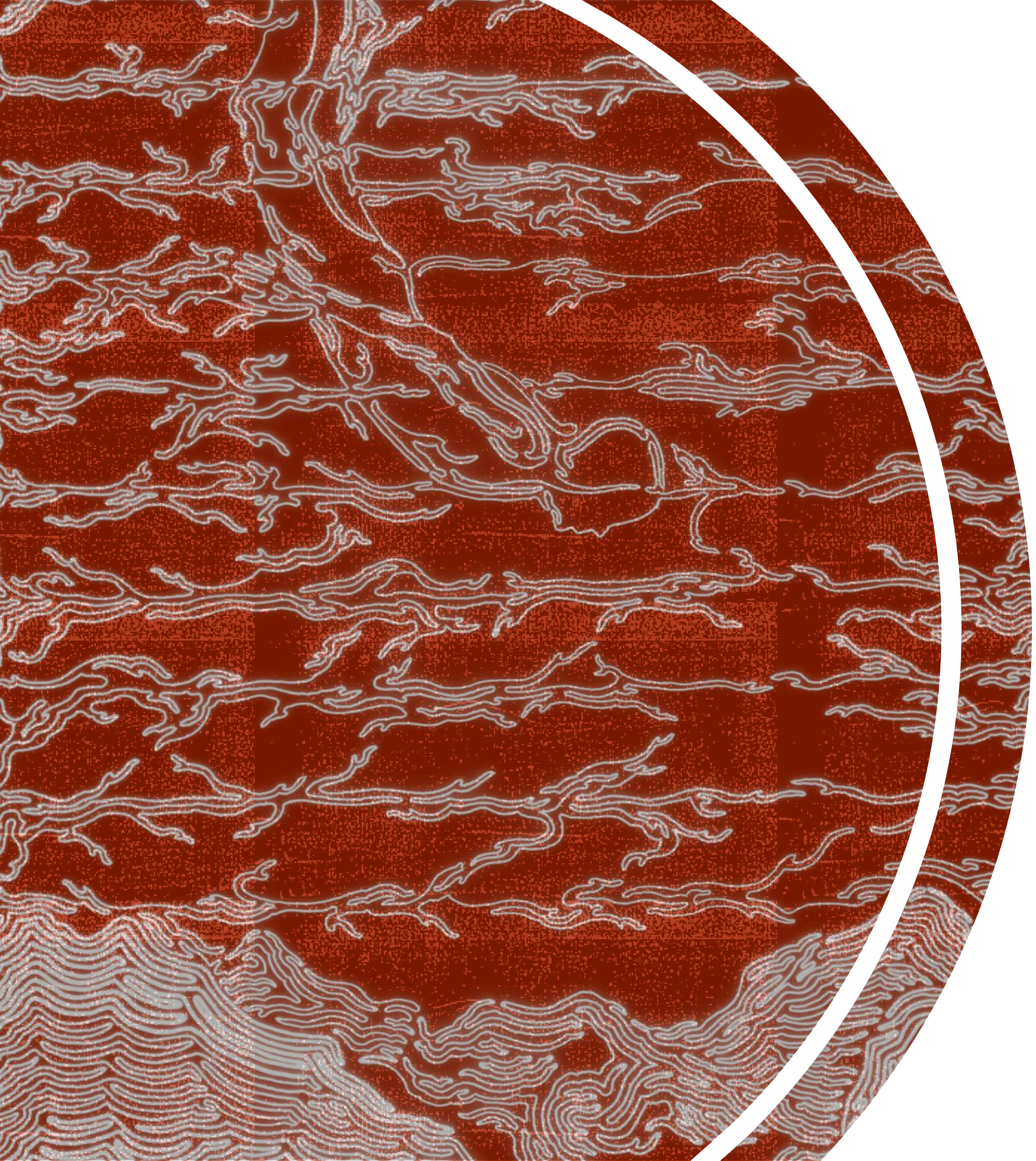


INTRODUCTION TO COMPUTATIONAL TOPOLOGY

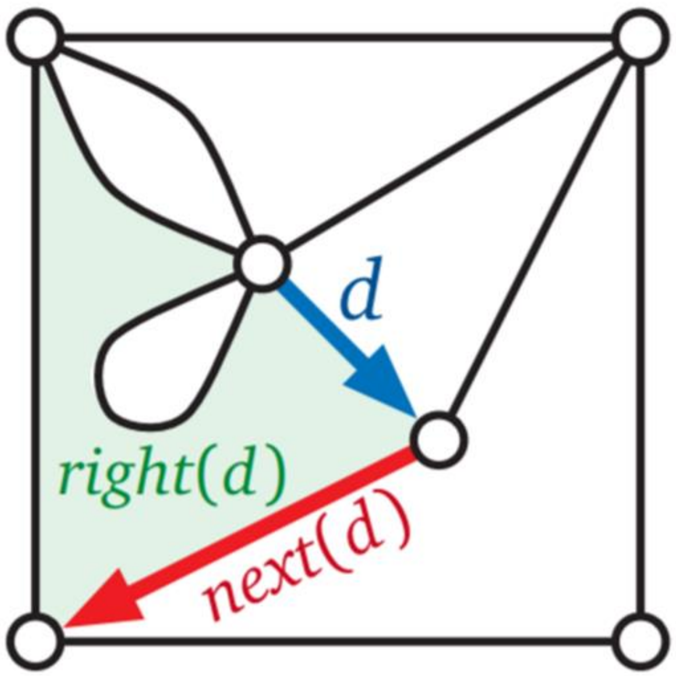
Hsien-Chih Chang

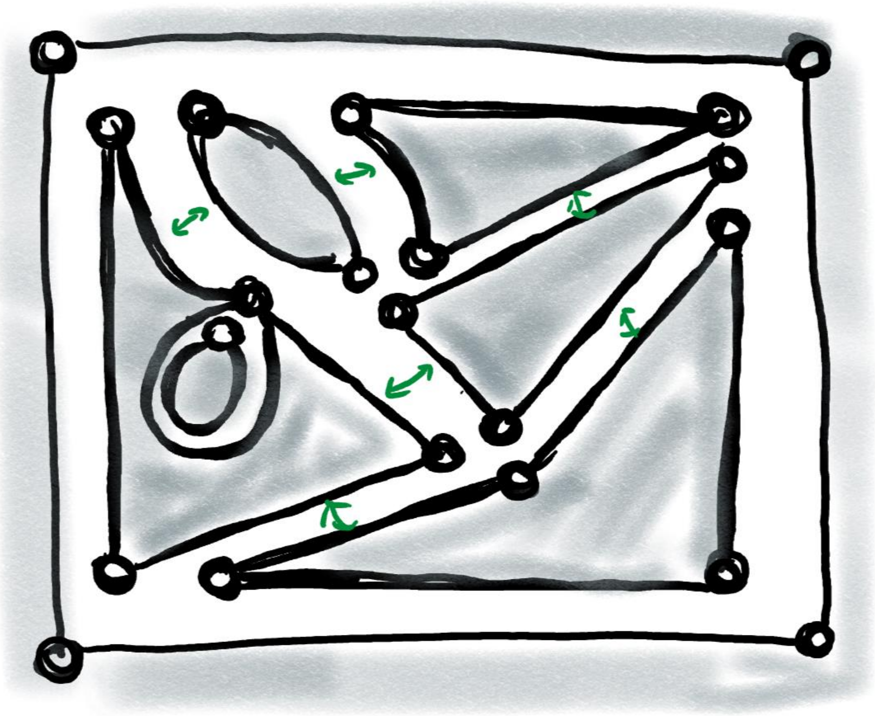
Lecture 4, September 23, 2021

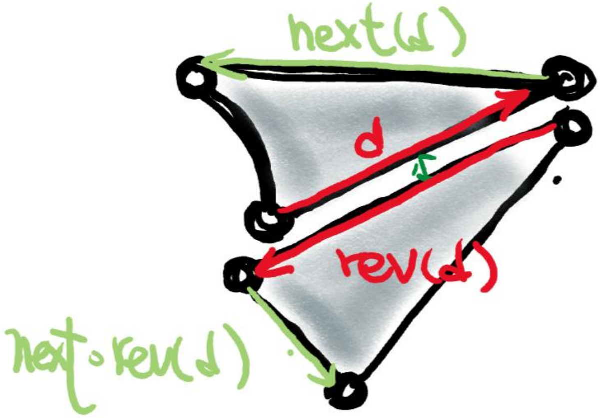
Acknowledgement

- Most of the figures today comes from
  - Jeff Erickson, *One-Dimensional Computational Topology*
  - Robert Ghrist, *Elementary Applied Topology*
  - Keenan Crane, *Discrete differential geometry: An applied introduction*

Recap: Polygonal Schema IS Rotation System





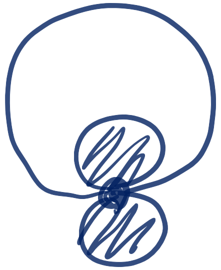


Euler's Formula

Let's Focus on Plane Graphs

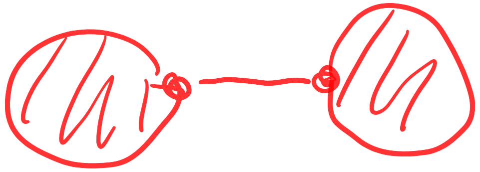
Spanning Tree Terminology

- Loop



edge who two incident vertices are the same

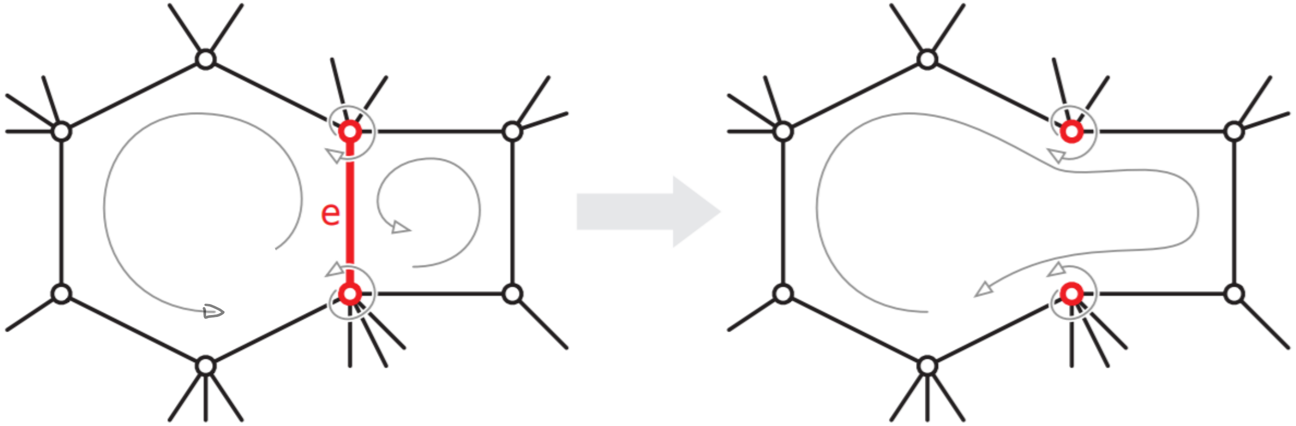
- Bridge



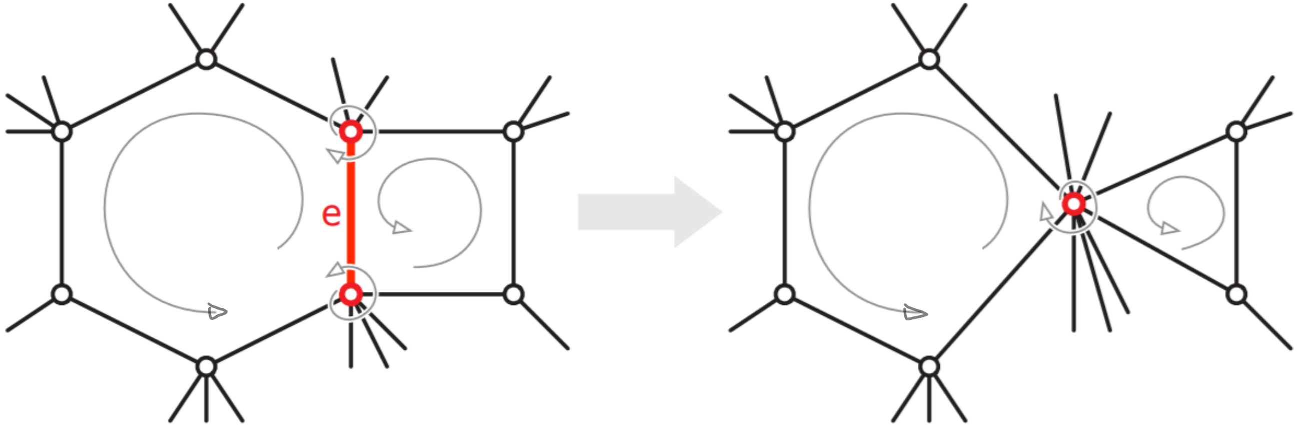
edge whose two incident faces are the same

Spanning Tree Terminology

- Delete



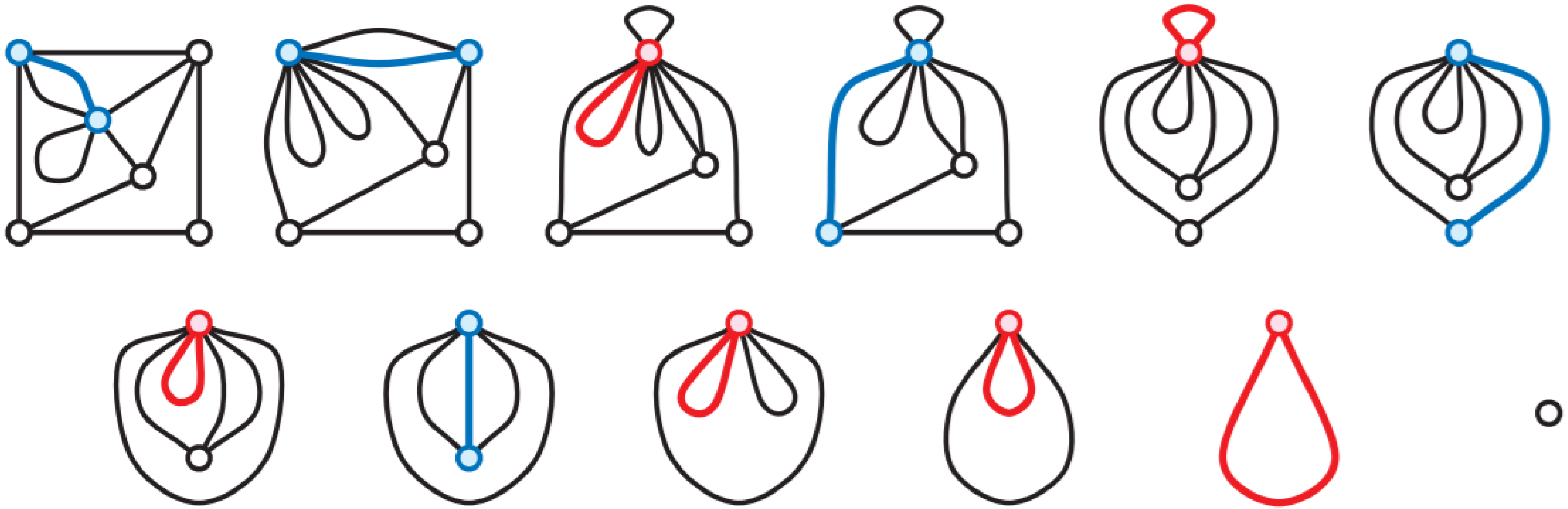
- Contract



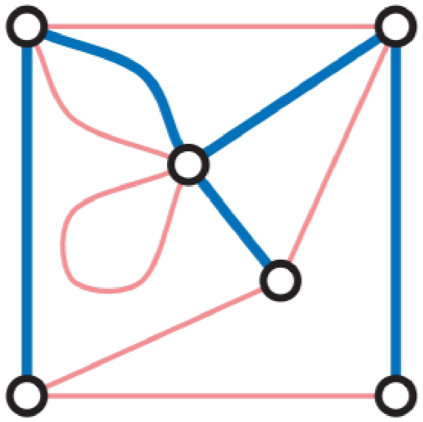
THE SPANNING TREE ALGORITHM

```
SpanningTree(G):
    for any edge e in G:
        if e is a loop:
            delete e
        if e is a bridge:
            contract e
        otherwise:
            delete or contract e
    return all contracted edge T
```

The Spanning Tree Algorithm



Computing a spanning tree of a graph.



The Spanning Tree Algorithm

```
WhateverFirstSearch(s):
  put (∅, s) in bag
  while the bag is not empty
      take (p, v) from the bag                      (⋆)
      if v is unmarked
          mark v
          parent(v) ← p
          for each edge vw                          (†)
              put (v, w) into the bag               (⋆⋆)
```

What Happened in the Dual?

SpanningTree(G):

for any edge e in G:
  if e is a loop:
    delete e.
  if e is a bridge:
    contract e
  o.w.
    delete or contract e
return all contracted edges T

SpanningTree(G*):

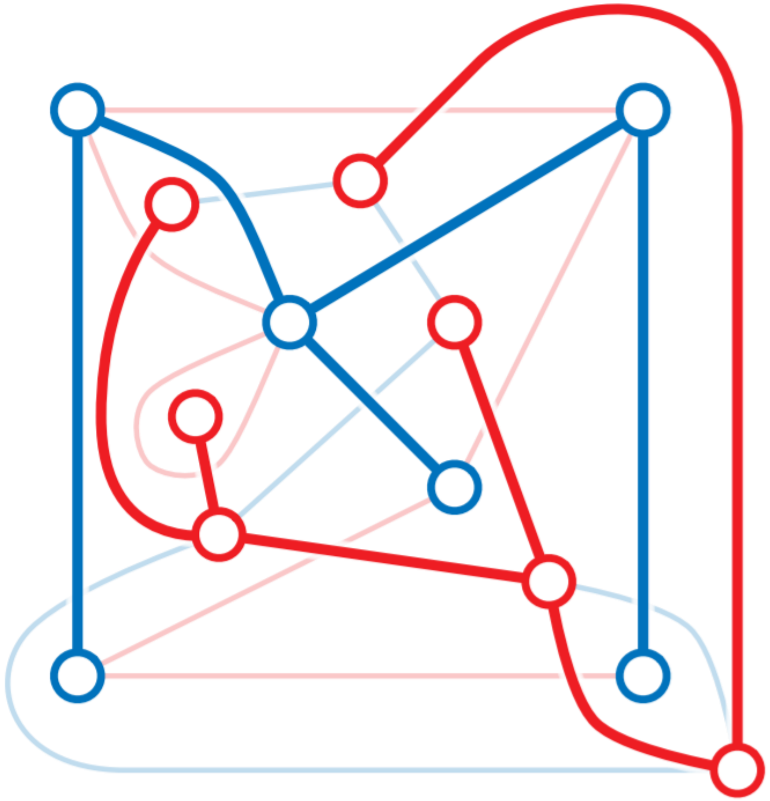
for any edge e* in G*:
  if e* is a ~~loop~~ bridge:
    ~~delete~~ contract e*
  if e* is a ~~bridge~~ loop:
    ~~contract~~ delete e*
  o.w.
    ~~delete~~ contract or ~~contract~~ delete e
return all contracted edges T^C

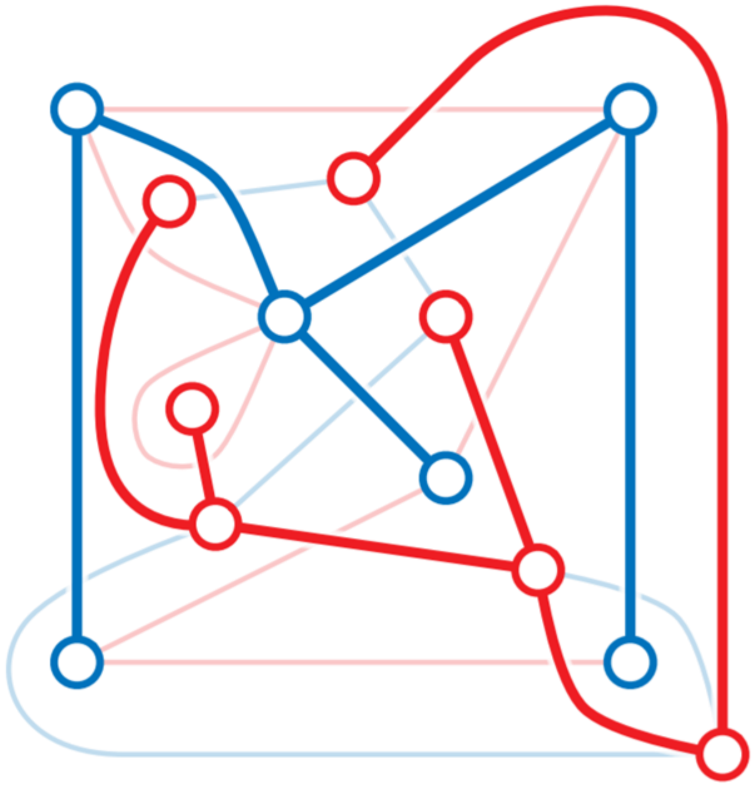
Tree-Cotree Decomposition

- Plane graph G decomposes into
  - Primal spanning tree T
  - Dual spanning cotree C

$$E = (V-1) + (F-1)$$

$$V - E + F = 2$$





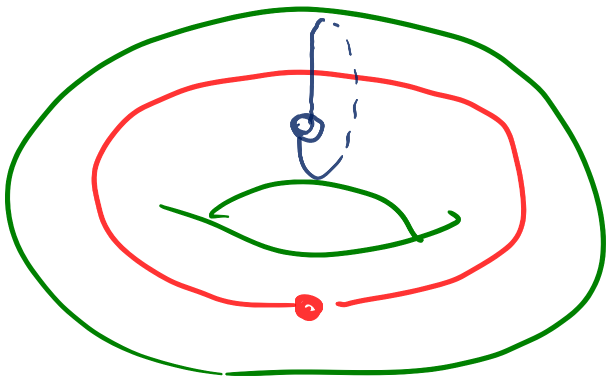
Euler's Formula [Euler 1750] [Legendre 1794] [Cayley-Listing 1861]

For any plane graph G,

$$V_G - E_G + F_G = 2$$

What about Surface Graph?

Spanning Tree



SpanningTree(G):

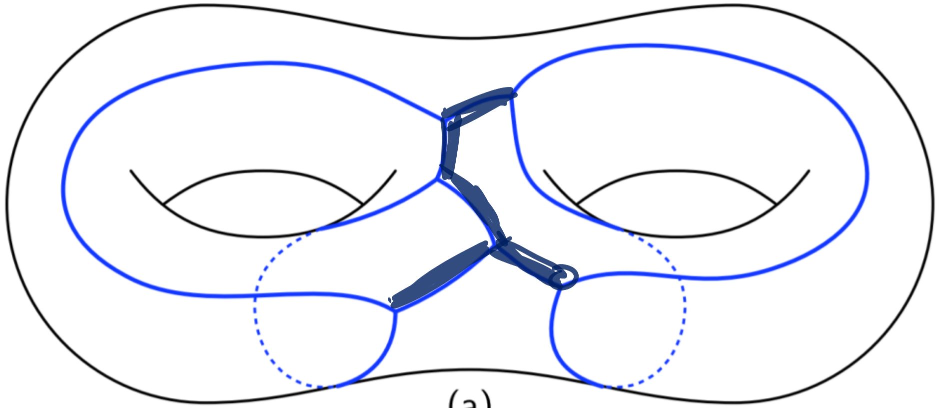
for any edge e in G:
  if e is a loop w/ diff. incident faces
    delete e.
  if e is a bridge: has diff. endpoints.
    contract e
  o.w.
    delete or contract e
return all contracted edges T

System of Loops

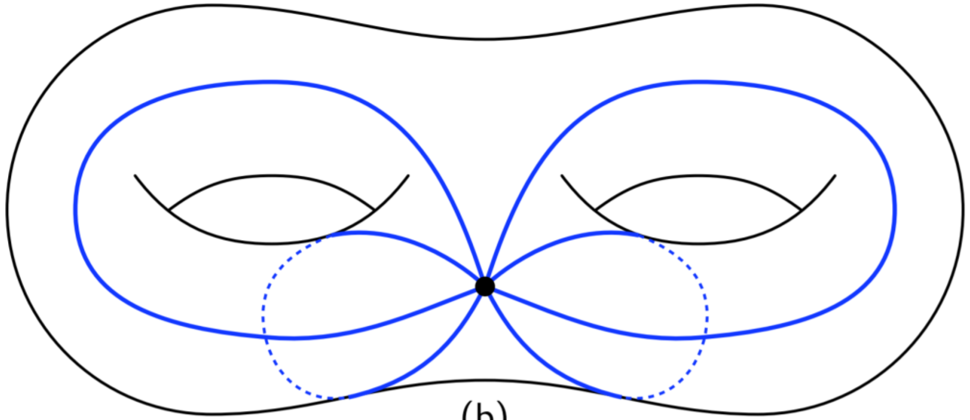
$$E = (V-1) + (F-1) + L$$

Euler characteristic

$$\chi := V - E + F = 2 - L$$



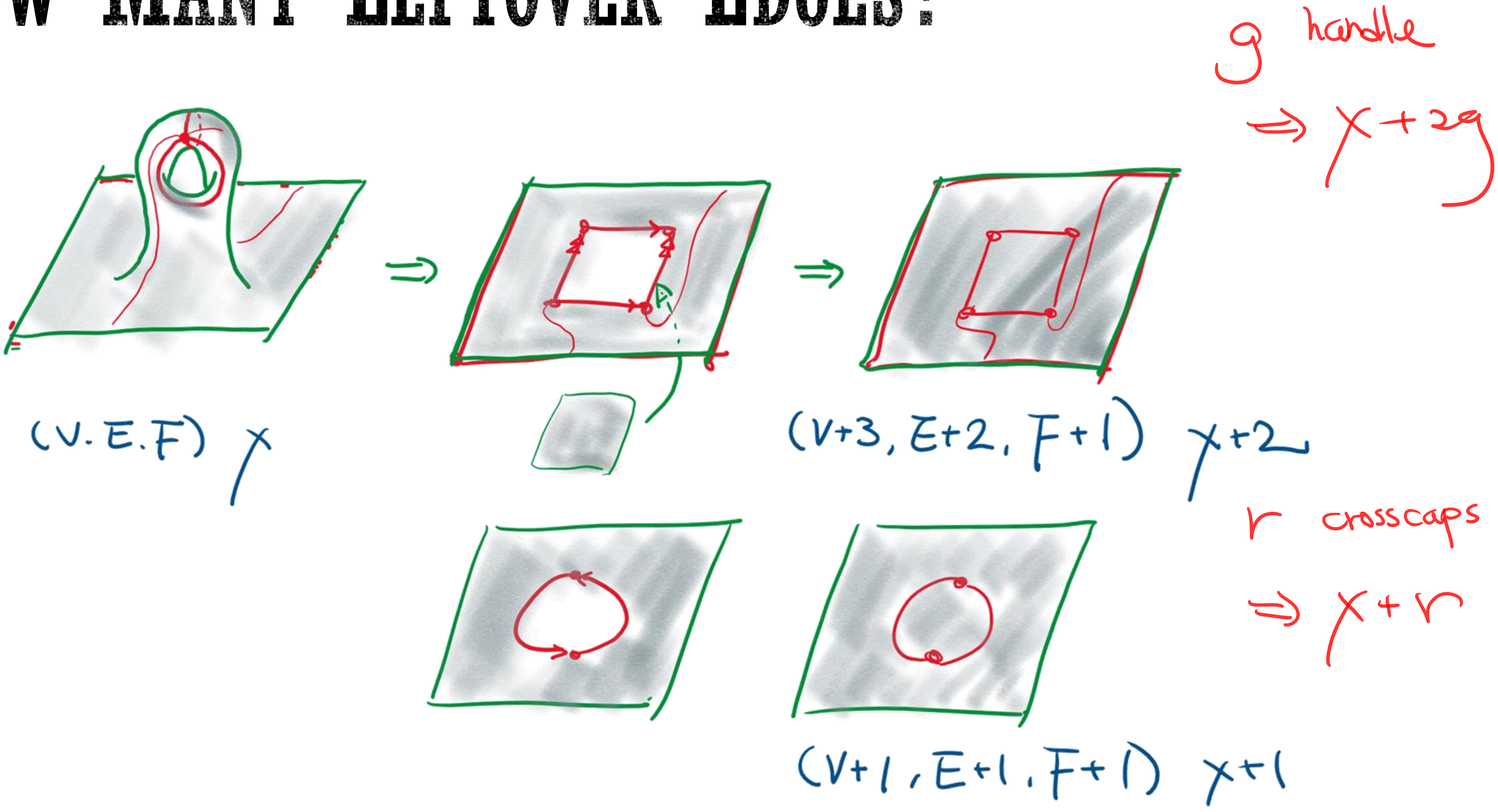
(a)

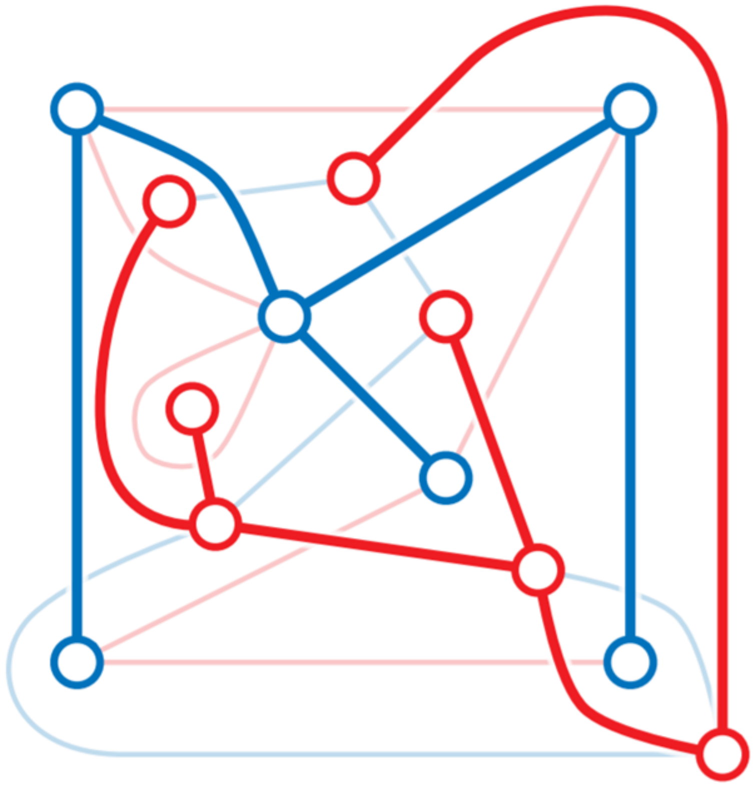


(b)

leftovers L

How Many Leftover Edges?





Euler's Formula

[Euler 1750] [Legendre 1794] [Cayley-Listing 1861]

For any graph G embedded on surface Σ(g,r,b),

$$V_G - E_G + F_G = 2 - 2g - r - b$$

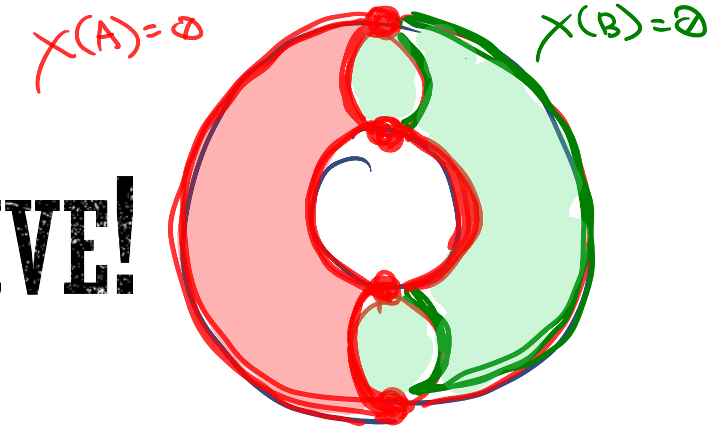
Euler Characteristic is a "Complete" Invar. of Surfaces

Ponder.
Torus and Möbius band have the same χ?

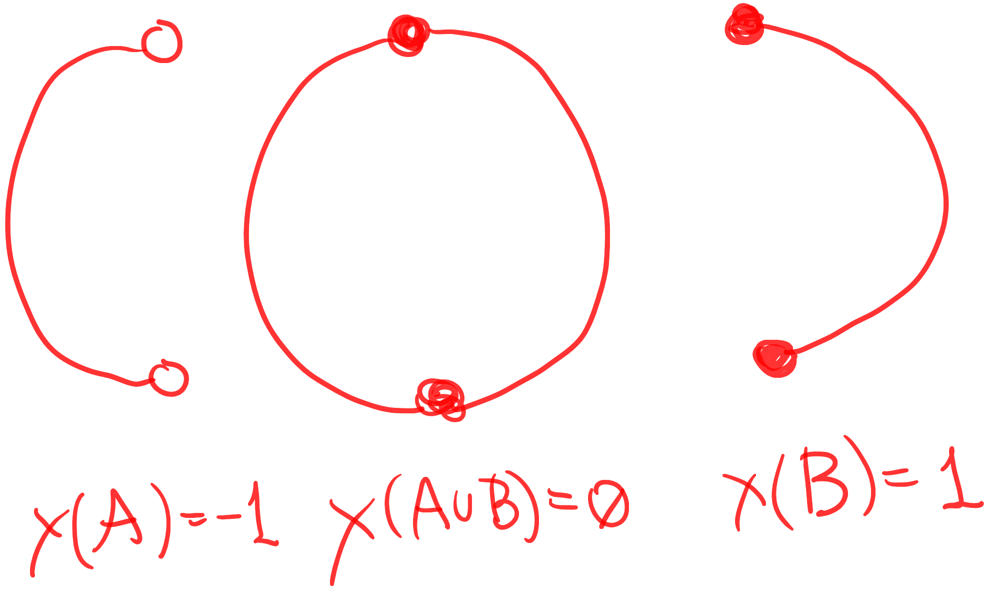
Intermission

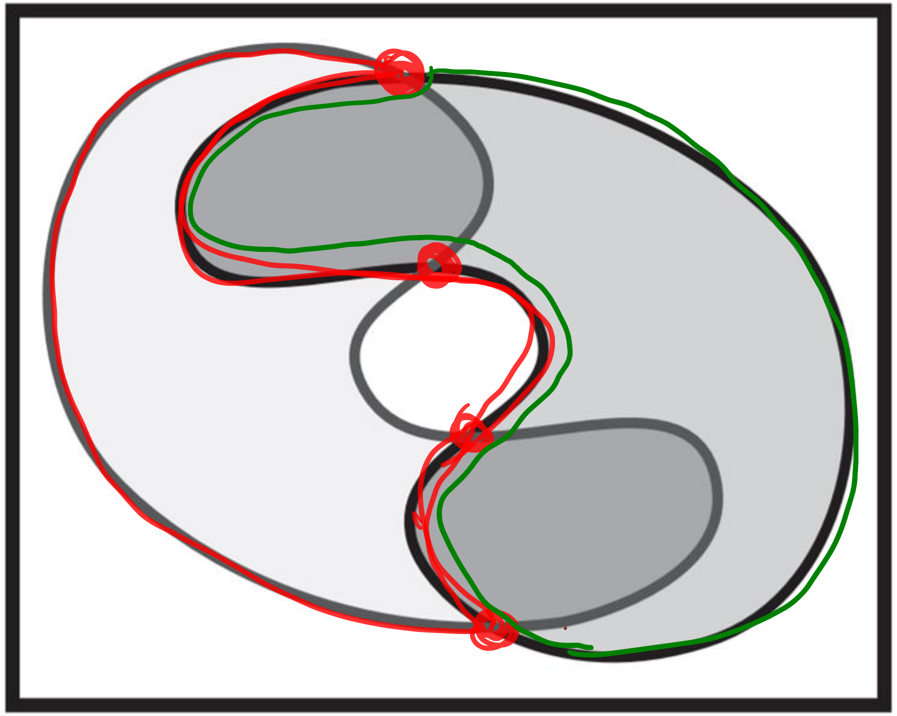
Euler Calculus and Curvature

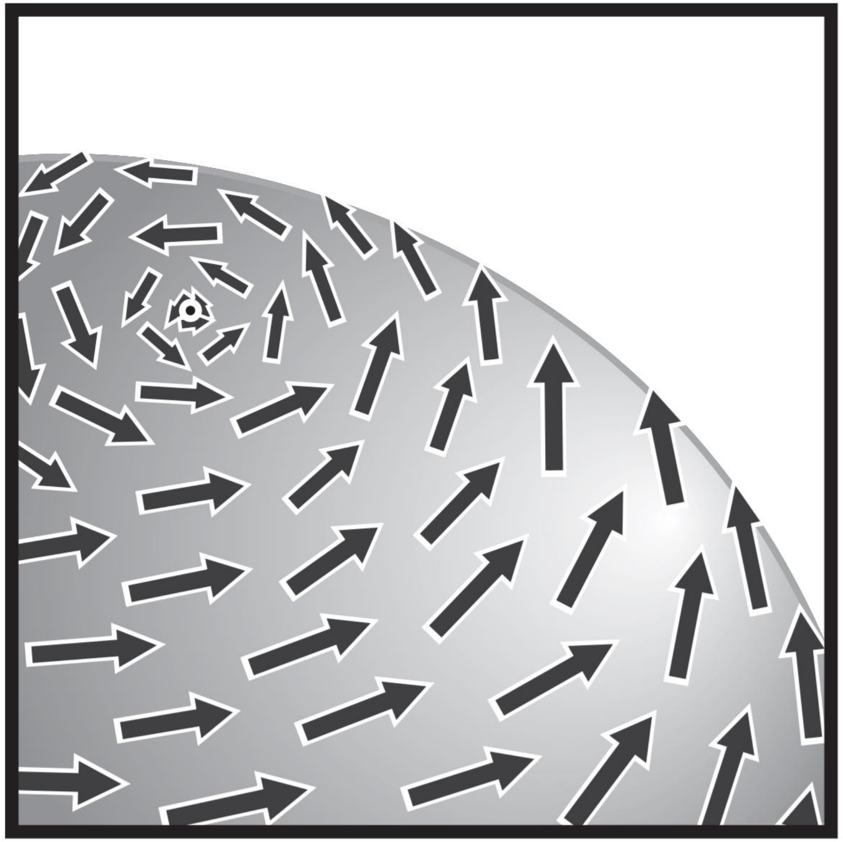
Euler Characteristic is Additive!



- $\chi(A \cup B) = \chi(A) + \chi(B) - \chi(A \cap B)$
- Watch out for open/closed





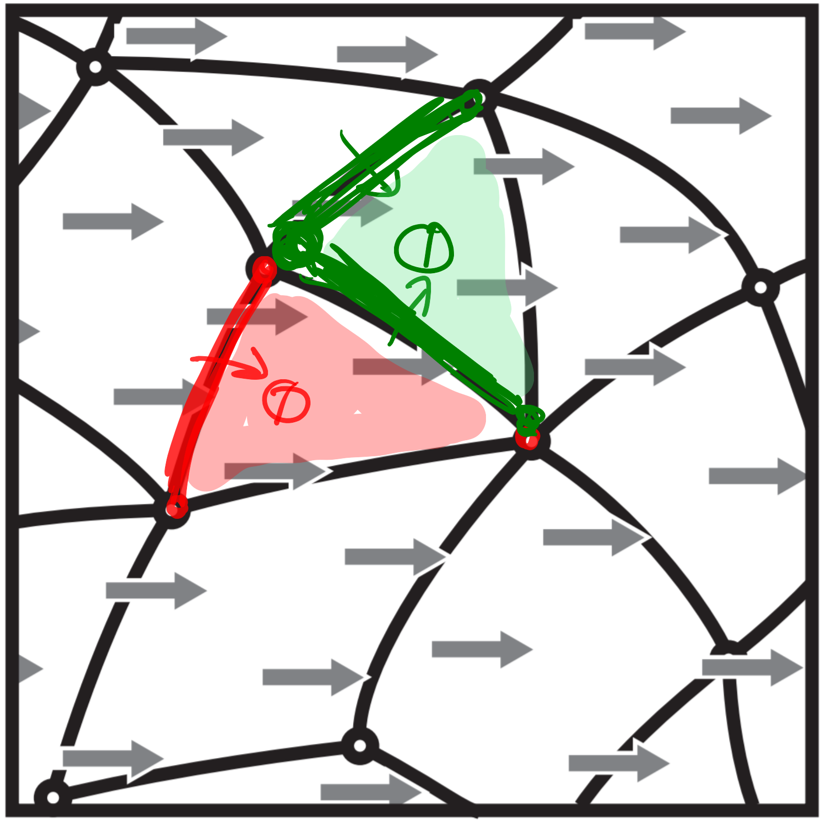


Hedgehog Theorem

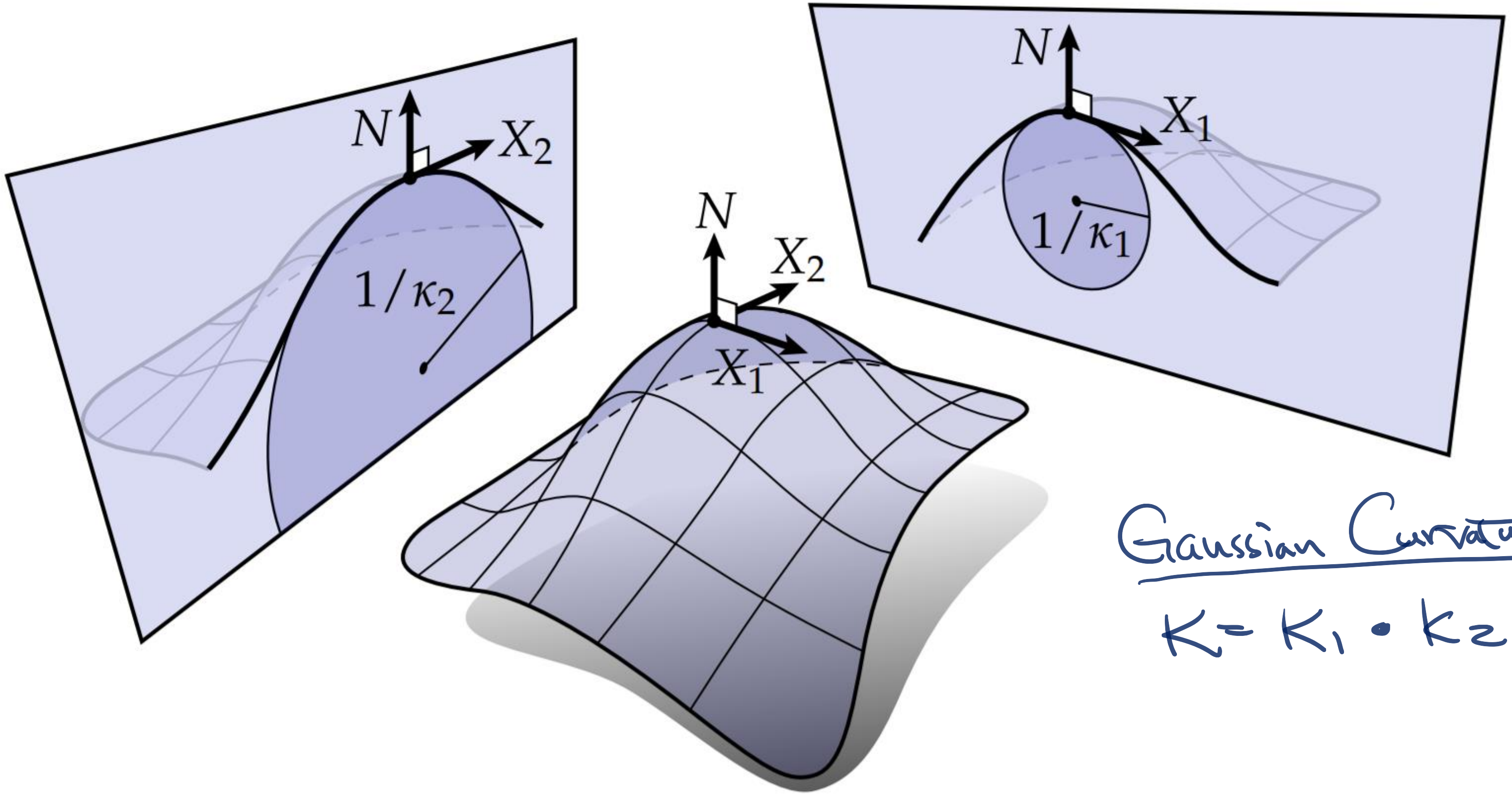
[Poincaré 1885] [Brouwer 1912]

There is no non-vanishing continuous vector field on closed surfaces of non-zero Euler characteristics.

Euler Calculus Proof



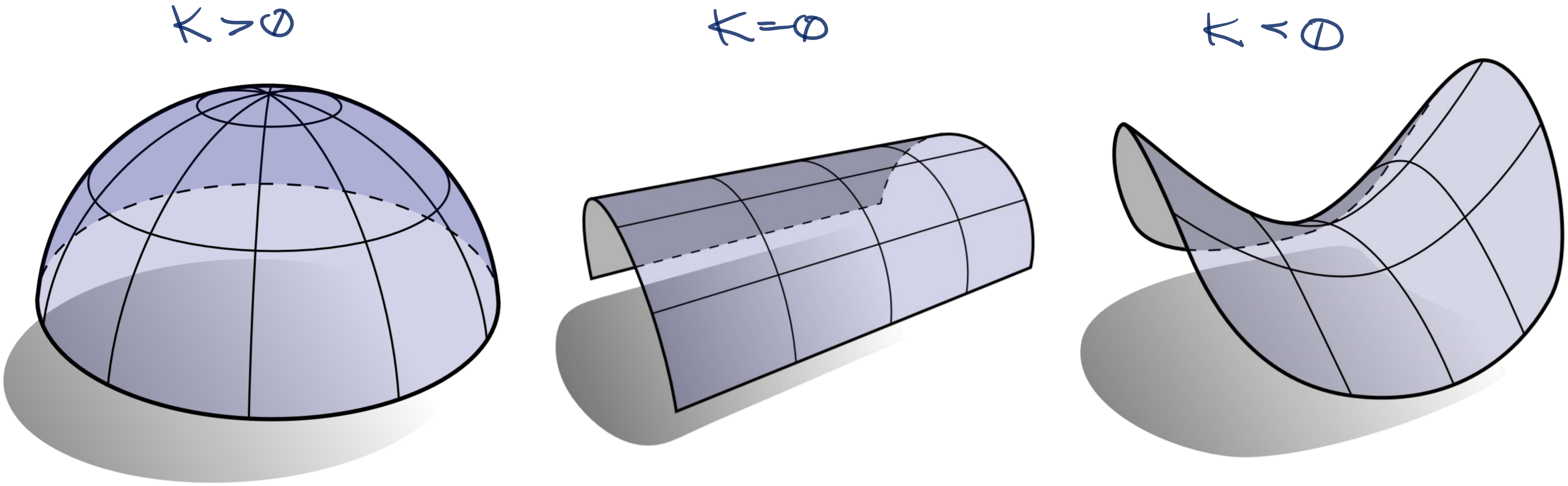
Curvature

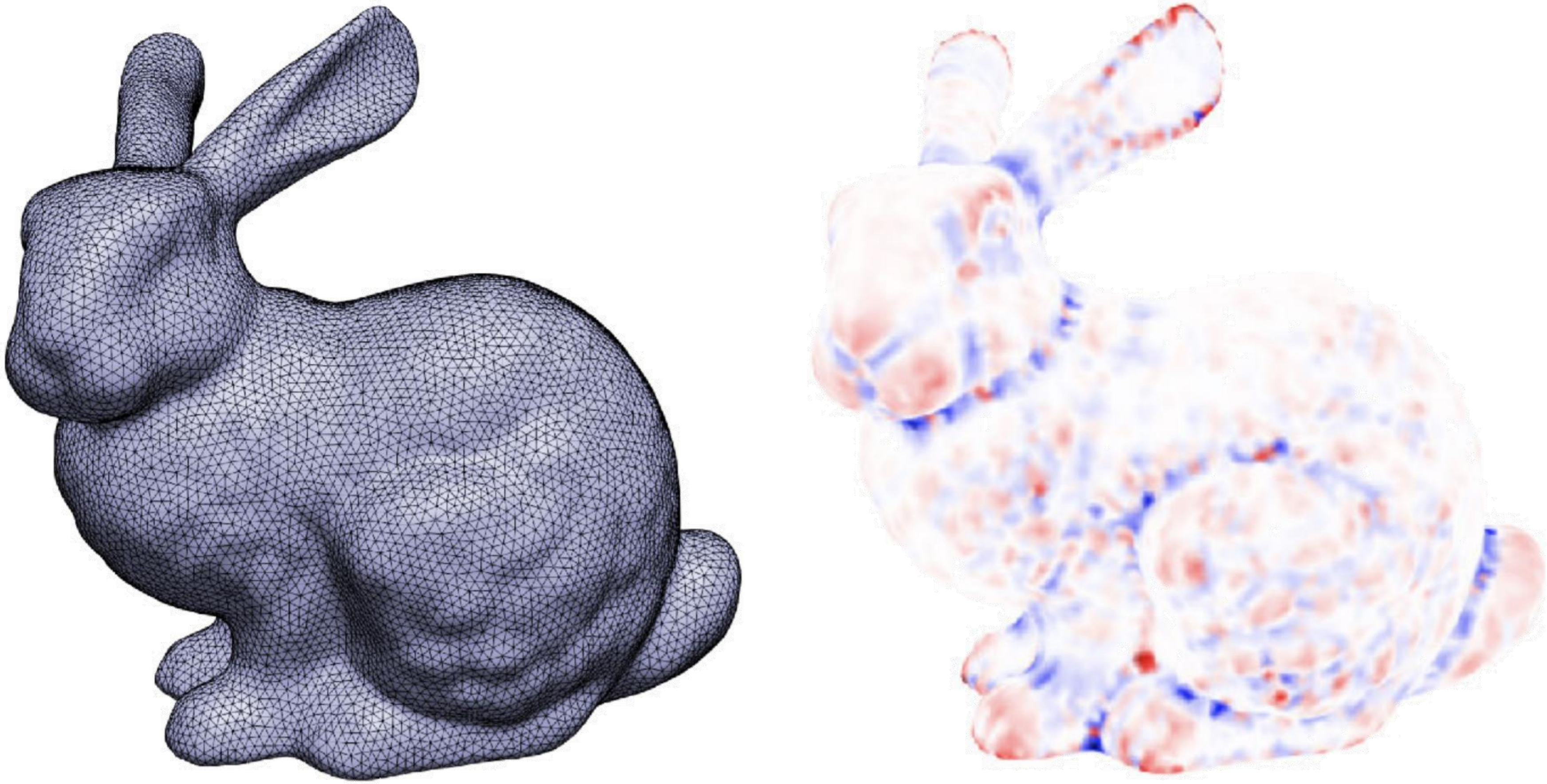


Gaussian Curvature

$K = K_1 \cdot K_2$

Curvature





Gauss-Bonnet Theorem

[Gauß 1827] [Bonnet 1848]

For any surface Σ of Euler char. χ and curvature κ,

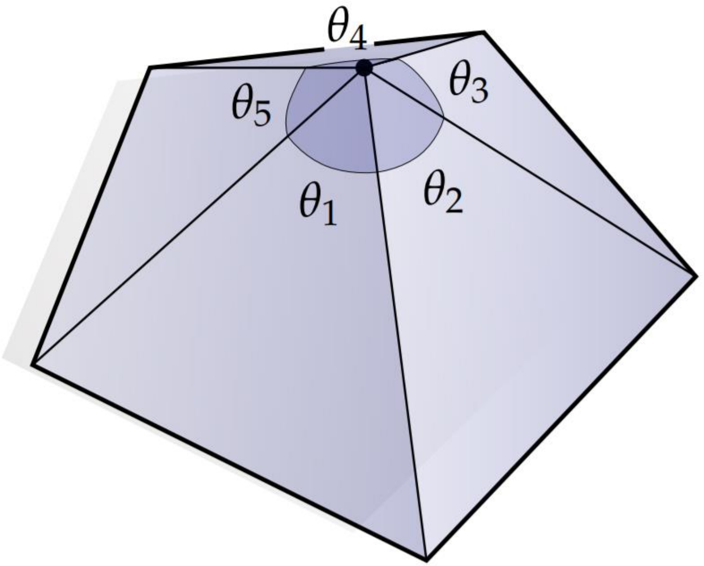
$$\int_\Sigma \kappa \, d\sigma = 2\pi\chi$$

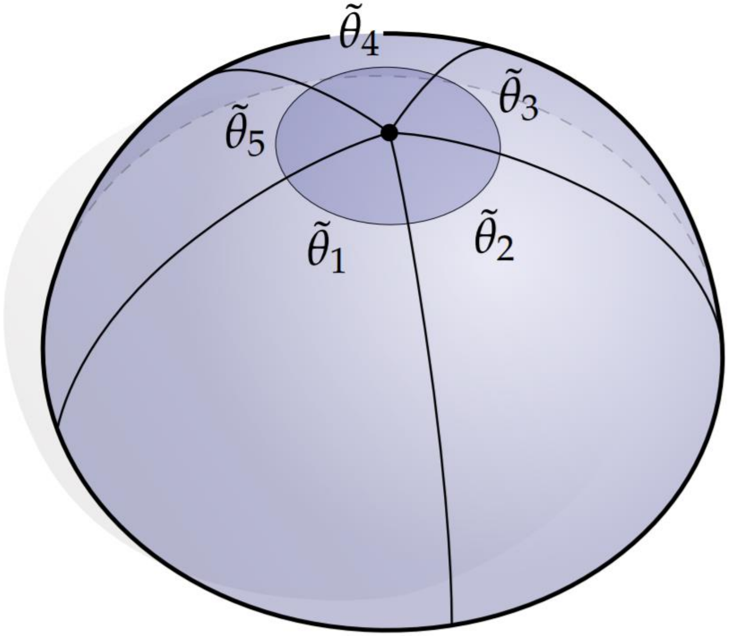
Curvature

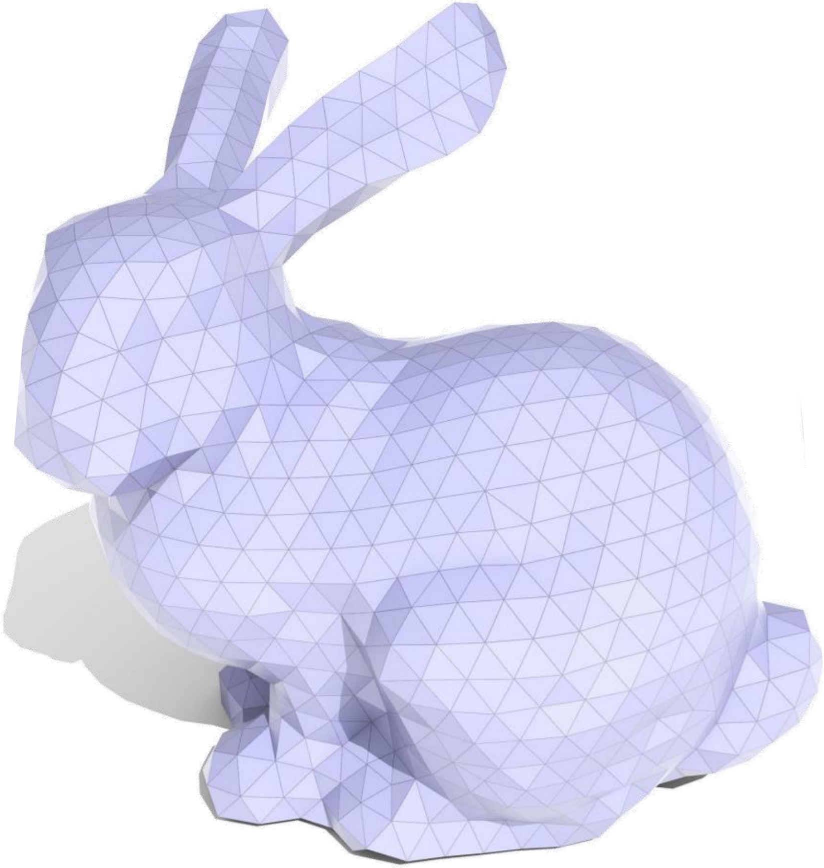
- Curvature $\kappa_\theta(x)$

$$K_\theta(x) = 2\pi - \sum_i \theta_i$$

angle defect







Discrete Gauss-Bonnet Theorem

For any discrete surface Σ of Euler char. χ and angles θ,

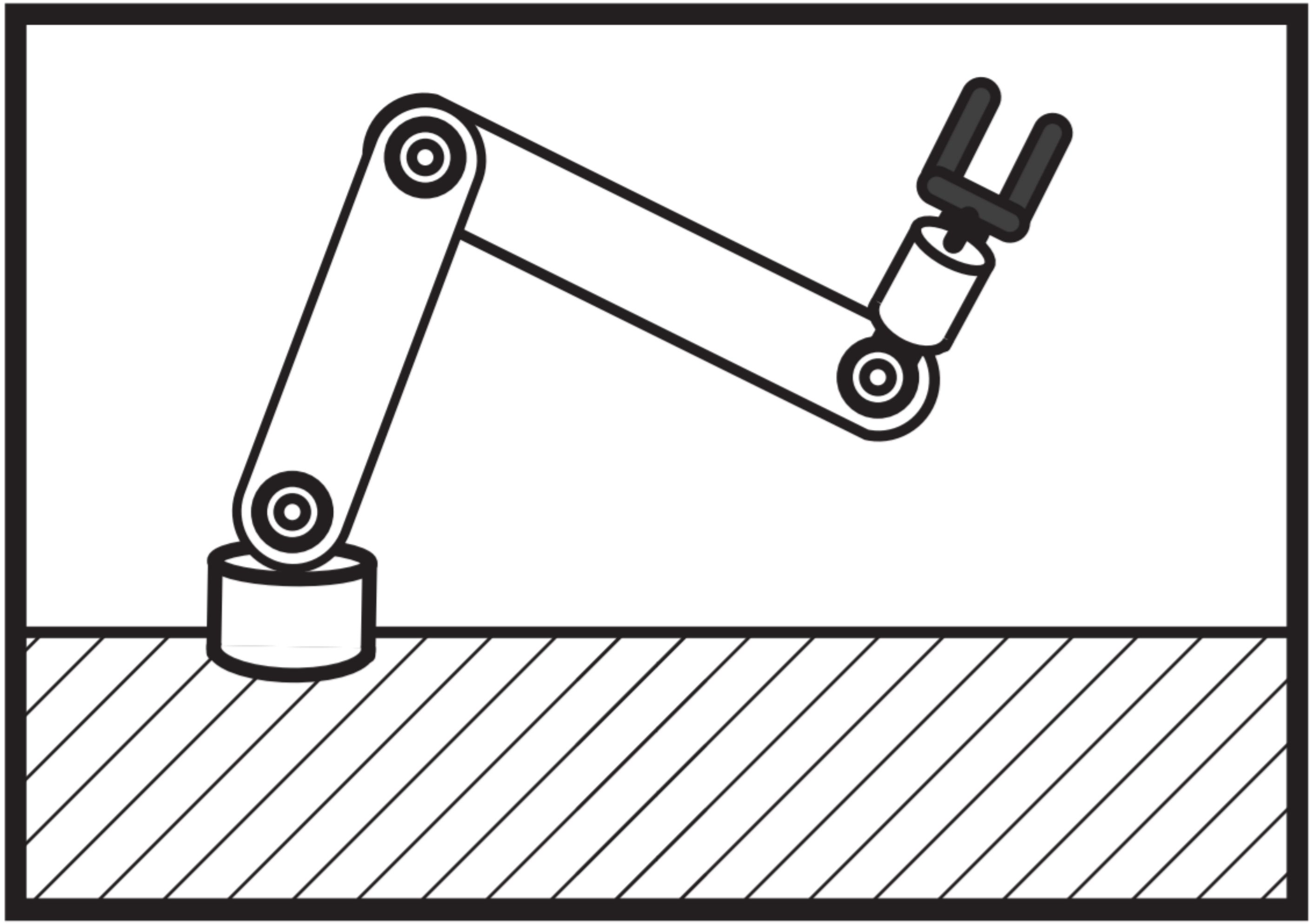
$$\sum_{x} \kappa_\theta(x) = 2\pi\chi$$

Curvatures can be Moved Around, but not Removed

To think about later.

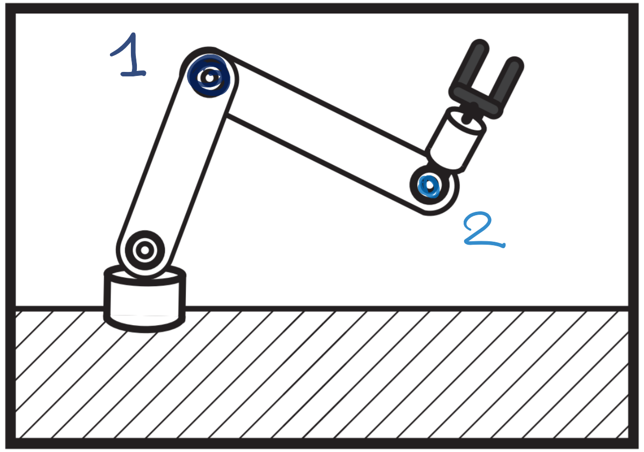
Can you prove Hedgehog Theorem using Gauss-Bonnet?

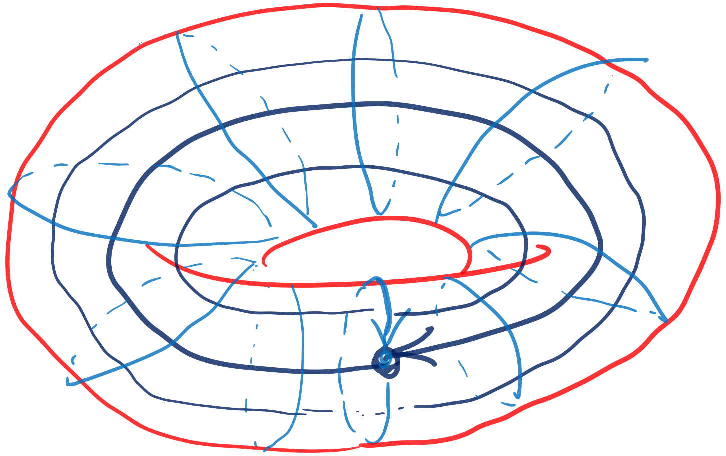
Configuration Space and Complex

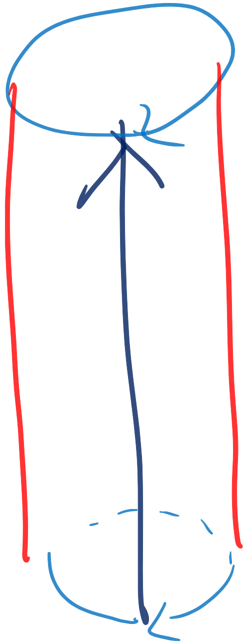


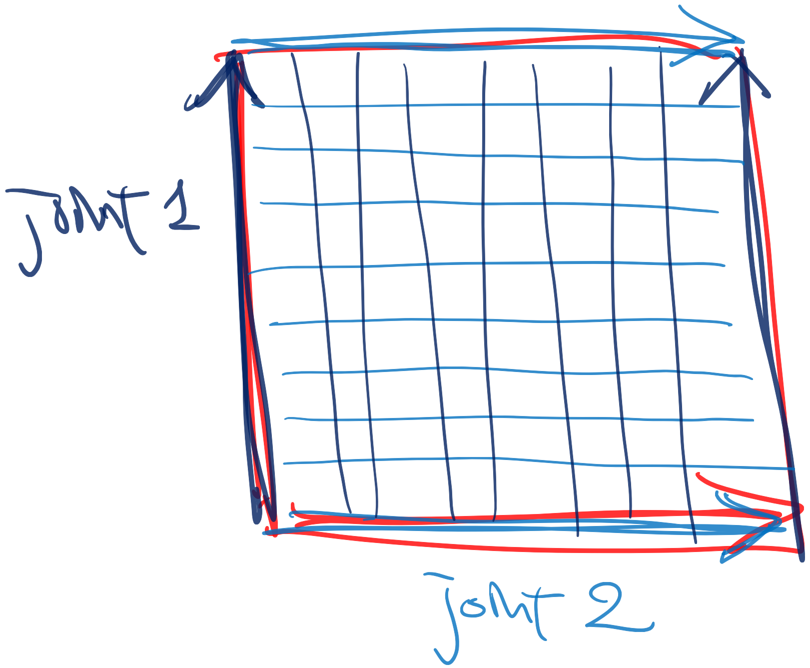
Example: Robot Arms

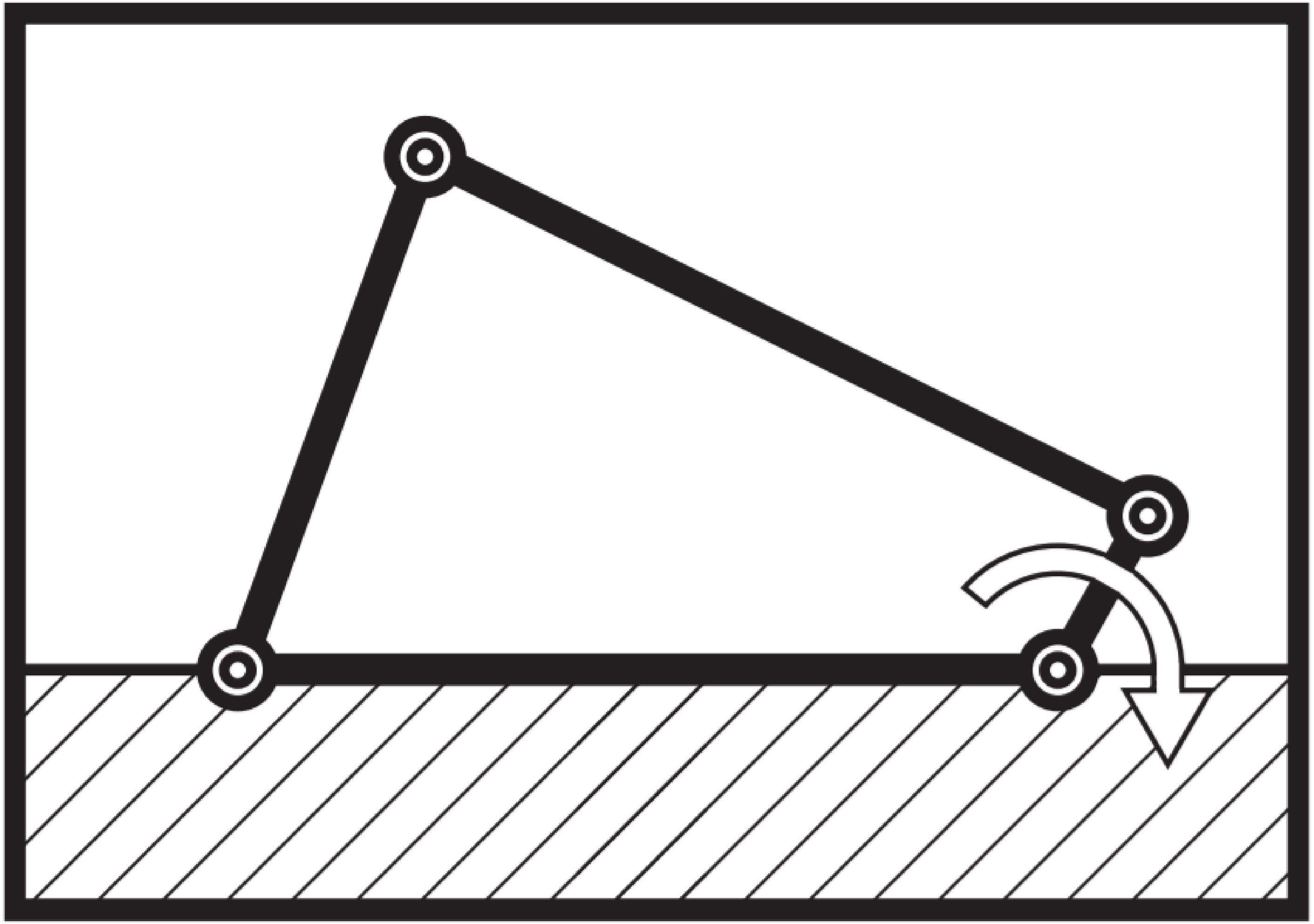
Example: Robot Arms





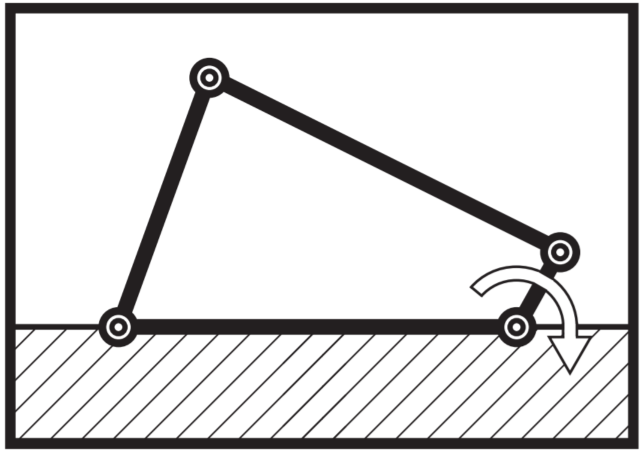


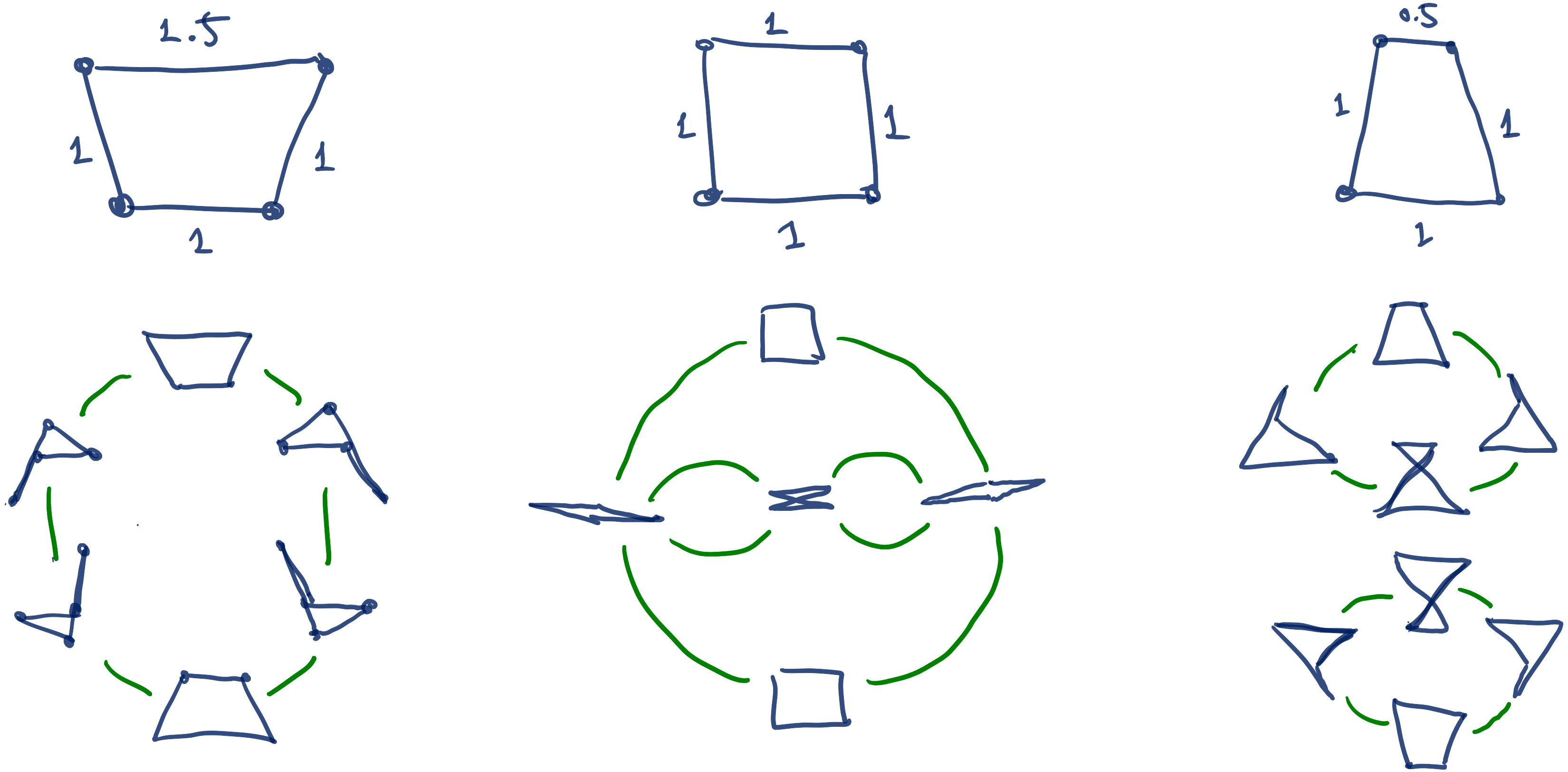


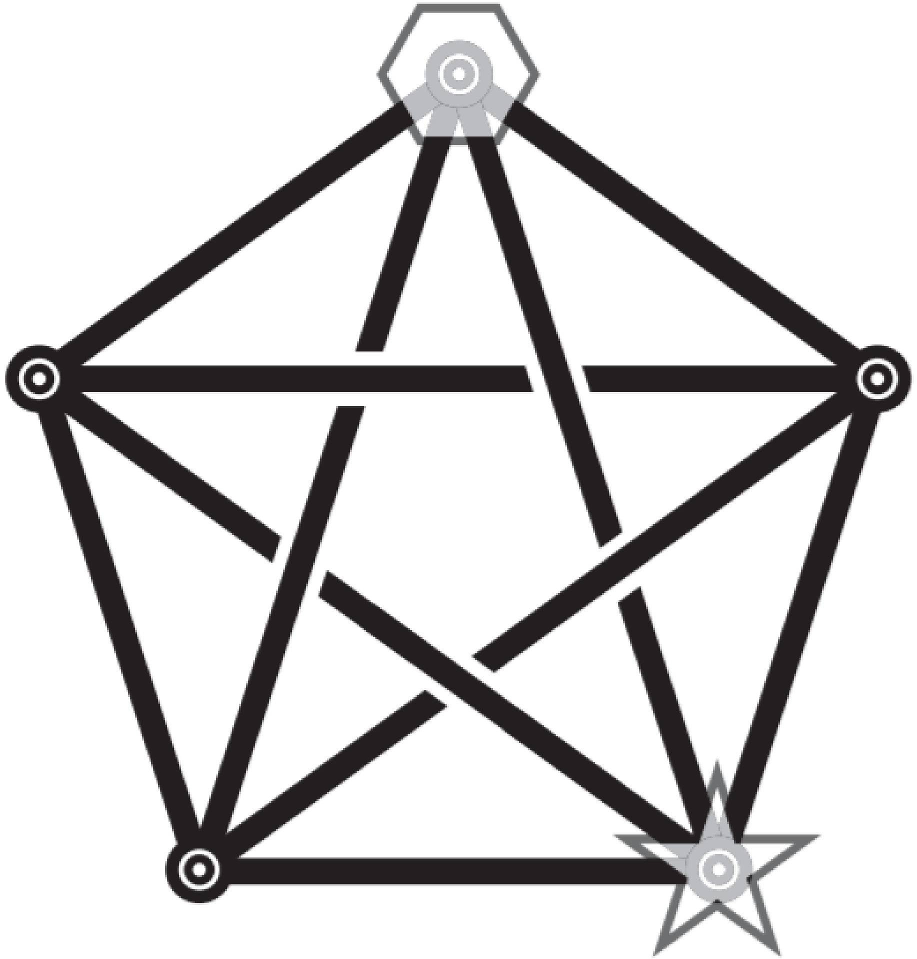


Example: 4-Bar Linkage

Example: 4-Bar Linkage

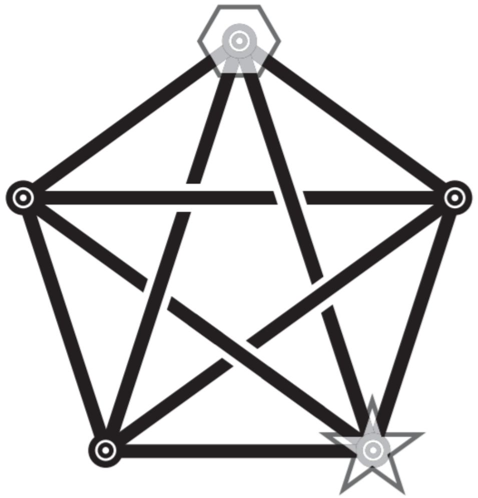






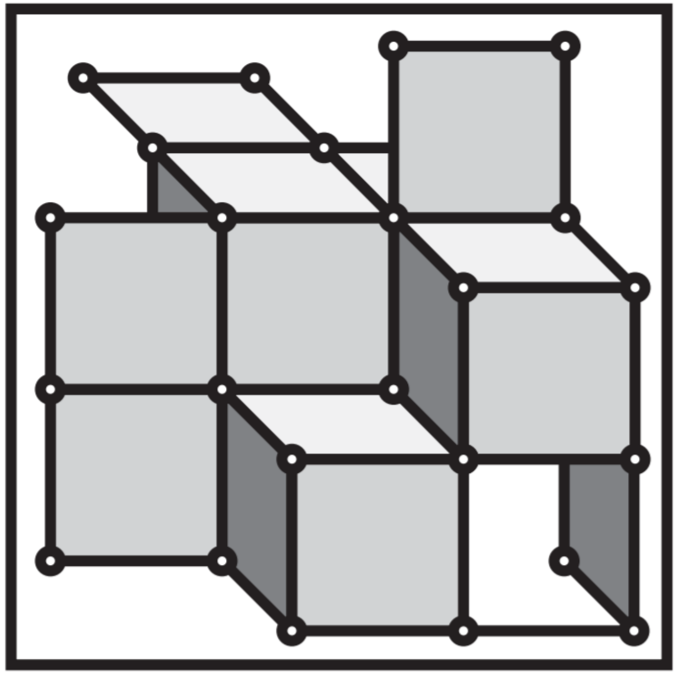
Example: Space of Vertex Pairs

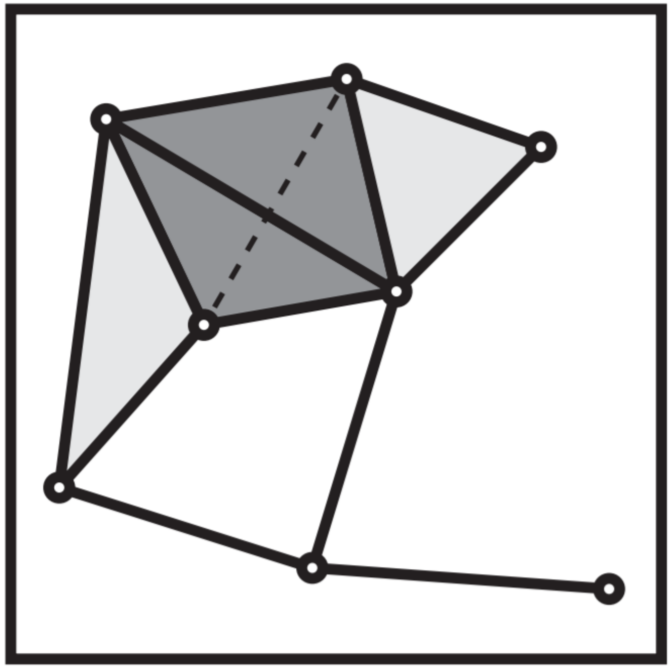
Example: Space of Vertex Pairs



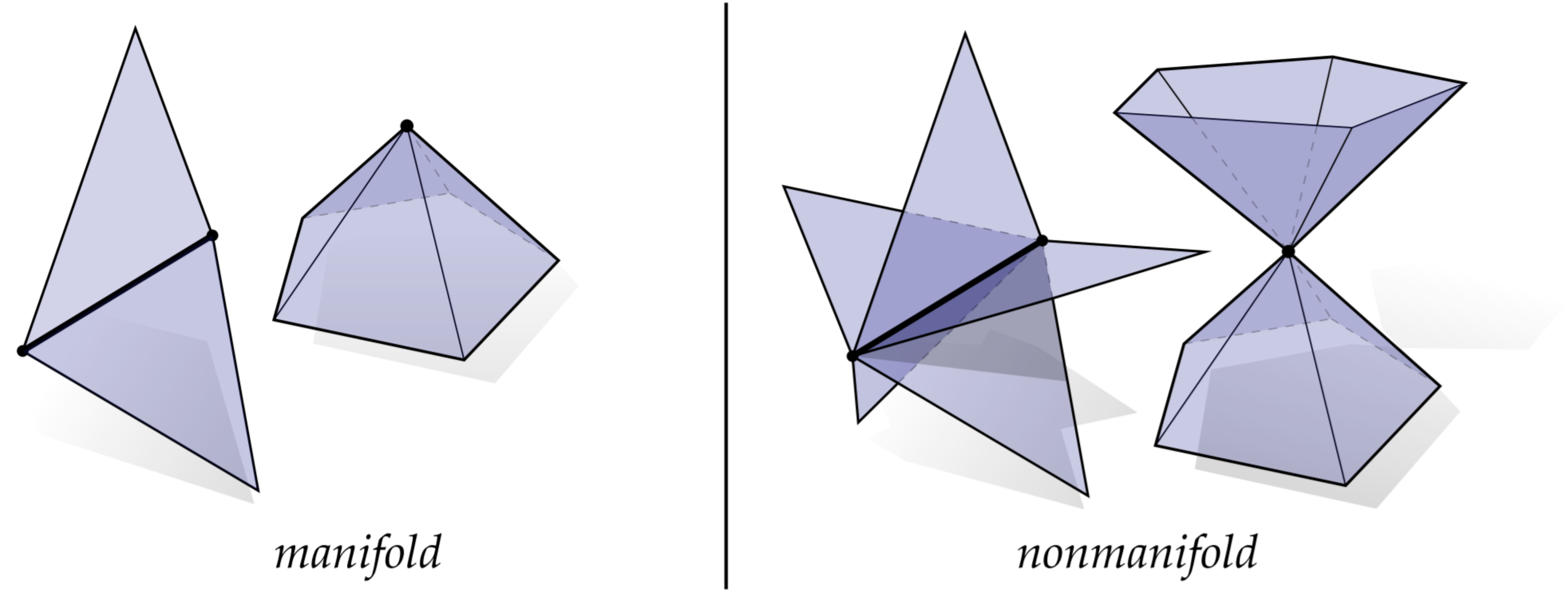
Complexes

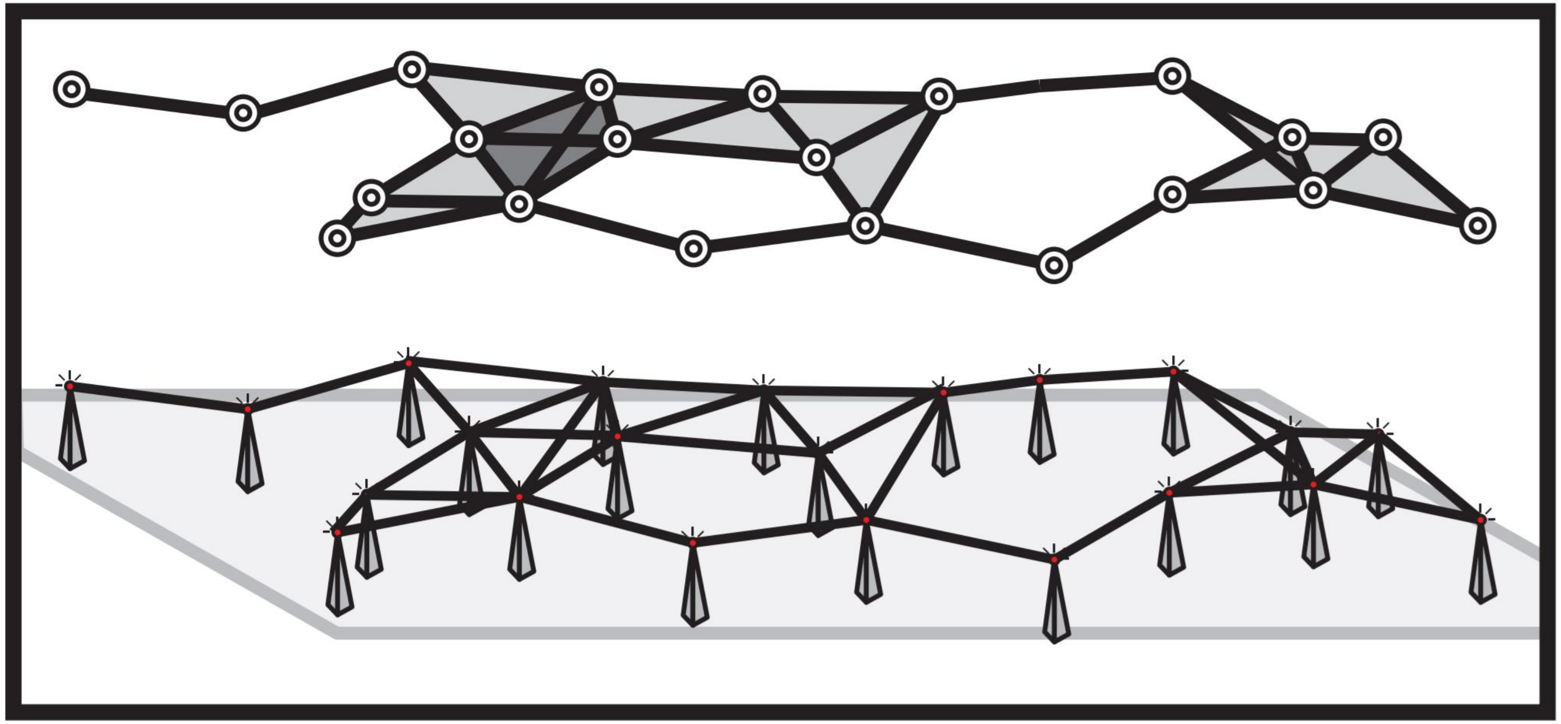
- Gluing a bunch of simplexes together





Extra Features





Vietoris-Rips Complex

- Connect any two points of distance at most r
- Add all simplexes inside a clique

Closing Q. How do we Tell Apart Complexes?

Choose your own adventure:

(A) applications of curves and surfaces: Linkage and Folding, or
(B) behavior of closed curves dictates the shape of space