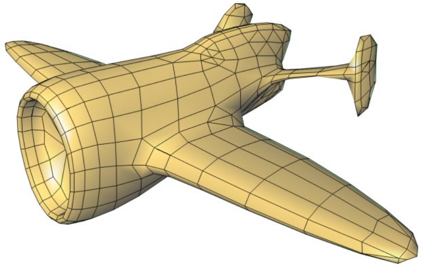
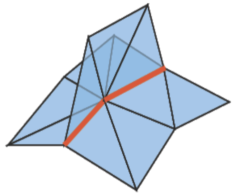
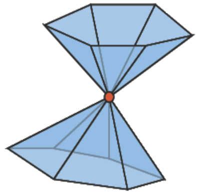
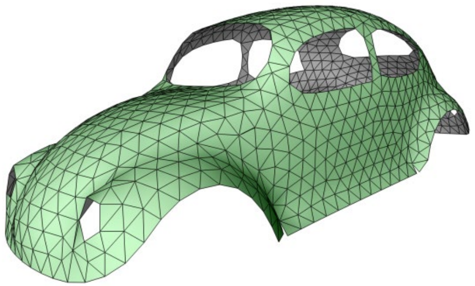
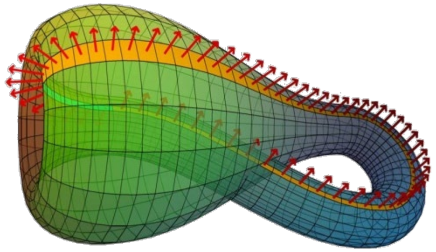






Basic Concepts



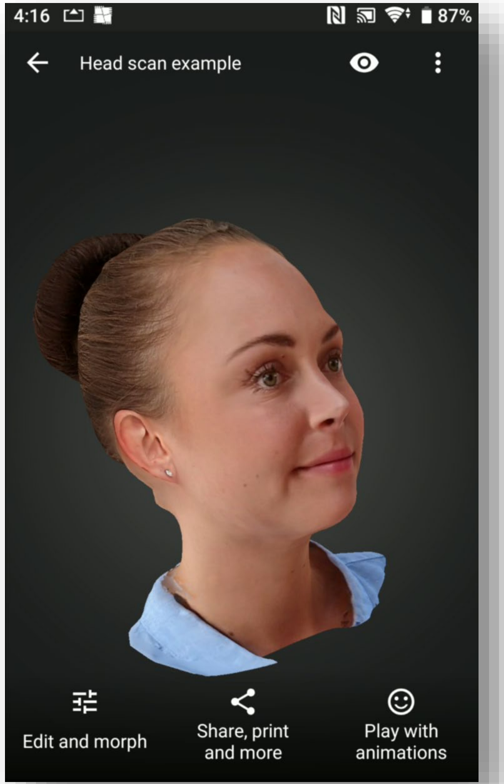


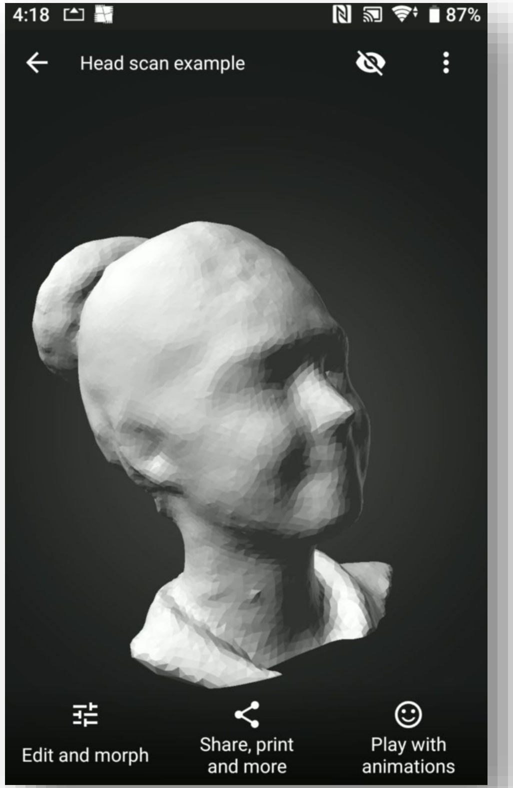
Last Time

Cool applications

Today

How is the 3D data represented computationally?

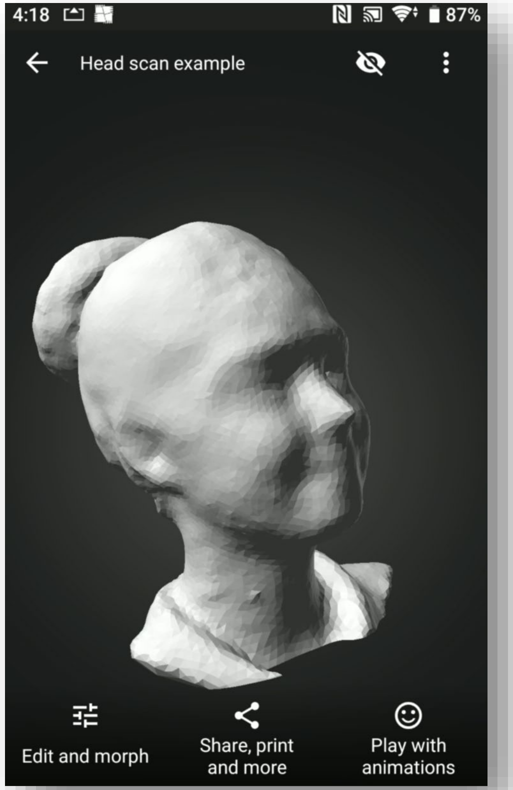




Motivation

Requirements:

- Efficient = surface only
- arbitrary shapes
  - Holes, boundaries, sharp creases
- more data = better accuracy
- adaptive



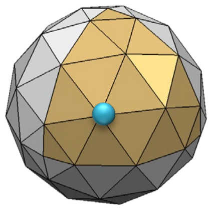
Triangle Meshes

- Connectivity: vertices, edges, triangles

$$\mathcal{V} = \{v_1, \ldots, v_n\}$$

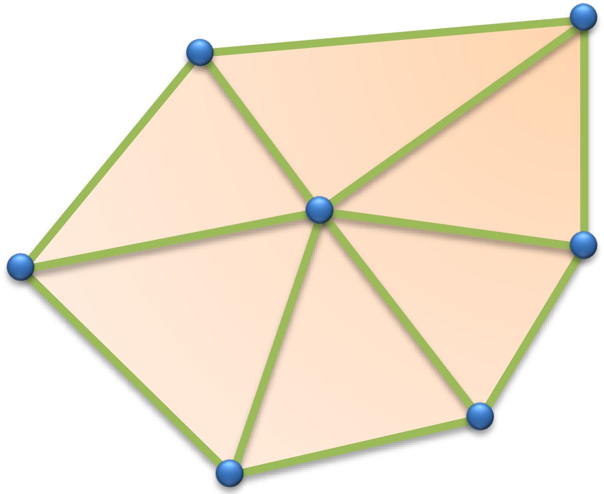
$$\mathcal{E} = \{e_1, \ldots, e_k\} \;, \quad e_i \in \mathcal{V} \times \mathcal{V}$$

$$\mathcal{F} = \{f_1, \ldots, f_m\} \;, \quad f_i \in \mathcal{V} \times \mathcal{V} \times \mathcal{V}$$



- Geometry: vertex positions

$$\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\} \;, \quad \mathbf{p}_i \in \mathbb{R}^3$$



Polygonal ~~Triangle~~ Meshes

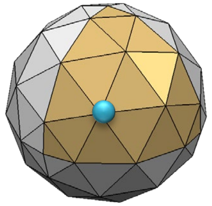
- Connectivity: vertices, edges, triangles

$$\mathcal{V} = \{v_1, \ldots, v_n\}$$

$$\mathcal{E} = \{e_1, \ldots, e_k\} \;, \quad e_i \in \mathcal{V} \times \mathcal{V}$$

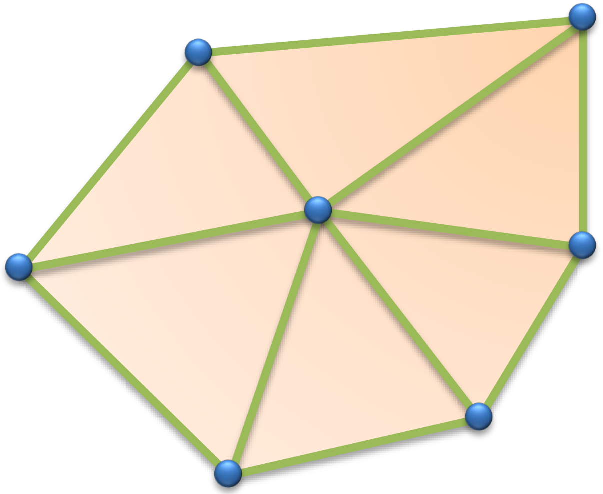
$$\mathcal{F} = \{f_1, \ldots, f_m\} \;, \quad \cancel{f_i \in \mathcal{V} \times \mathcal{V} \times \mathcal{V}}$$

$$f_i \in \mathcal{V} \times \mathcal{V} \times \ldots \times \mathcal{V}$$

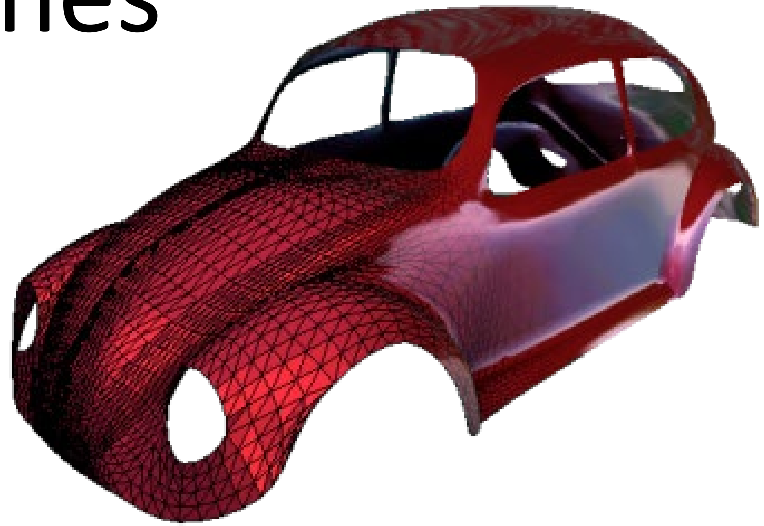


- Geometry: vertex positions

$$\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\} \;, \quad \mathbf{p}_i \in \mathbb{R}^3$$

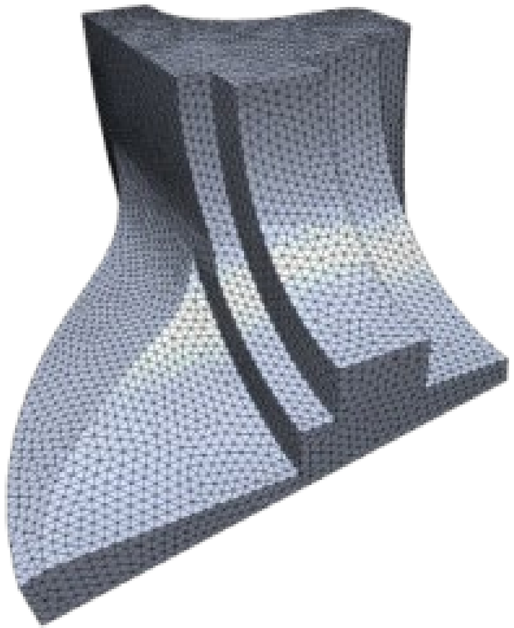


Polygonal Meshes

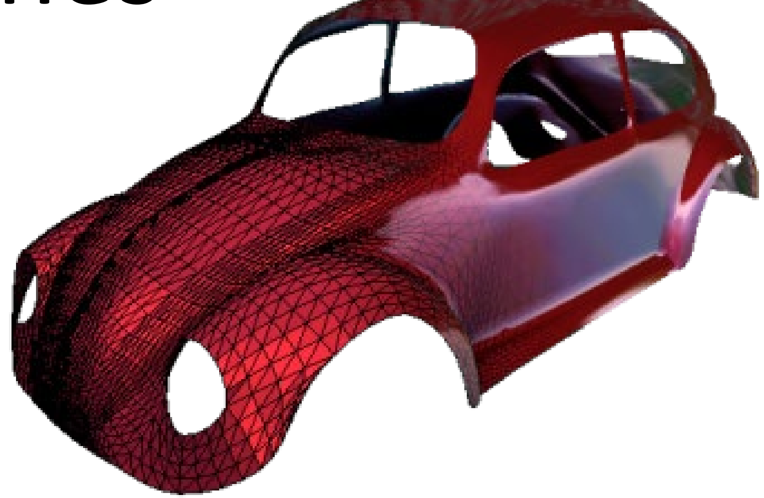


A good representation:

- arbitrary topology (holes, boundaries)
- piecewise smooth surfaces

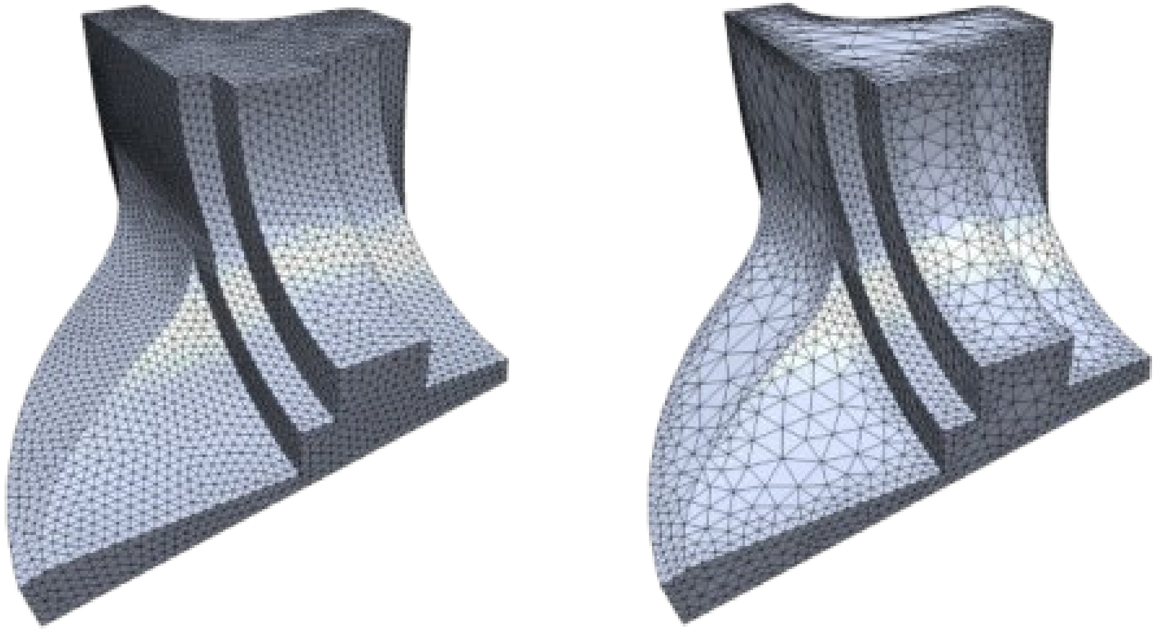


Polygonal Meshes

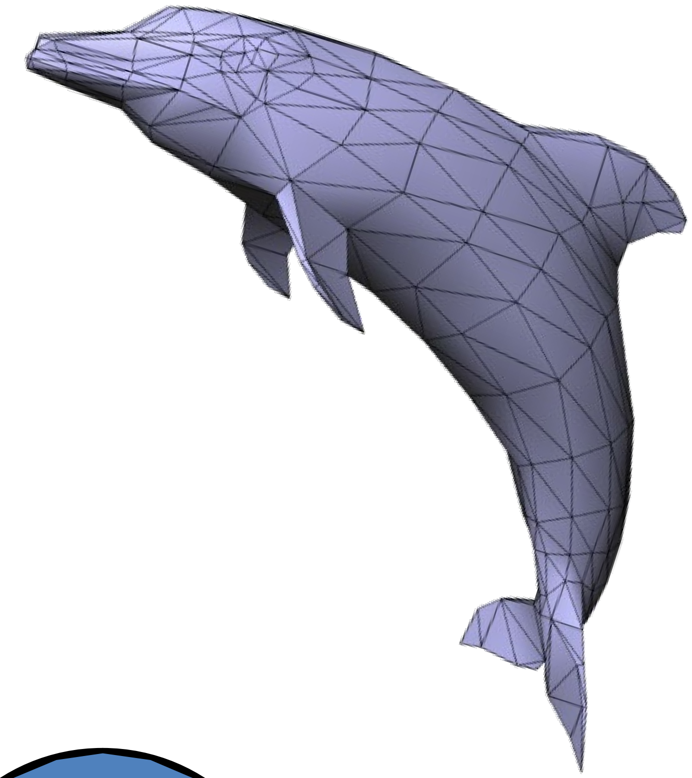


A good representation:

- arbitrary topology (holes, boundaries)
- piecewise smooth surfaces
- adaptive refinement
- efficient rendering

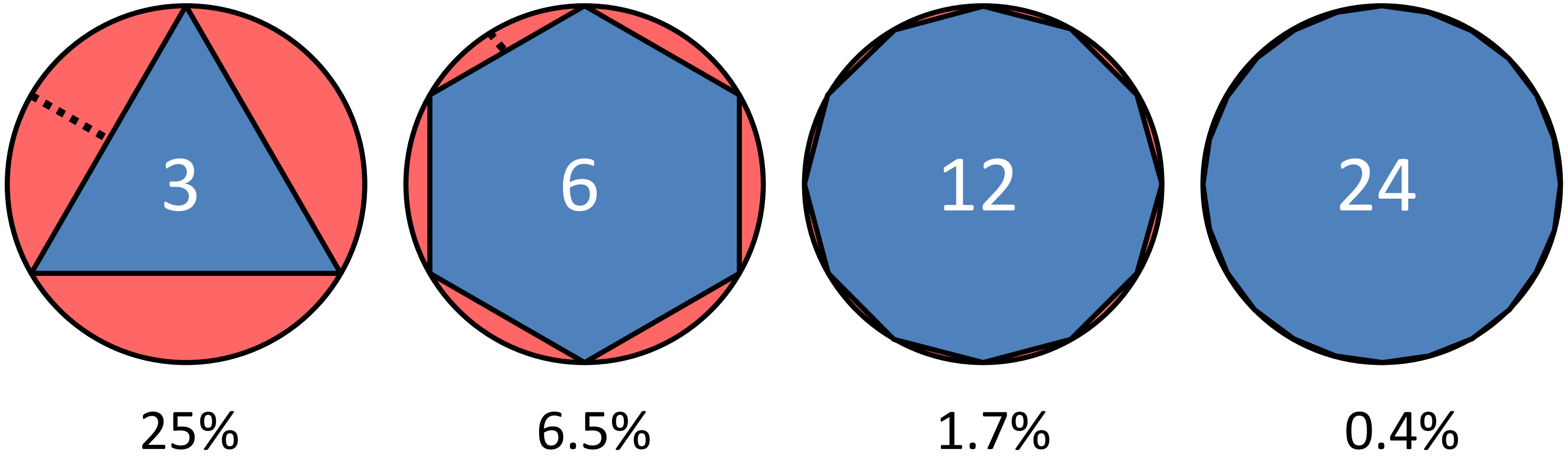


Polygonal Meshes

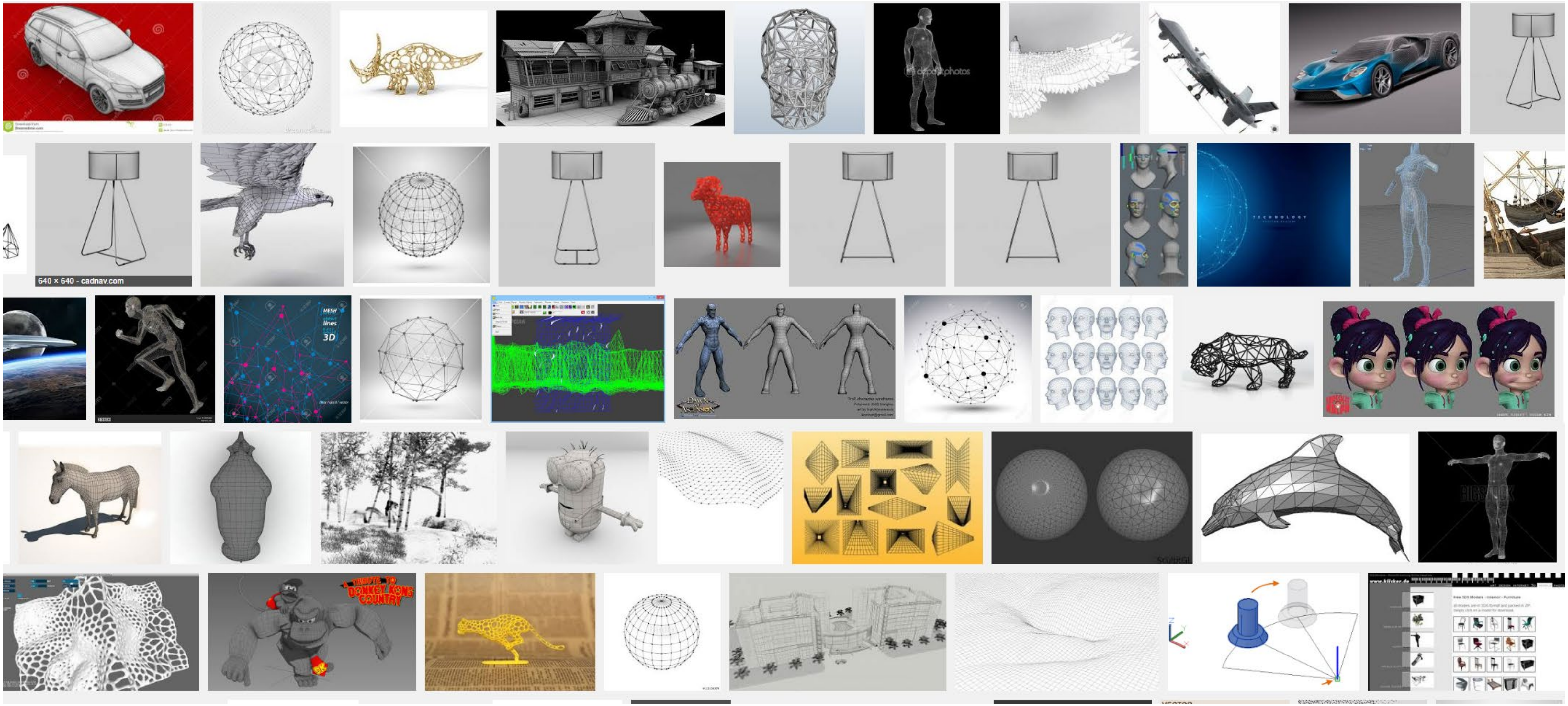


Piecewise linear approximation

- More data = better accuracy
- Error is $O(h^2)$
  - *Edge length divided by 2 → Error divided by 4*



Available Meshes



"Good" Meshes

Real time compression of **triangle mesh** connectivity
S Gumhold, W Straßer - Proceedings of the 25th annual conference on …, 1998 - dl.acm.org
... To allow the encoding of an arbitrarily connected and **oriented triangle mesh** in one run, several
basic ... of the current cut-border part is known during compression and decompression, the "**close**
cut-bc
Cited

The ball-pivoting algorithm for surface reconstruction
F Bernardini, J Mittleman, H Rushmeier… - IEEE transactions on …, 1999 - ieeexplore.ieee.org
... By checking that **triangle** and data point normals are consistently **oriented**, we avoid generating a **triangle** in this **...** In all cases, the BPA outputs an **orientable**, triangulated **manifold**. **...** other potential seed **triangles**, leading to the construction of small sets of **triangles** lying **close** to the **...**
Cit

Fair morse functions for extracting the topological structure of a surface **mesh**
X Ni, M Garland, JC Hart - ACM Transactions on Graphics (TOG), 2004 - dl.acm.org
... Its values are defined on the vertices of an **oriented** 2- **manifold triangle mesh** M, and **...** Such **manifolds** contain one or more boundaries each repre- sented by a simple **closed** loop of **...** properties of the field, and depend on a smooth and well-shaped function over the **manifold**. **...**
Cited by 201 Related articles All 13 versions Web of Science: 61 Cite Save

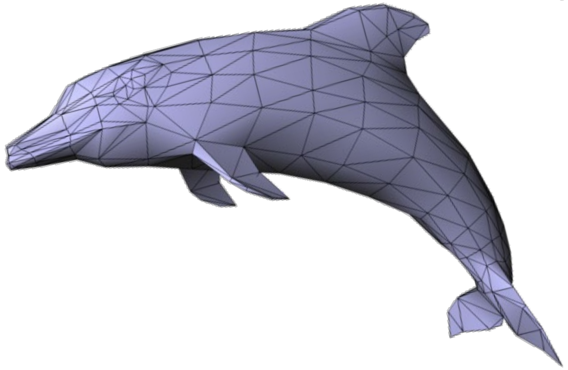
Tasks / Applications

- Determine inside/outside
  - Collision detection
    - Animation
    - Mesh design
    - Fabrication

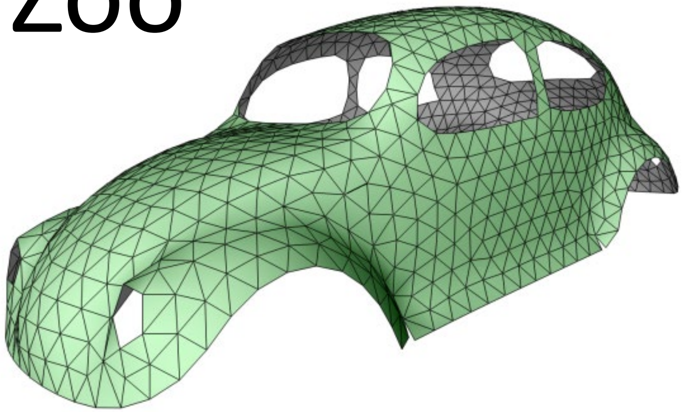
From: [Computational Design of Reconfigurables, Garg et al., 2016]

- Find boundaries
  - Preserve boundaries during processing
  - Set boundary values for numerically solving differential equations

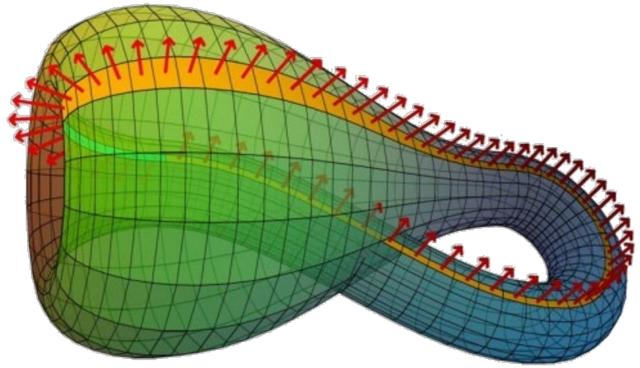
Mesh Zoo



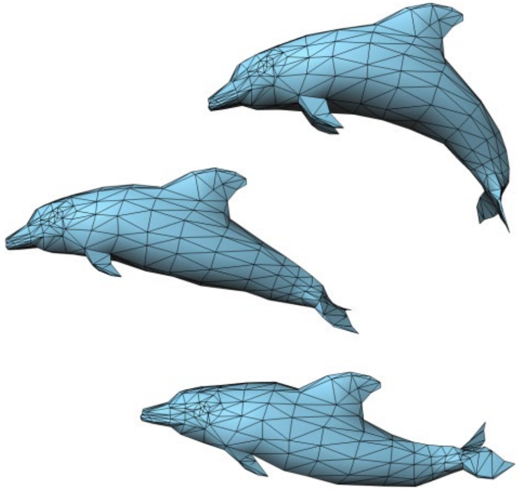
Single component, closed, triangular, orientable manifold



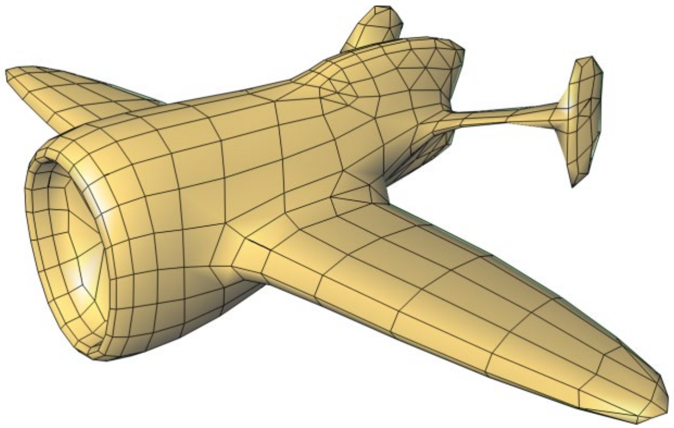
With boundaries



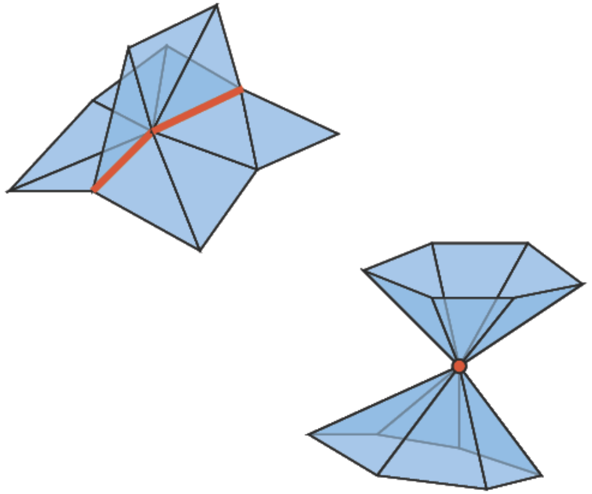
Not orientable



Multiple components



Not only triangles

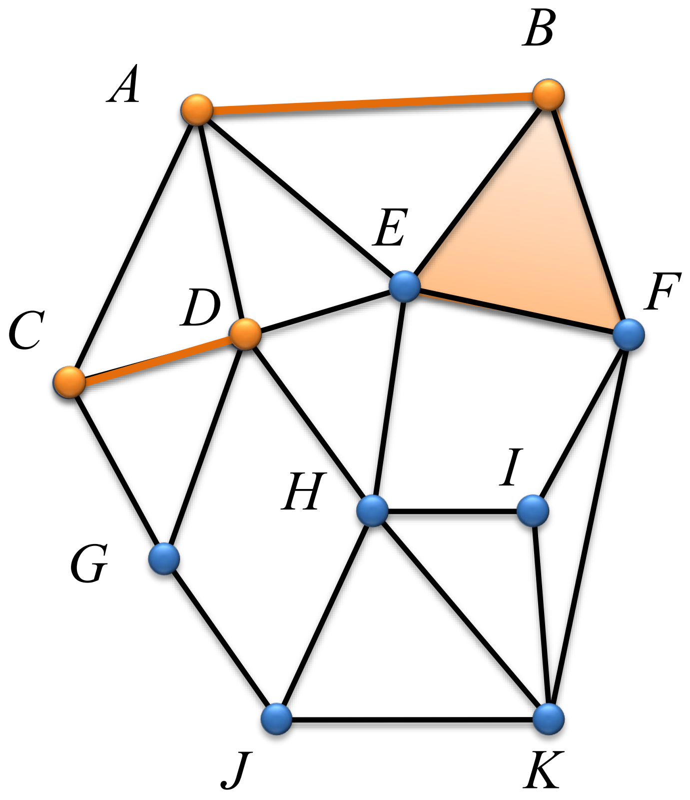


Non manifold

Mesh Definitions

- Need vocabulary to describe zoo meshes
- The connectivity of a mesh is just a graph
- We’ll start with some graph theory

Graph Definitions



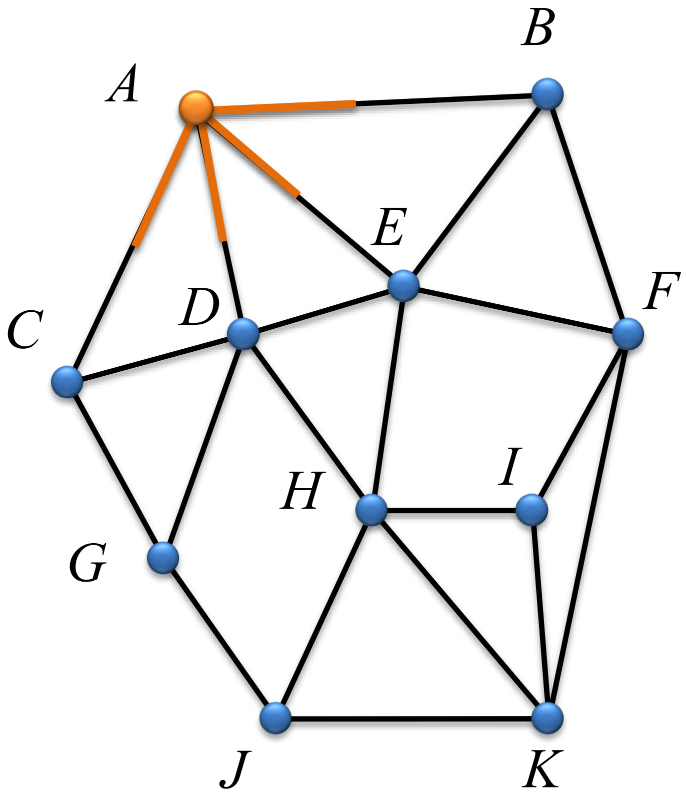
G = graph = $\langle V, E \rangle$

V = vertices = $\{A, B, C, \ldots, K\}$

E = edges = $\{(AB), (AE), (CD), \ldots\}$

F = faces = $\{(ABE), (DHJG), \ldots\}$

Graph Definitions



Vertex degree or **valence** =
number of incident edges

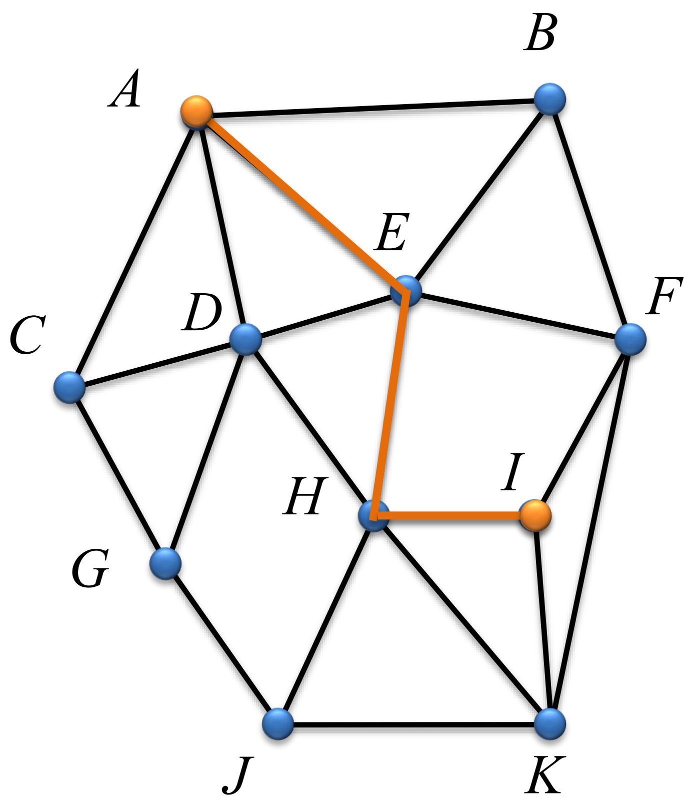
$\deg(A) = 4$

$\deg(E) = 5$

Regular mesh =

all vertex degrees are equal

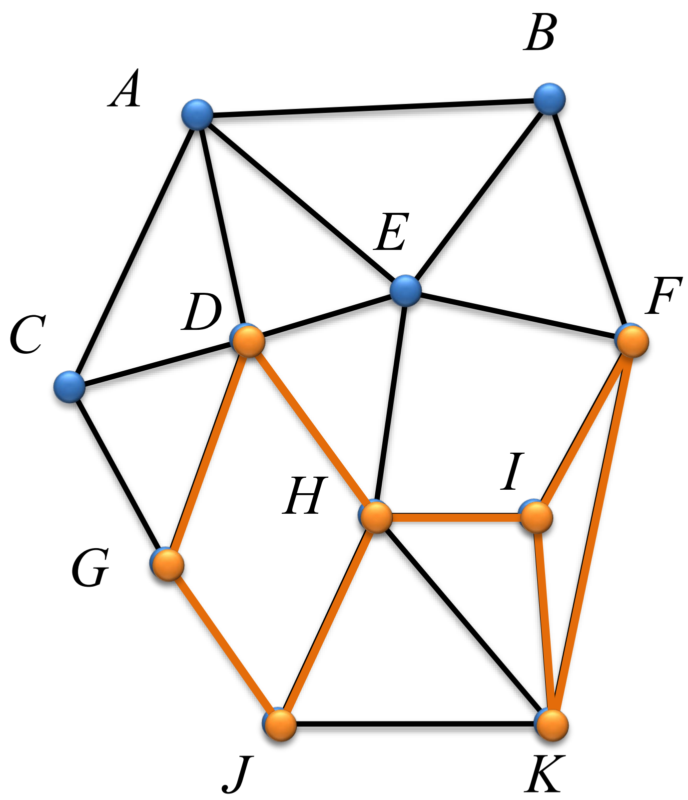
Connectivity



Connected =

path of edges connecting every two vertices

Connectivity



Connected =

path of edges connecting every two vertices

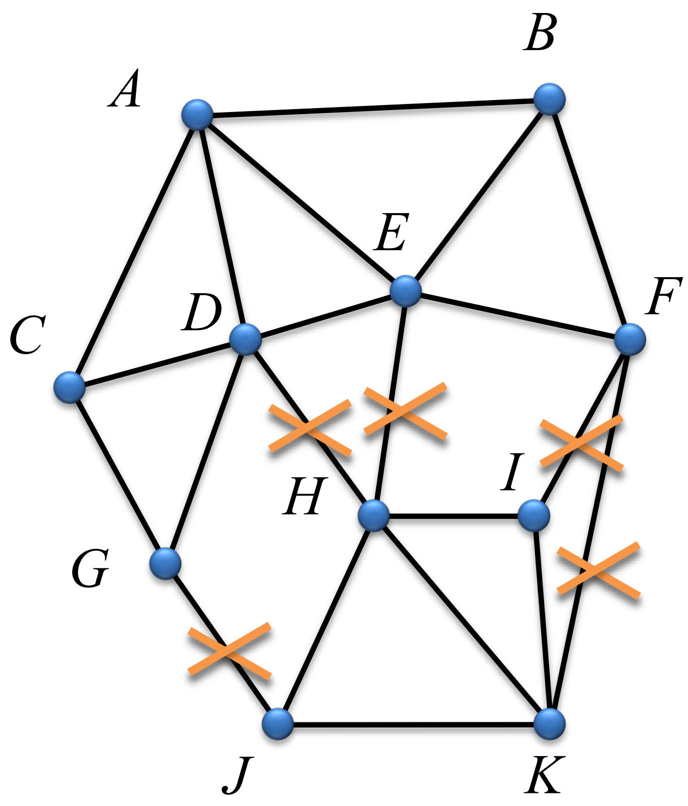
Subgraph =

$G'=<V',E'>$ is a subgraph of $G=<V,E>$ if

V' is a subset of V and

E' is the subset of E incident on V'

Connectivity



Connected =

path of edges connecting every two vertices

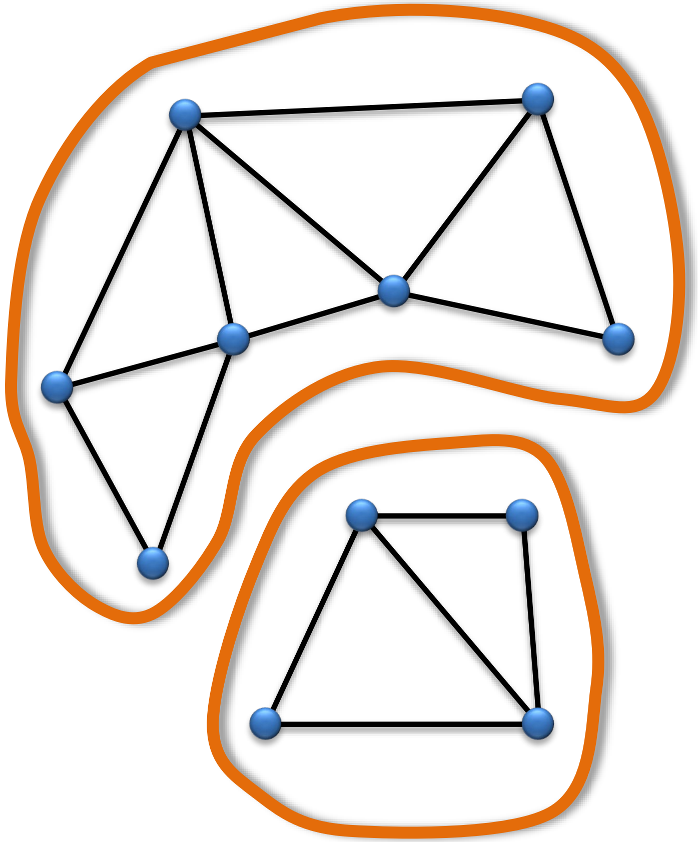
Subgraph =

G'=<V',E'> is a subgraph of G=<V,E> if

V' is a subset of V and

E' is the subset of E incident on V'

Connectivity



Connected =

path of edges connecting every two vertices

Subgraph =

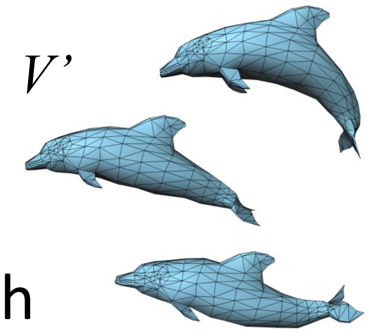
G'=<V',E'> is a subgraph of G=<V,E> if

V' is a subset of V and

E' is the subset of E incident on V'

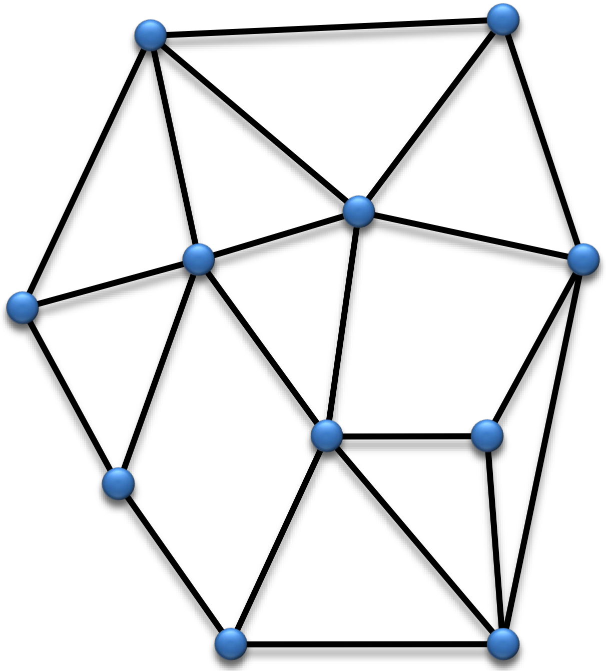
Connected Component =

maximally connected subgraph

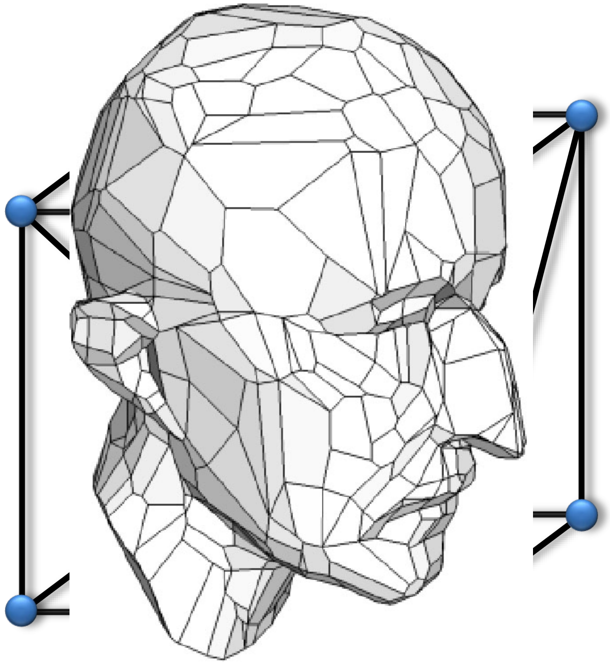


Graph Embedding

Embedding: G is *embedded* in R^d, if each **vertex** is assigned a position in R^d



Embedded in R^2

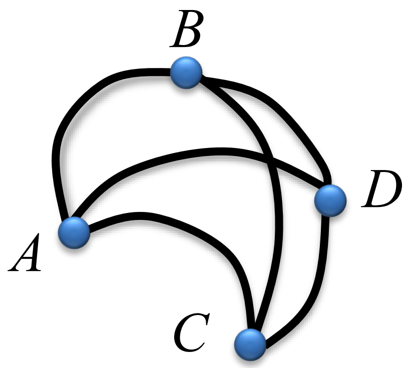


Embedded in R^3

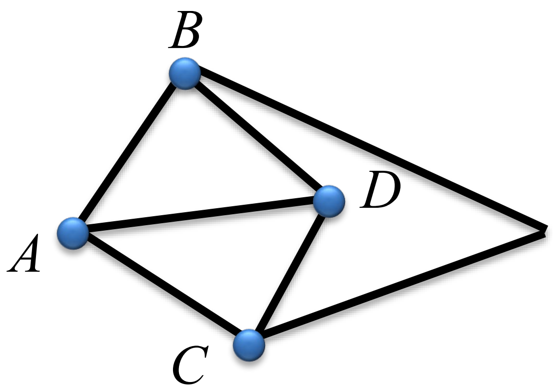
Planar Graphs

Planar Graph =
Graph whose vertices and edges *can be* embedded in R^2 such that its edges *do not intersect*

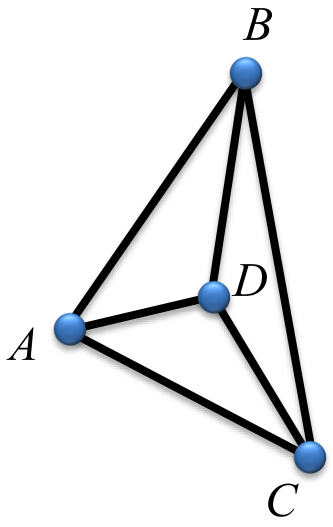
Planar Graph



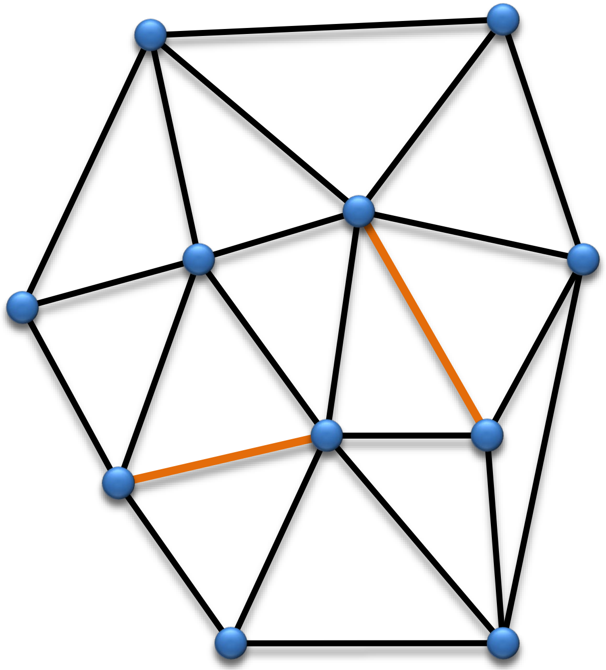
Plane Graph



Straight Line Plane Graph



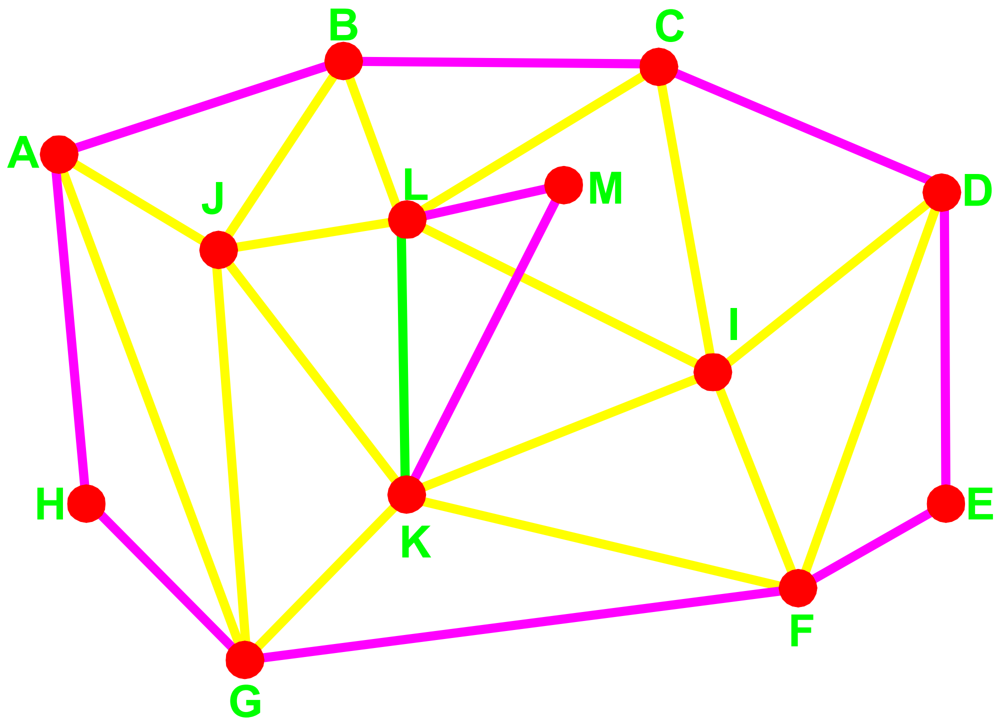
Triangulation



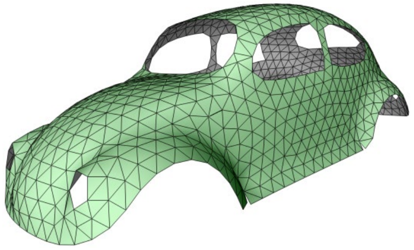
Triangulation:

Straight line plane graph where every face is a *triangle*.

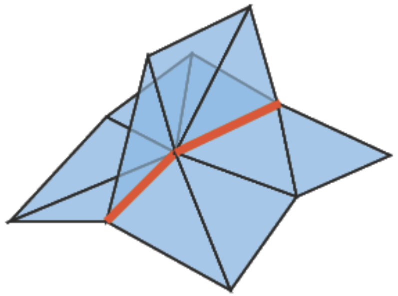
Mesh



Mesh:
straight-line graph embedded in R^3

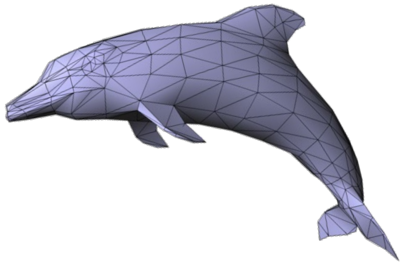


Boundary edge:
adjacent to exactly *one* face



Regular edge:
adjacent to exactly *two* faces

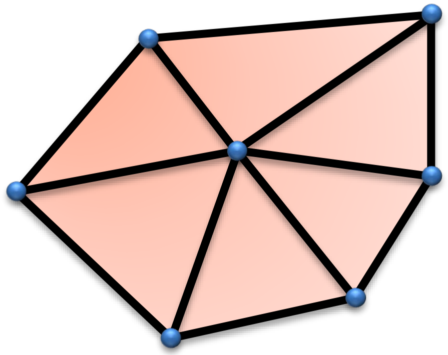
Singular edge:
adjacent to more than two faces

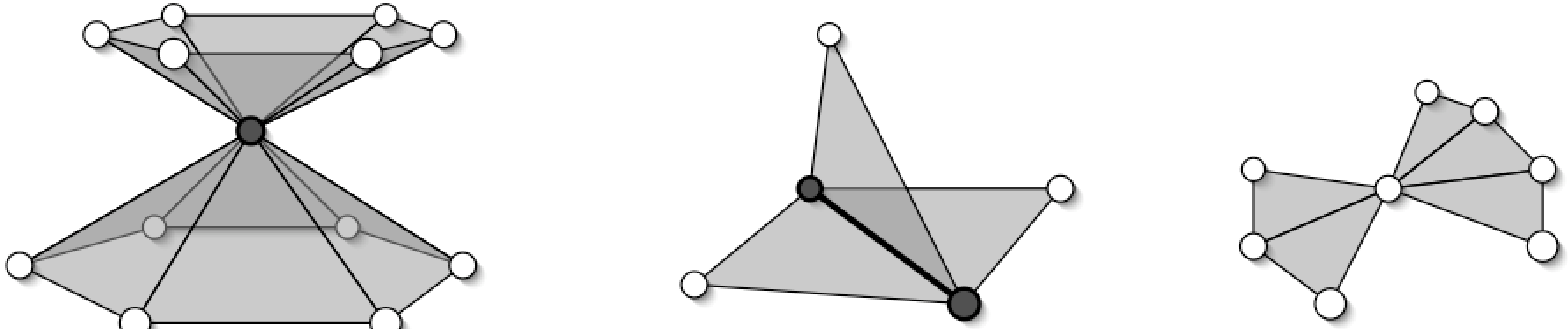


Closed mesh:
mesh with no boundary edges

2-Manifolds Meshes

Disk-shaped neighborhoods

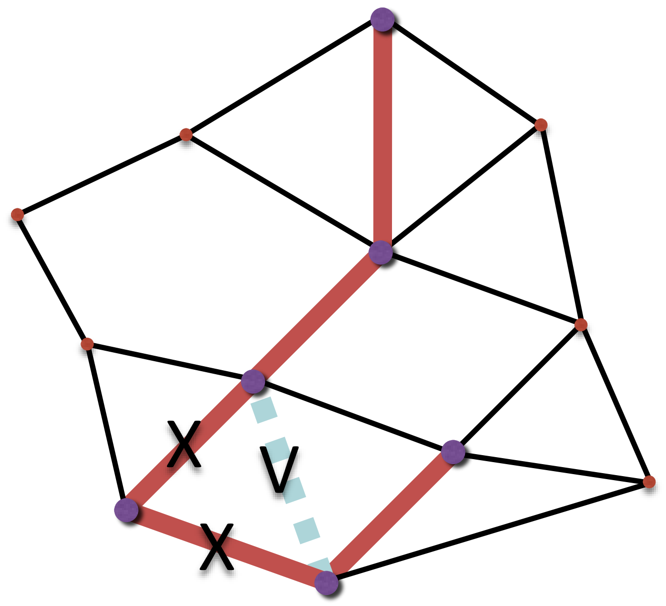




non-manifolds

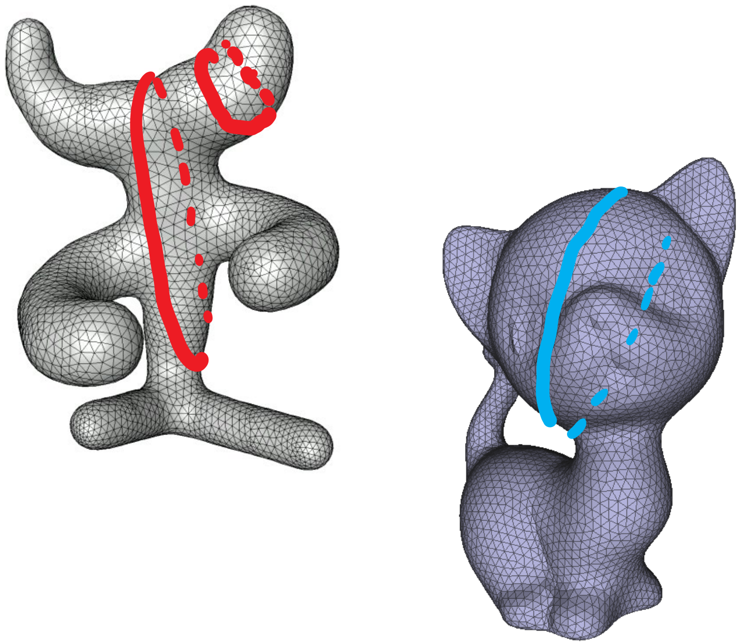
Topology

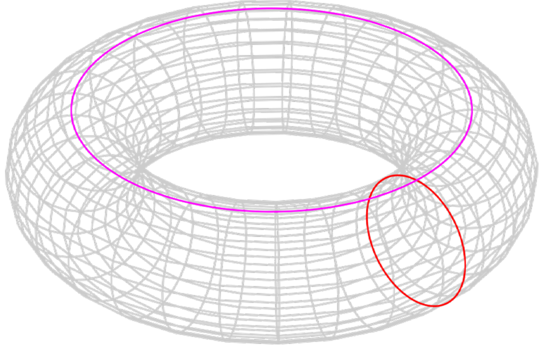
- **Path morphing**: local editing without breaking the path connectivity.
- **Independent loops:** cannot be edited to each other.



Topology

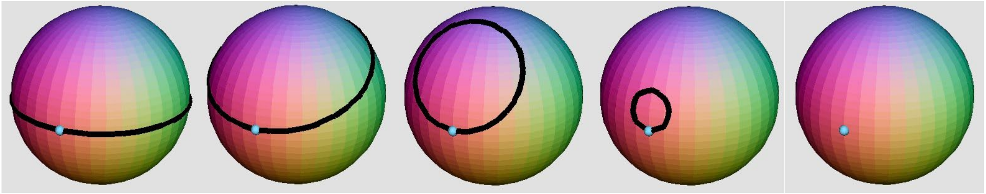
- **Contractible loop**: closed path that can be **contracted** to a single vertex.
- Otherwise, **incontractible** loops.
- **Handles**: incontractible loops that do not disconnect the mesh.
  - Which incontractible loops are not handles?





Topology

- **Simply-connected patch**: all loops are contractible.



- **Genus**, Informal: #independent handles.

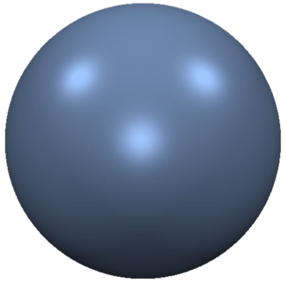
- Genus 0 **does not** imply simply connected! Why?

Global Topology: Genus

Genus:

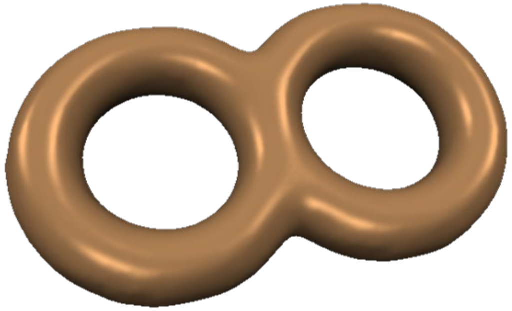
Half the maximal number of closed paths that do not disconnect the mesh

= the number of holes



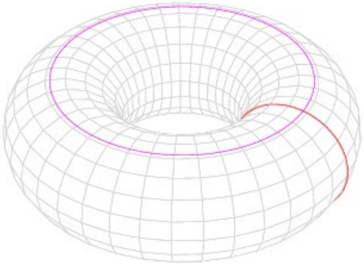
Genus 0

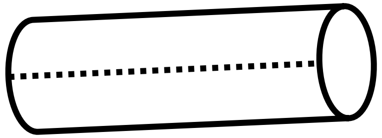
Genus 1



Genus 2

Genus ?





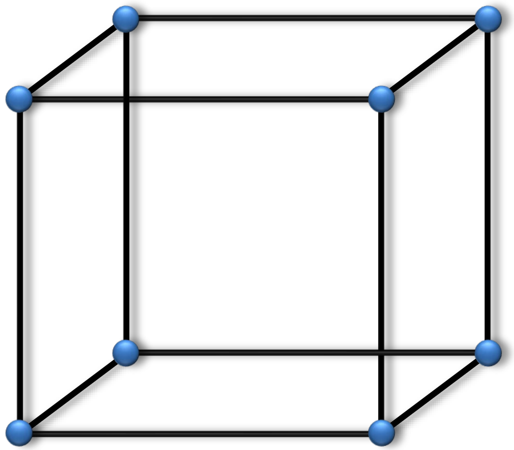
Closed 2-Manifold Polygonal Meshes

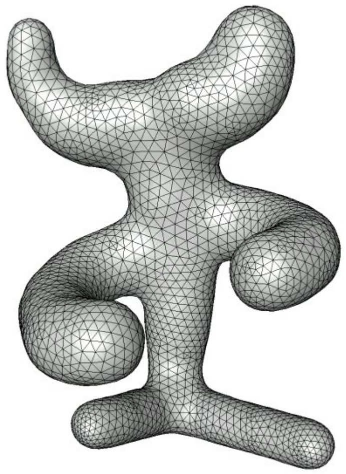
[20 proofs](#)

[Video of proof](#)

Euler-Poincaré formula

$$V + F - E = \boxed{\chi}$$ **Euler characteristic**





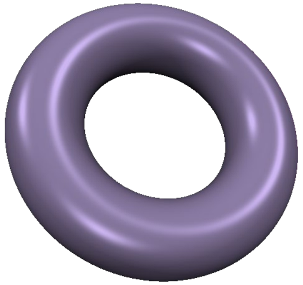
$$V = 8$$
$$E = 12$$
$$F = 6$$
$$\chi = 8 + 6 - 12 = \mathbf{2}$$

$$V = 3890$$
$$E = 11664$$
$$F = 7776$$
$$\chi = \mathbf{2}$$

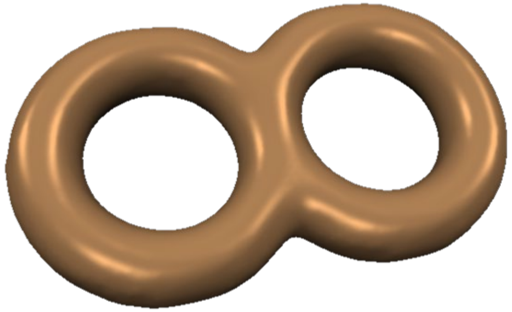
Closed 2-Manifold Polygonal Meshes

Euler-Poincaré formula

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



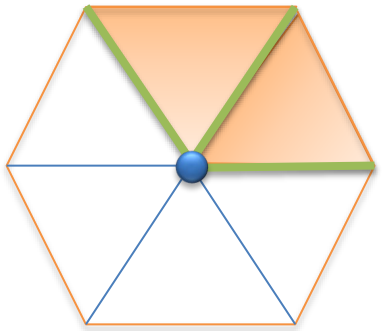
$V = 1500$, $E = 4500$
$F = 3000$, $\mathbf{g = 1}$
$\chi = \mathbf{0}$



$\mathbf{g = 2}$
$\chi = \mathbf{-2}$

Closed 2-Manifold Triangle Meshes

- *Triangle* mesh statistics



$$E \approx 3V$$
$$F \approx 2V$$

- Avg. valence ≈ 6
  Show using Euler Formula

- When can a closed triangle mesh be 6-regular?

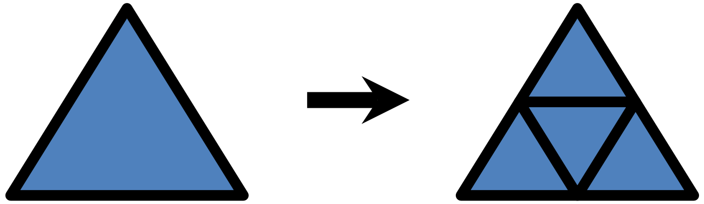
How Many Pentagons?

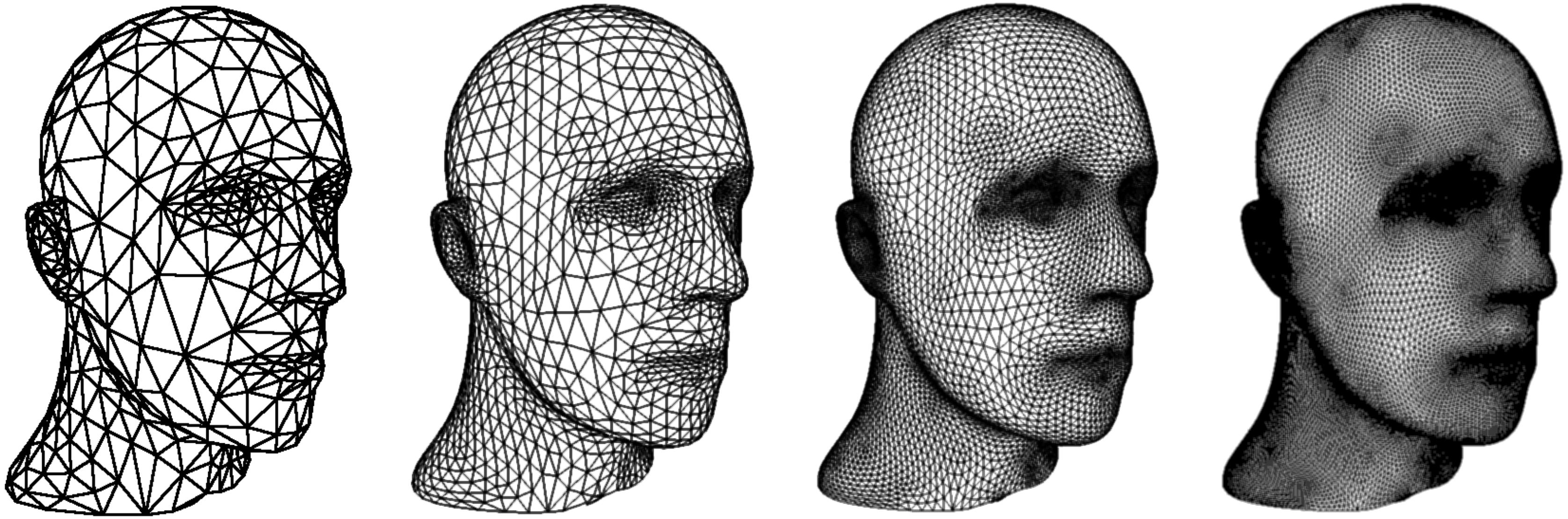
- Closed genus 0 surface
- Only hexagons and pentagons
- 3-Regular vertices

- How many pentagons?

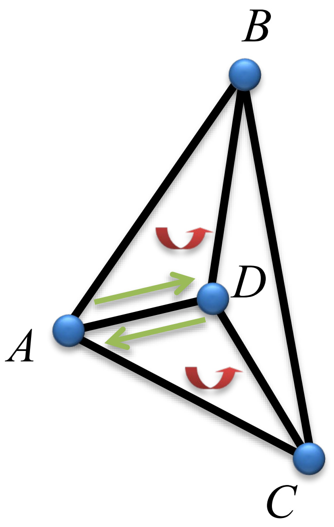
Regularity

- semi-regular





Orientability



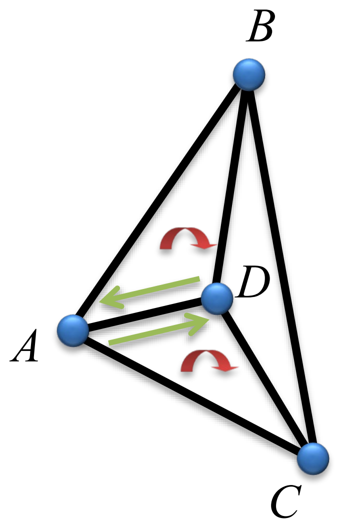
Face Orientation =
clockwise or anticlockwise order in which the vertices listed

defines direction of face **normal**

Oriented **CCW**:
{(C,**D,A**), (**A,D**,B), (C,B,D)}

Oriented **CW**:
{(C,A,D), (D,A,B), (B,C,D)}

Orientability



Face Orientation =
clockwise or anticlockwise order in which the vertices listed

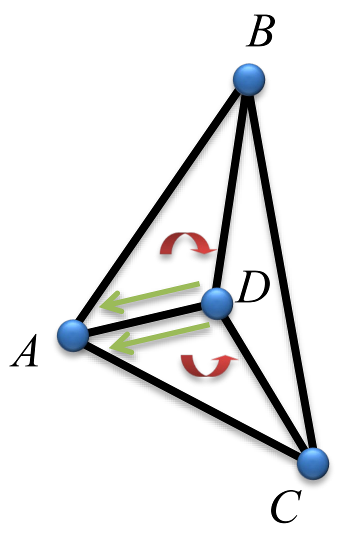
defines direction of face **normal**

Oriented **CCW**:
{(C,D,A), (A,D,B), (C,B,D)}

Oriented **CW**:
{(C,**A,D**), (**D,A**,B), (B,C,D)}

Not oriented:
{(C,D,A), (D,A,B), (C,B,D)}

Orientability



Face Orientation =
clockwise or anticlockwise order in which the vertices listed

defines direction of face **normal**

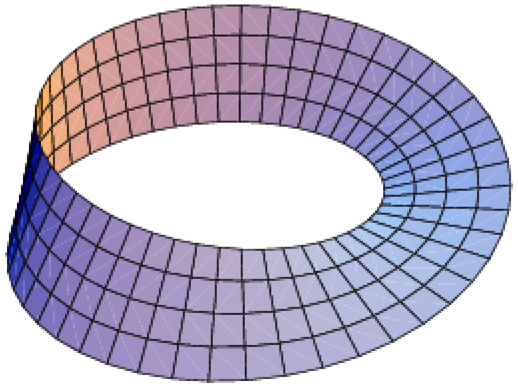
Oriented **CCW**:
{(C,D,A), (A,D,B), (C,B,D)}

Oriented **CW**:
{(C,A,D), (D,A,B), (B,C,D)}

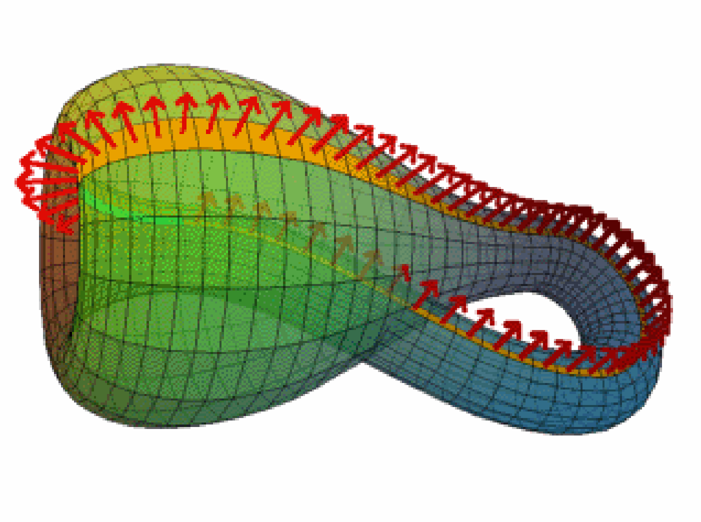
Not oriented:
{(C,**D,A**), (**D,A**,B), (C,B,D)}

Orientable Plane Graph =
orientations of faces can be chosen so that each non-boundary edge is oriented in *both* directions

Non-Orientable Surfaces

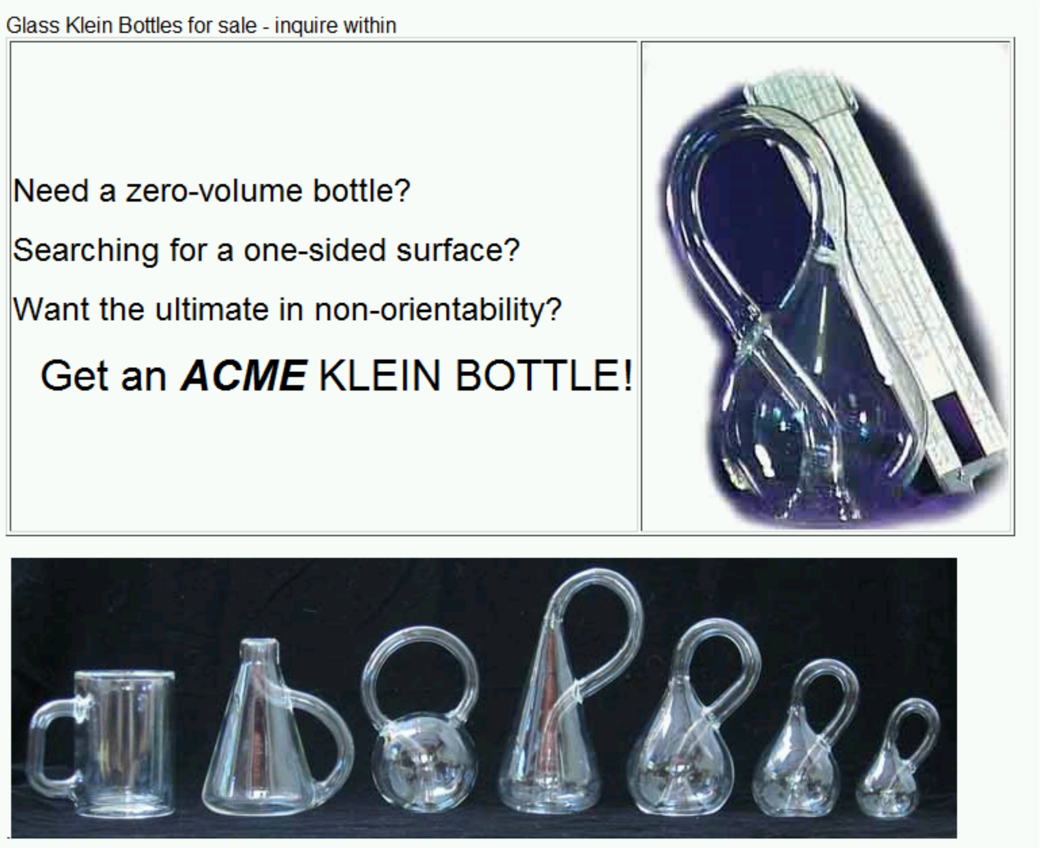


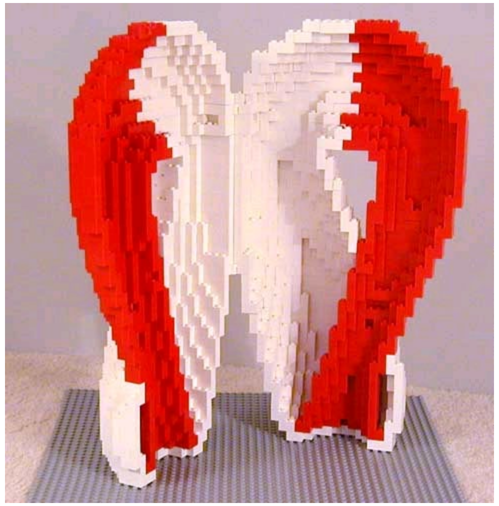
Mobius Strip



Klein Bottle

Garden Variety Klein Bottles





http://www.kleinbottle.com/
http://andrewlipson.com/kleinbottle.htm