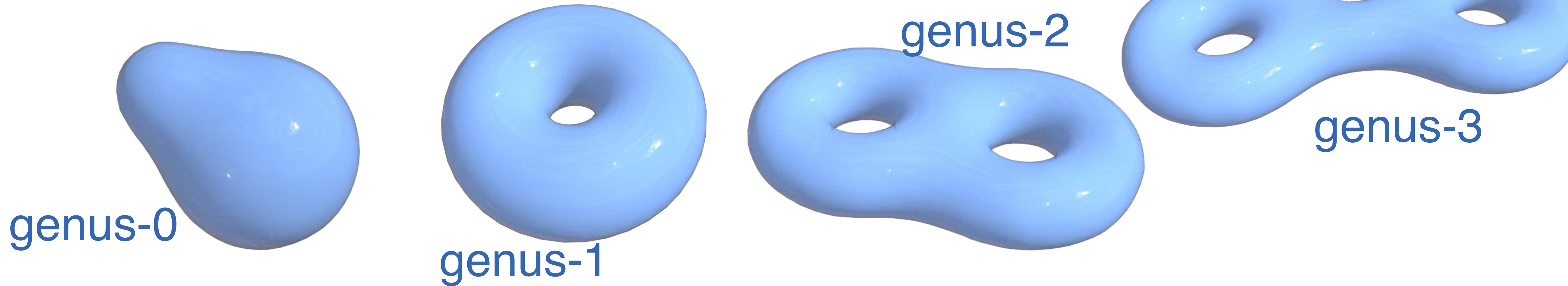
# Topology of surfaces

- Genus (  $g$  ) describes number of holes



# Topology of surfaces

- Genus (  $g$  ) describes number of holes

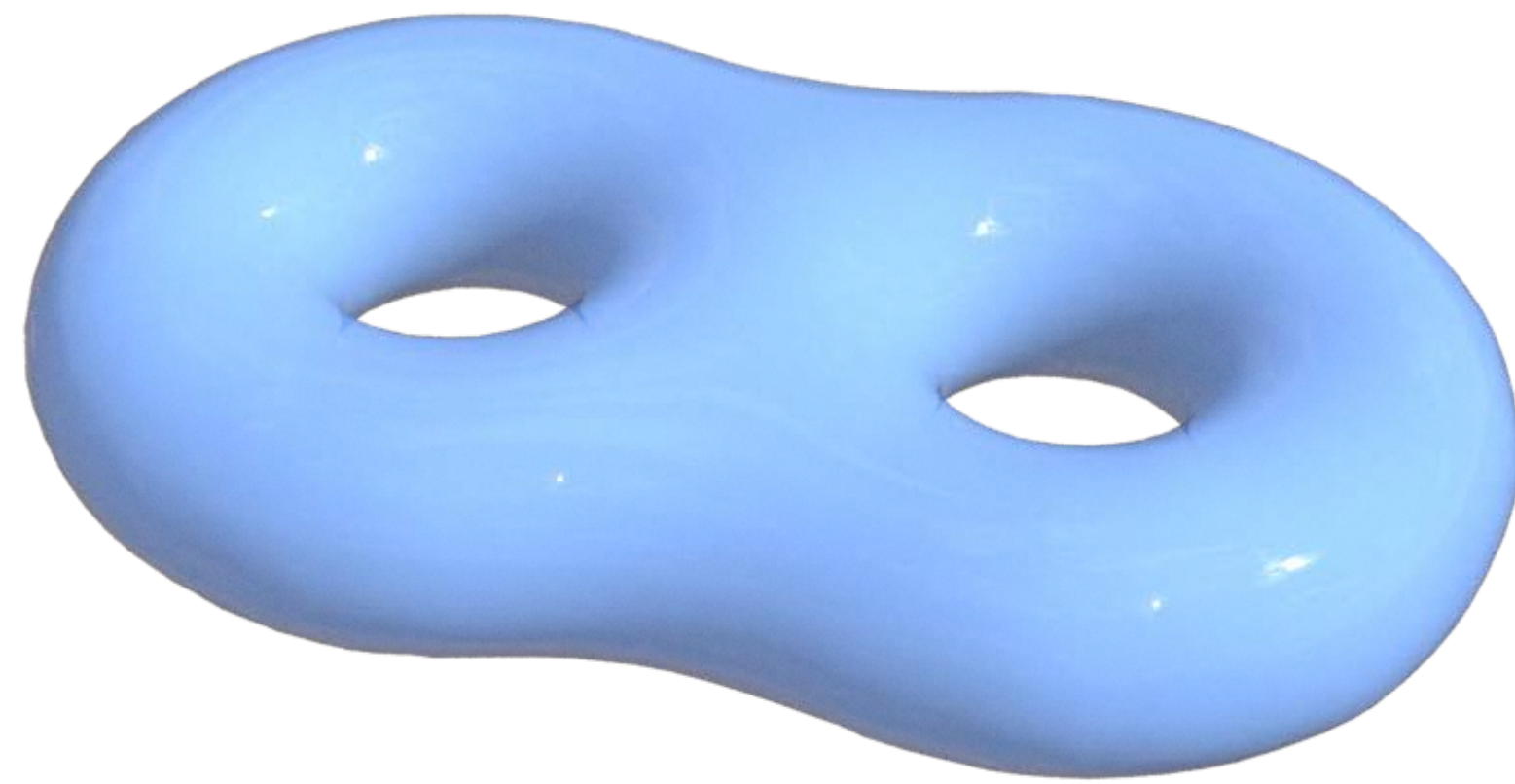


## Definition

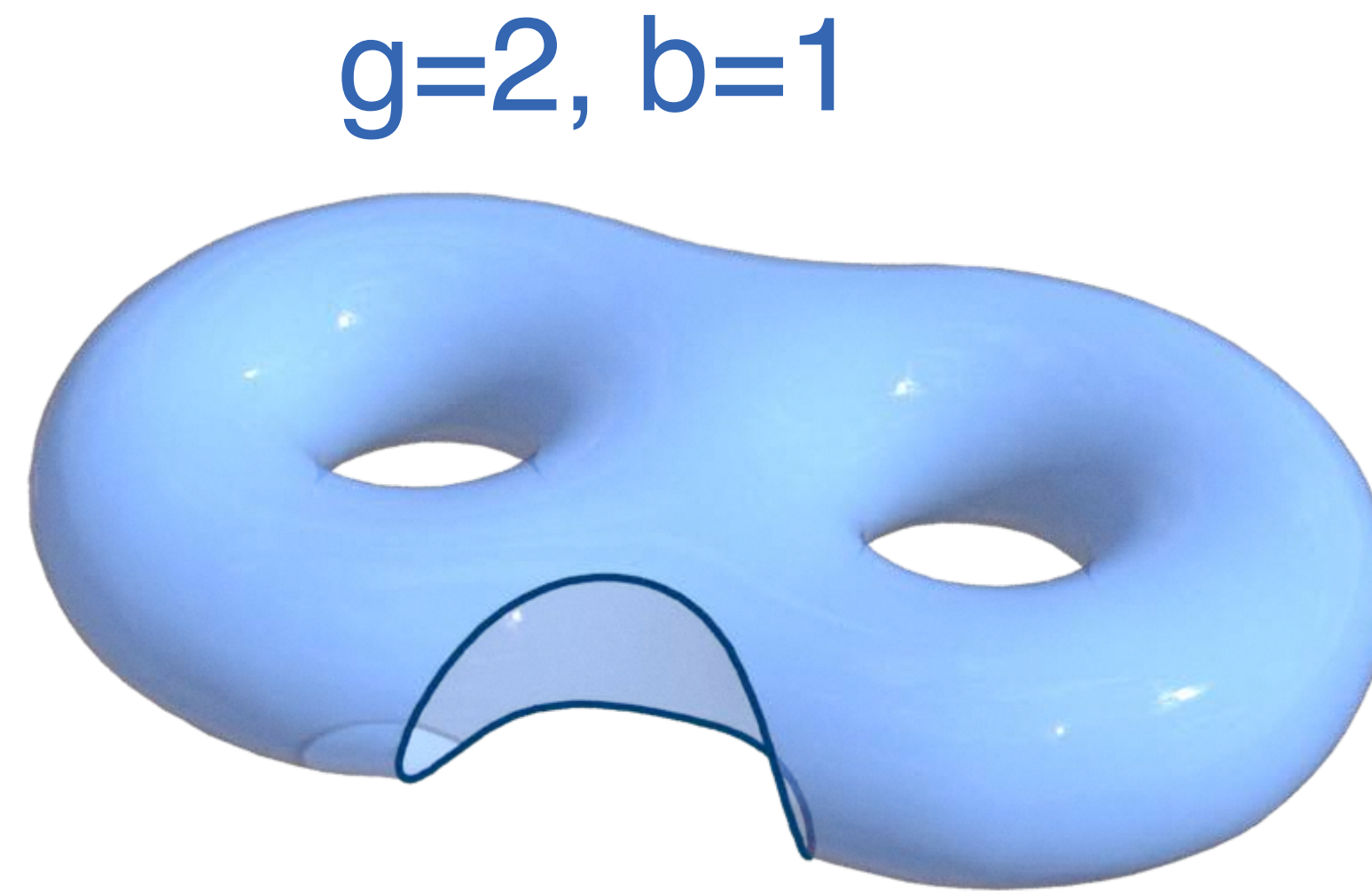
The *genus* of a connected orientable surface is the maximal number of cuttings along non-intersecting closed curves without making the surface disconnected.

# Topology of surfaces

- A genus  $g$  surface with  $b$  boundary components



$g=2, b=0$



$g=2, b=1$



$g=2, b=4$

- Two surfaces with different  $(g,b)$  profile cannot be continuously mapped to each other
- that is, same topology must have same  $(g,b)$  profile

# Classification of oriented surface

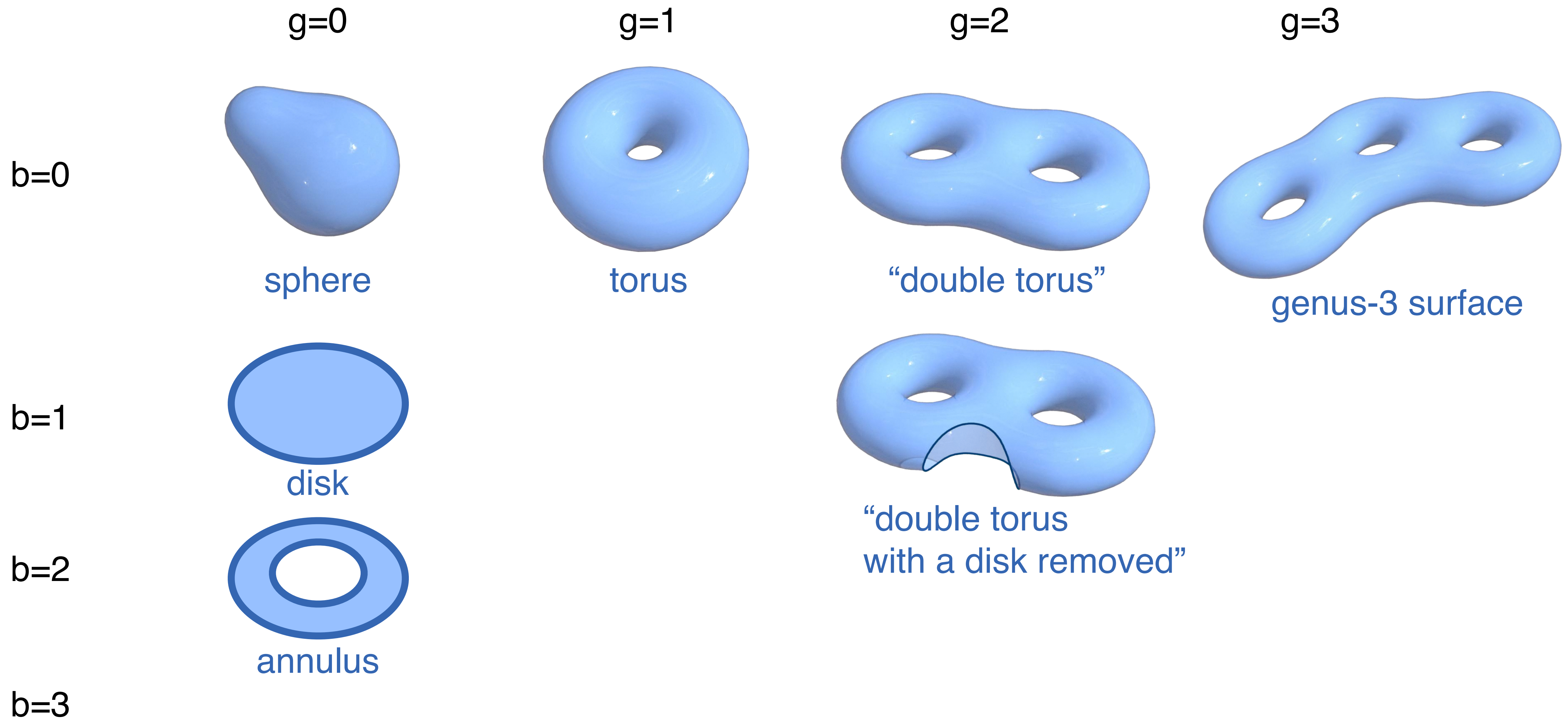
- **Theorem**

*Two orientable surfaces have the same topology if and only if they share the same genus and number of boundaries*

We only need to know genus  $g$  and number of boundaries  $b$  to describe the topological type of an oriented surface.

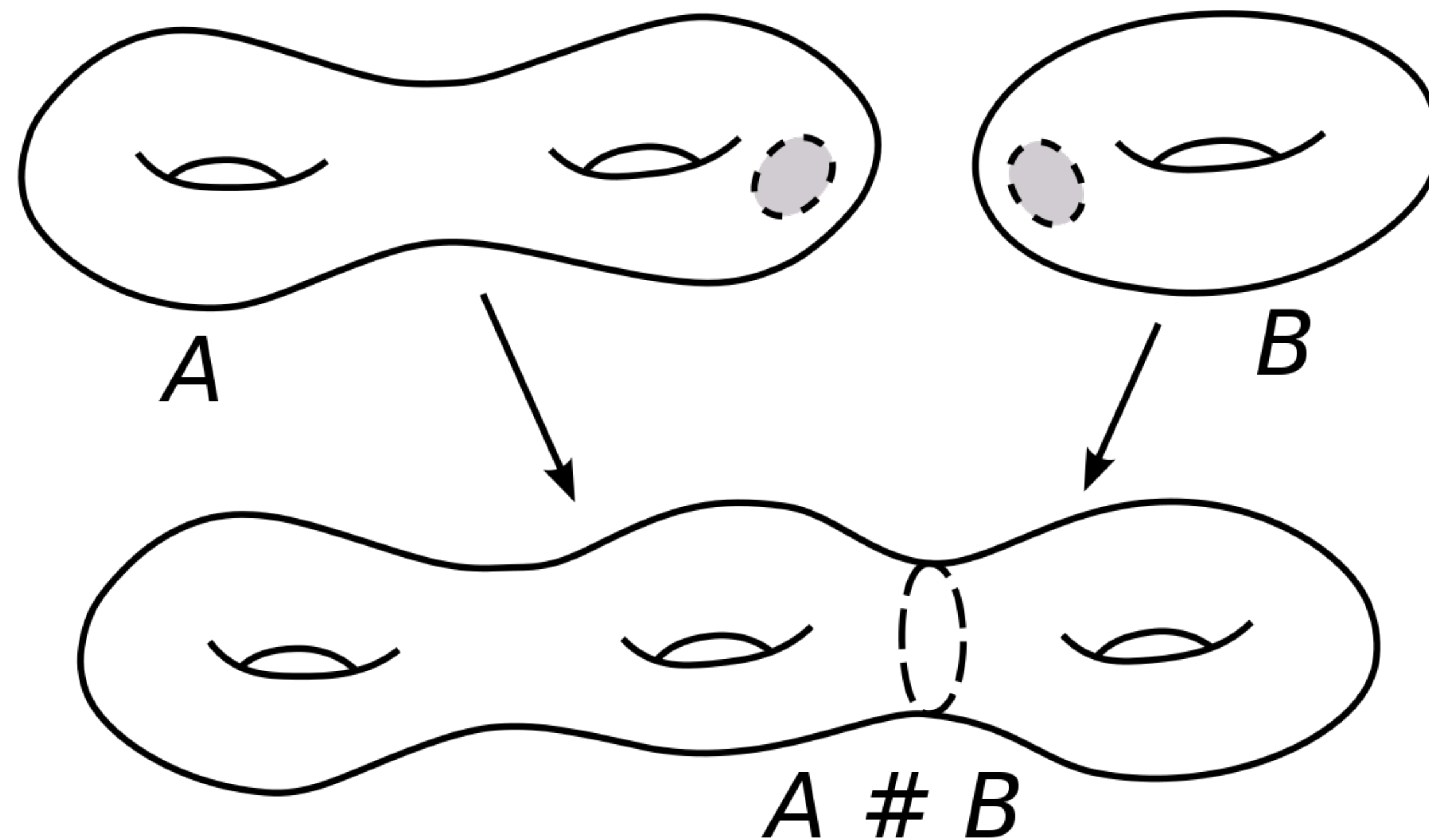
Use  $(g,b)$  to build the “periodic table” of surfaces

# Classification of oriented surface



# Fusion

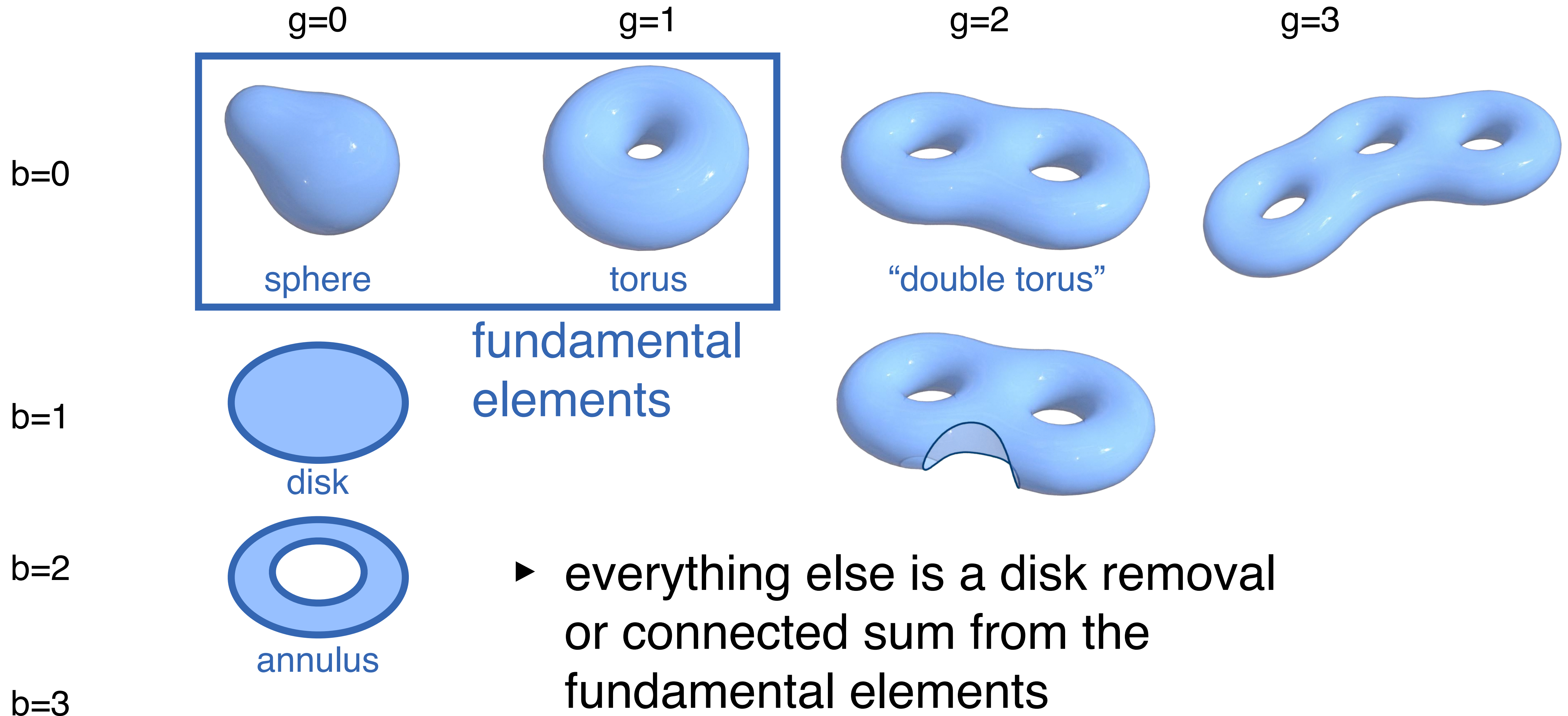
- There is one operation which we often use to describe change of topology: **connected sum**



- The  $g$  and  $b$  components just add up during the connected sum

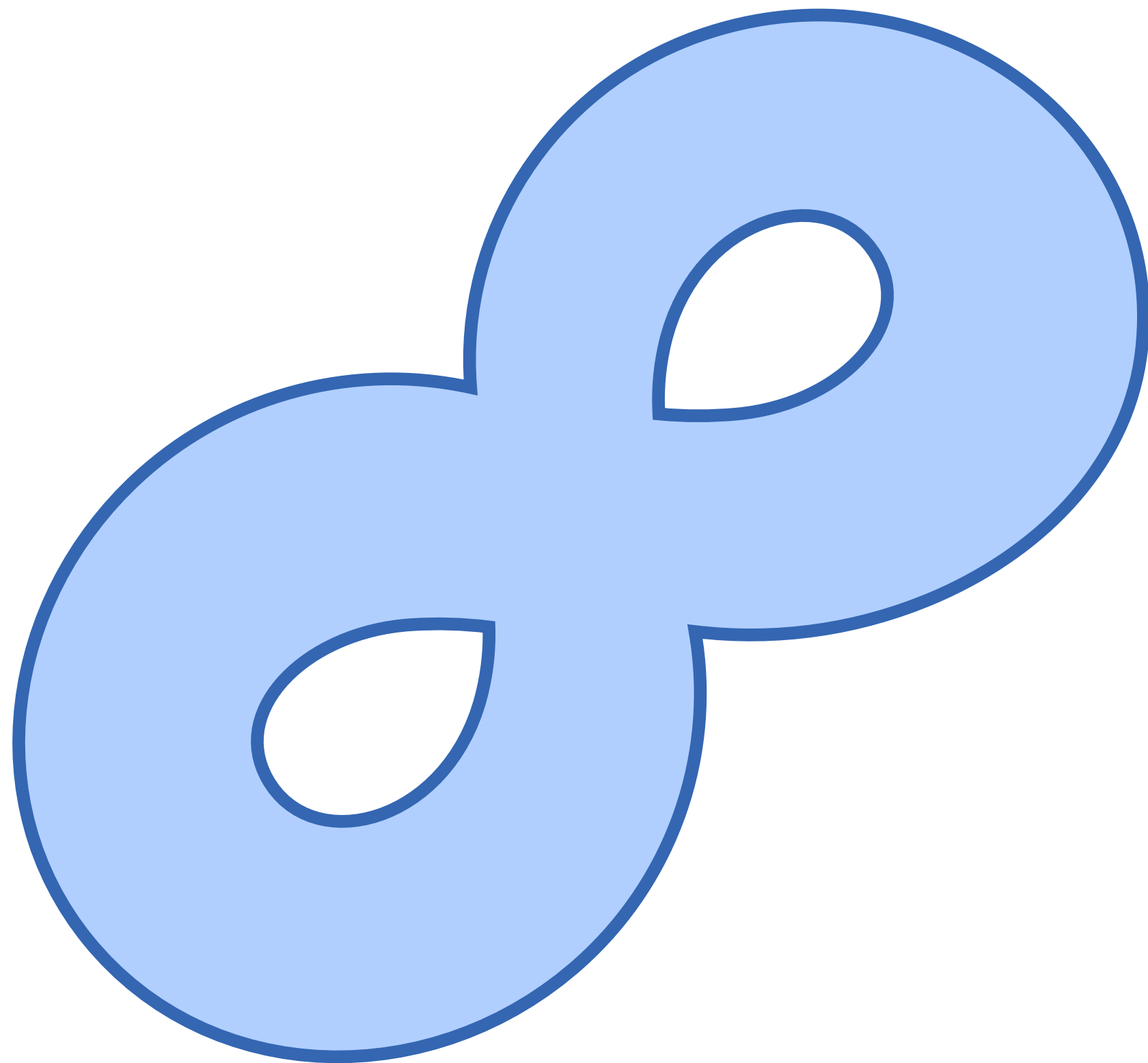
$$(g_{A\#B}, b_{A\#B}) = (g_A + g_B, b_A + b_B)$$

# Classification of oriented surface

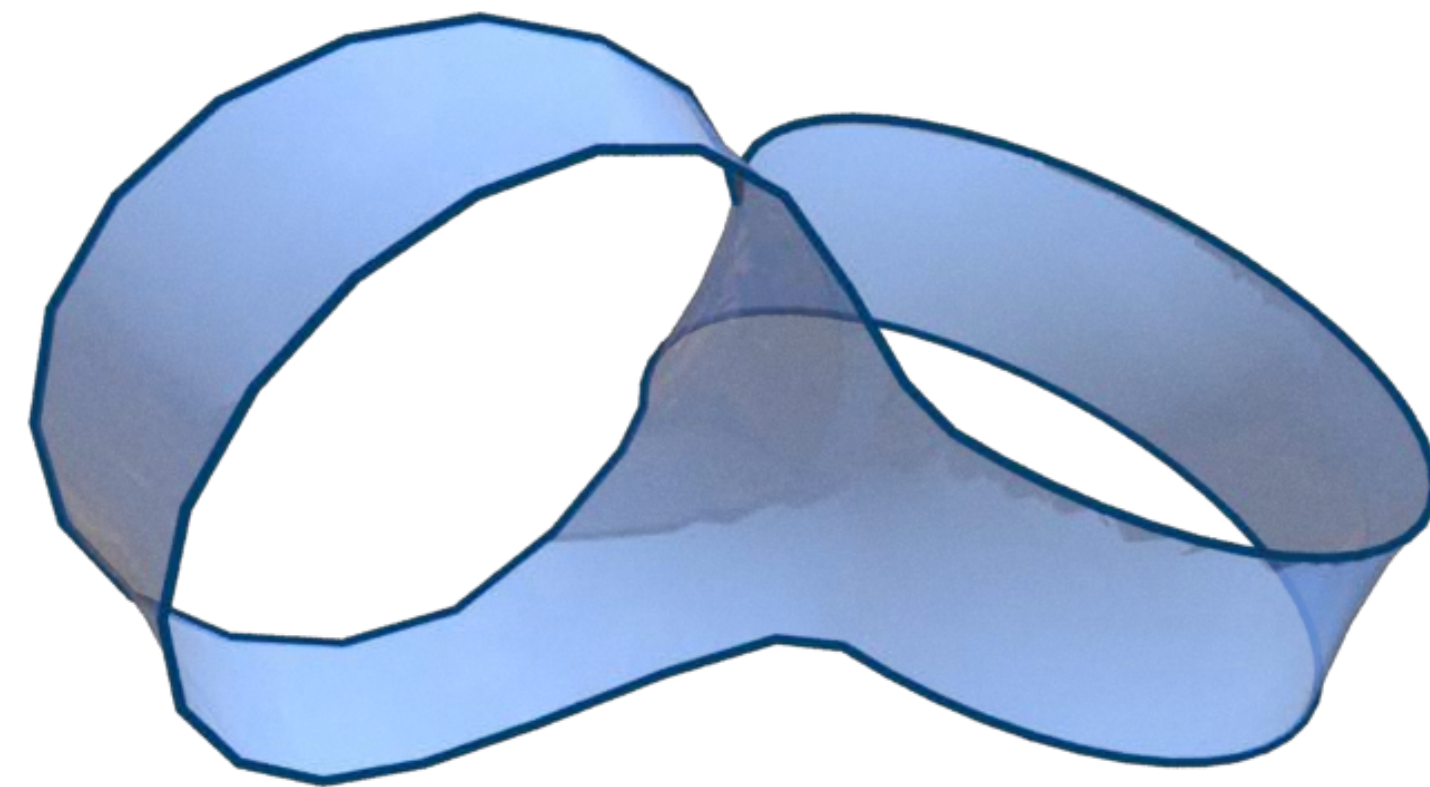


# What are these?

figure-8 ribbon glued at the crossing

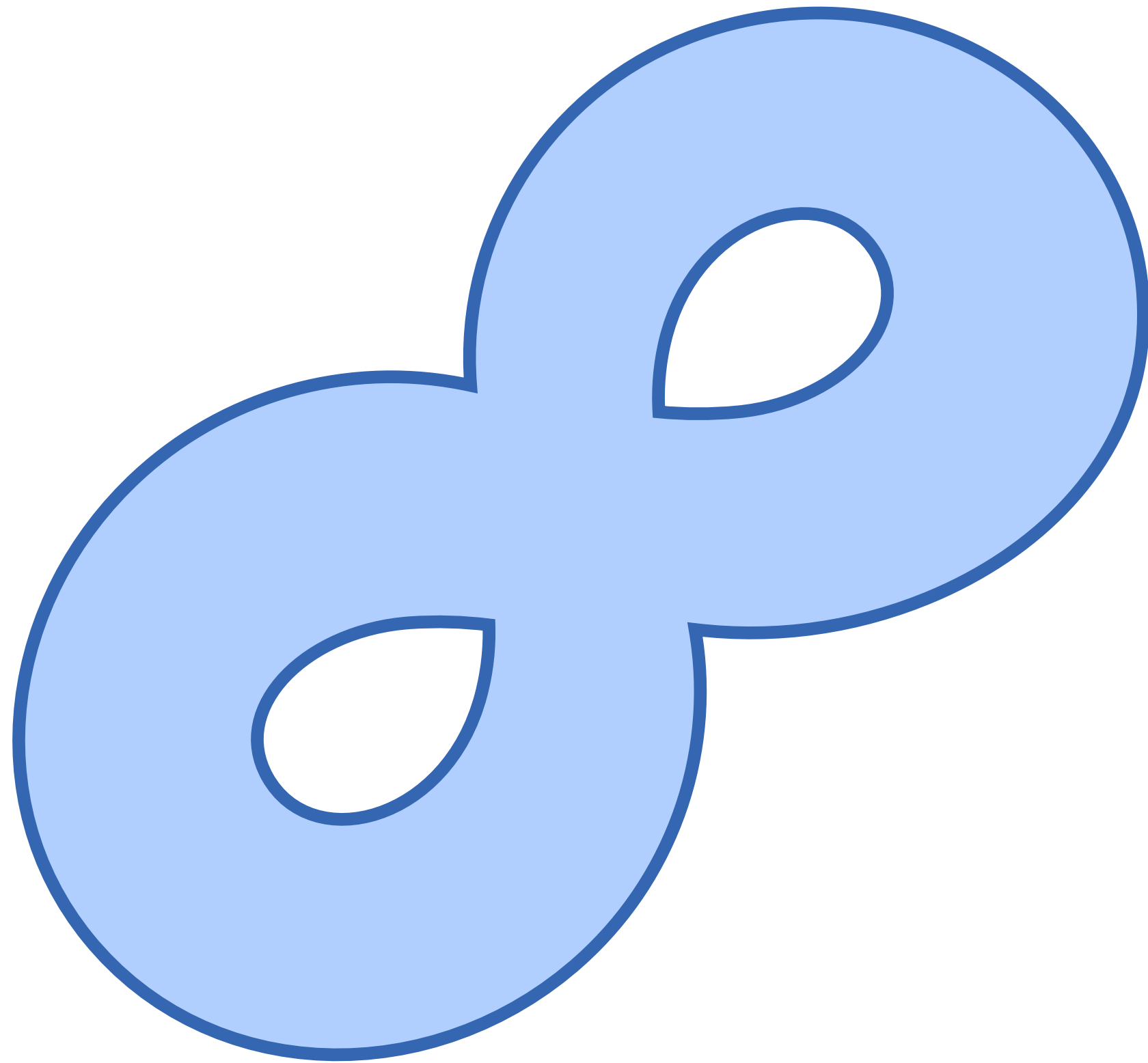


two ribbons glued together



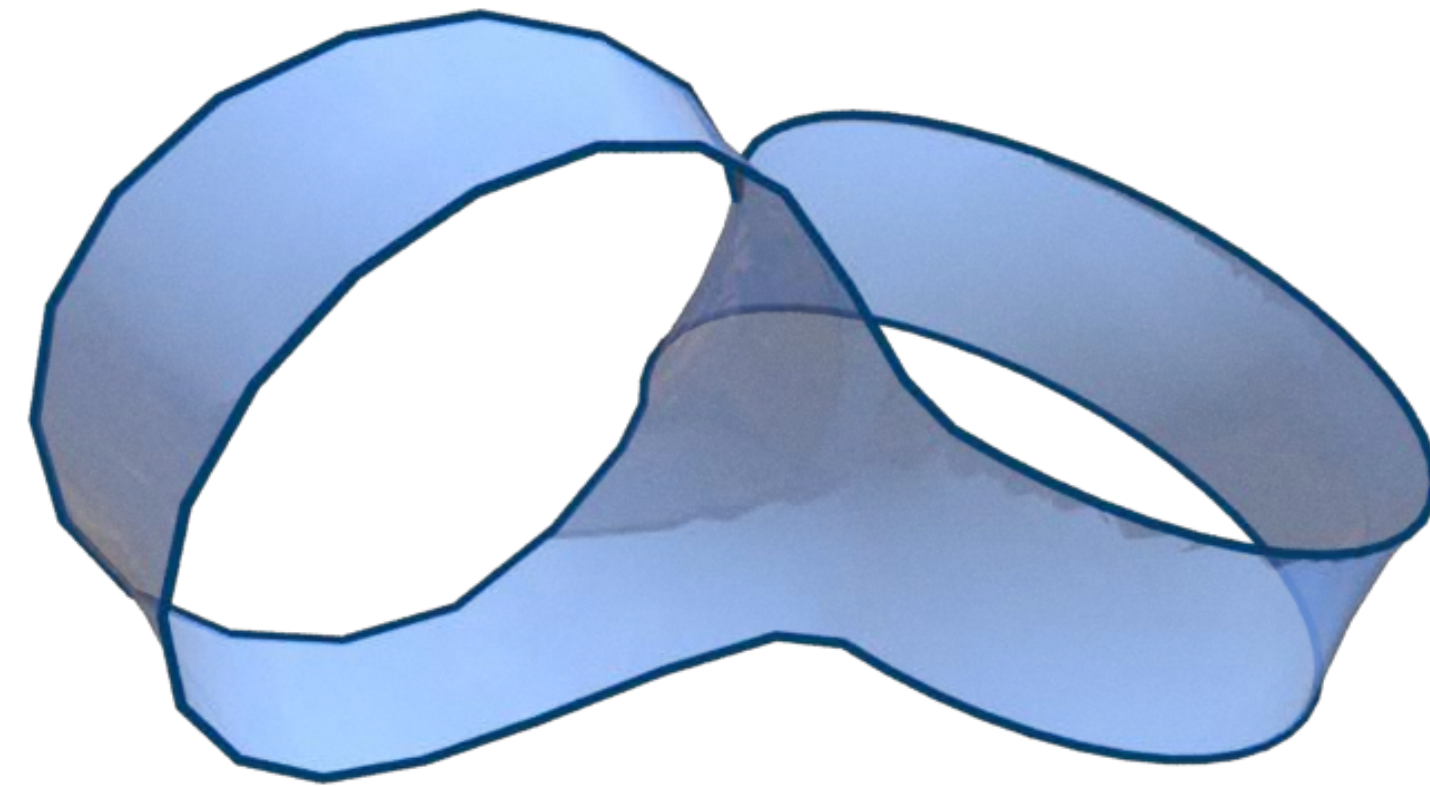
# What are these?

figure-8 ribbon glued at the crossing



$g=0, b=3$

two ribbons glued together



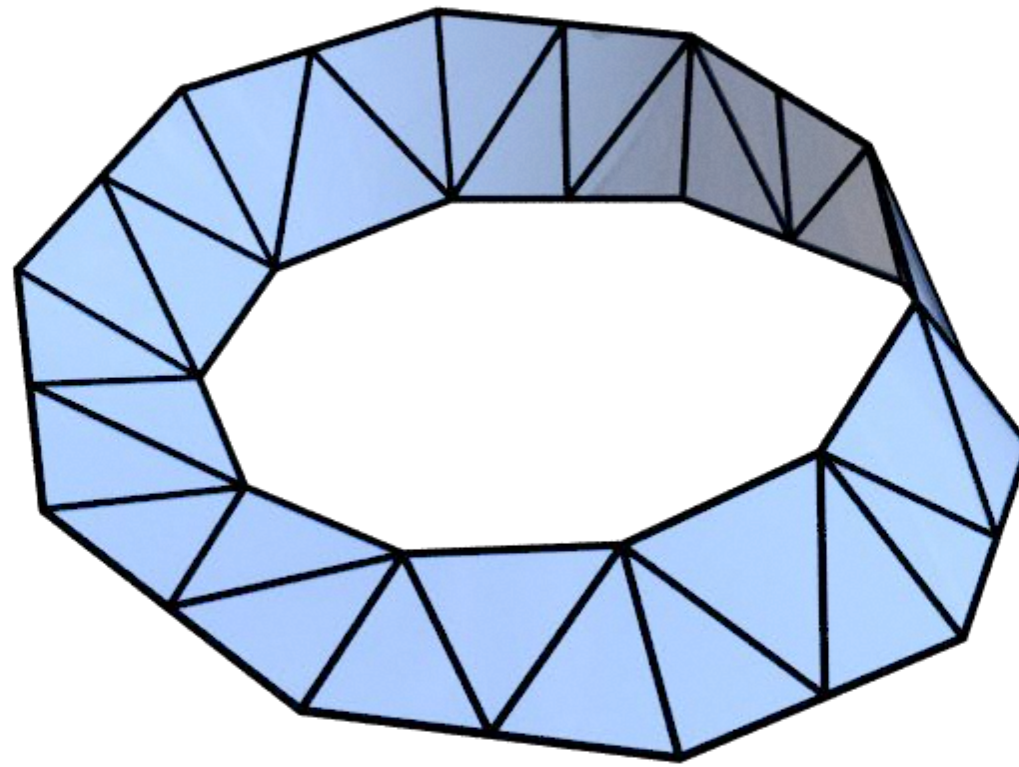
$g=1, b=1$

# Classification of surface

- **Theorem**

*Two orientable surfaces have the same topology if and only if they share the same genus and number of boundaries*

What about the classification of non-orientable surfaces?



# Classification of surface

- **Theorem**

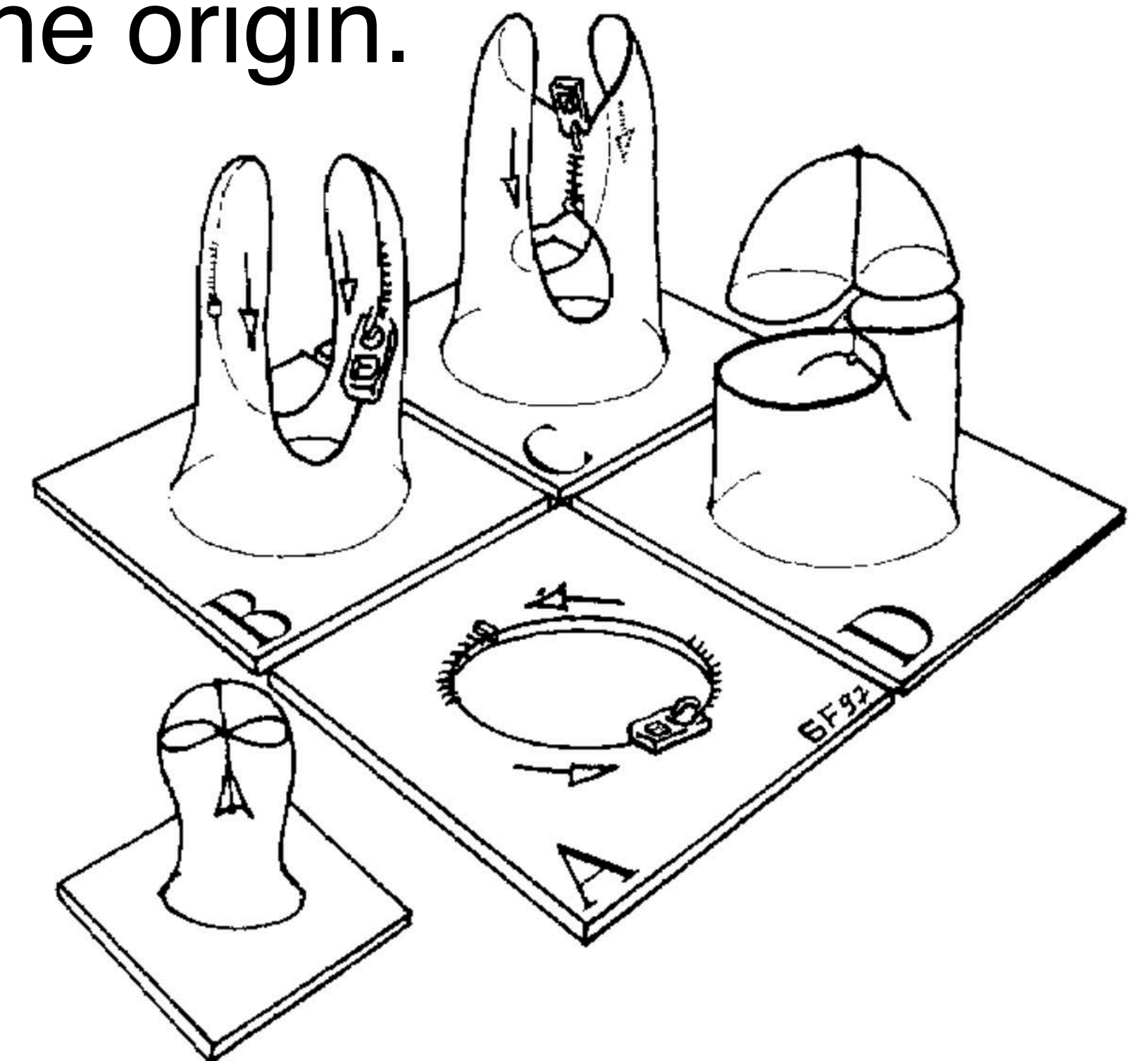
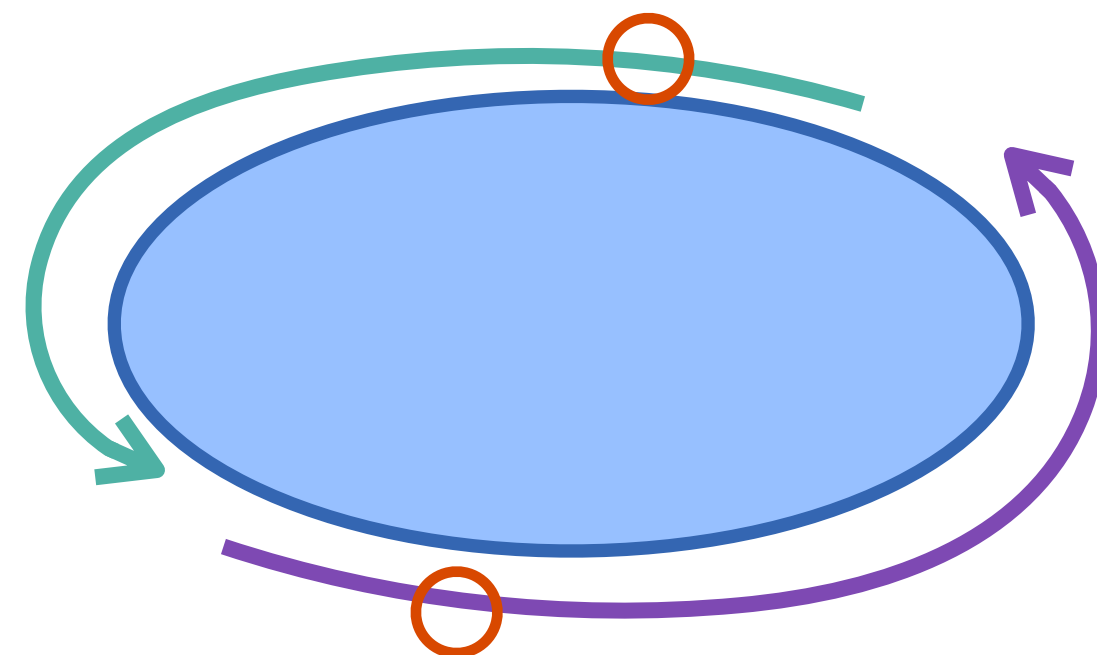
*Two surfaces have the same topology if and only if they share the same orientability, genus and number of boundaries*

- ▶ every non-orientable surface is the connected sum of the fundamental element (followed by some disk removal)

# Non-orientable surface

## Real projective plane

- Elementary non-orientable 2D space.
- **Definition** the space of all lines through the origin.
- It is the same as the sphere with antipodal points viewed as the same point.
- Same as the hemisphere (disk) with the antipodal points of the boundary identified.

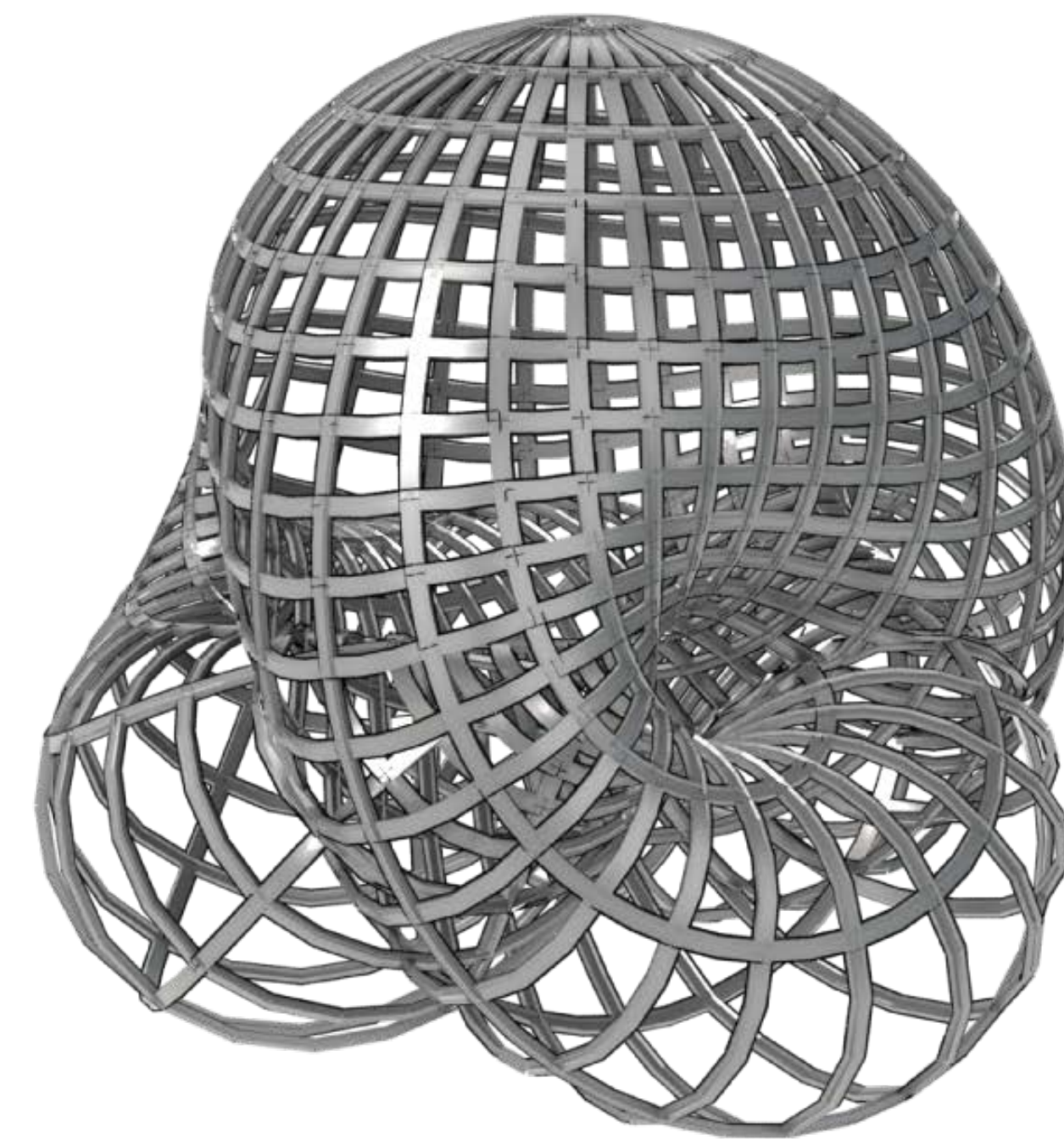
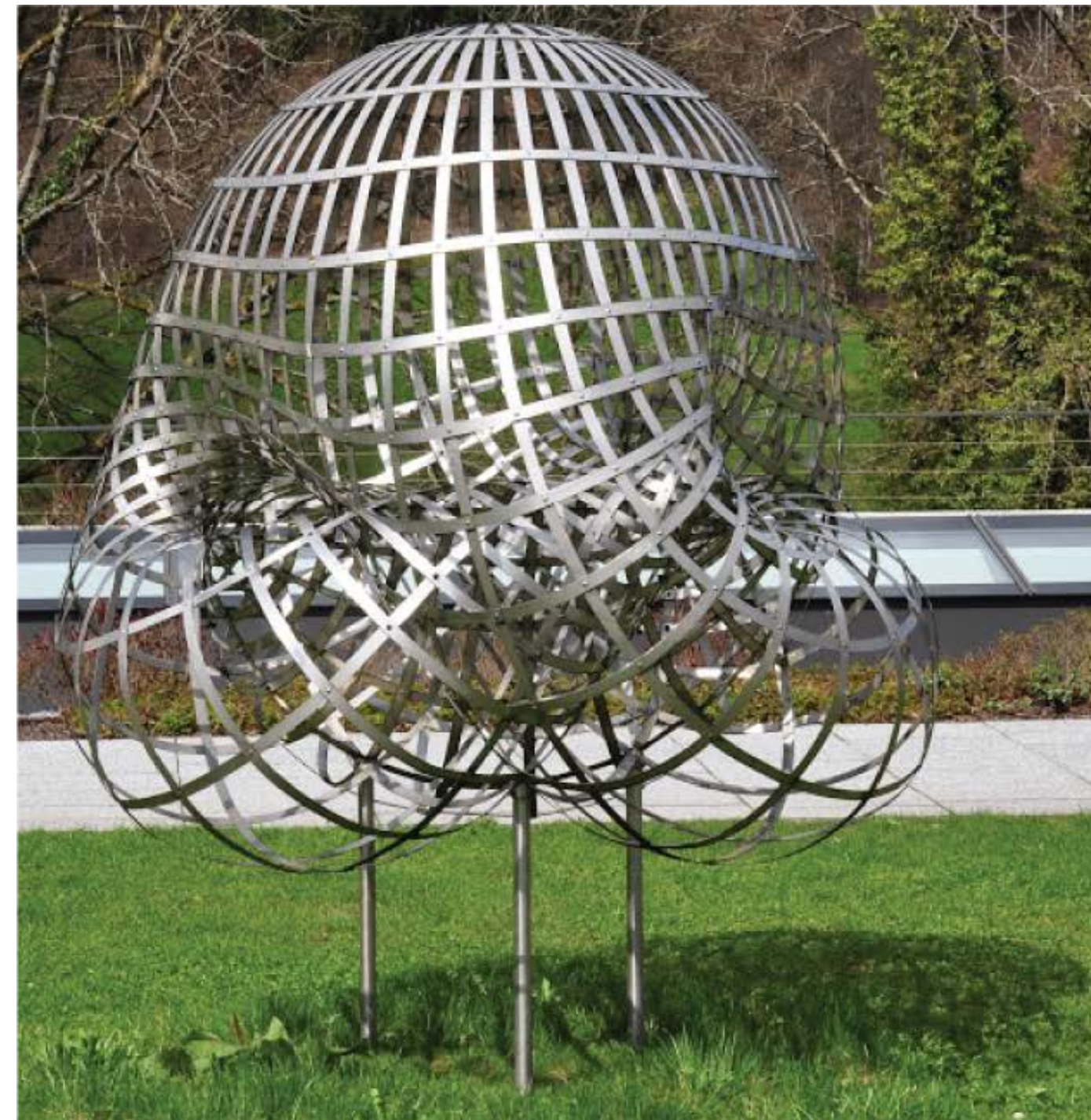


cross-cap

# Non-orientable surface

Boy's Surface (discovered by Werner Boy in 1901)

- It is a smooth realization of the real projective plane.
- “genus = 0.5,” (and  $b = 0$ )
- $k=1$  (number of projective planes)
- $k$  is also called the *non-orientable genus*

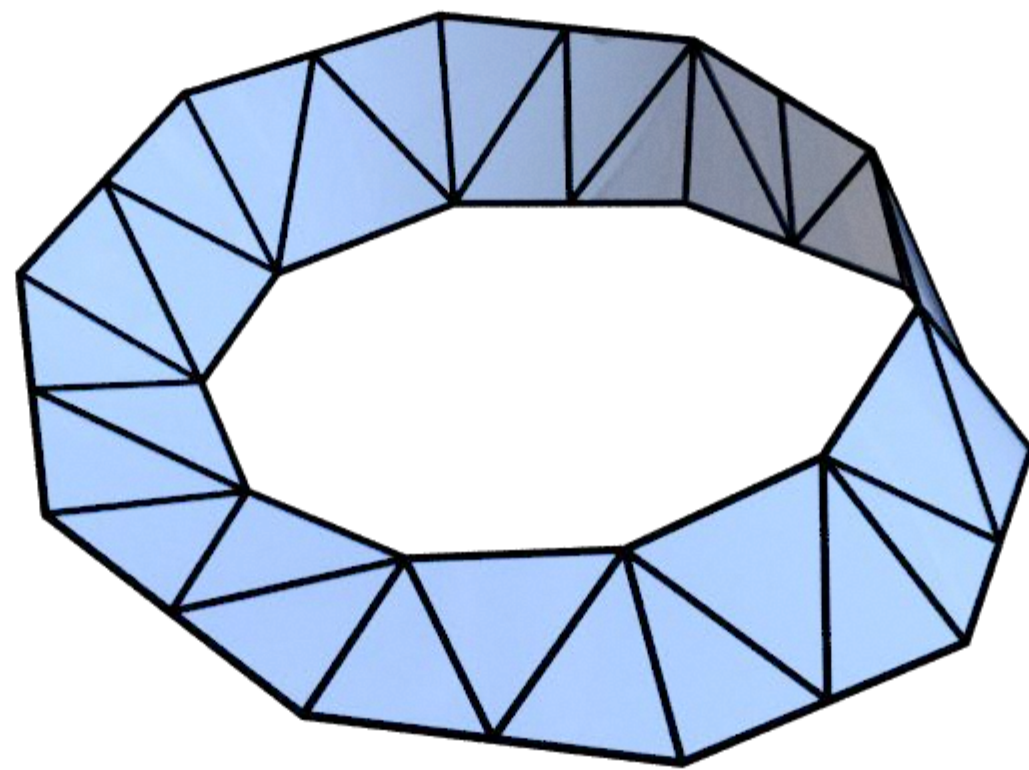


# Non-orientable surface

Boy's Surface (discovered by Werner Boy in 1901)

- Other non-orientable surfaces can be made by Boy's surfaces

$k=1, b=1$

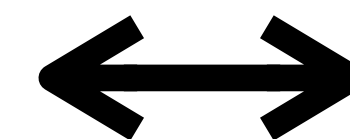


Möbius strip is  
a Boy's surface with  
one disk removed

$k=2, b=0$



Fusing 2 Boys



Klein bottle

# Euler Characteristics

- Simplicial complex
- Chain complex
- Manifold surface mesh
- Surface topology
- Euler characteristics

# Euler characteristic

- For any discrete surface

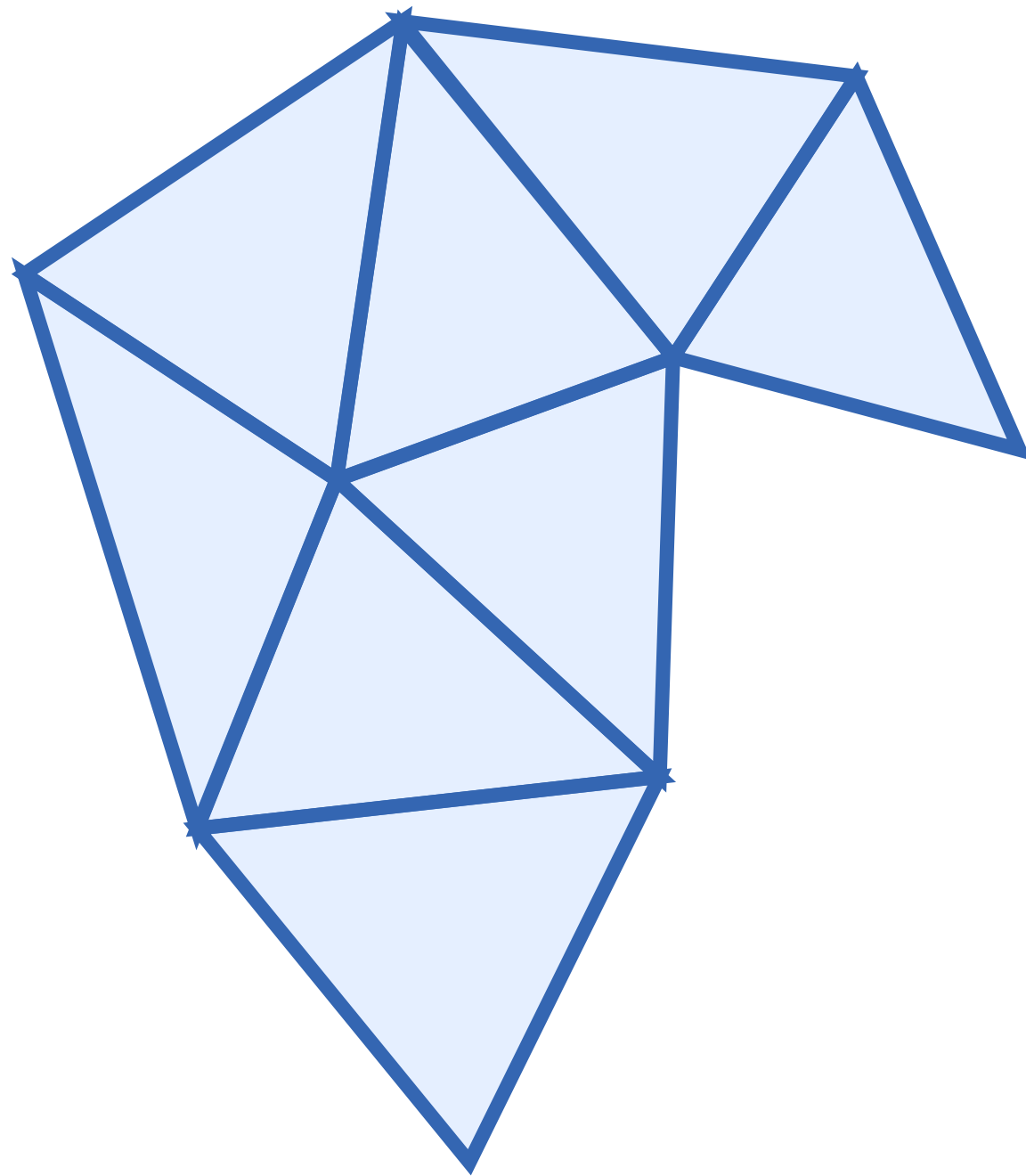
$$M = (V, E, F)$$

define the **Euler characteristic**

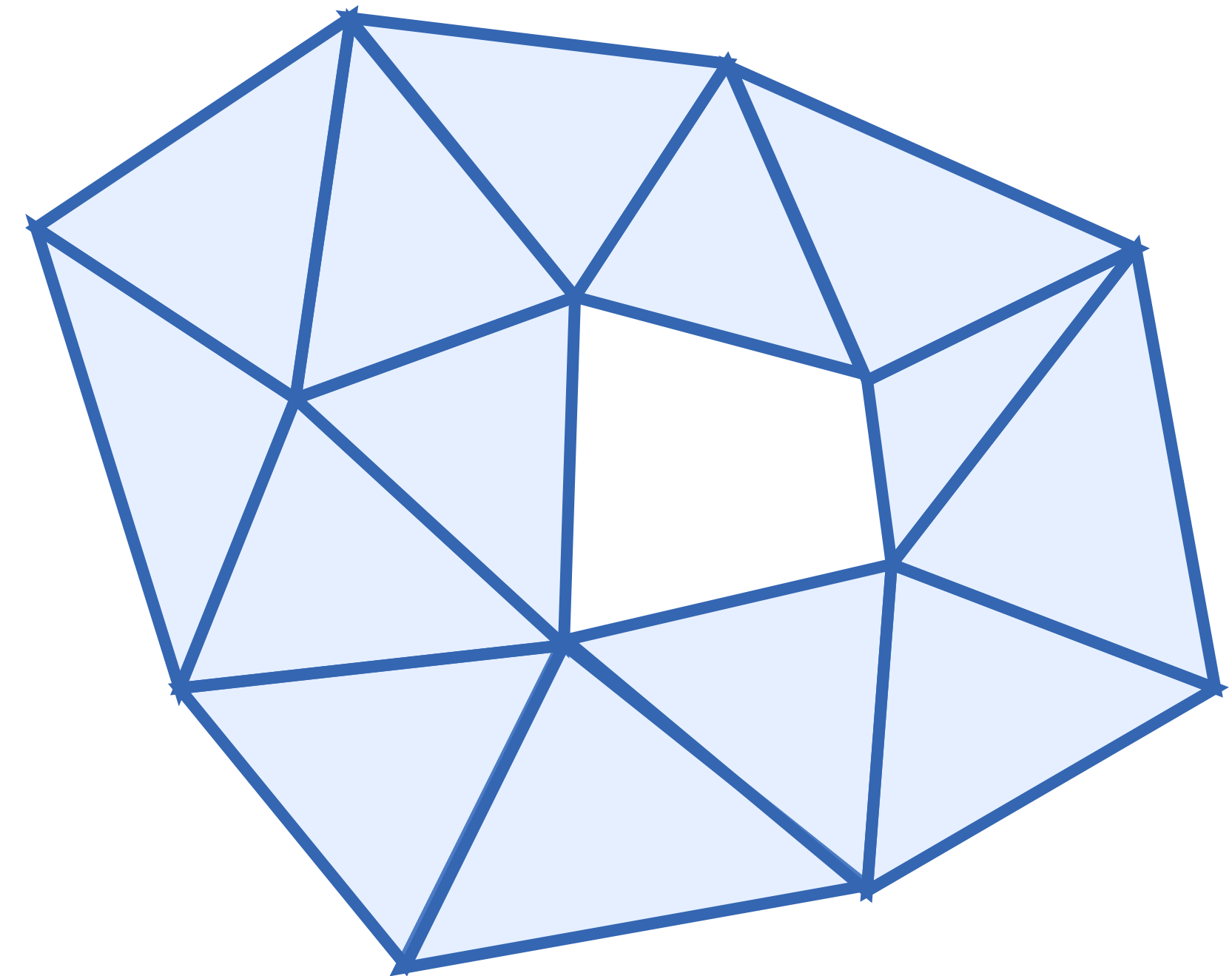
$$\chi(M) = |V| - |E| + |F|$$

# Euler characteristic

Discrete disk



Discrete annulus



# Euler characteristic

- For any discrete surface

$$M = (V, E, F)$$

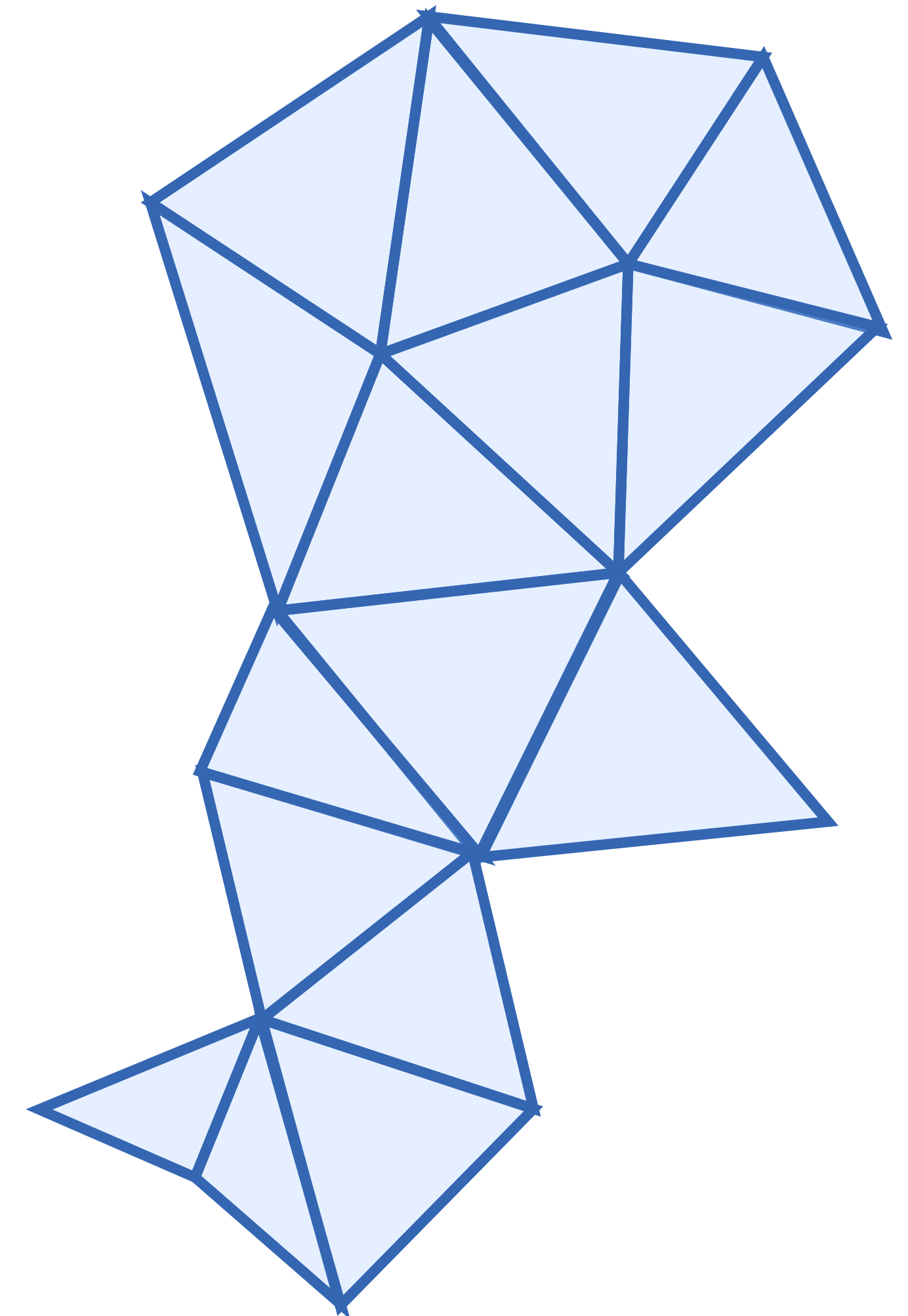
define the **Euler characteristic**

$$\chi(M) = |V| - |E| + |F|$$

- The Euler characteristic is a **topological invariant**
- No matter how big the mesh is,  $\chi(M)$  is determined by its topological type

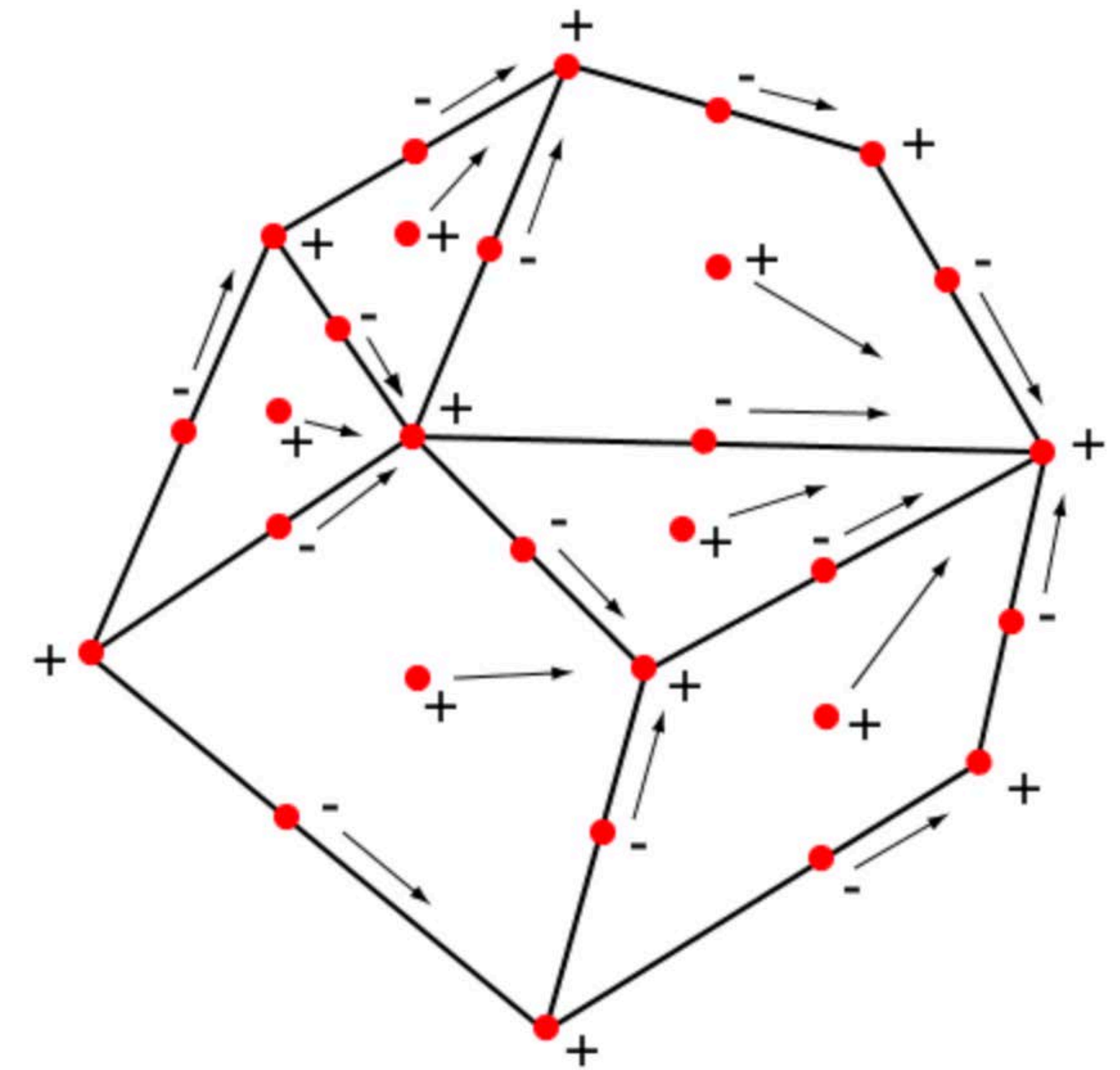
# Euler characteristic

- Show that topological disk has Euler characteristic = 1
  - ▶ Start peel off triangle from the boundary while maintaining being disk
  - ▶ Whenever a triangle is removed, show Euler characteristic is the same
  - ▶ When there is only one triangle left, we get Euler characteristic = 1



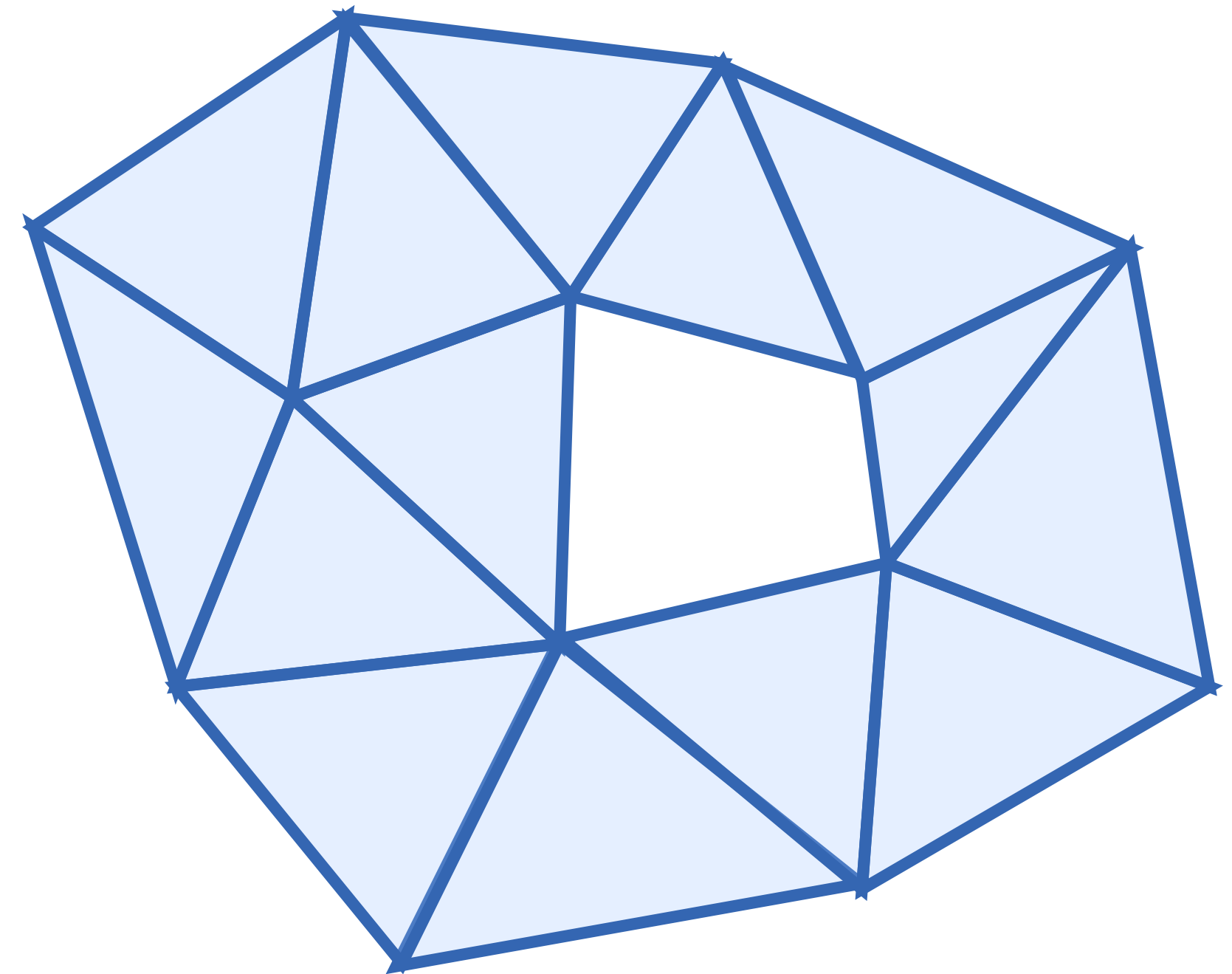
# Euler characteristic

- Show that topological disk has Euler characteristic = 1
  - ▶ Place a positive charge on each vertex and face
  - ▶ negative charges on each edge
  - ▶ Slide all face and edge charges to the right
  - ▶ All charges will cancel except for 1 left-most vertex



# Euler characteristic

- For other topology
  - ▶ Cut, remove disk, or fill in holes so that it becomes a disk

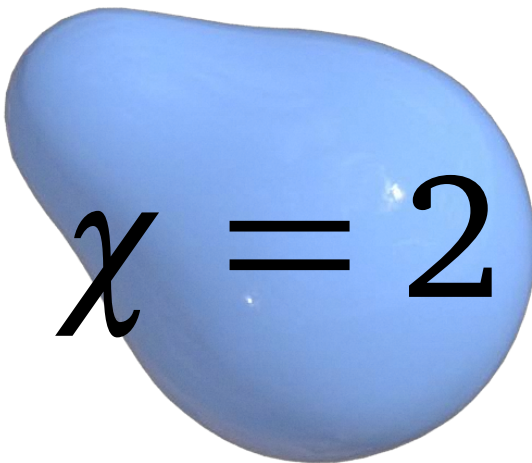
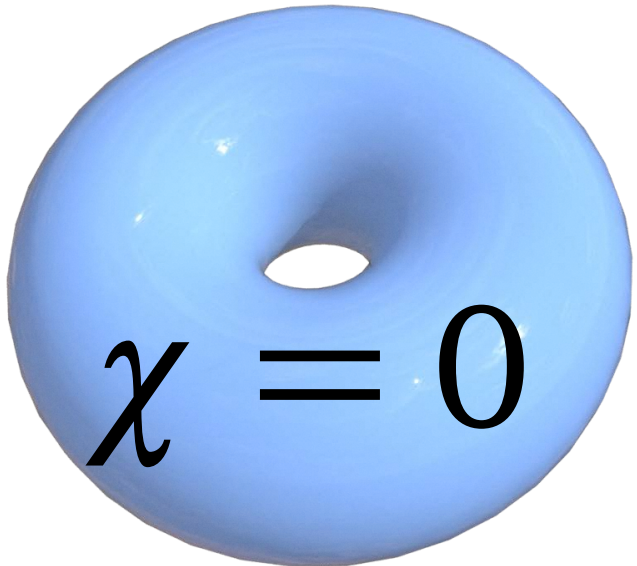
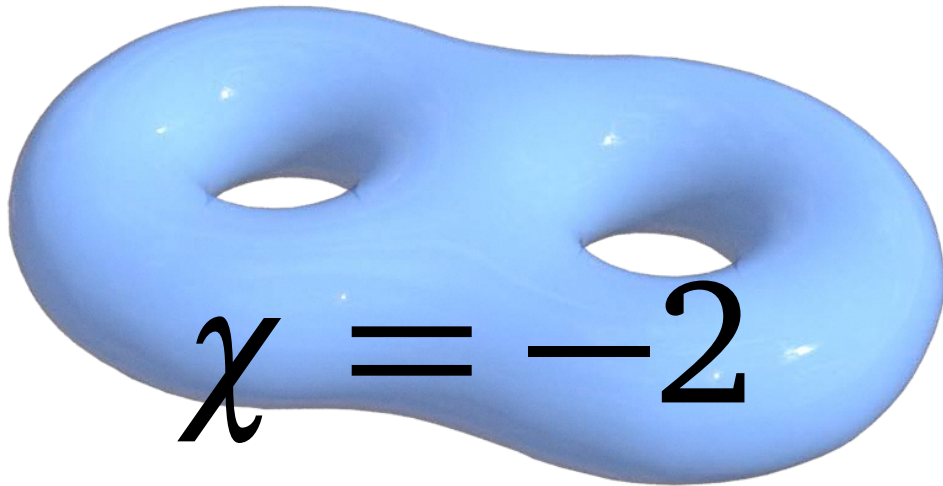
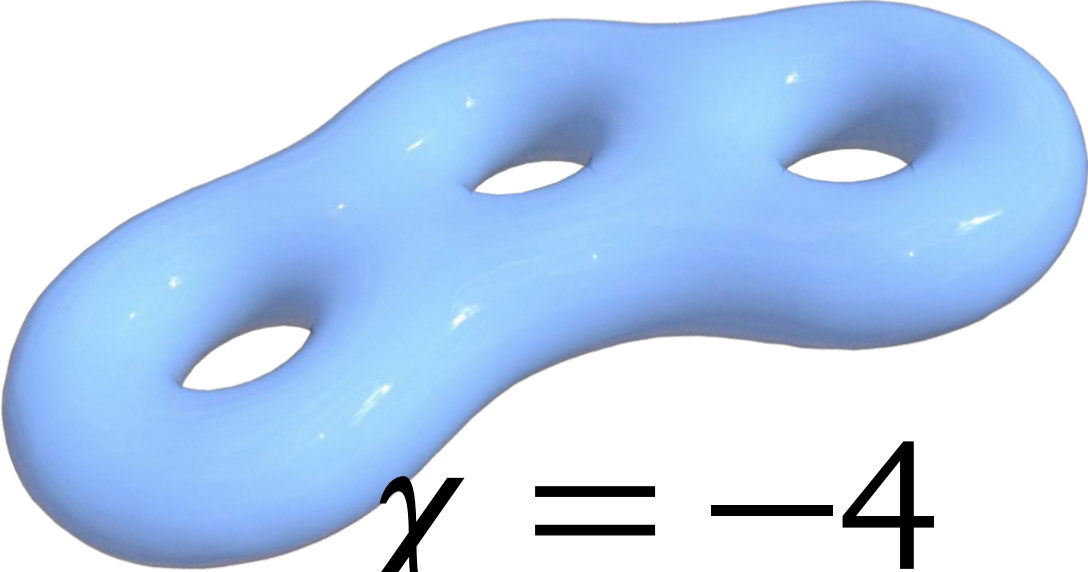
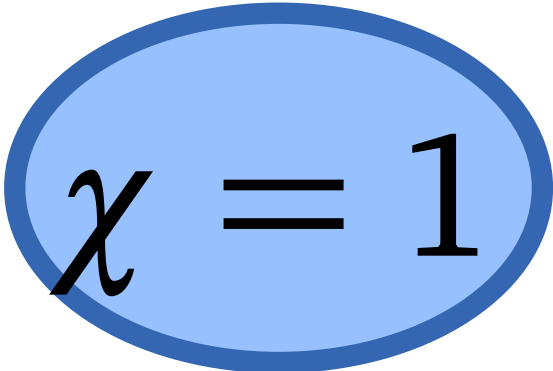
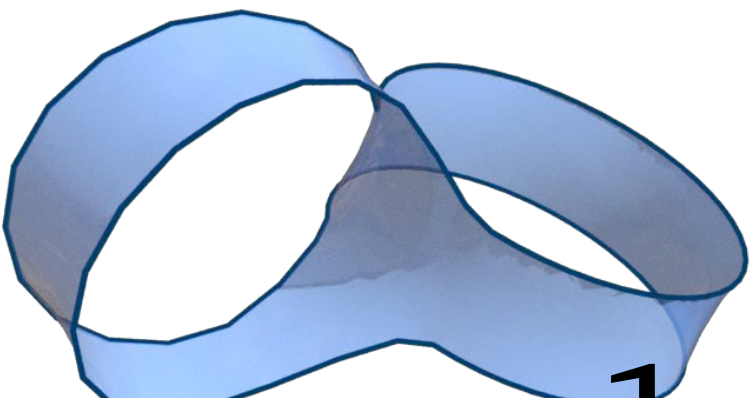
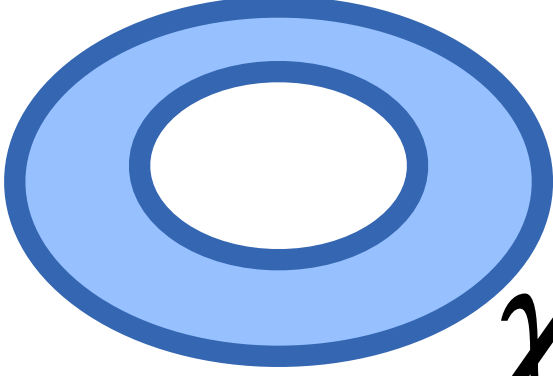
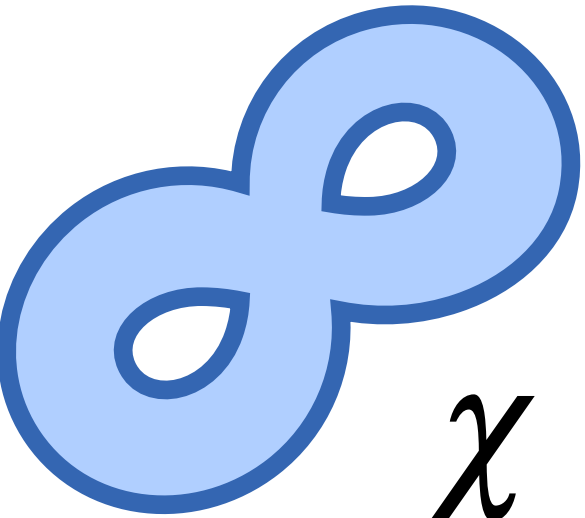


# Euler–Poincaré Theorem

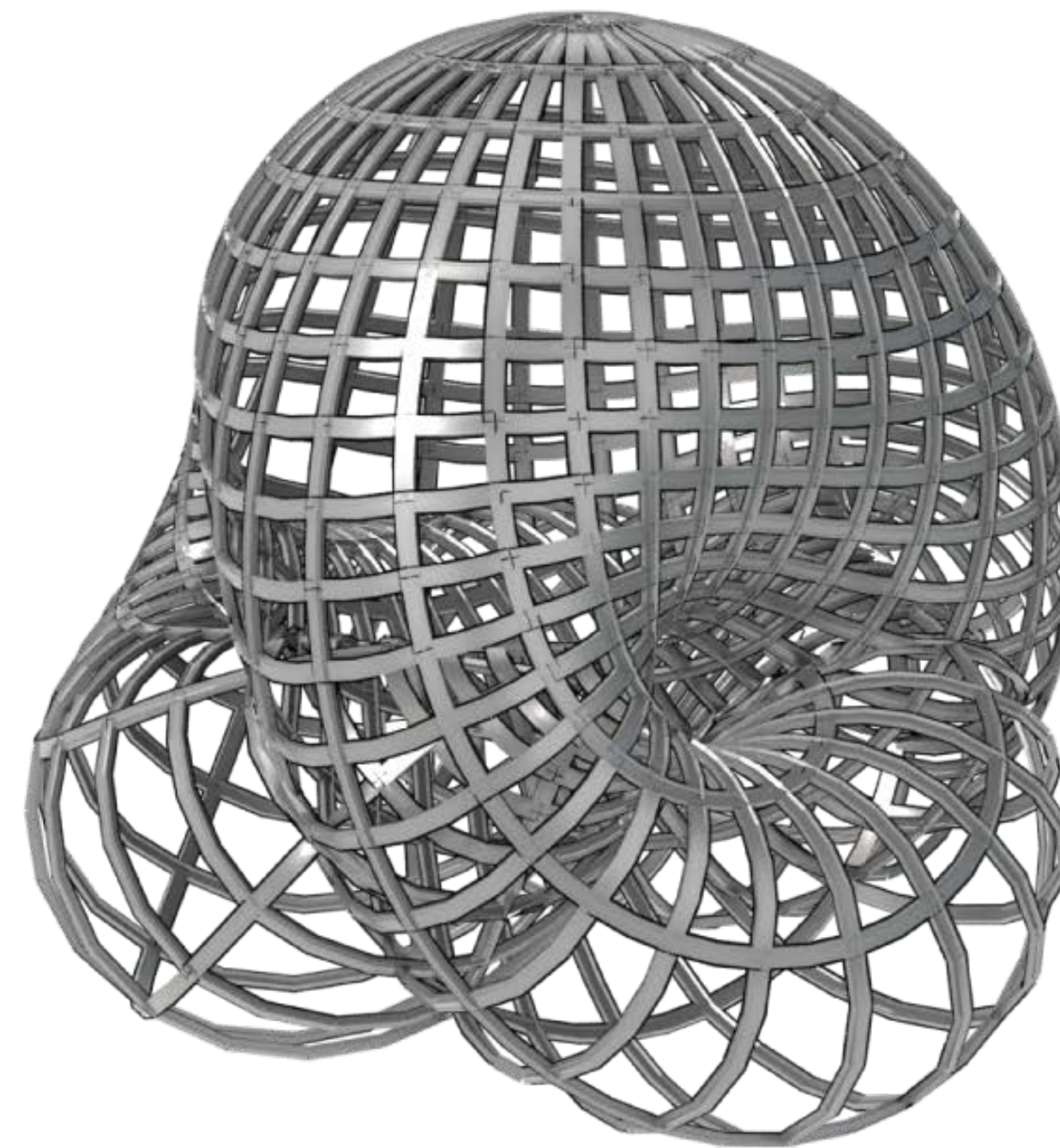
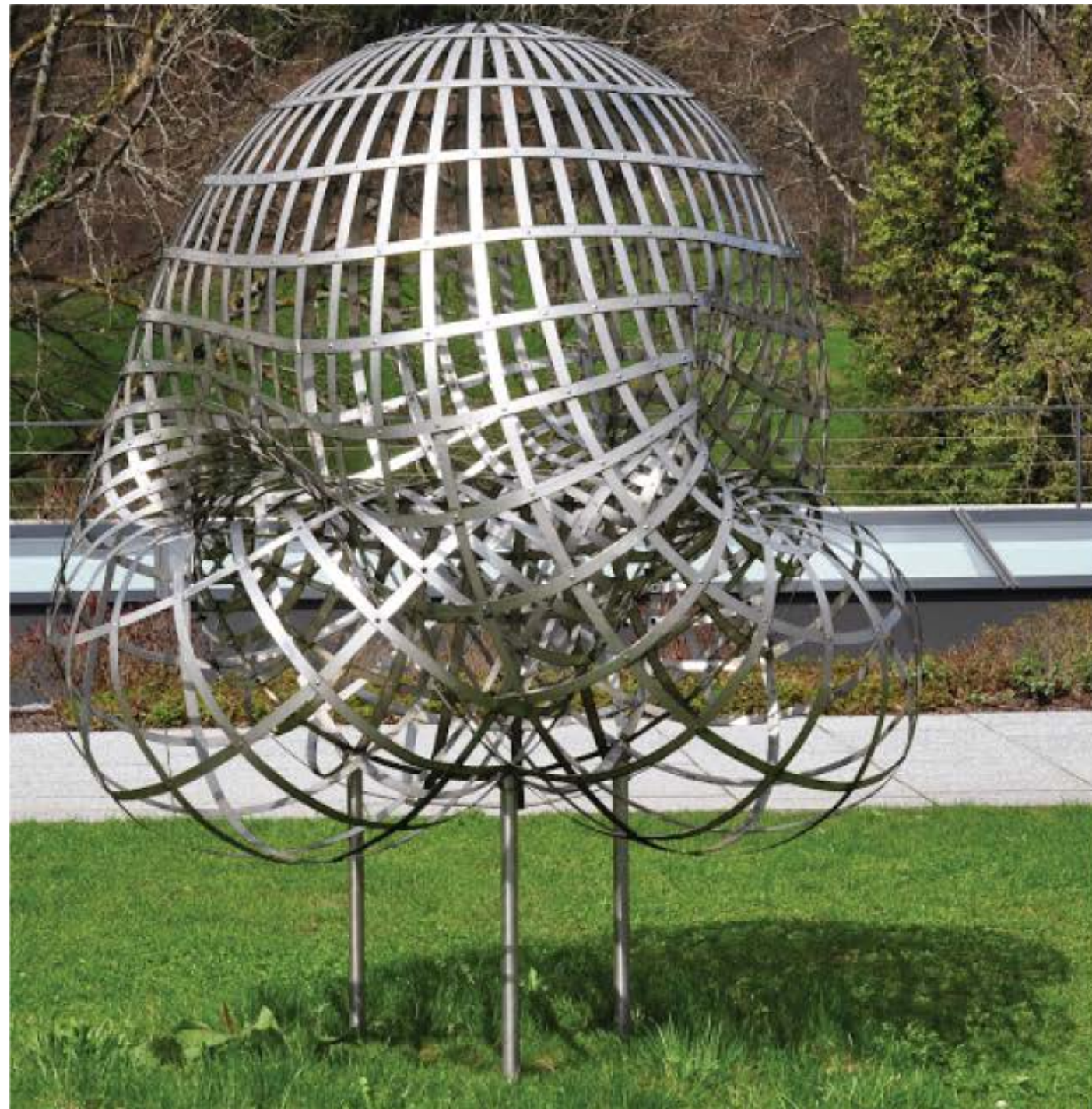
$$\chi(M) = 2 - 2g - b$$

$$|V| - |E| + |F| = 2 - 2g - b$$

# Euler–Poincaré Theorem

|       | $g=0$                                                                                                     | $g=1$                                                                                                      | $g=2$                                                                                                                | $g=3$                                                                                              |
|-------|-----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| $b=0$ | <br>$\chi = 2$<br>sphere | <br>$\chi = 0$<br>torus | <br>$\chi = -2$<br>“double torus” | <br>$\chi = -4$ |
| $b=1$ | <br>$\chi = 1$<br>disk  | <br>$\chi = -1$        | <br>$\chi = -3$                  |                                                                                                    |
| $b=2$ | <br>annulus $\chi = 0$ |                                                                                                            |                                                                                                                      |                                                                                                    |
| $b=3$ | <br>$\chi = -1$        |                                                                                                            |                                                                                                                      |                                                                                                    |

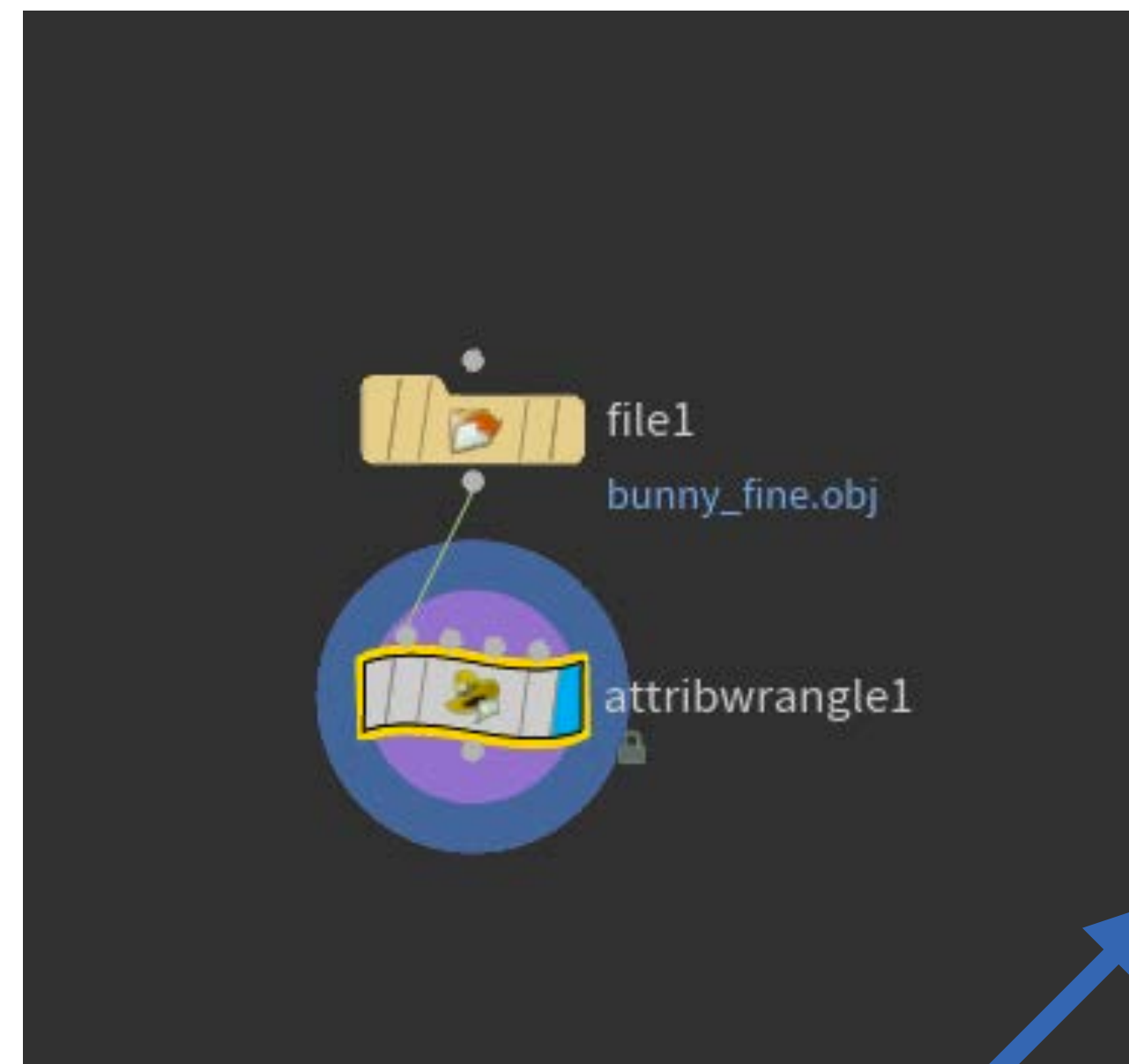
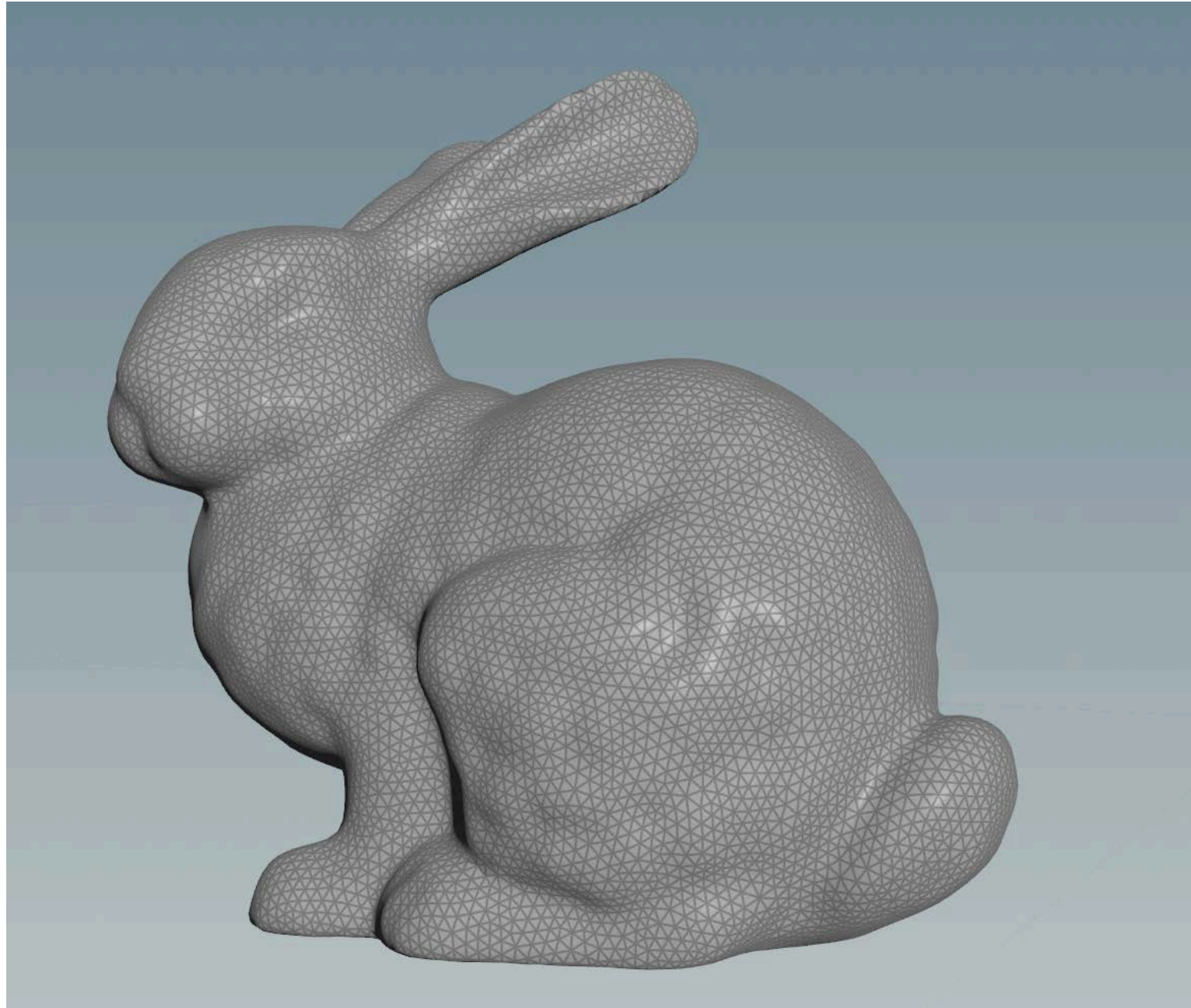
# Euler–Poincaré Theorem



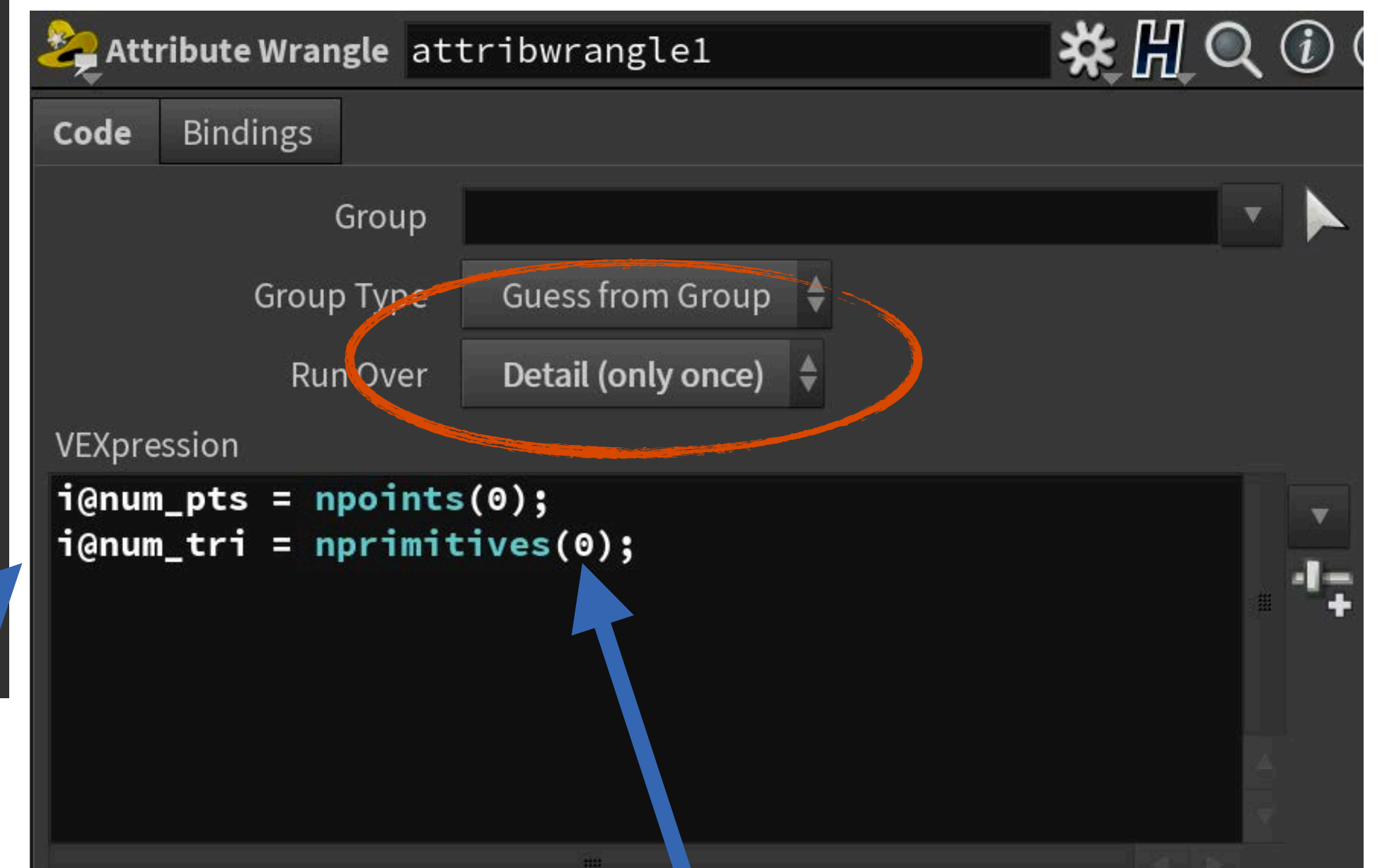
$$\chi = 1$$

# Count $|V|$ and $|F|$ in Houdini

1. read a triangle mesh
2. put an attribute wrangle
3. (important) Select “Detail” for “RunOver”



This defines new integer detail attributes  
(with prefix `i@`)



0 here means the geometry from  
the 0th input

# vex functions are quite easy to look up

(if you know what to search for...)



vex npoints

<https://www.sidefx.com> › [houdini](#) › [vex](#) › [functions](#) › [n...](#) ⋮

## [npoints VEX function - SideFX](#)

**npoints** VEX function. Returns the number of points in the input or geometry file. int  
**npoints**(<geometry>geometry). Show/hide arguments. <geometry>.

[Houdini 19.0](#) › [VEX](#) › [Functions](#) › [npoints](#) ›

## **npoints** VEX function

*Returns the number of points in the input or geometry file.*

```
int npoints(<geometry>geometry)
```

[Show/hide arguments](#)

### <geometry>

When running in the context of a node (such as a wrangle SOP), this argument can be an integer representing the input number (starting at 0) to read the geometry from.

Alternatively, the argument can be a string specifying a geometry file (for example, a **.bgeo**) to read from. When running inside Houdini, this can be an **op:** /path/to/sop reference.

# Count $|V|$ and $|F|$ in Houdini

## Geometry spreadsheet

point  
attributes

vertex  
attributes

primitive  
attributes

detail attributes

|         | Detail |  |
|---------|--------|--|
| num_pts | 14290  |  |
| num_tri | 28576  |  |
|         |        |  |
|         |        |  |
|         |        |  |
|         |        |  |

So, for our mesh, we have

$$|V| = 14,290$$

$$|F| = 28,576$$

# Count $|V|$ and $|F|$ in Houdini

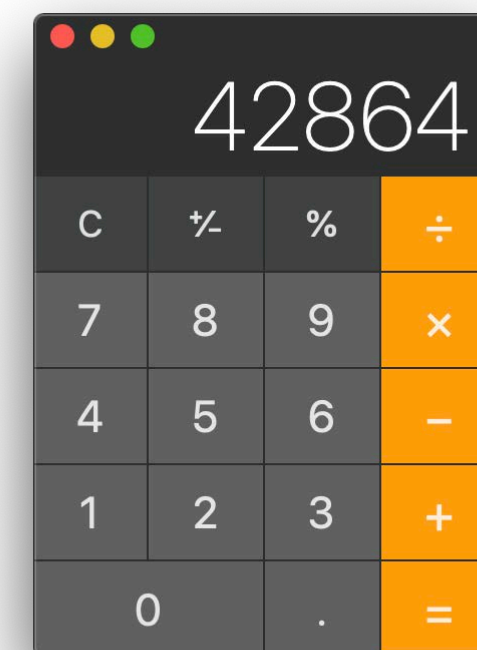
So, for our mesh, we have

$$|V| = 14,290$$

$$|F| = 28,576$$

- You can compute number of edges by  $3|F| = 2|E|$ 
  - ▶ For triangle surface without boundary, every face owns three edges, and each edge is owned by 2 faces.

$$|E| =$$



# Count $|V|$ and $|F|$ in Houdini

$$|V| = 14,290$$

$$|F| = 28,576$$

$$|E| = 42,864$$

$$|V| - |E| + |F| = 2$$

Run Over Detail (only once)

VEXpression

```
i@num_pts = npoints(0);  
i@num_tri = nprimitives(0);  
  
i@num_edge = i@num_tri * 3 / 2;  
  
i@chi = i@num_pts - i@num_edge + i@num_tri;
```

| Node: attribwr... |        |  |
|-------------------|--------|--|
|                   | Detail |  |
| chi               | 2      |  |
| num_edge          | 42864  |  |
| num_pts           | 14290  |  |
| num_tri           | 28576  |  |
|                   |        |  |
|                   |        |  |

# Count $|V|$ and $|F|$ in Houdini

Rewire the Euler characteristic calculator and see the number changes according to the topology!

