

# IFT 6112

## MESH DATA STRUCTURES

<http://www-labs.iro.umontreal.ca/~bmpix/teaching/6112/2018/>

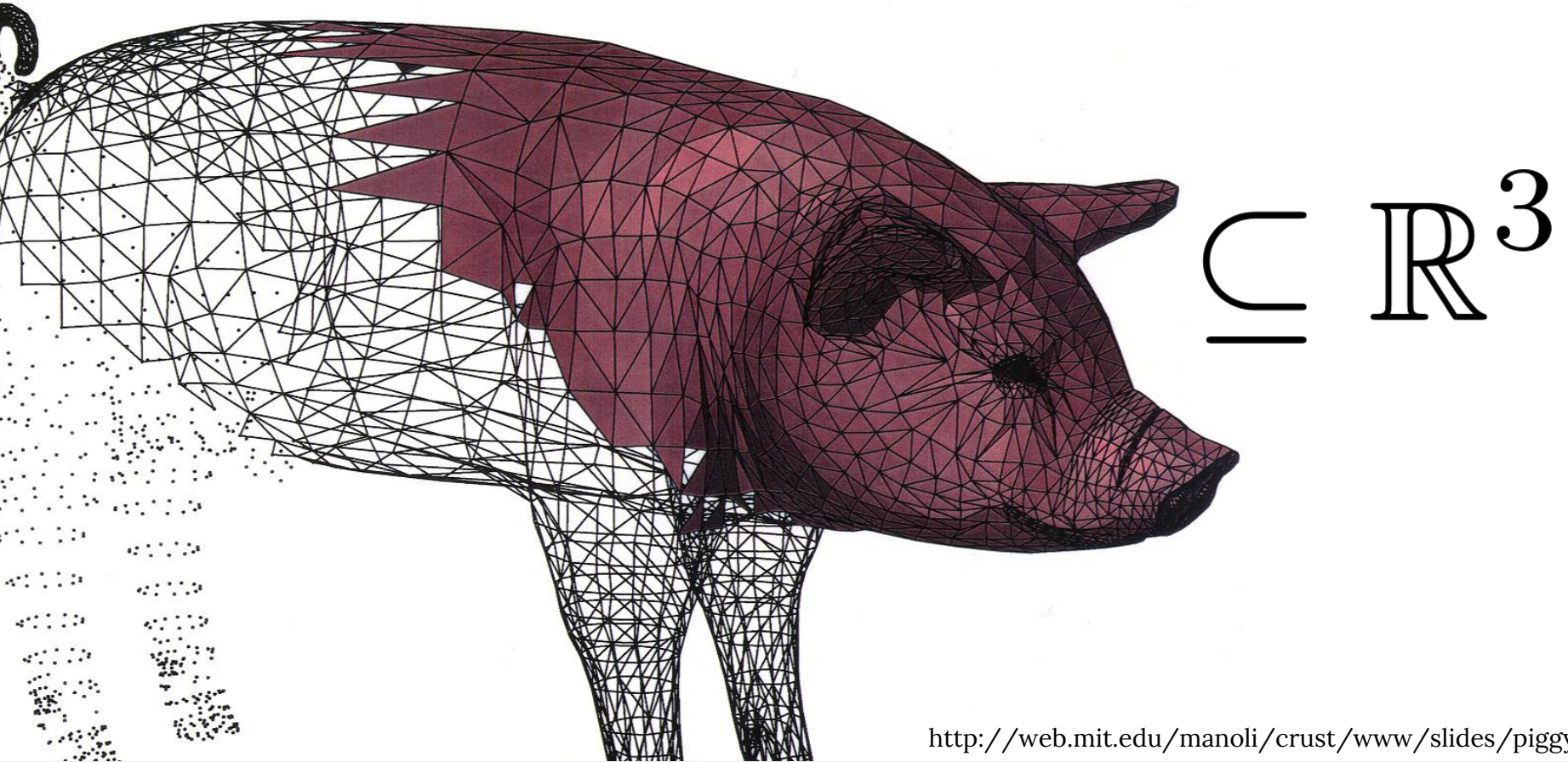


Mikhail Bessmeltsev

# Today's Plan

- **Theoretical definition of surfaces**
  - **Discrete representations**
  - **Higher dimensionality**

# Our Focus

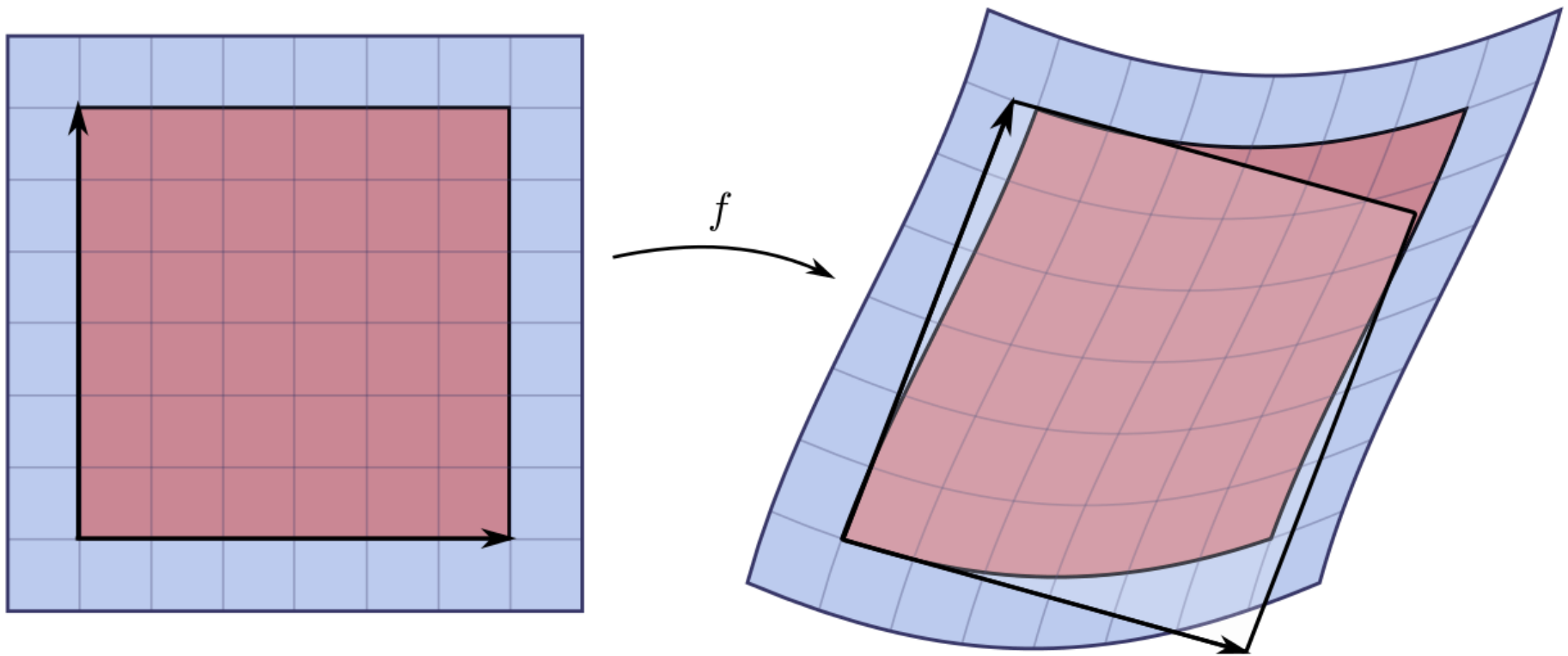


<http://web.mit.edu/manoli/crust/www/slides/piggy.jpg>

# Embedded geometry

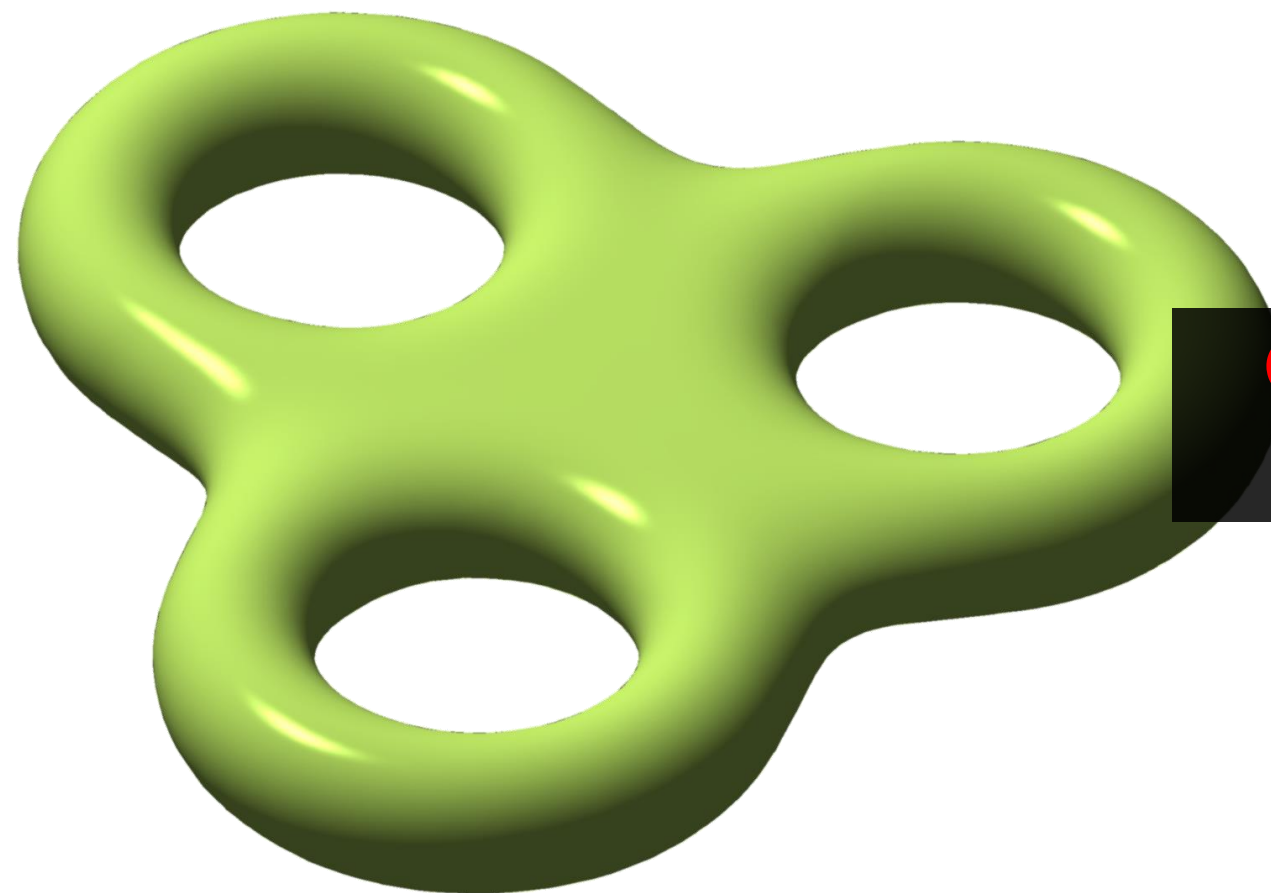
**What is an  
embedded surface?**

# Parametric Surface



# Next Issue

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \dots ?$$



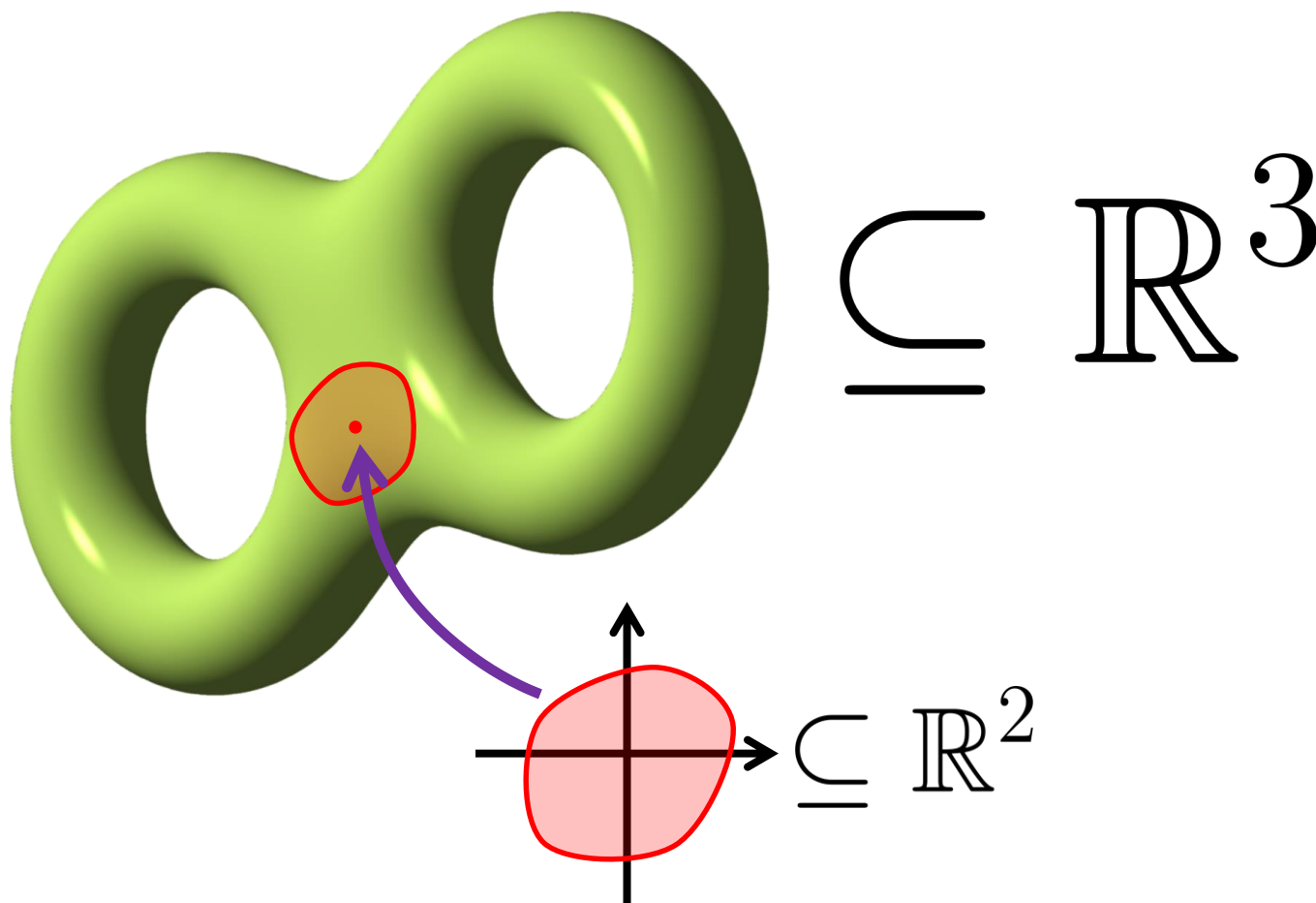
**One function  
isn't enough!**

A surface is a  
**set of points**  
with certain properties.

It is not a function.



# Theoretical Definition of Surface



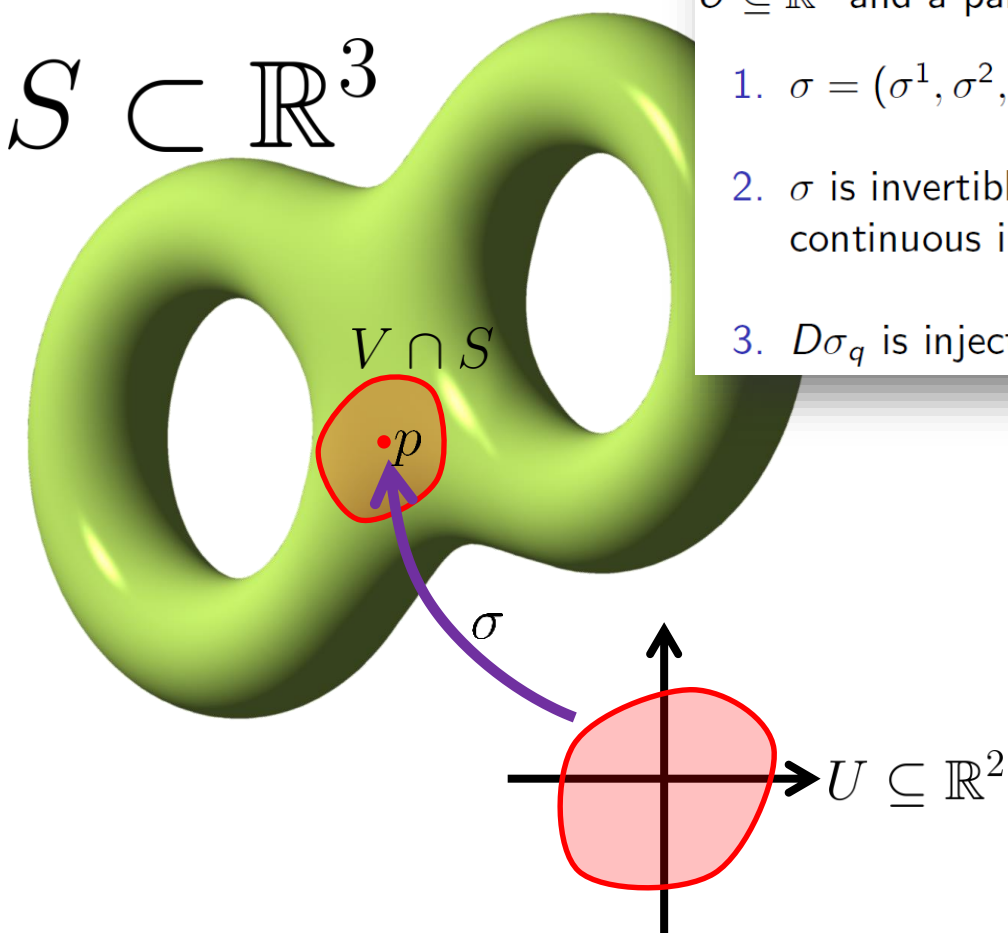


# Theoretical Definition of Surface

A set  $S \subset \mathbb{R}^3$  is a *regular surface* if for each  $p \in S$  there exists an open neighbourhood  $V \subseteq \mathbb{R}^3$  containing  $p$ , an open neighbourhood  $U \subseteq \mathbb{R}^2$  and a parametrization  $\sigma : U \rightarrow V \cap S$  such that:

1.  $\sigma = (\sigma^1, \sigma^2, \sigma^3)$ .
2.  $\sigma$  is invertible as a map from  $U$  onto  $V \cap S$  and has a continuous inverse.
3.  $D\sigma_q$  is injective  $\forall q$ . (If and only if  $\det((D\sigma_q)^\top D\sigma_q) \neq 0$ .)

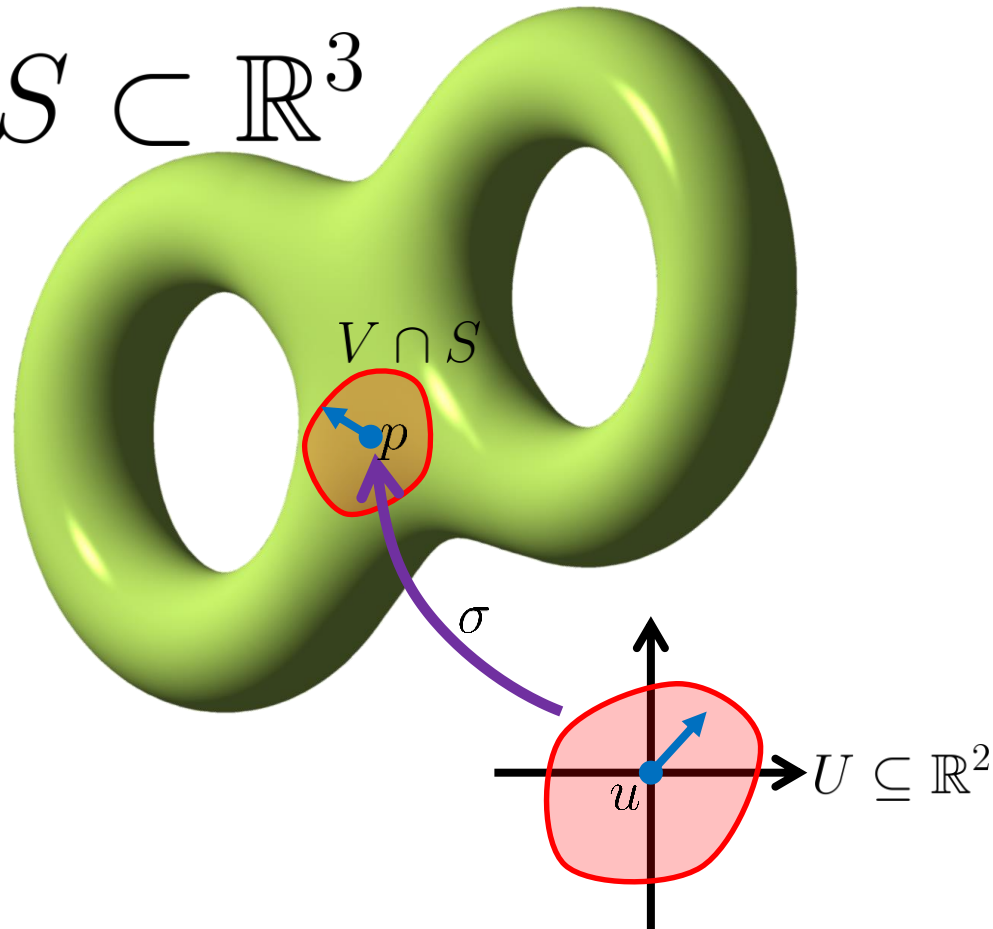
Text from Stanford CS 468 lecture 5 (2013), A. Butscher



# Tangent Space

$$T_p S := \text{Image}(D\sigma_u)$$

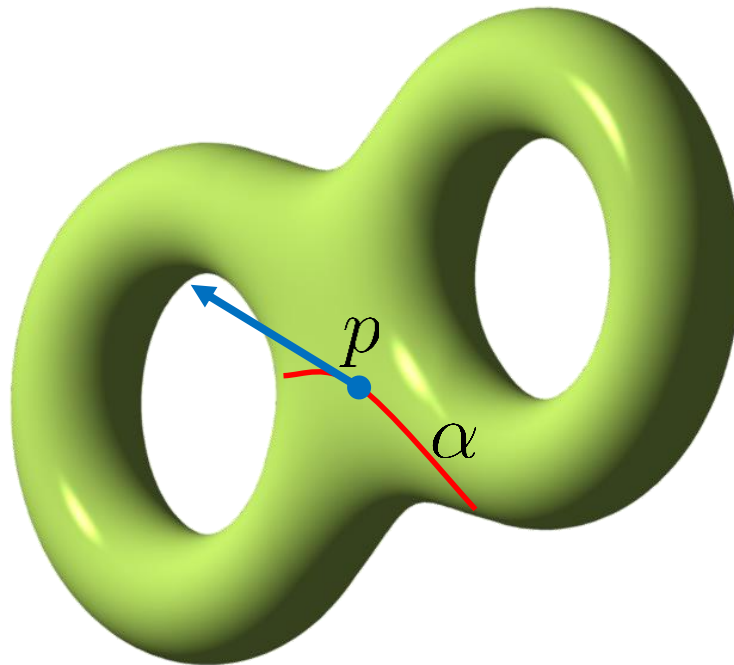
$$S \subset \mathbb{R}^3$$



# Tangent Space: Coordinate-Free

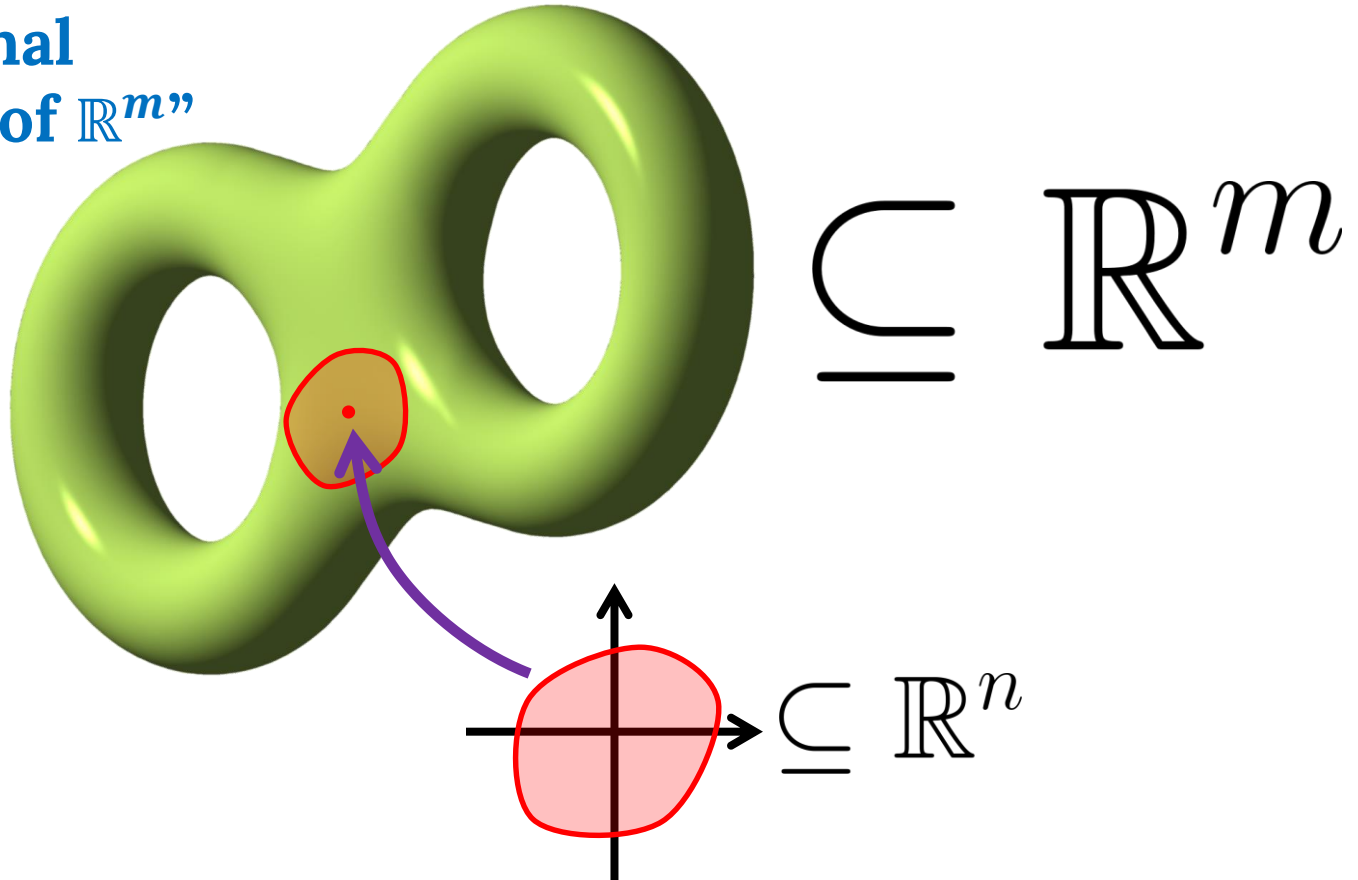
$$v \in T_p S \iff$$

there exists curve  $\alpha : (-\varepsilon, \varepsilon) \rightarrow S$   
with  $\alpha(0) = p, \alpha'(0) = v$



# $n$ -D Embedded Manifold

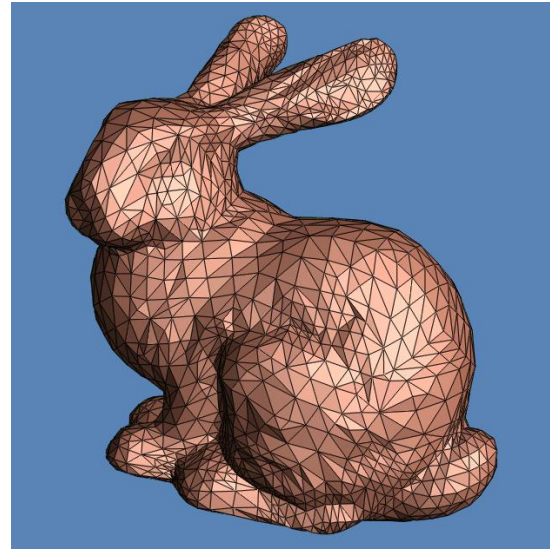
“ $n$ -dimensional  
submanifold of  $\mathbb{R}^m$ ”



**Includes curves,  
surfaces, 3-manifolds,**

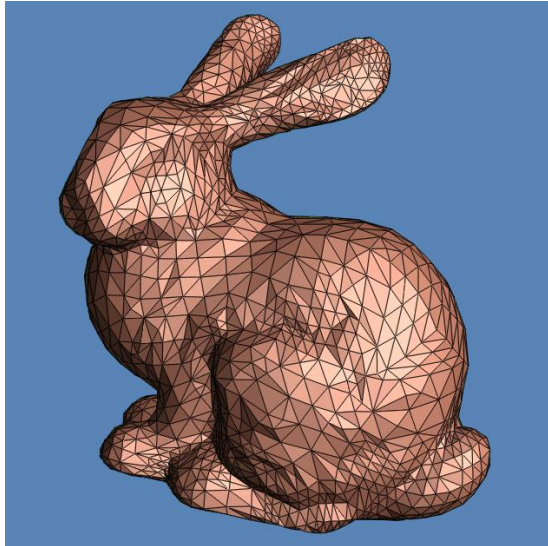
# Discrete Problem

**What is a discrete surface?  
How do you store it?**



# SHAPE REPRESENTATIONS

Boundary



Volume

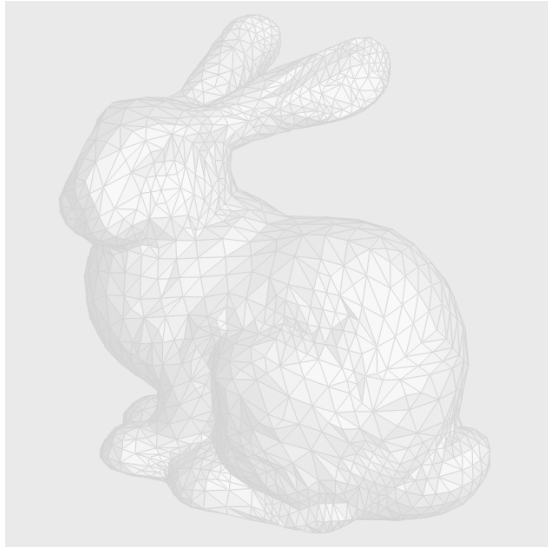


<https://mathgrrl.com/hacktastic/2015/10/low-voxel-stanford-bunny-voxelizationminecraft-tutorial/>



# SHAPE REPRESENTATIONS

Boundary



Volume

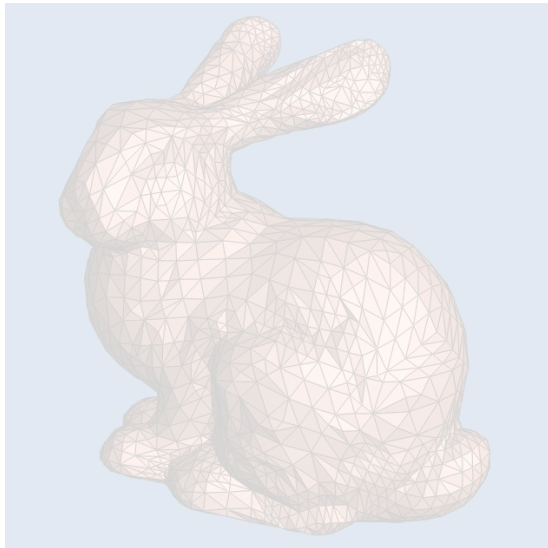


<https://mathgrrl.com/hacktastic/2015/10/low-voxel-stanford-bunny-voxelizationminecraft-tutorial/>

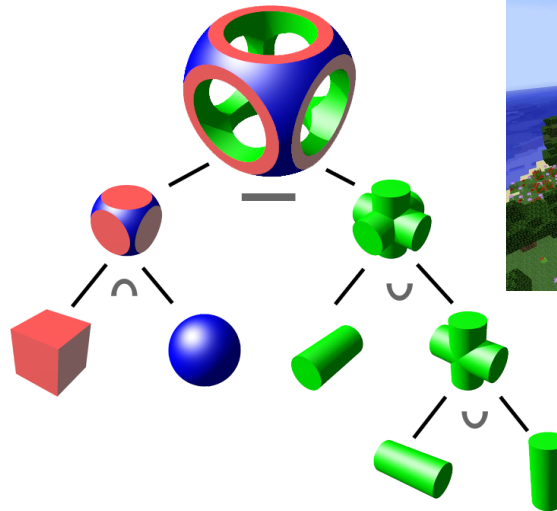


# SHAPE REPRESENTATIONS

Boundary



Volume

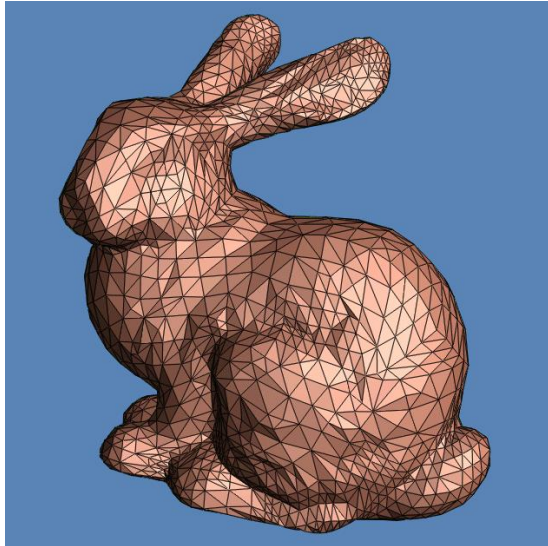


Voxelization

Constructive Solid Geometry

# SHAPE REPRESENTATIONS

Boundary



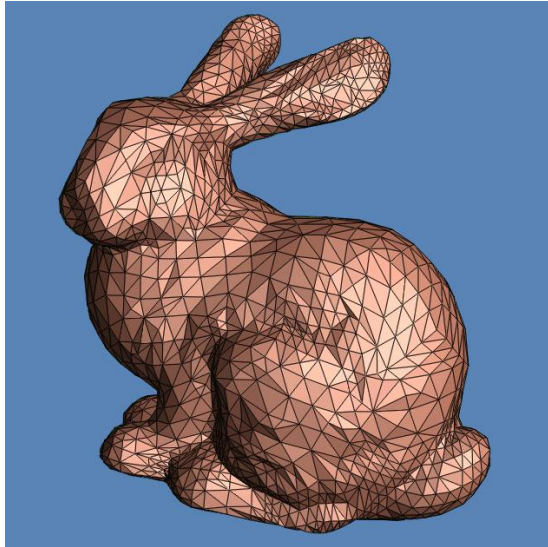
Volume



<https://mathgrrl.com/hacktastic/2015/10/low-voxel-stanford-bunny-voxelizationminecraft-tutorial/>

# SHAPE REPRESENTATIONS

Boundary

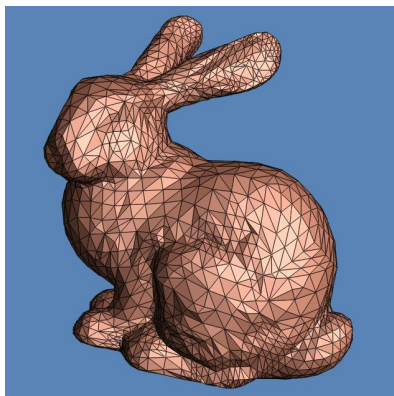


Volume

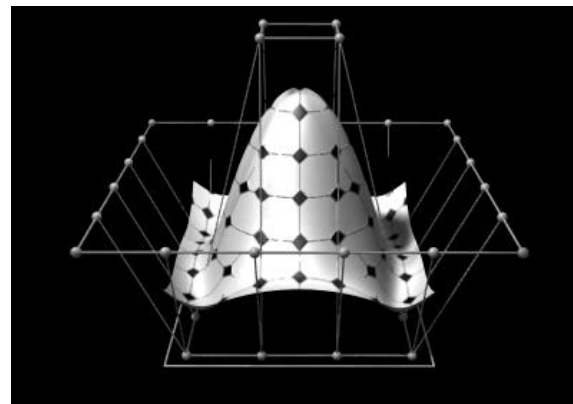


<https://mathgrrl.com/hacktastic/2015/10/low-voxel-stanford-bunny-voxelizationminecraft-tutorial/>

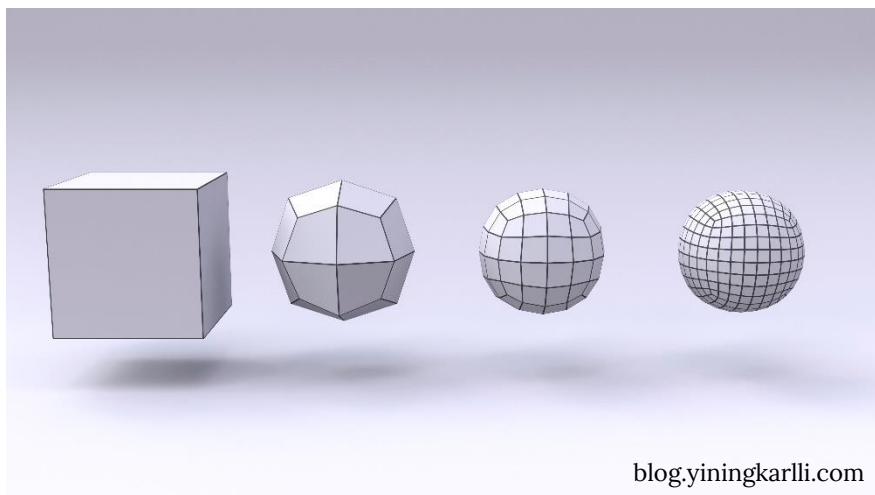
# BOUNDARY SHAPE REPRESENTATIONS



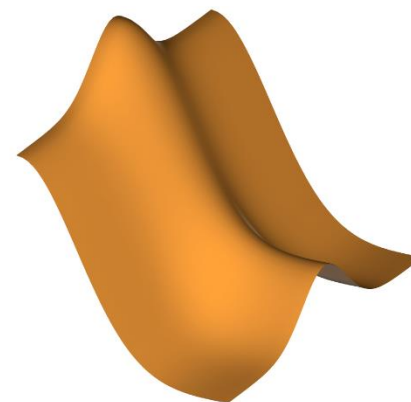
Meshes



NURBS



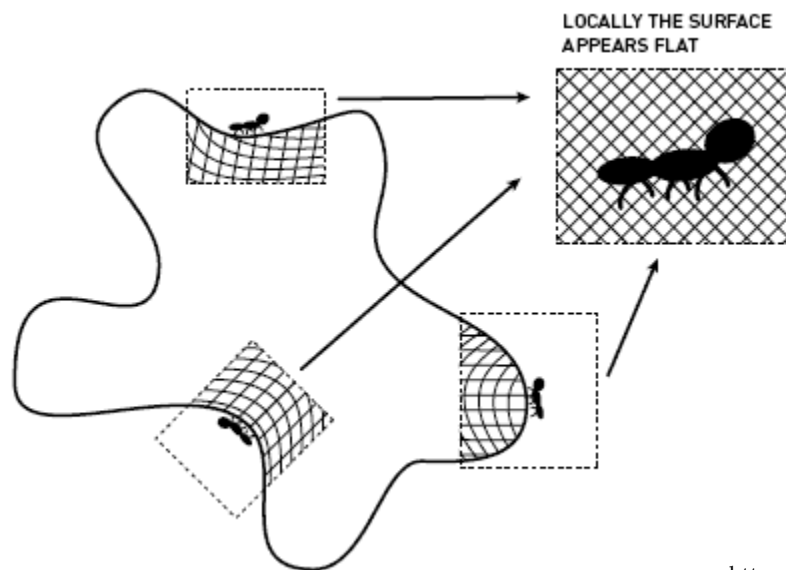
Subdivision surfaces



Tensor product surfaces

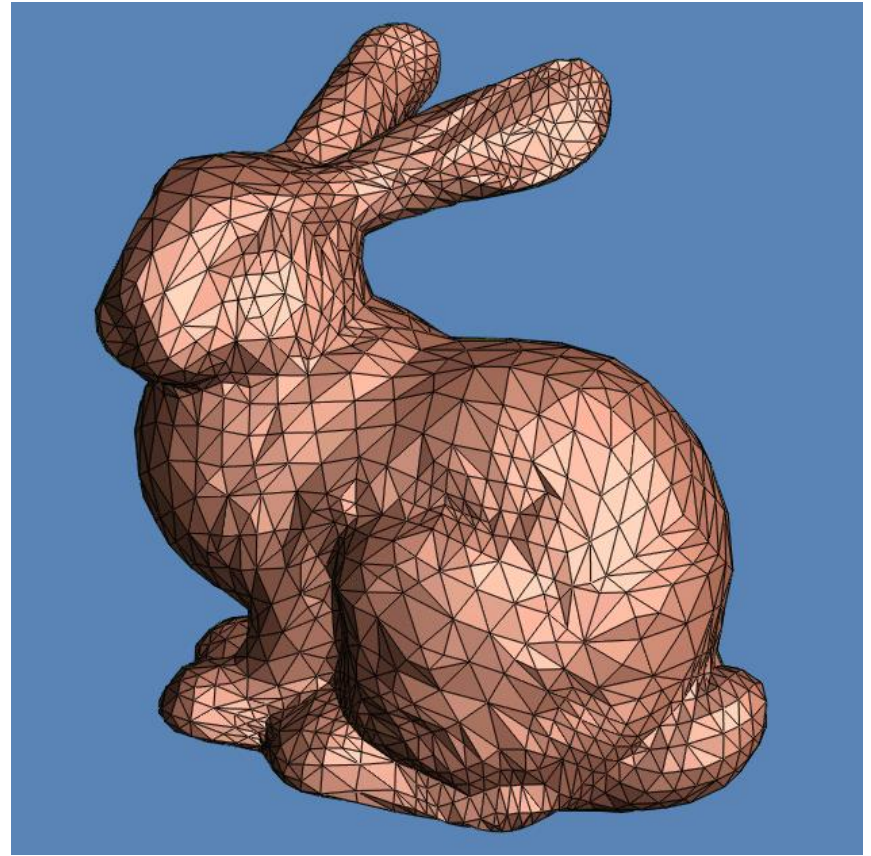
# Differential Geometer's Mantra

A surface is  
**locally planar.**





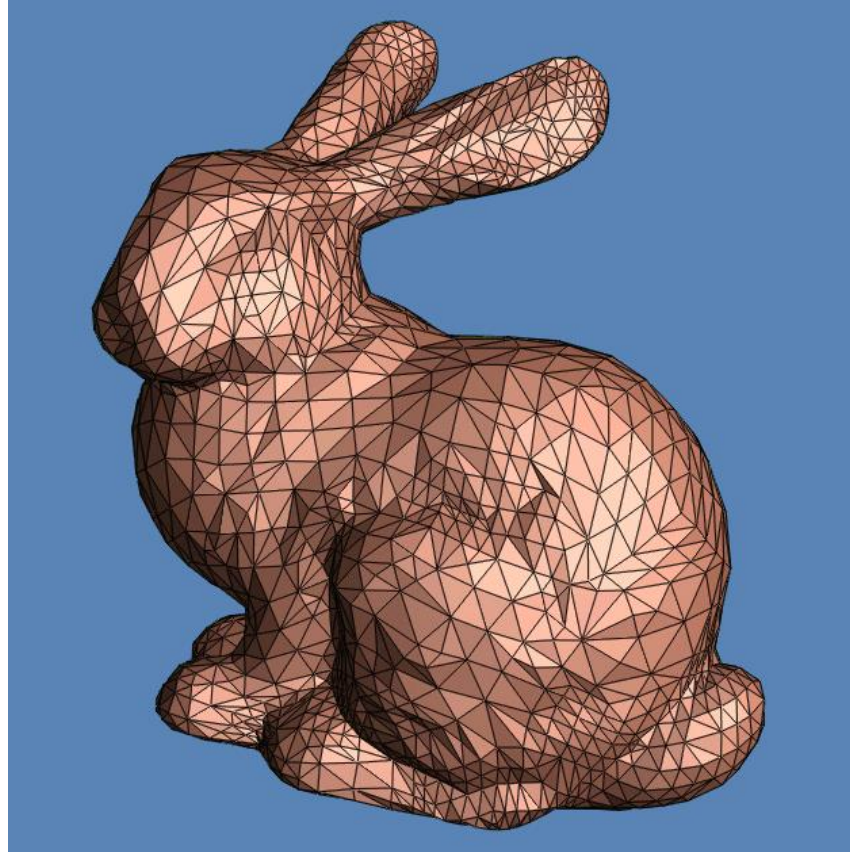
# Common Representation



<http://graphics.stanford.edu/data/3Dscanrep/stanford-bunny-cebal-ssh.jpg>  
<http://www.stat.washington.edu/wxs/images/BUNMID.gif>

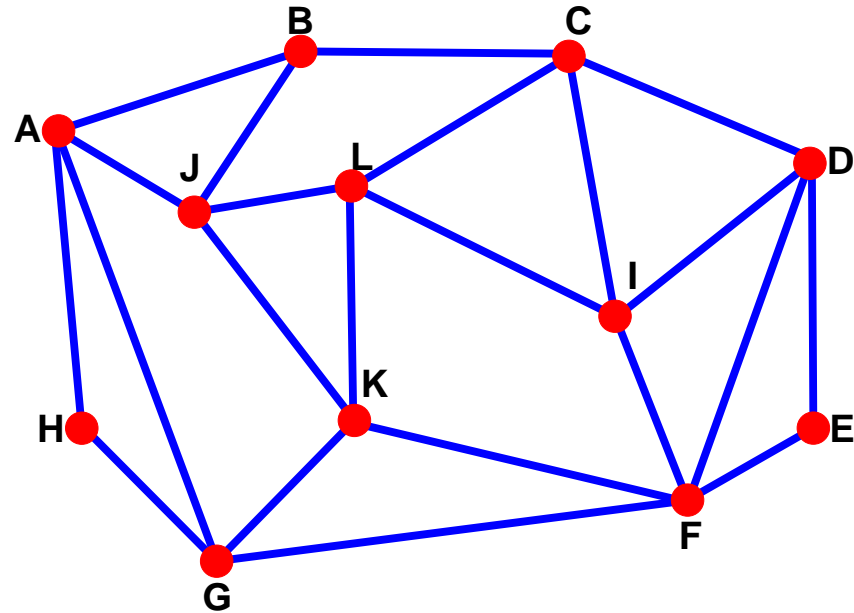
## Triangle mesh

# Mesh = Embedded Graph?





# GRAPH

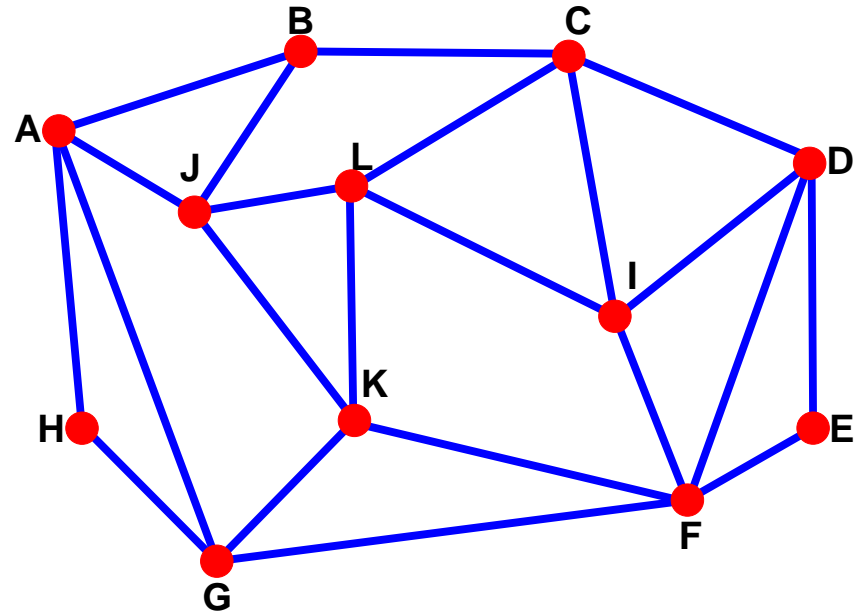


$$G = \langle V, E \rangle$$

$$V = \text{vertices} = \{A, B, C, \dots, L\}$$

$$E = \text{edges} = \{(A, B), (B, C), (C, D), \dots, (L, I)\}$$

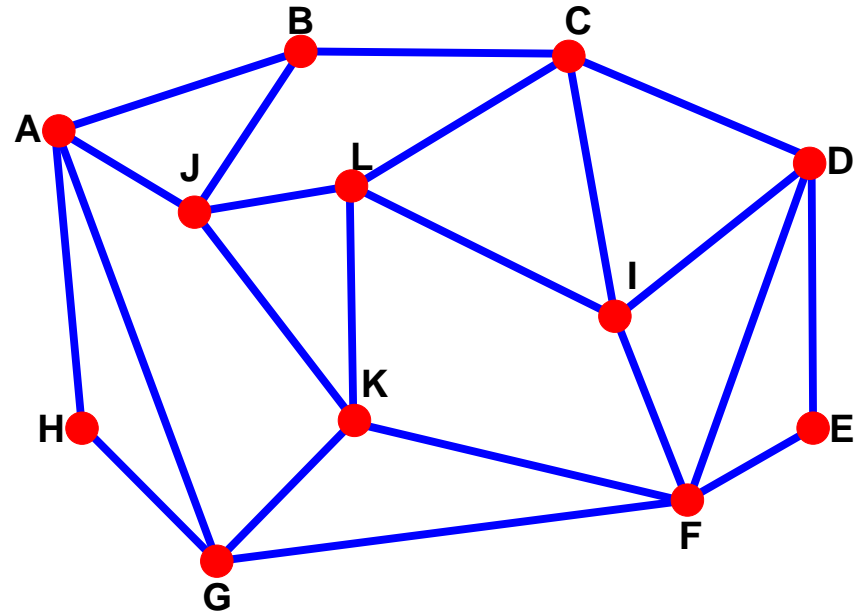
# GRAPH



**Face:** cycle of vertices/edges which cannot be shortened

$\mathbf{F} = \text{faces} = \{(A,H,G), (A,J,K,G), (B,A,J), (B,C,L,J), (C,I,L), (C,D,I), (D,E,F), (D,I,F), (L,I,F,K), (L,J,K), (K,F,G)\}$

# GRAPH

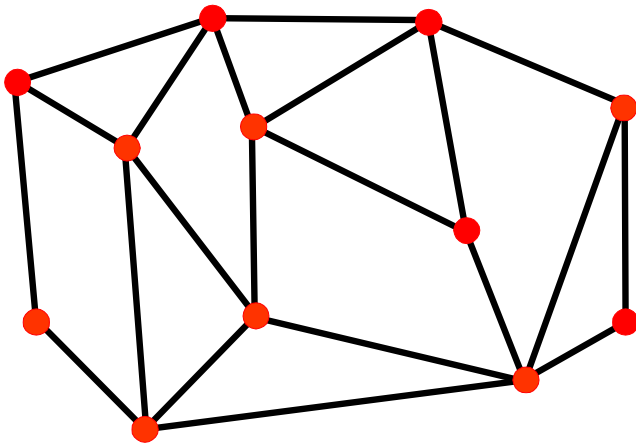


***Vertex degree (valence)*** = number of edges adjacent to vertex

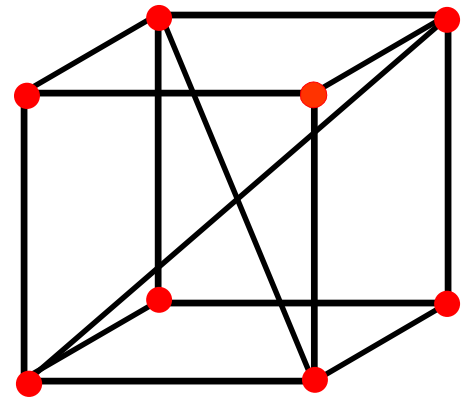
$$\deg(J) = 4, \deg(H) = 2$$

# GRAPH EMBEDDING

Graph is embedded in  $\mathbb{R}^d$  if  
each vertex is assigned a position in  $\mathbb{R}^d$ .



Embedding in  $\mathbb{R}^2$



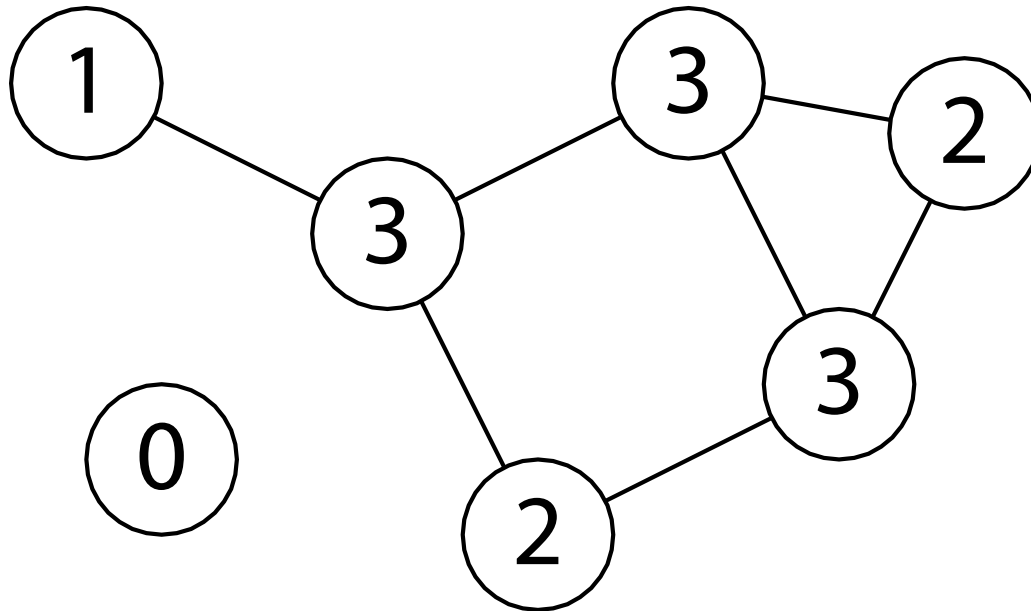
Embedding in  $\mathbb{R}^3$

# TRIANGLE MESH

A graph, embedded in 3D,  
such that each face is a triangle?

# CONNECTED GRAPH

$\Leftrightarrow$  there is a path of edges connecting every two vertices



A simpler approach?

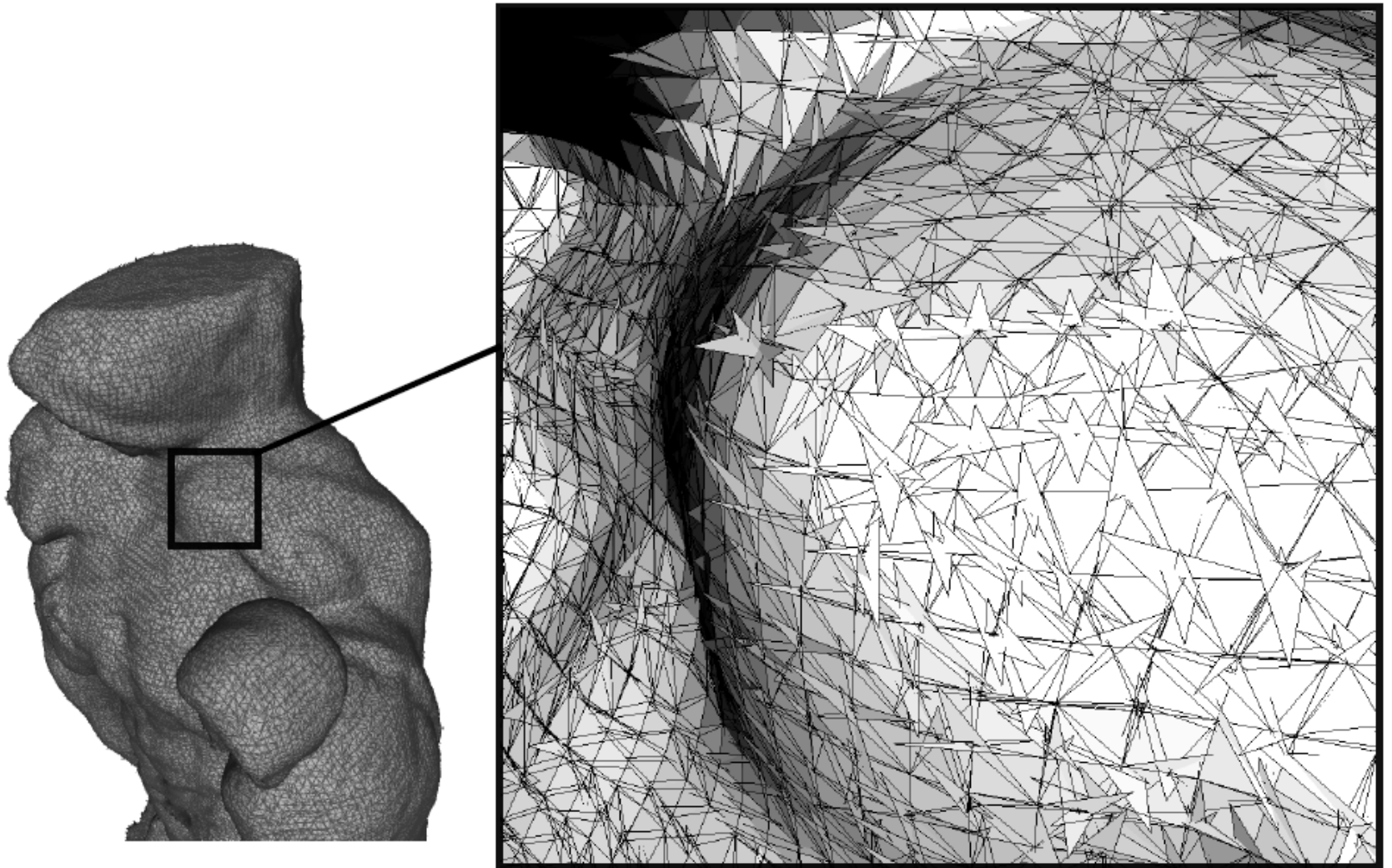
$$M = (V, T)$$

What conditions are needed?

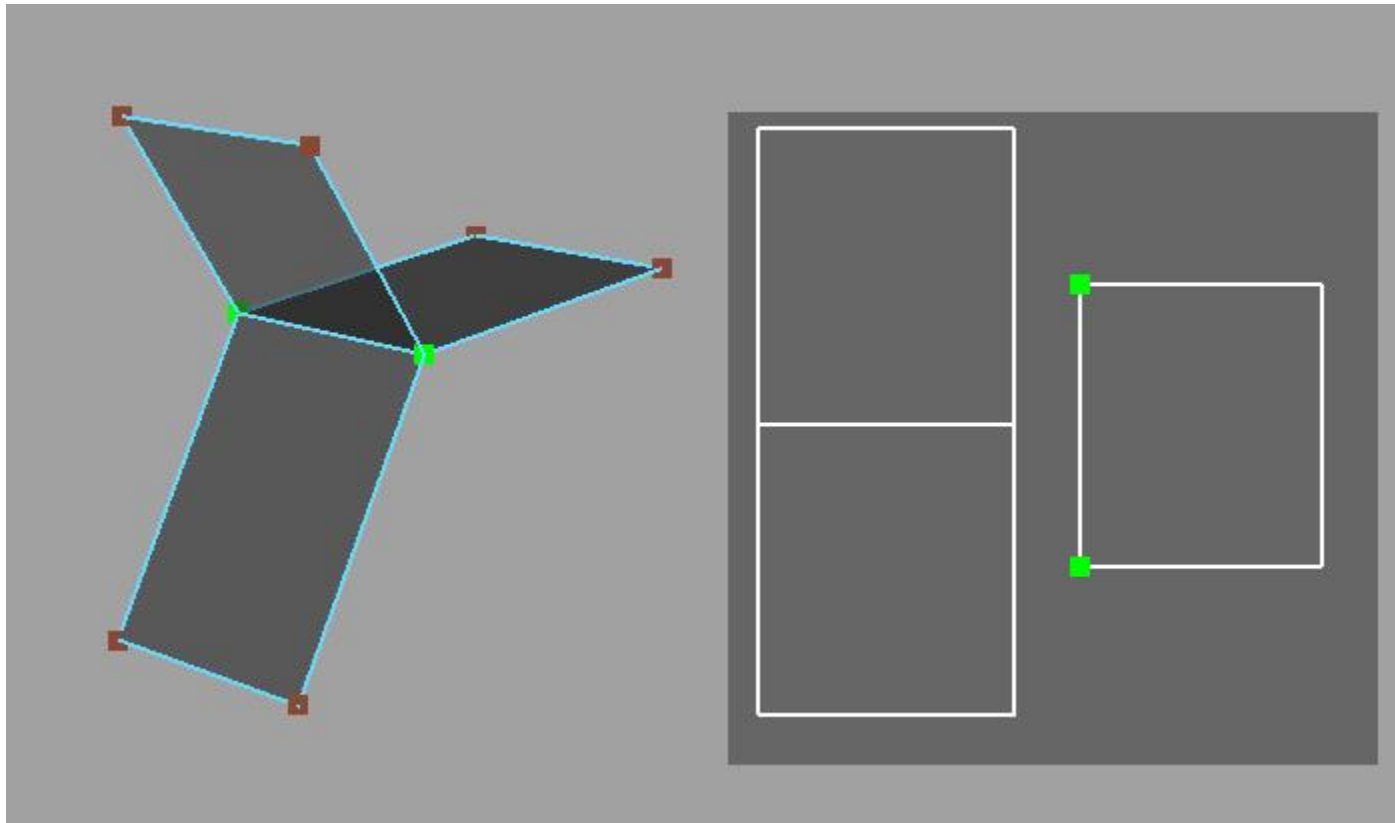
**Triangle mesh**



# Is this a surface?

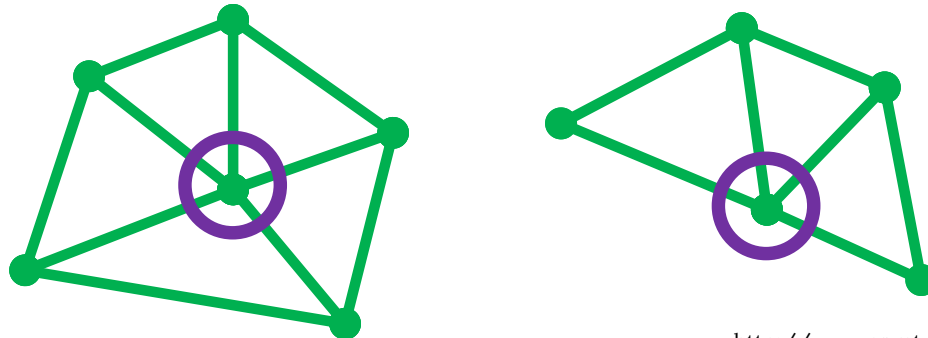


# Nonmanifold Edge



# Manifold Mesh

1. Each **edge** is incident to one or two faces
2. **Faces** incident to a vertex form a closed or open fan



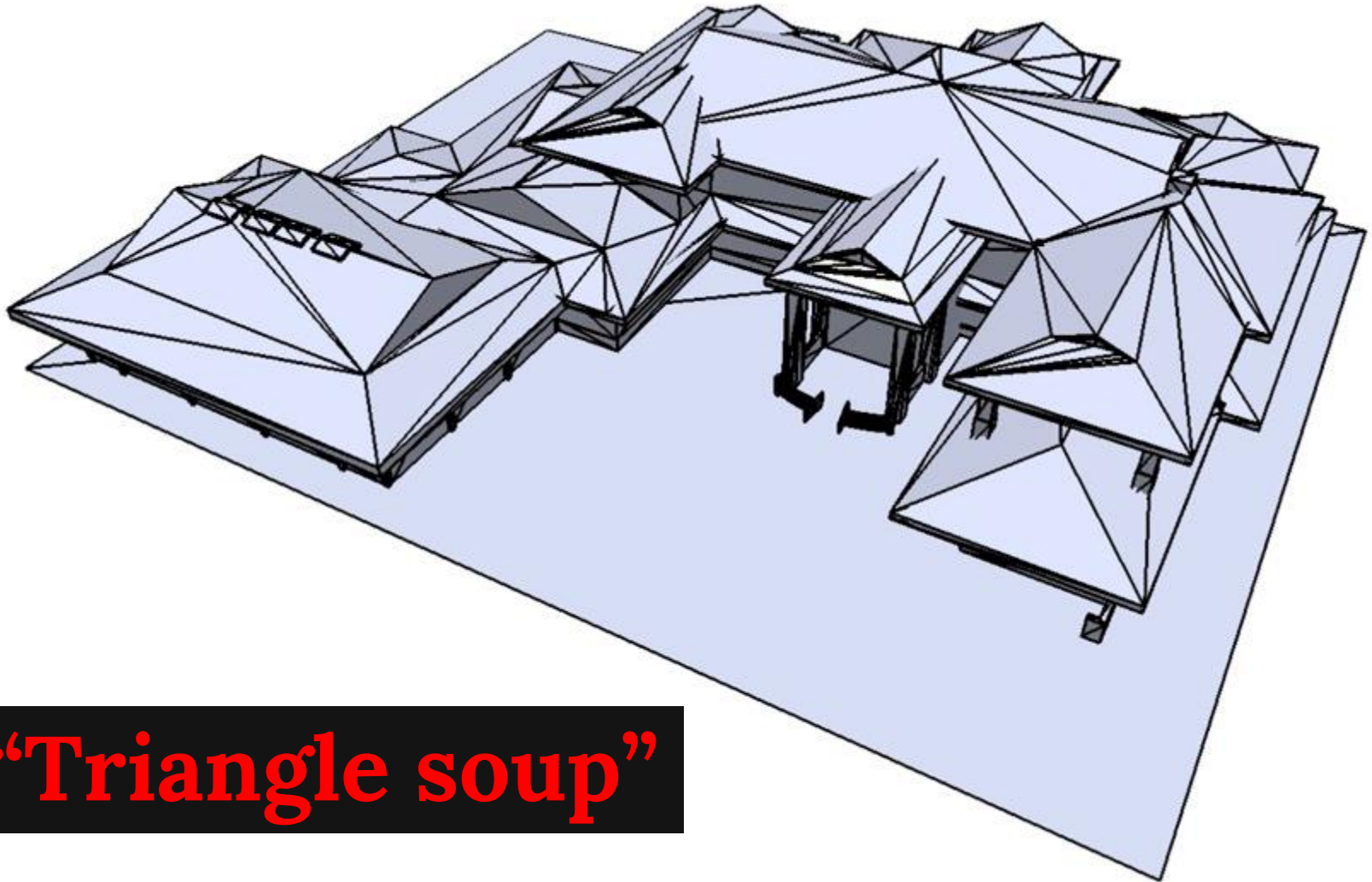
# Manifold Mesh

1. Each **edge** is incident to one or two faces
2. **Faces** incident to a vertex form a closed or open fan

**Assume meshes are manifold  
(for now)**



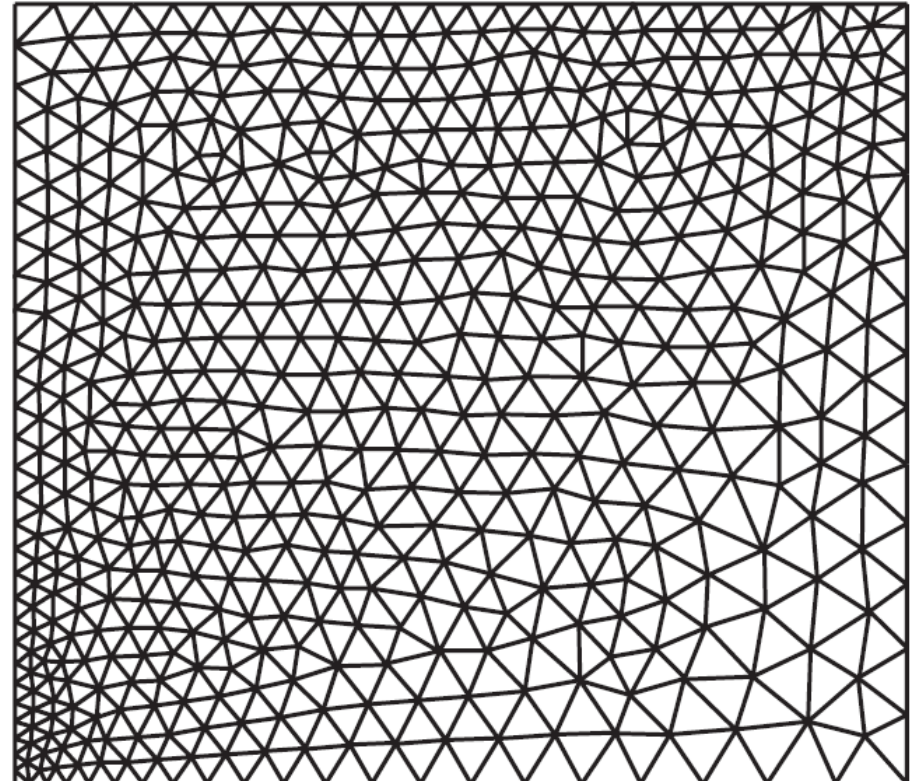
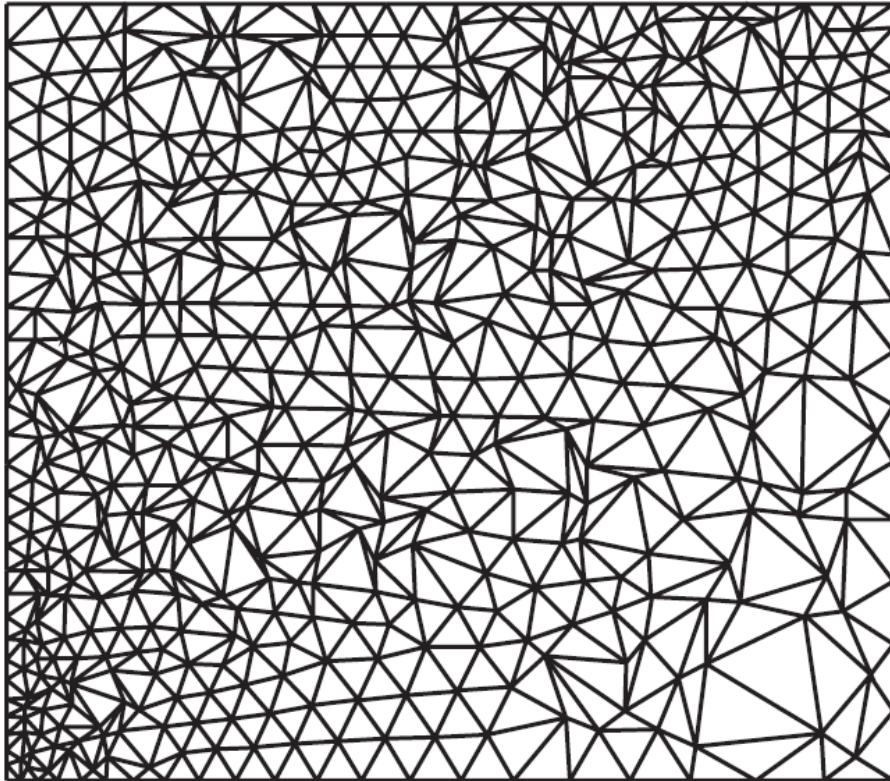
# Easy-to-Violate Assumption



**“Triangle soup”**

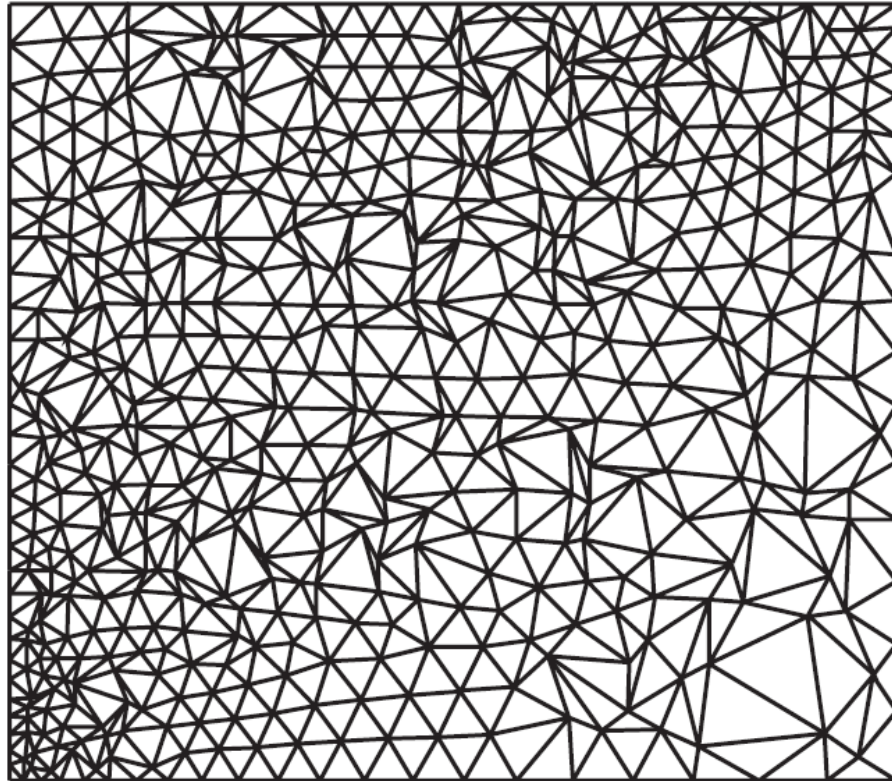


# Invalid Meshes vs. Bad Meshes



**Nonuniform**  
**areas and angles**

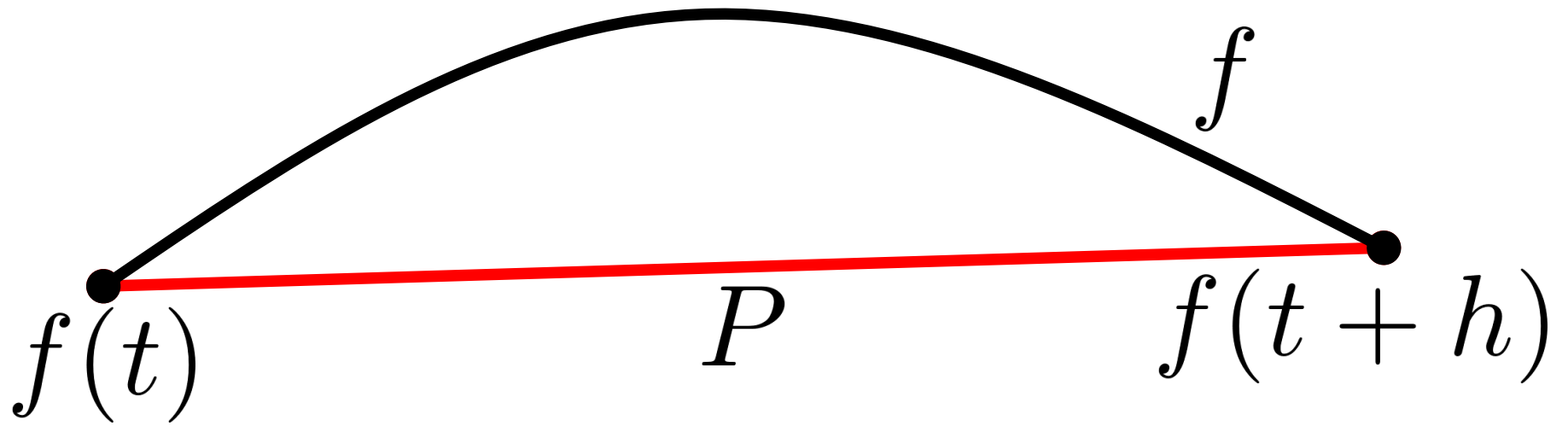
# Why is Meshing an Issue?



**How to you interpret  
one value per vertex?**

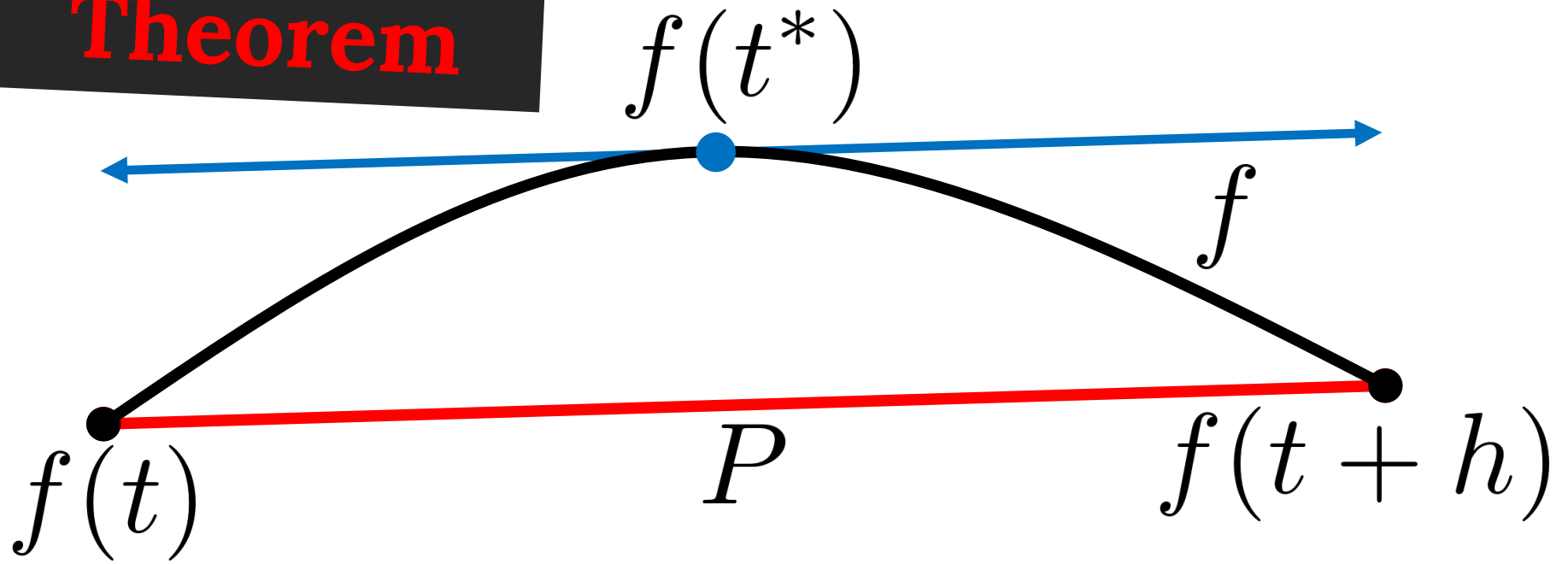


# Approximation Properties

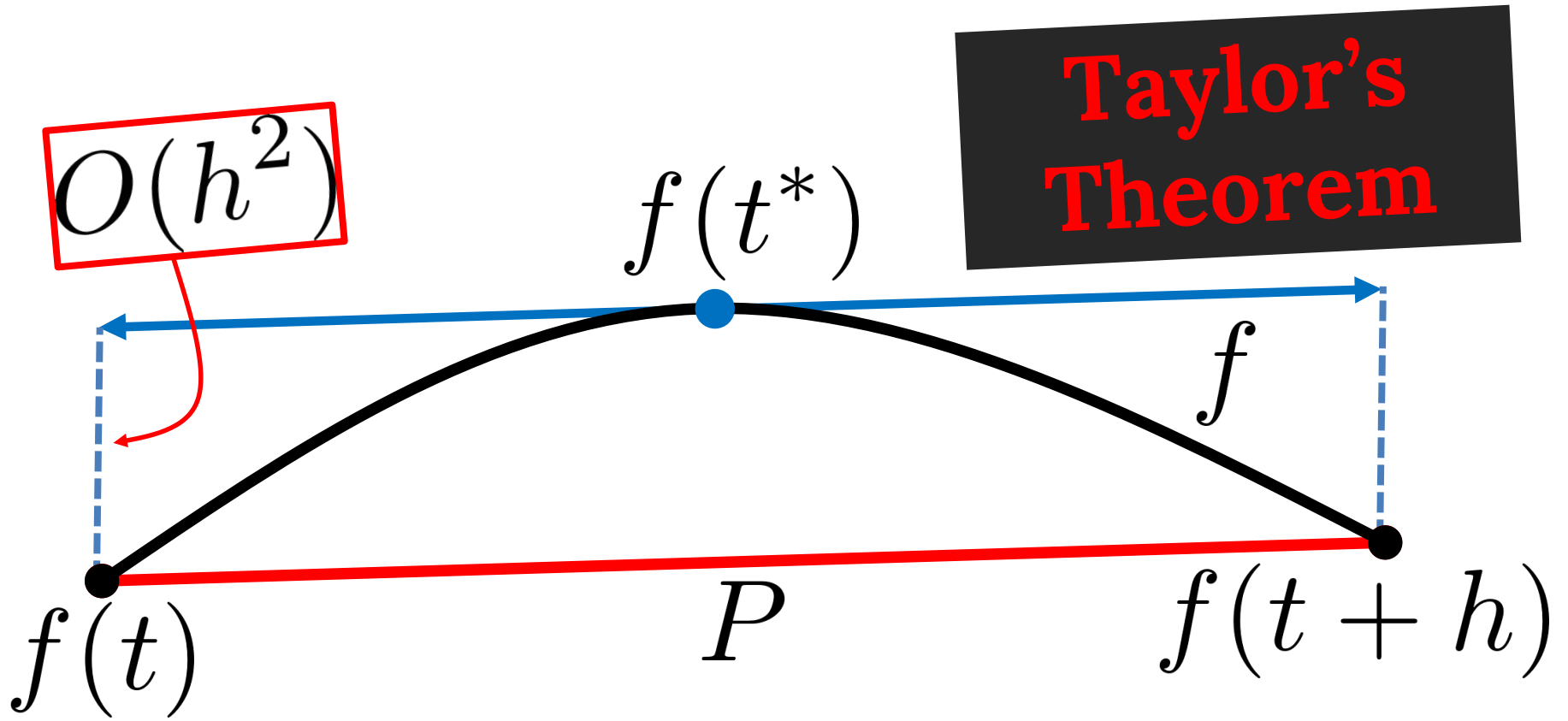


# Approximation Properties

**Mean Value  
Theorem**



# Approximation Properties



# Conclusion

**Piecewise linear faces are  
reasonable building blocks.**

# Additional Advantages

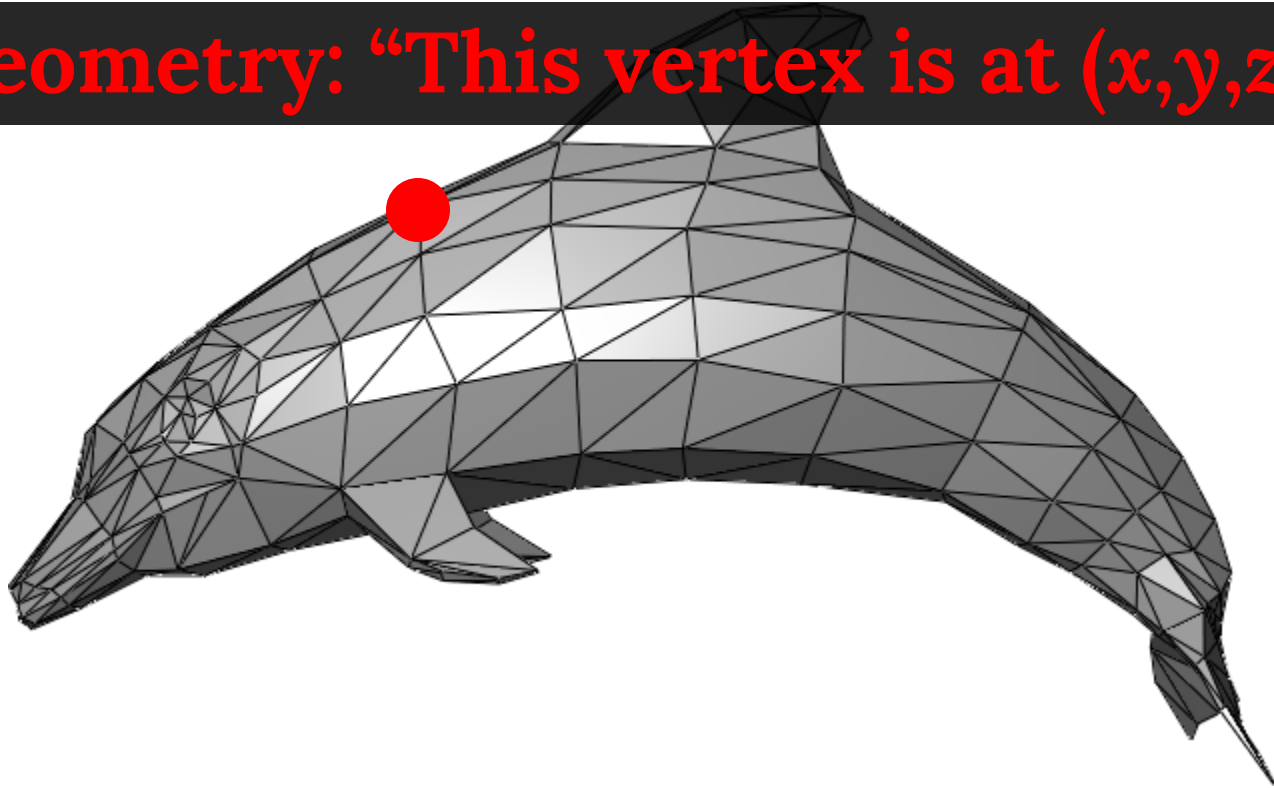
- Simple to render
- *Arbitrary topology* possible
- Basis for subdivision, refinement

# Topology

The study of geometric properties that remain invariant under certain transformations

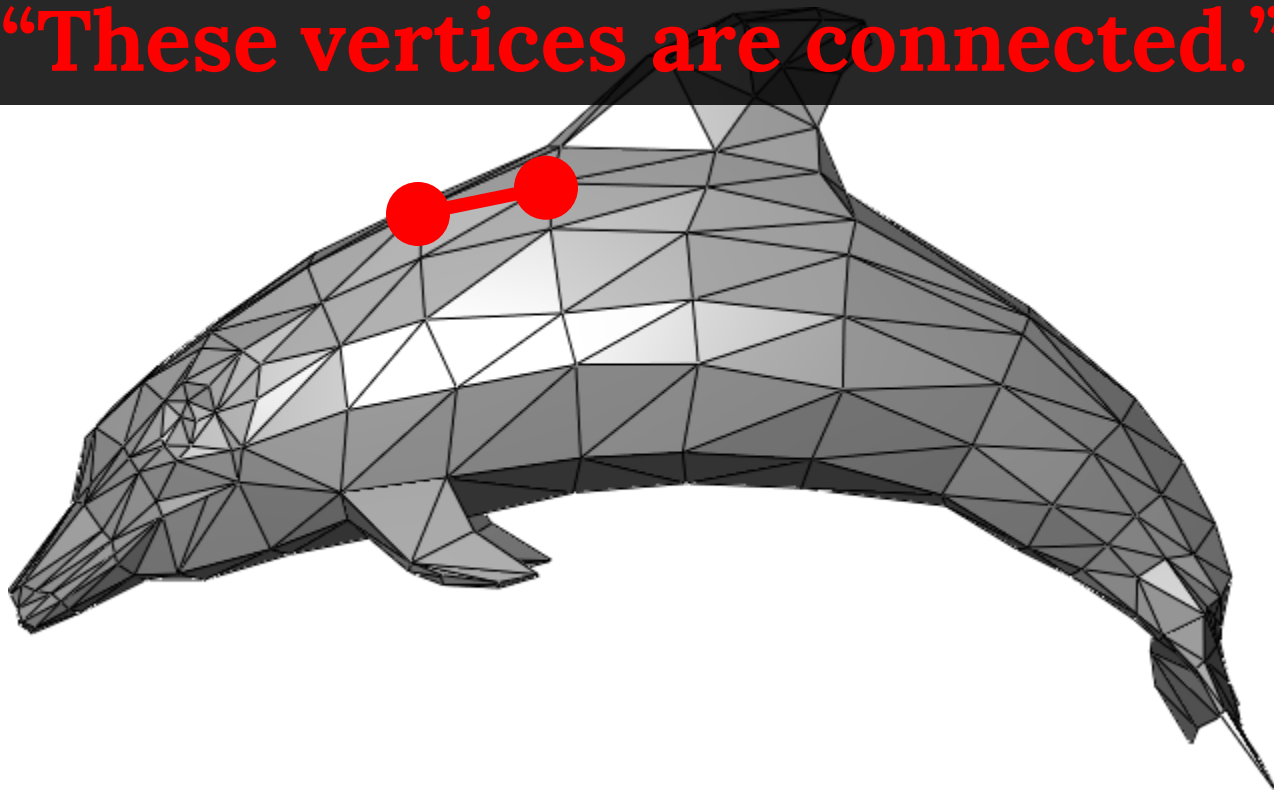
# Mesh Topology vs. Geometry

**Geometry: “This vertex is at  $(x,y,z)$ .”**



# Mesh Topology vs. Geometry

**Topology:**  
**“These vertices are connected.”**





# Triangle Mesh

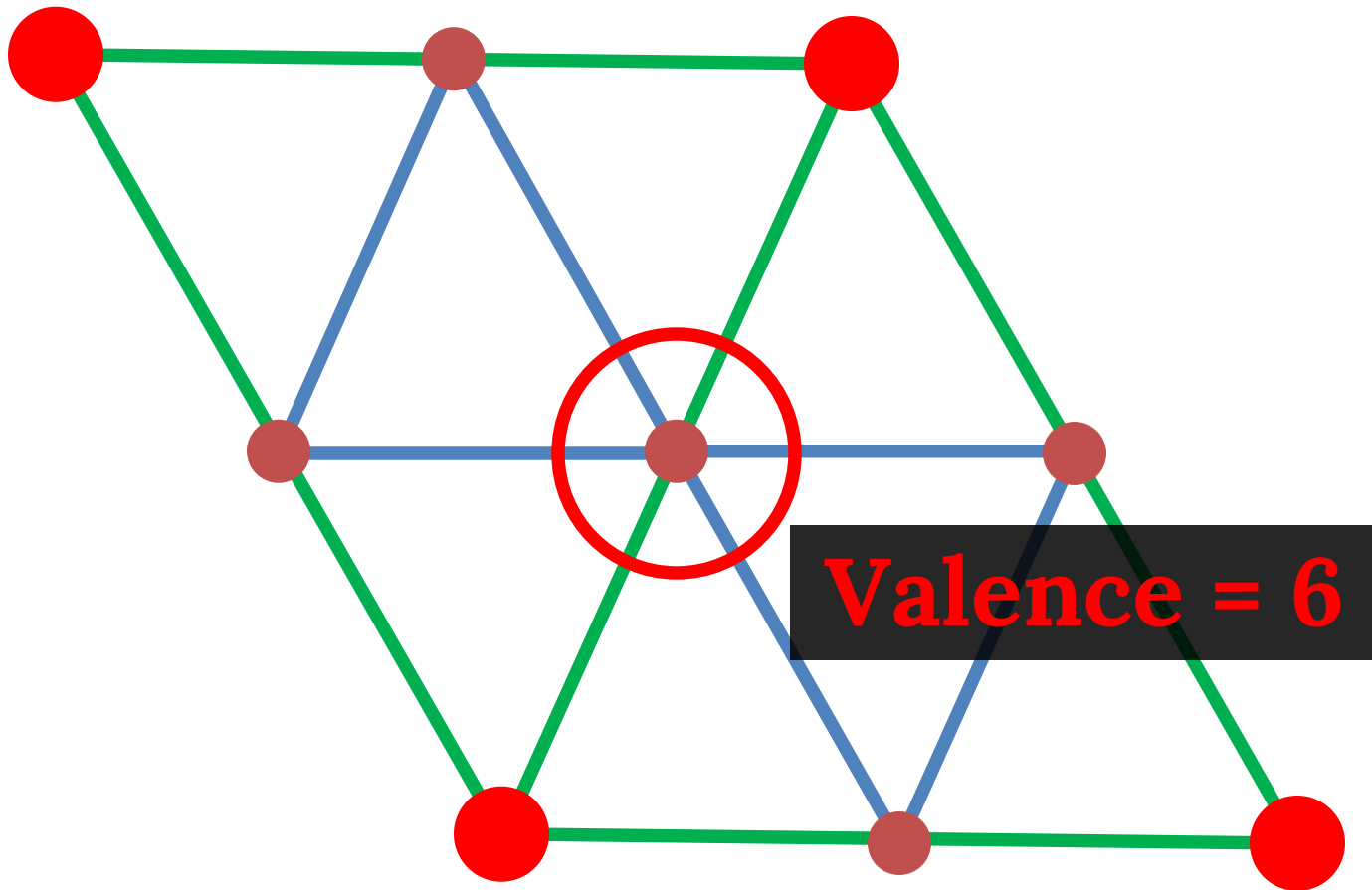
$$V = (v_1, v_2, \dots, v_n) \subset \mathbb{R}^n$$

$$E = (e_1, e_2, \dots, e_k) \subseteq V \times V$$

$$F = (f_1, f_2, \dots, f_m) \subseteq V \times V \times V$$

**Plus manifold conditions**

# Valence



# Euler Characteristic

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

$$\chi = 2 - 2g$$

**Defer proof**



$$g = 0$$



$$g = 1$$

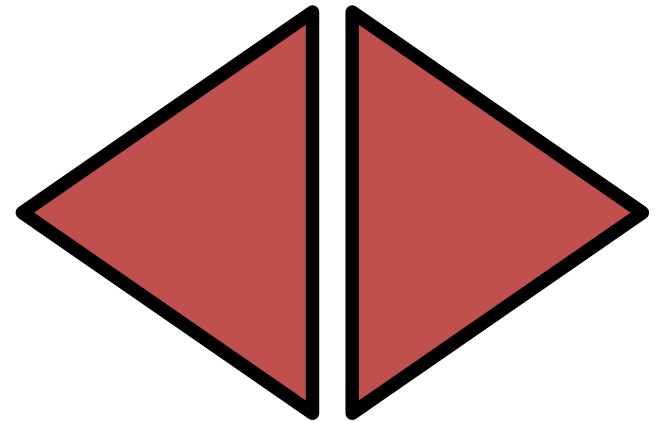


$$g = 2$$

# Consequences for Triangle Meshes

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

**“Each edge is adjacent to two faces. Each face has three edges.”**



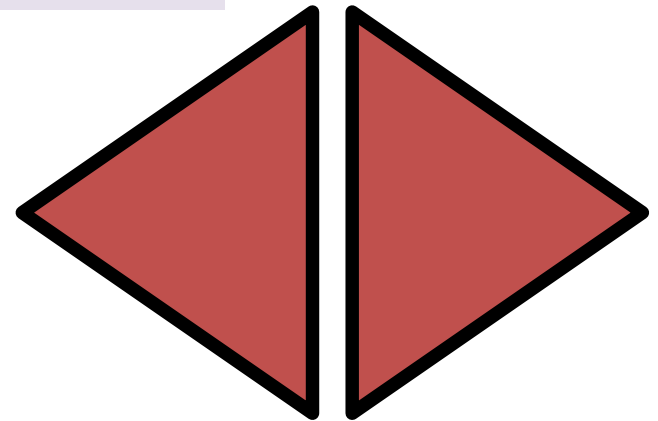
$$2E = 3F$$

**Closed mesh: Easy estimates!**

# Consequences for Triangle Meshes

$$V - \frac{1}{2}F := \chi$$

**“Each edge is adjacent to two faces. Each face has three edges.”**



$$2E = 3F$$

**Closed mesh: Easy estimates!**

# Consequences for Triangle Meshes

**Big  
number**

$$V - \frac{1}{2}F := \chi$$

**Small  
number**



$$F \approx 2V$$

**Closed mesh: Easy estimates!**

# Consequences for Triangle Meshes

$$E \approx 3V$$

$$F \approx 2V$$

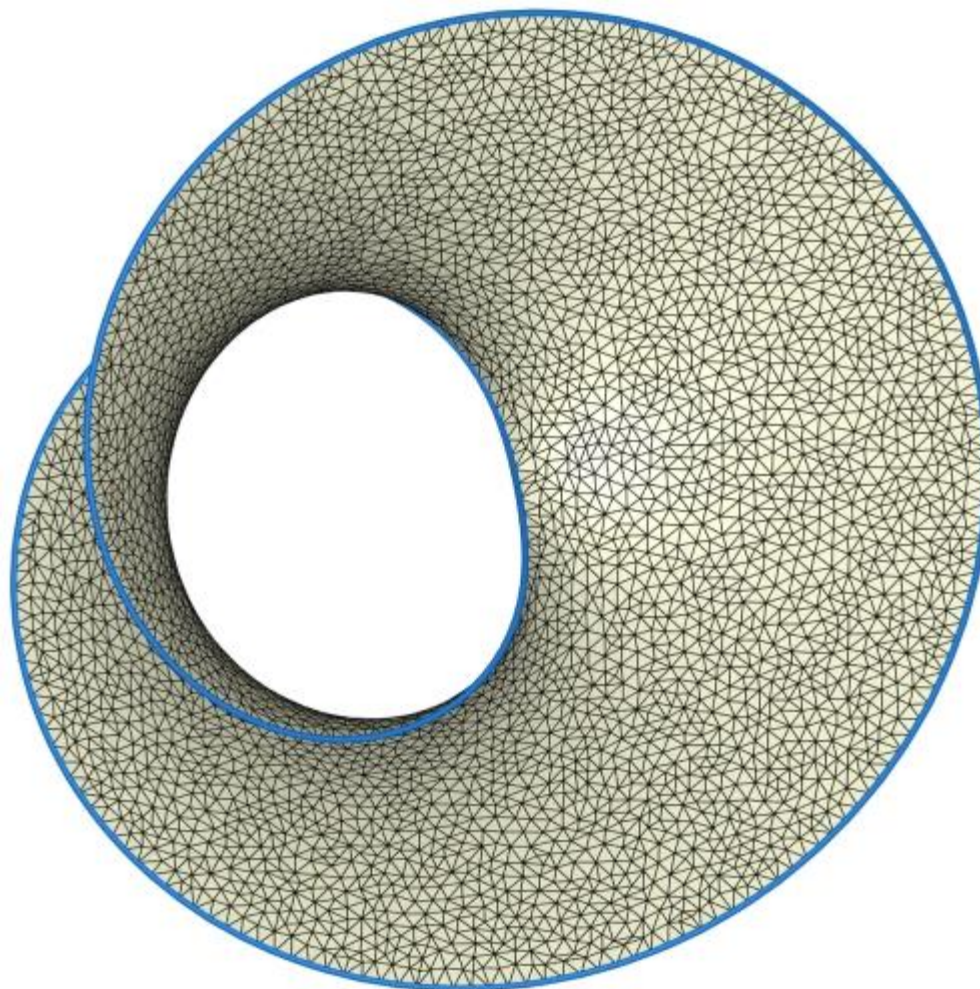
average valence  $\approx 6$

*Why?!*

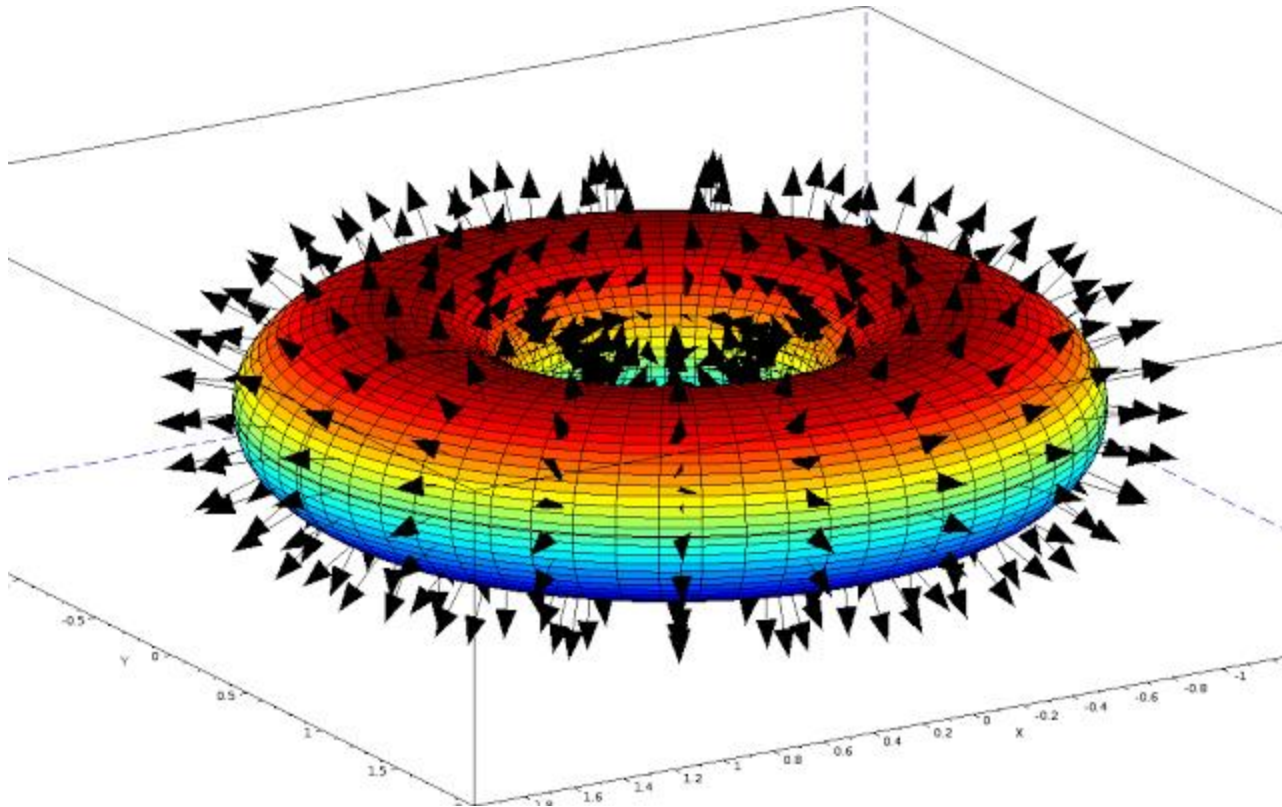
**General estimates**



# Orientability



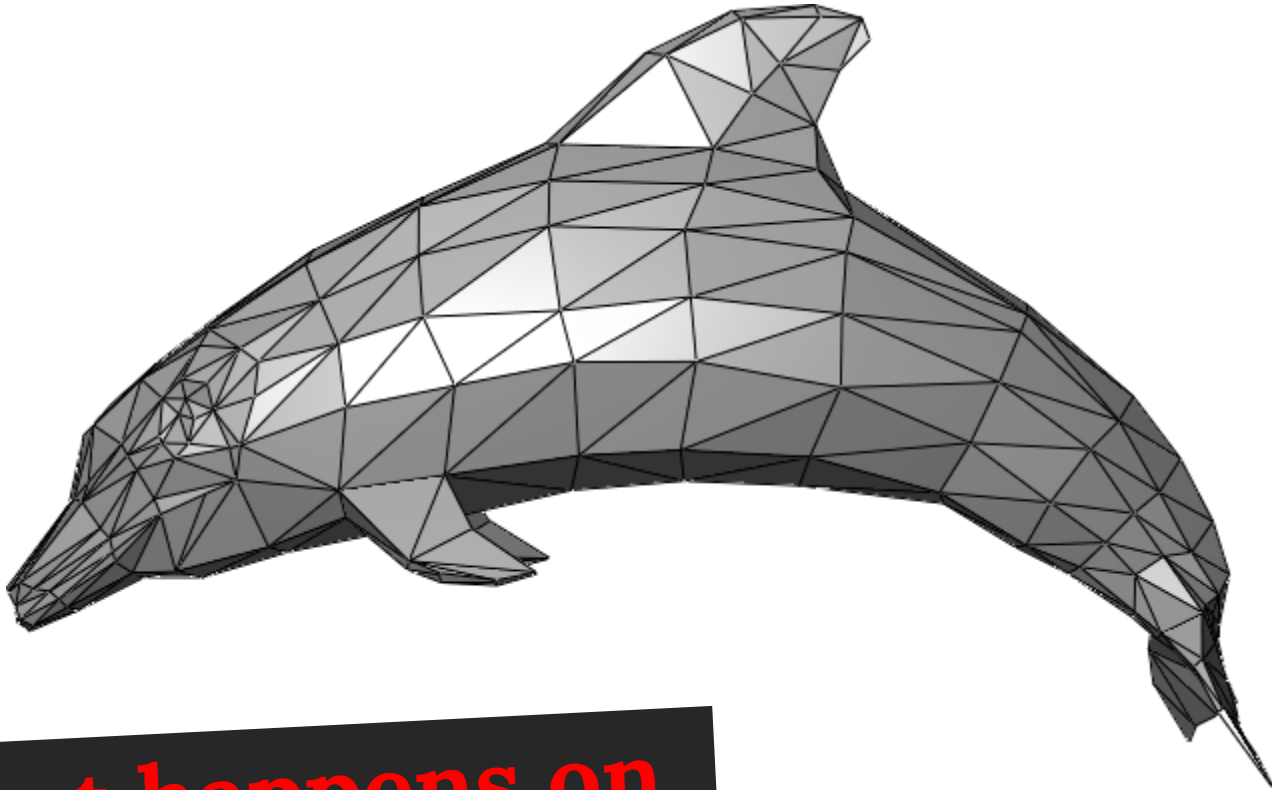
# Smooth Surface Definition



<https://lh3.googleusercontent.com/-njXPH7NSX5c/VV4PXu54n9I/AAAAAAAAAJM/m6TGg3ZVKGE/w640-h400-p->

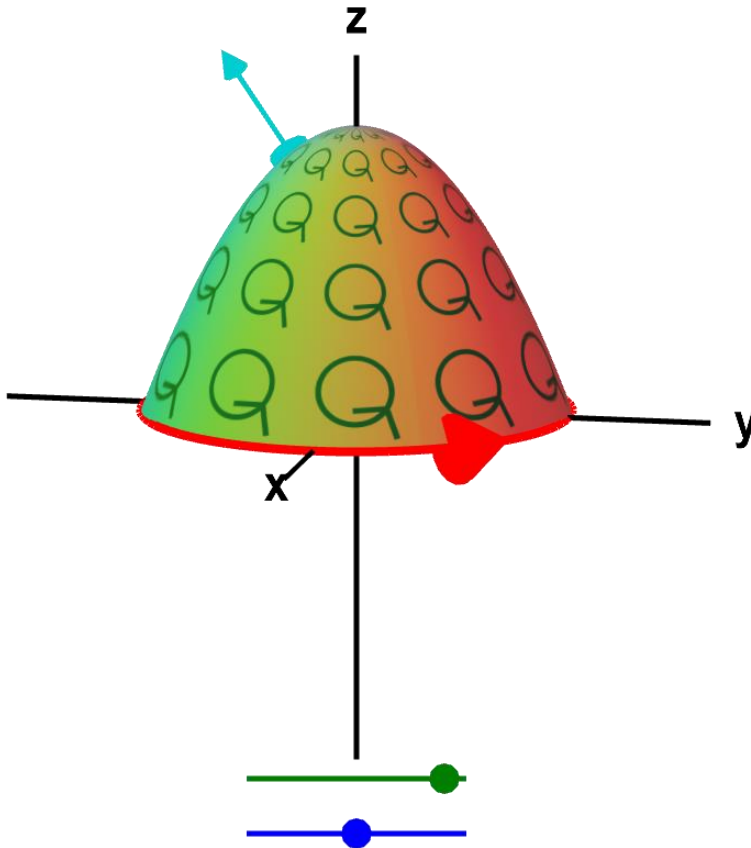
**Continuous field of normal**

# Issue on Triangle Mesh

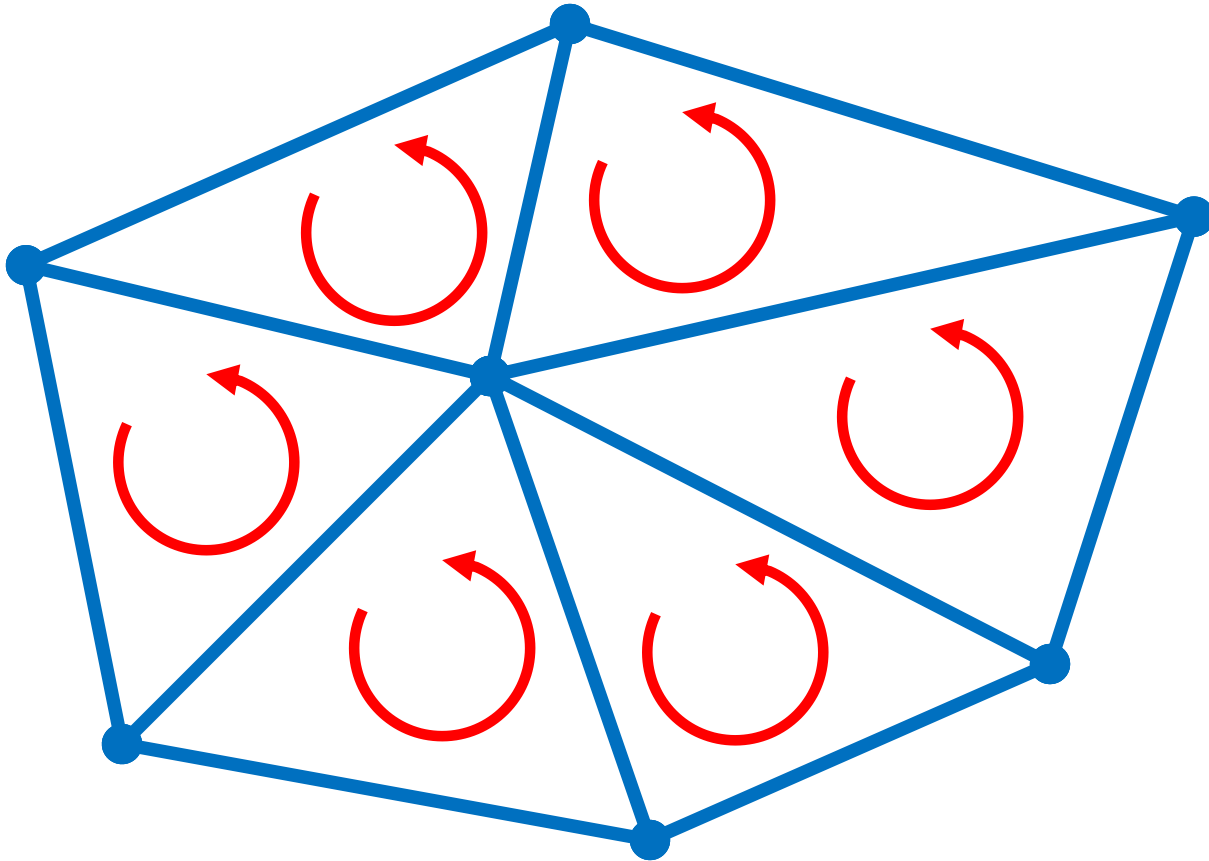


**What happens on  
edges/vertices?**

# Right-Hand Rule



# Discrete Orientability



**Normal field isn't continuous**

# Data Structures for Surfaces

**Must represent  
geometry and topology.**

# Simplest Format

x1	y1	z1	/	x2	y2	z2	/	x3	y3	z3
x1	y1	z1	/	x2	y2	z2	/	x3	y3	z3
x1	y1	z1	/	x2	y2	z2	/	x3	y3	z3
x1	y1	z1	/	x2	y2	z2	/	x3	y3	z3
x1	y1	z1	/	x2	y2	z2	/	x3	y3	z3

**No topology!**

# Triangle soup



# Simplest Format

```
x1 y1 z1 / x2 y2 z2 / x3 y3 z3  
x1 y1 z1 / x2 y2 z2 / x3 y3 z3  
x1 y1 z1 / x2 y2 z2 / x3 y3 z3  
x1 y1 z1 / x2 y2 z2 / x3 y3 z3  
x1 y1 z1 / x2 y2 z2 / x3 y3 z3
```

**No topology!**

```
glBegin(GL_TRIANGLES)
```

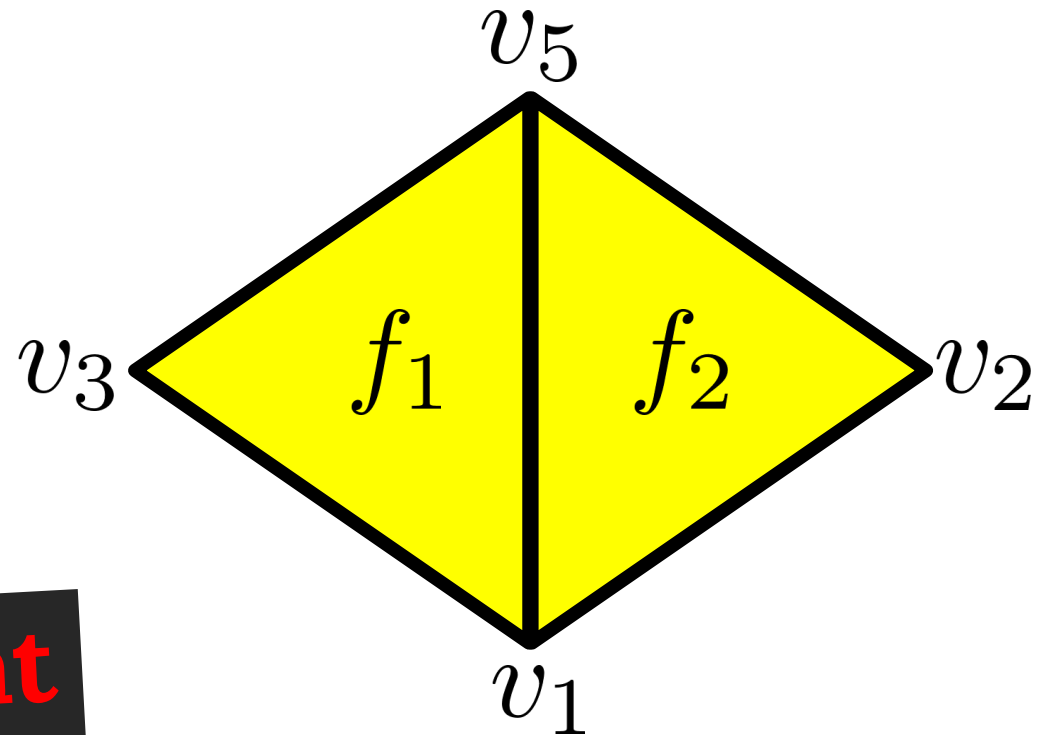
CS 468 2011 (M. Ben-Chen), other slides

# Triangle soup

# Factor Out Vertices

```
f 1 5 3
f 5 1 2
...
v 0.2 1.5 3.2
v 5.2 4.1 8.9
...
```

**.obj format**



## Shared vertex structure

# Simple Mesh Smoothing

```
for i=1 to n
  for each vertex v
    v = .5*v +
      .5*(average of neighbors) ;
```

# Typical Queries

- **Neighboring vertices to a vertex**
- **Neighboring faces to an edge**
- **Edges adjacent to a face**
- **Edges adjacent to a vertex**

**Mostly localized**

# Typical Queries

- **Neighboring** vertices to a vertex
- **Neighboring** faces to an edge
- Edges **adjacent** to a face
- Edges **adjacent** to a vertex

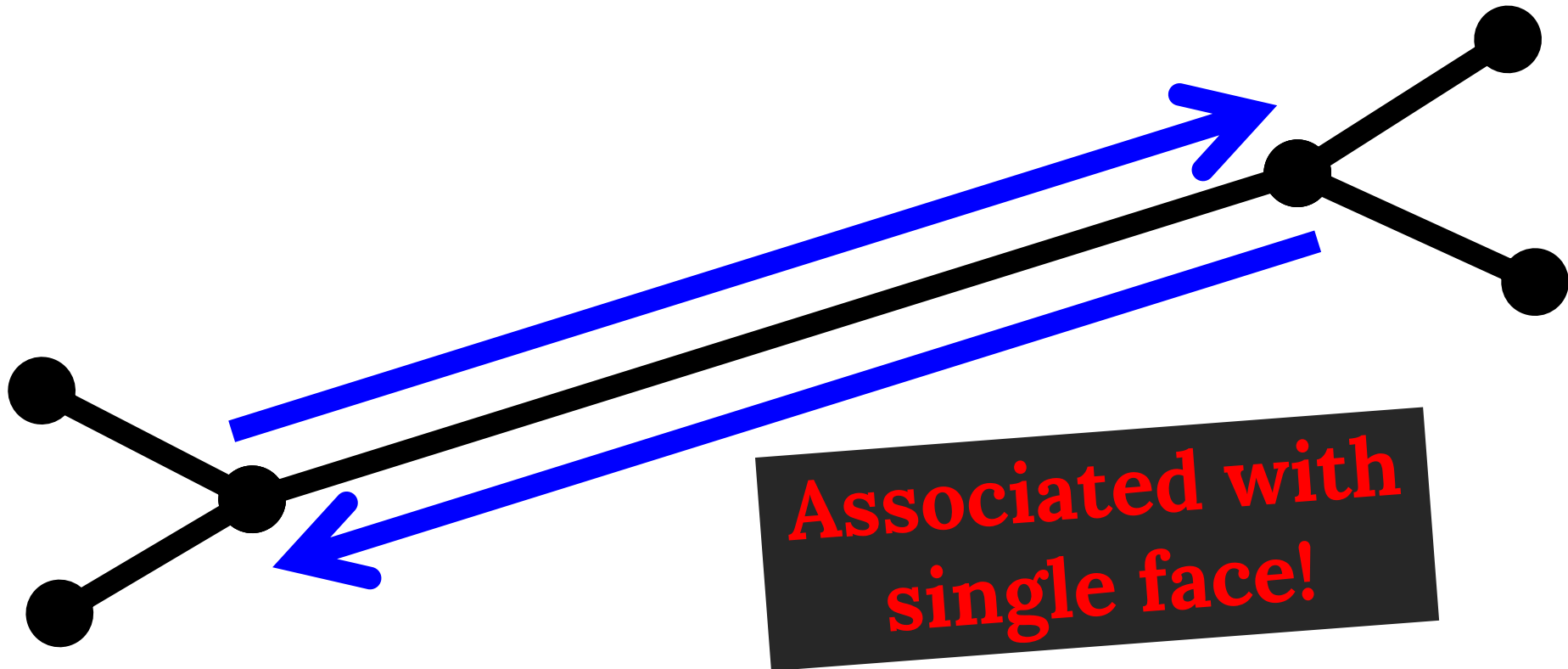
**Mostly localized**

# Pieces of Halfedge Data Structure

- **Vertices**
- **Faces**
- **Half-edges**

**Structure tuned for meshes**

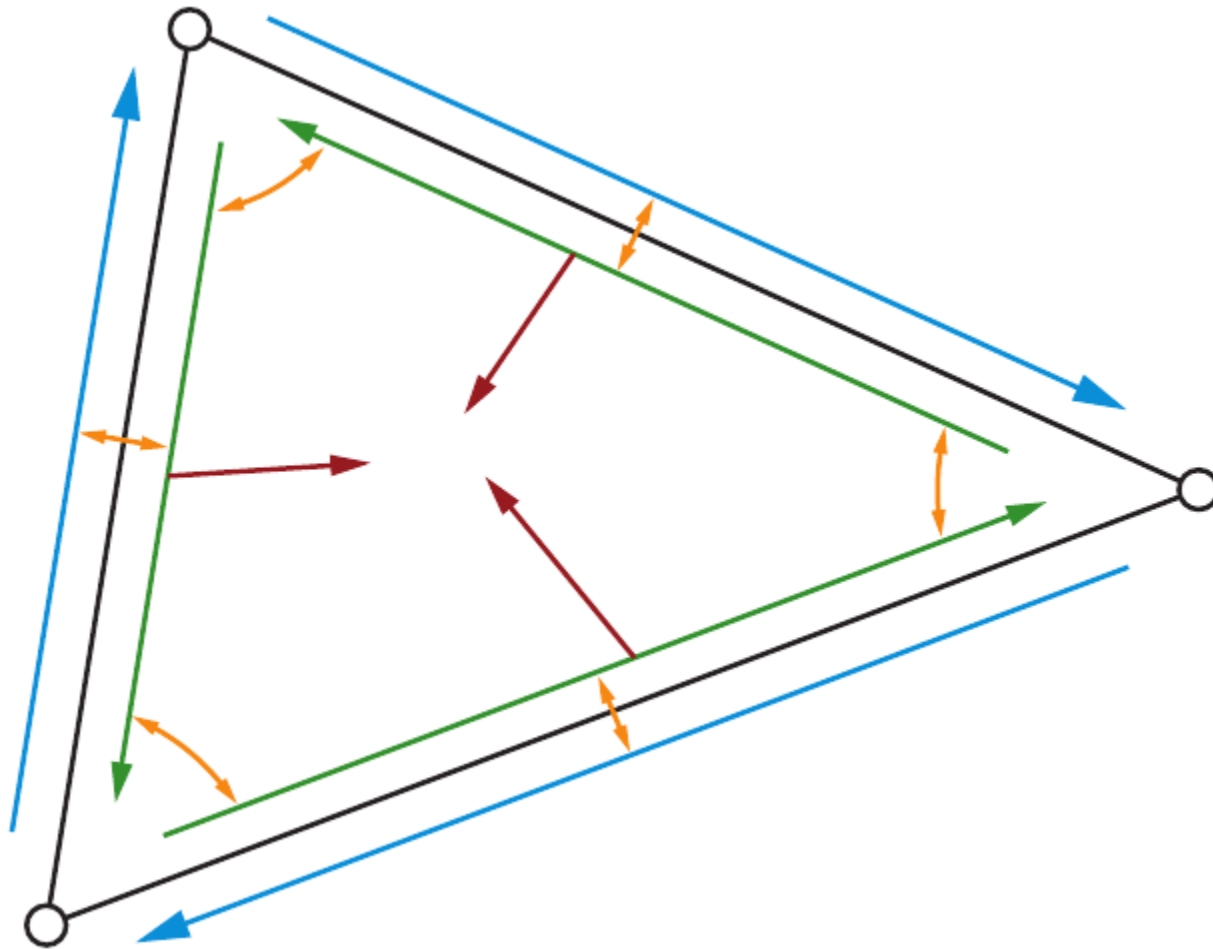
# Halfedge?



## Oriented edge



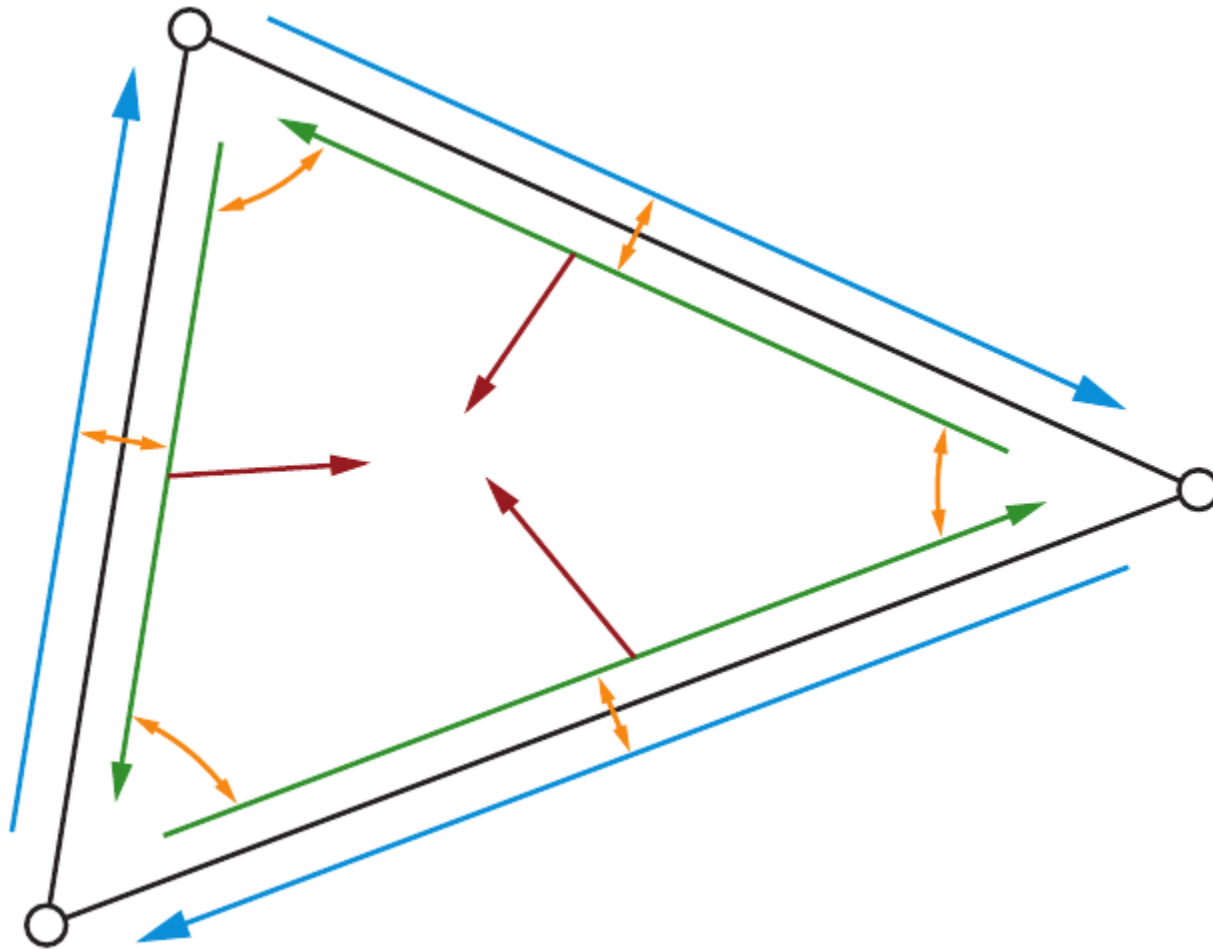
# Halfedge Data Types



**Vertex** stores:

- Arbitrary outgoing halfedge

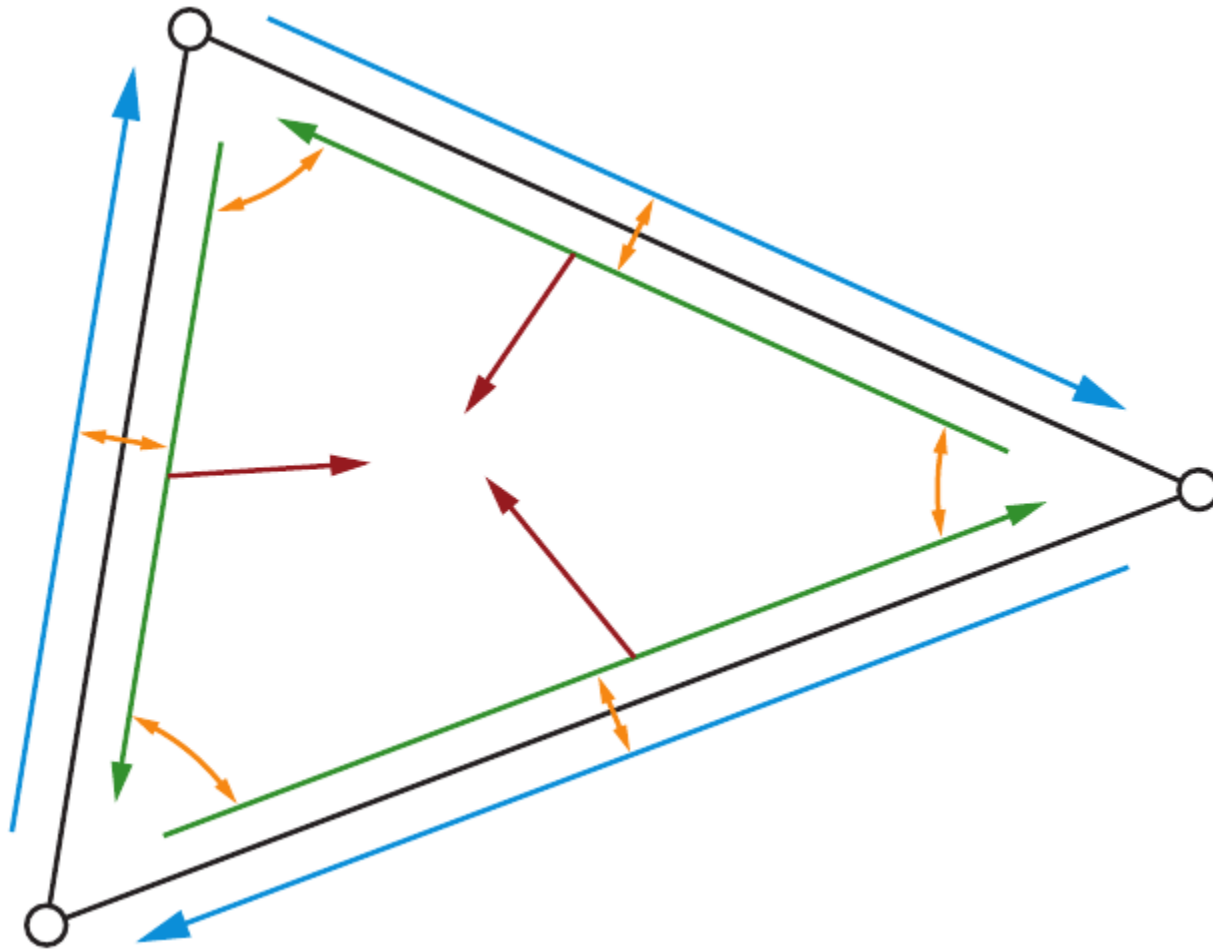
# Halfedge Data Types



**Face stores:**

- Arbitrary adjacent halfedge

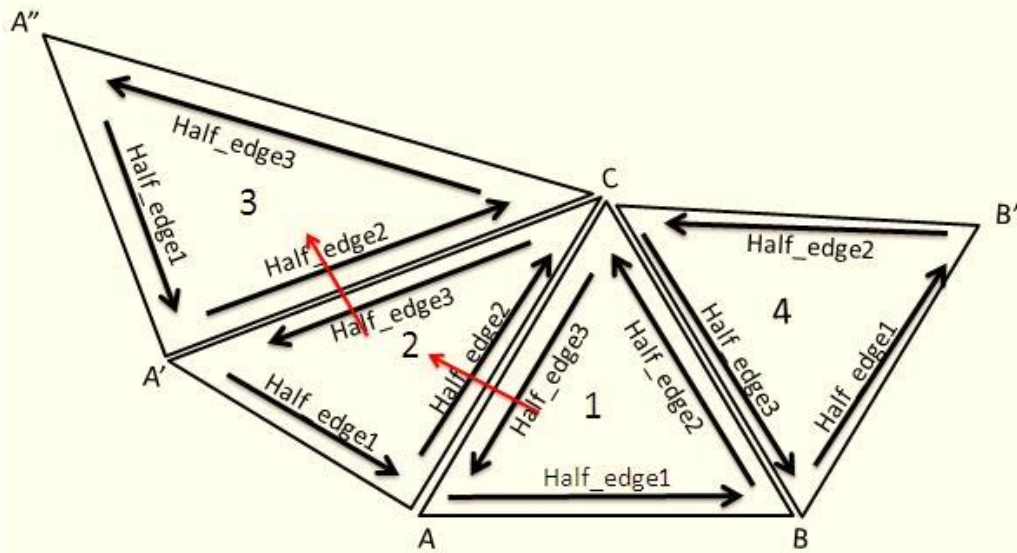
# Halfedge Data Types



**Halfedge** stores:

- **Flip**
- **Next**
- **Face**
- **Vertex**

# Iterating Over Vertex Neighbors



```
Iterate(v) :  
startEdge = v.out;  
e = startEdge;  
do  
    process(e.flip.from)  
    e = e.flip.next  
while e != startEdge
```

# Only Scratching the Surface

Eurographics Symposium on Geometry Processing (2005)  
M. Desbrun, H. Pottmann (Editors)

## Streaming Compression of Triangle Meshes

Martin Isenburg<sup>1†</sup>

Peter Lindstrom<sup>2</sup>

Jack Snoeyink<sup>1</sup>

<sup>1</sup> University of North Carolina at Chapel Hill

<sup>2</sup> Lawrence Livermore National Labs

EUROGRAPHICS 2011 / M. Chen and O. Deussen  
(Guest Editors)

*Volume 30 (2011), Number 2*

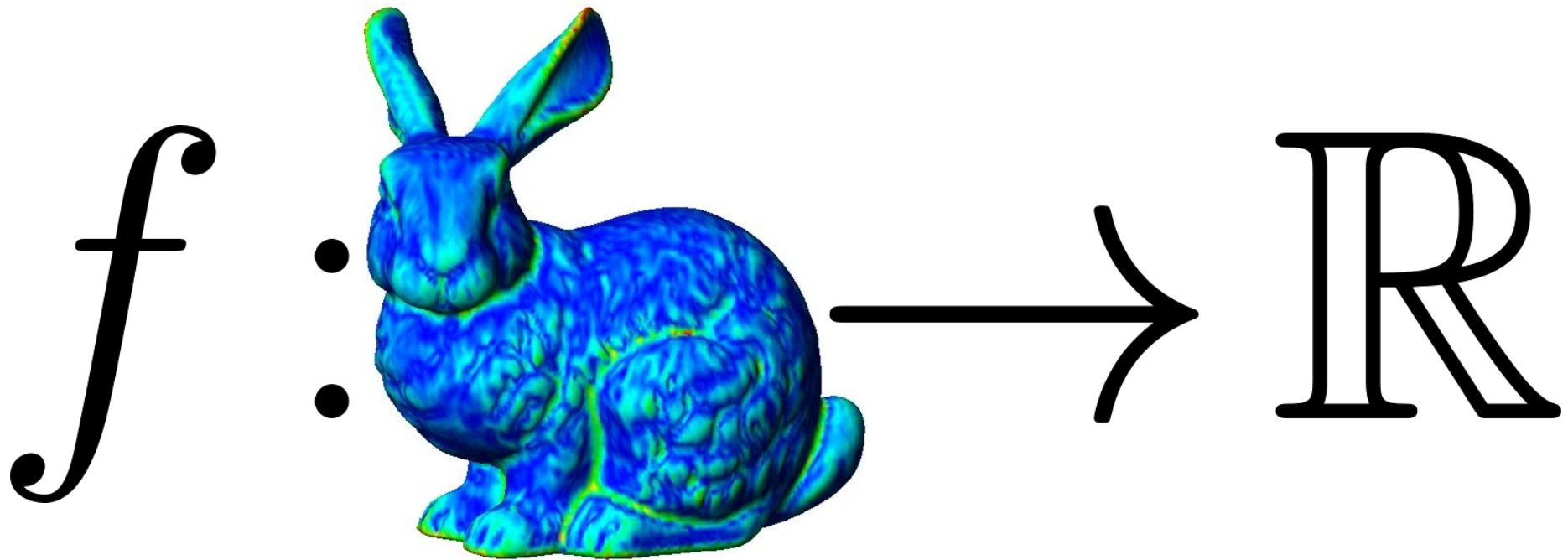
## SQuad: Compact Representation for Triangle Meshes

Topraj Gurung<sup>1</sup>, Daniel Laney<sup>2</sup>, Peter Lindstrom<sup>2</sup>, Jarek Rossignac<sup>1</sup>

<sup>1</sup>Georgia Institute of Technology

<sup>2</sup>Lawrence Livermore National Laboratory

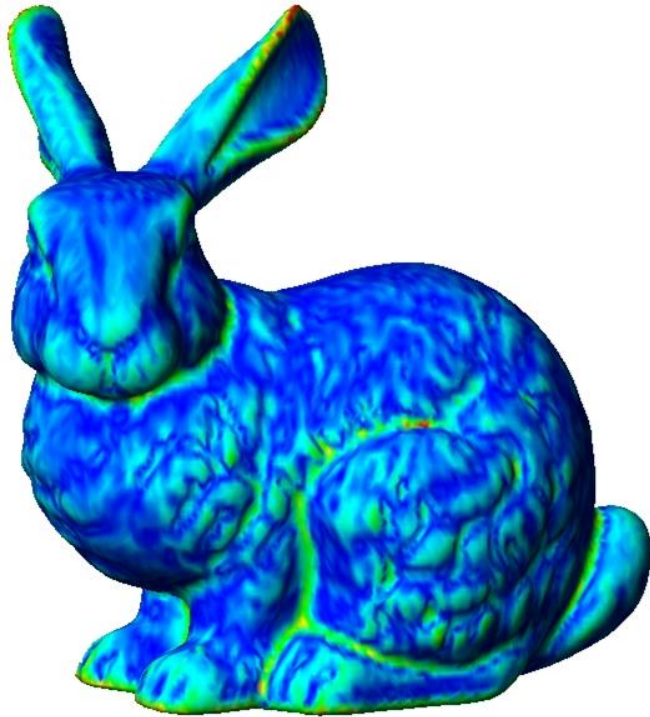
# Scalar Functions



[http://www.ieeta.pt/polymeco/Screenshots/PolyMeCo\\_OneView.jpg](http://www.ieeta.pt/polymeco/Screenshots/PolyMeCo_OneView.jpg)

**Map points to real numbers**

# Discrete Scalar Functions



$$f \in \mathbb{R}^{|V|}$$

[http://www.ieeta.pt/polymeco/Screenshots/PolyMeCo\\_OneView.jpg](http://www.ieeta.pt/polymeco/Screenshots/PolyMeCo_OneView.jpg)

**Map vertices to real numbers**

# Question

**What is the integral of  $f$ ?**

$$\int_M f \, dA$$



# Finite Elements Approach

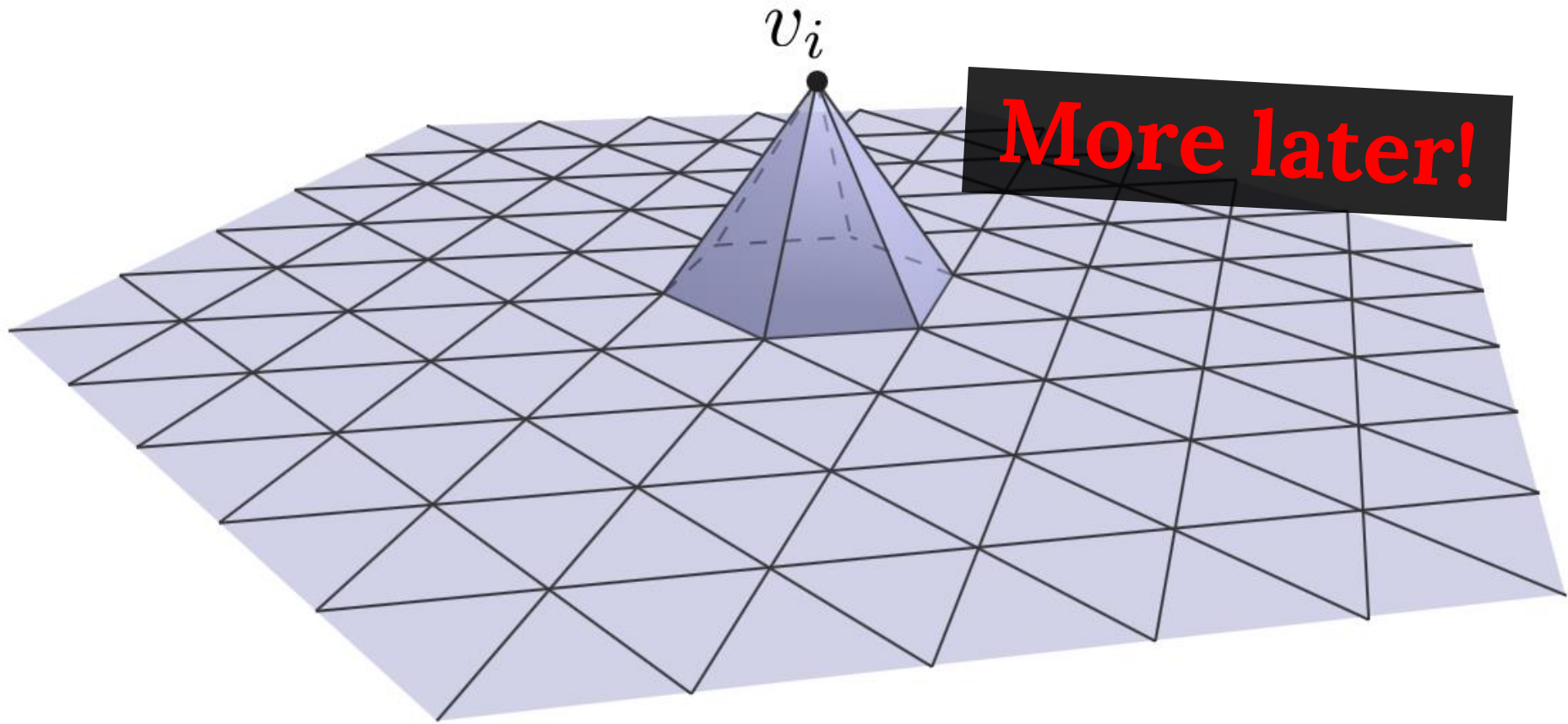
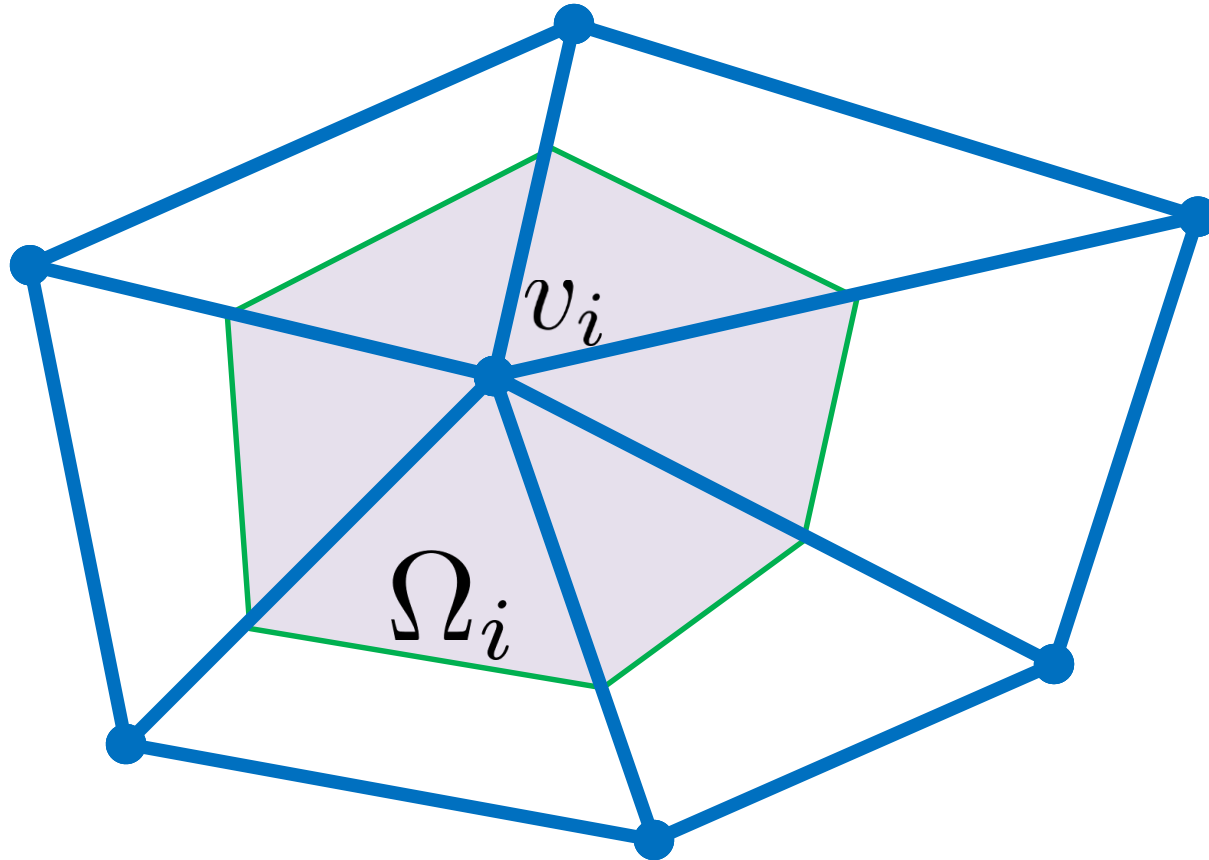


Image courtesy K. Crane

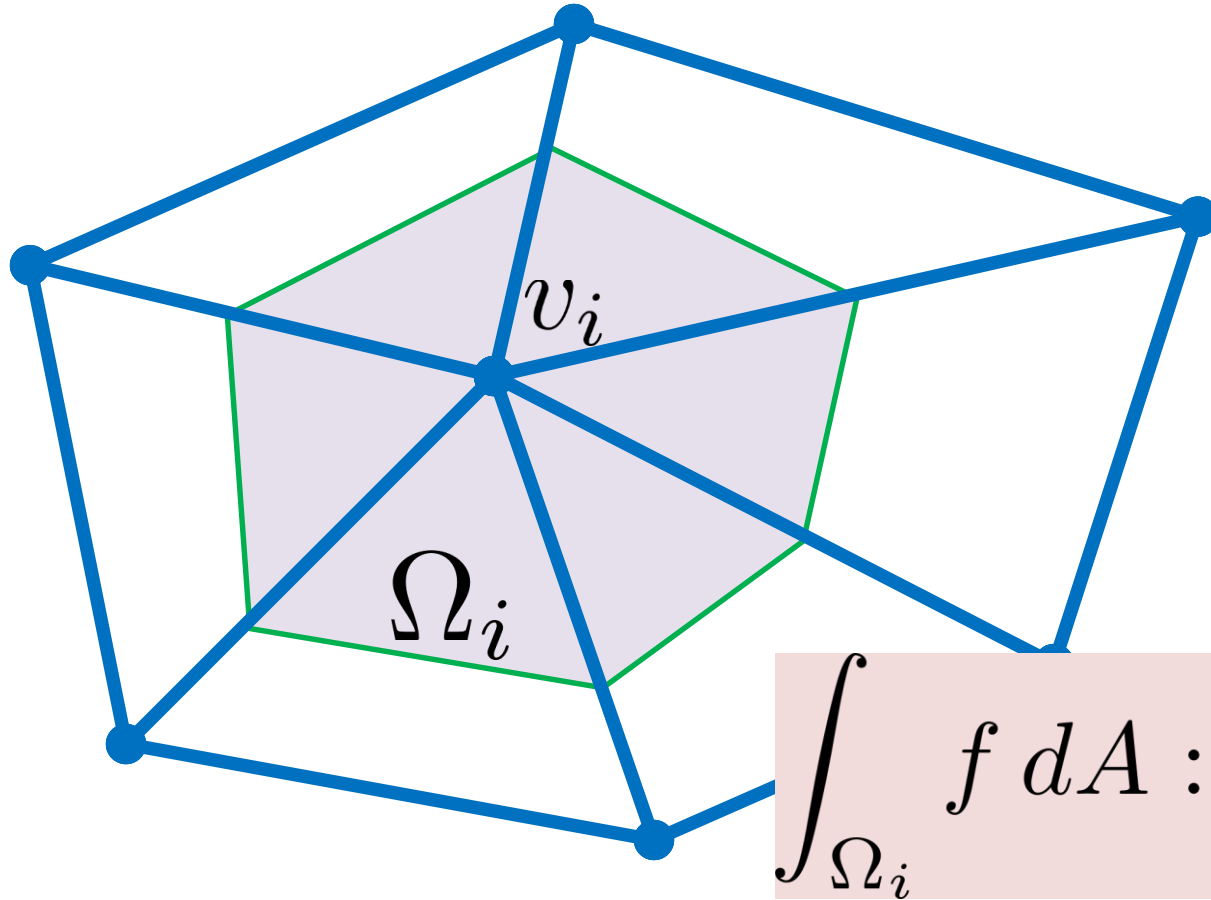
**Use hat functions to interpolate**

# Dual Cell



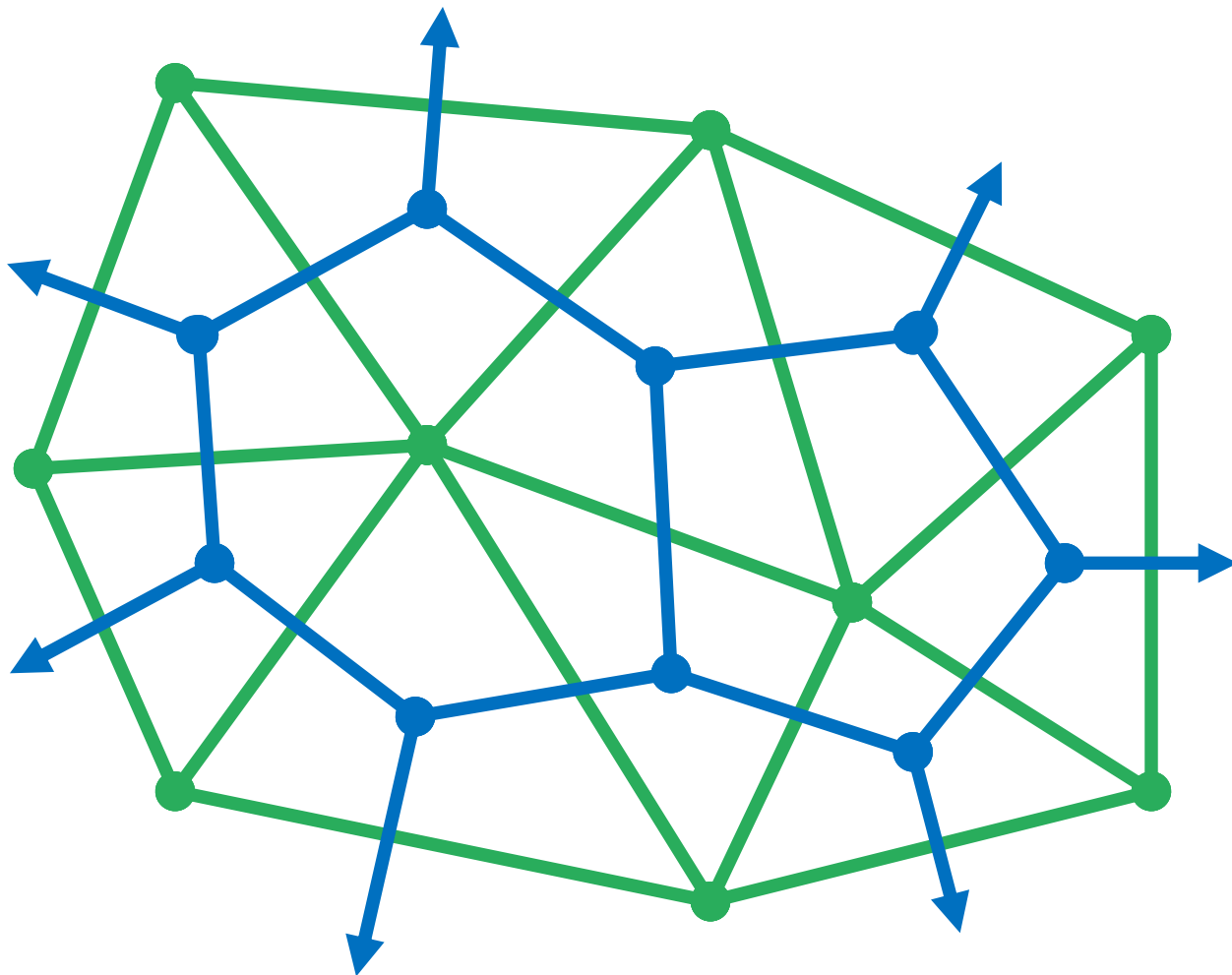
**Discrete version of  $dA$**

# Dual Cell

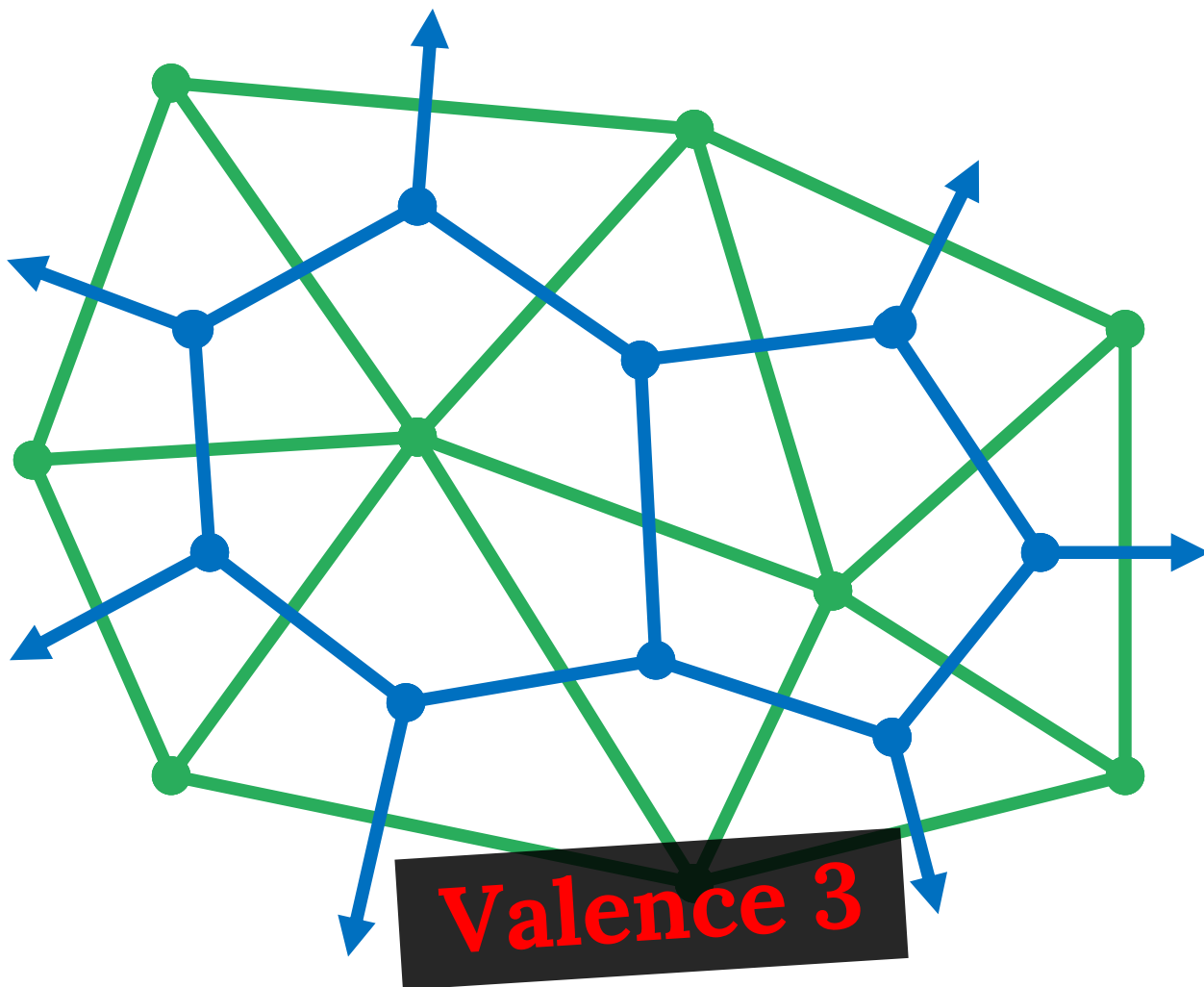


**Discrete version of  $dA$**

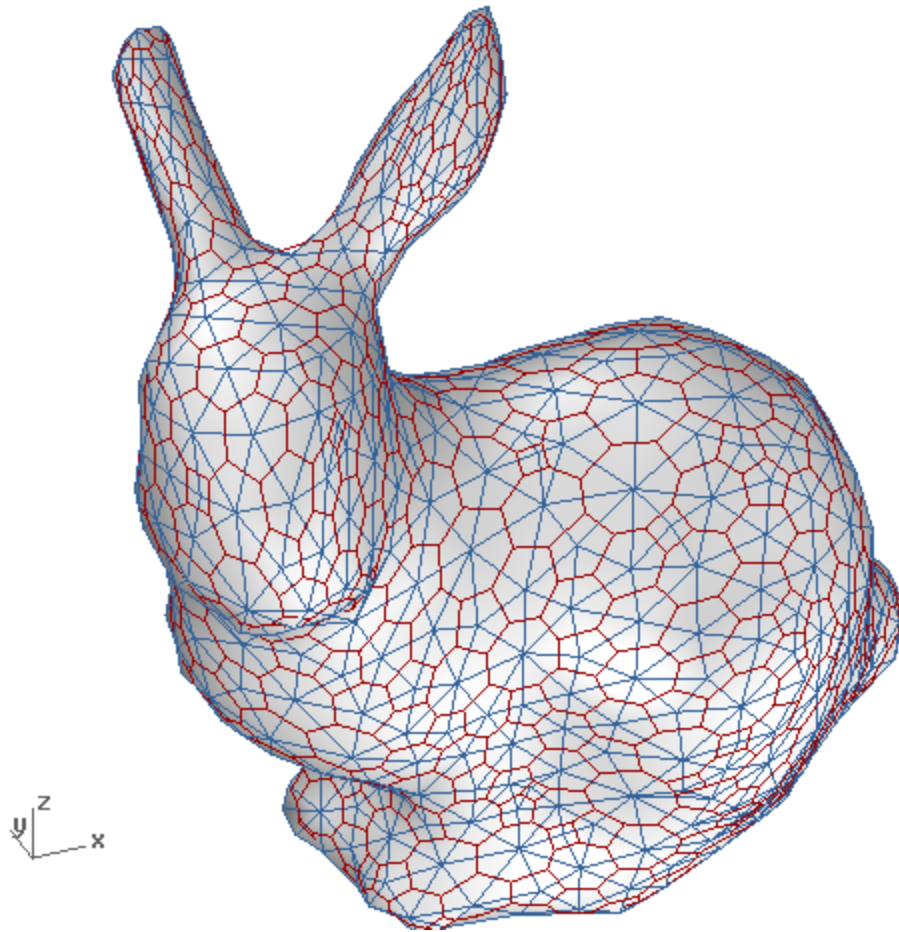
# Dual Complex



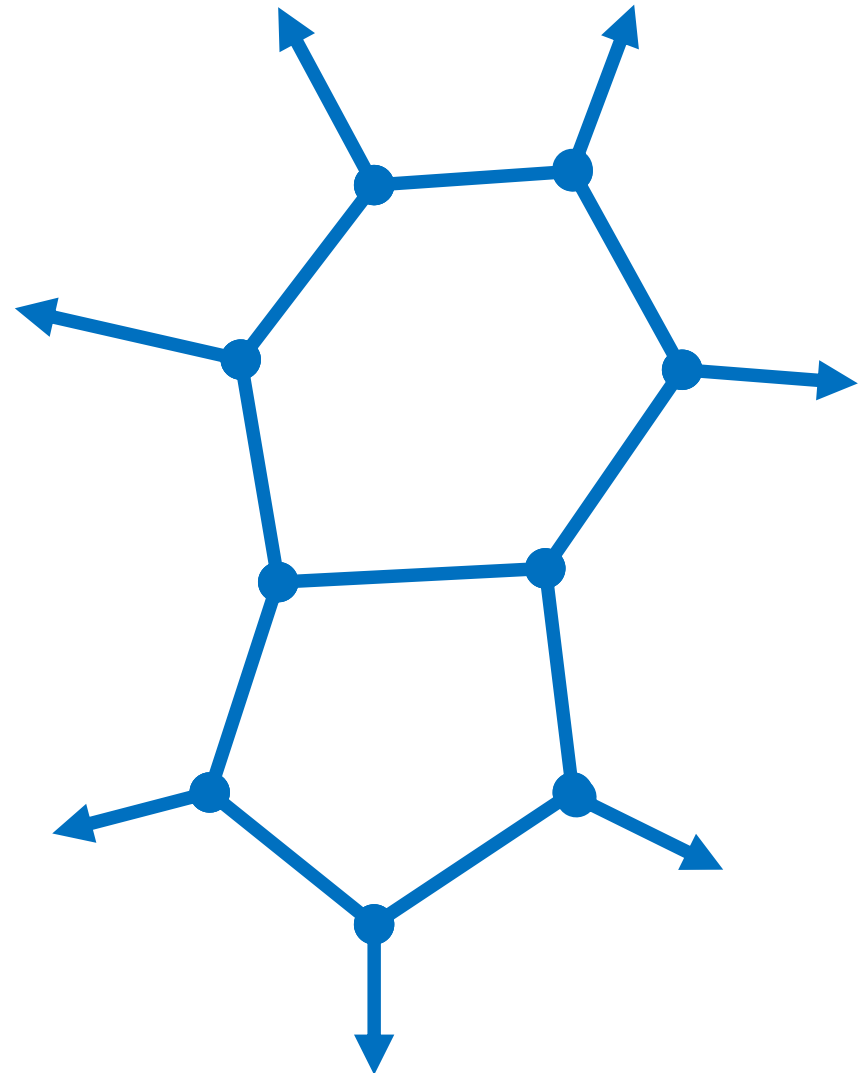
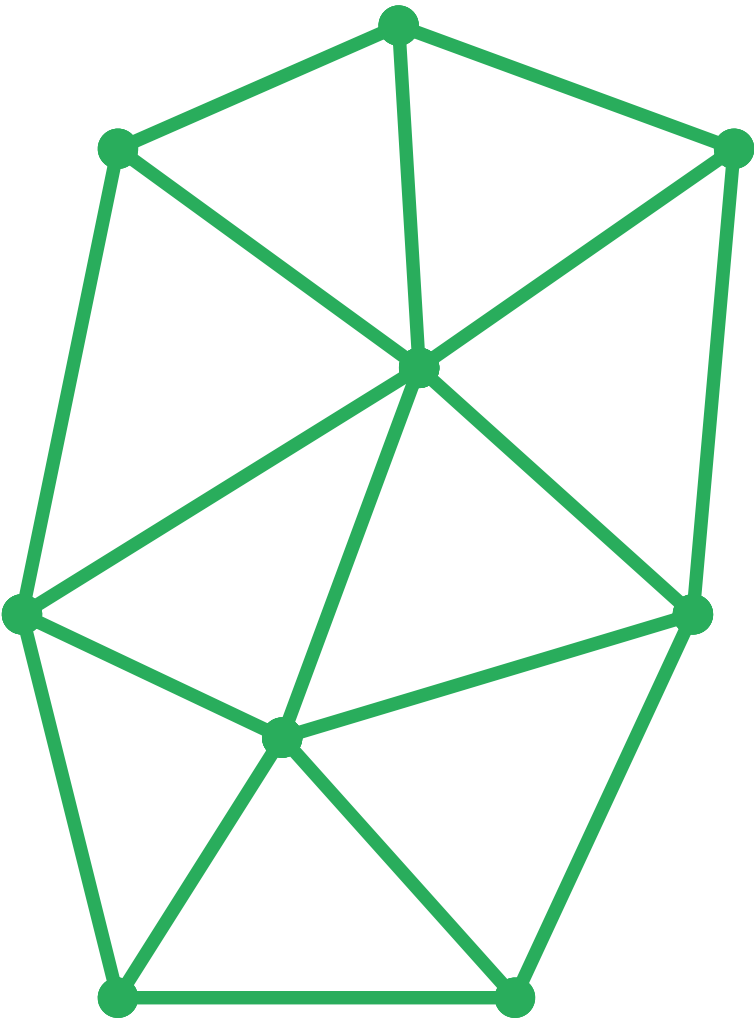
# Dual Complex



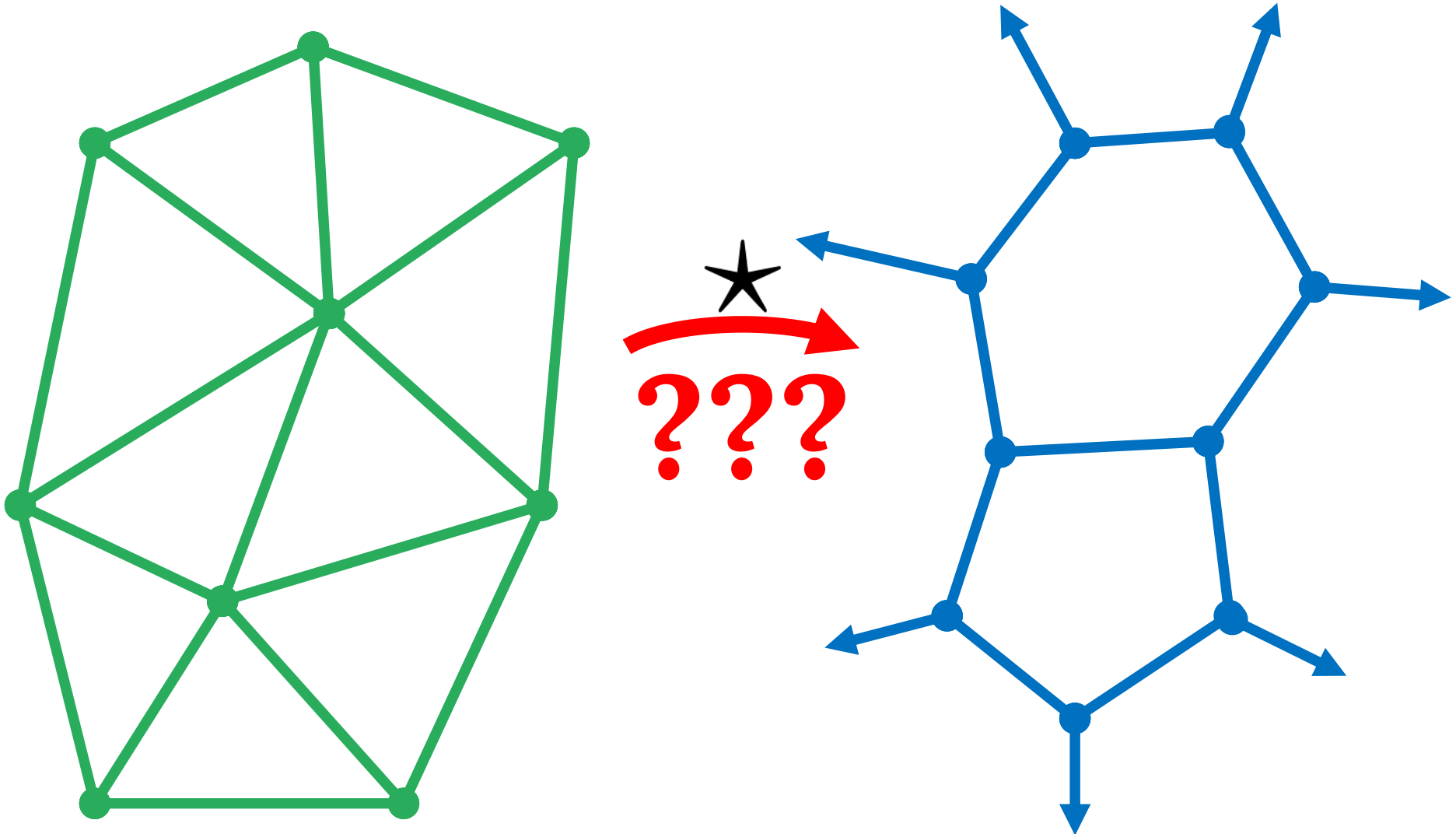
# One Surface, Two Meshes



# One Surface, Two Halfedges

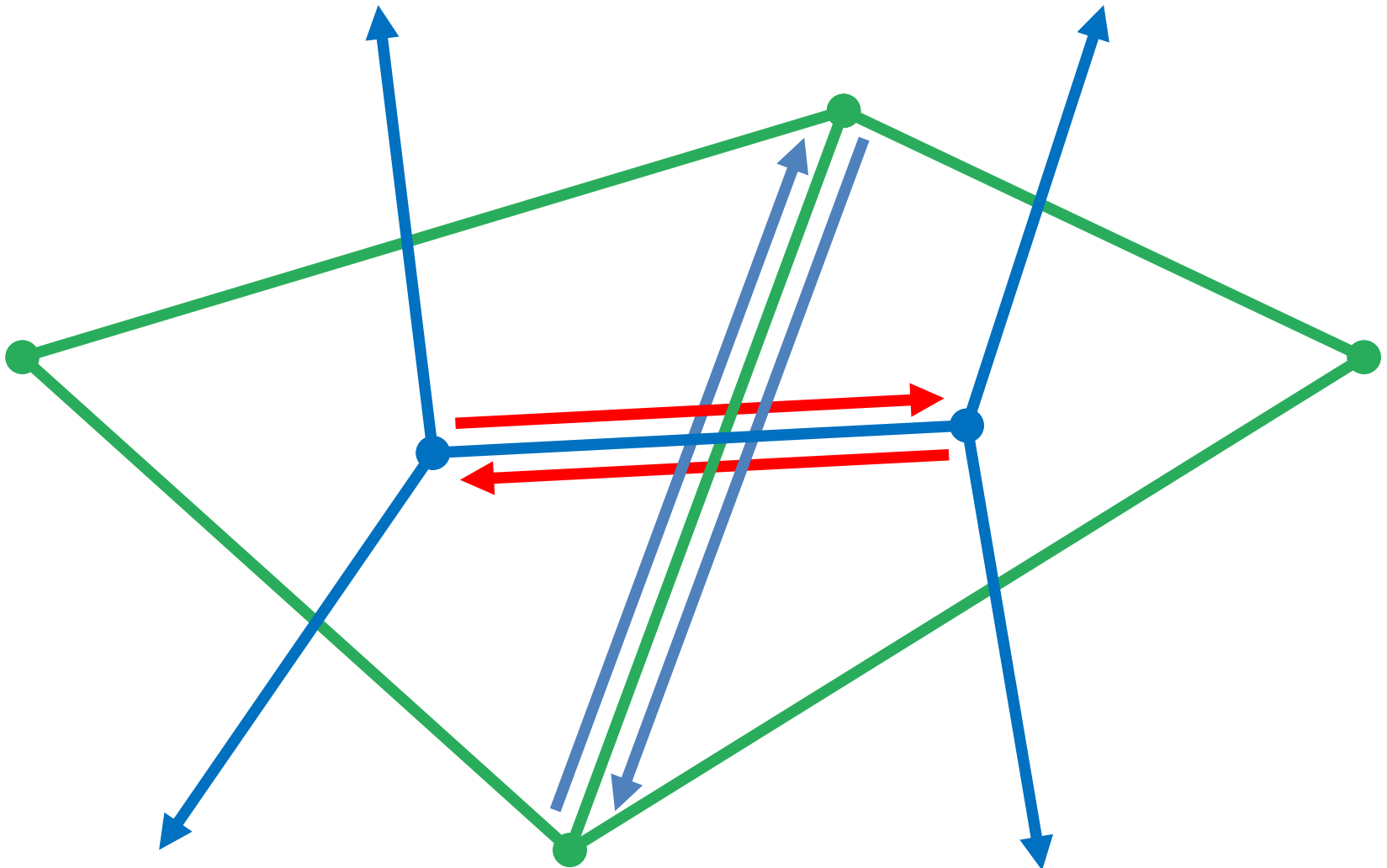


# Missing Operation



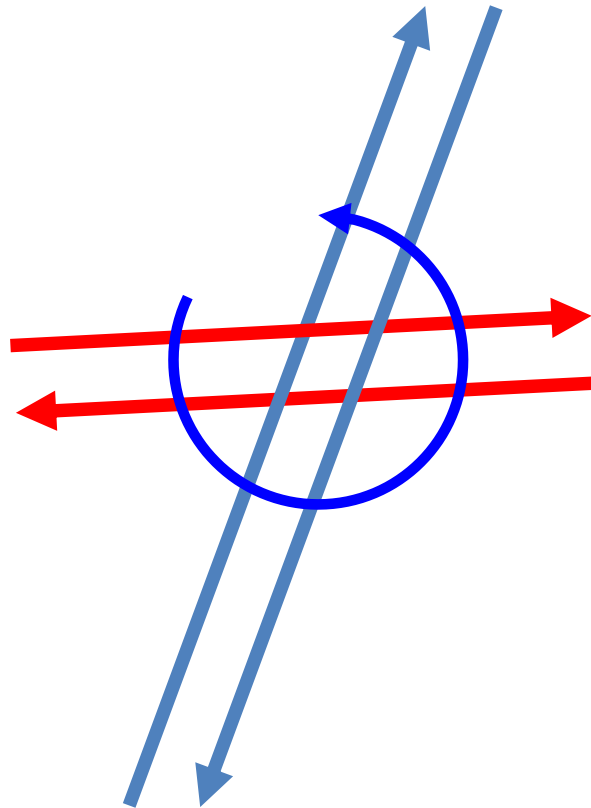


# Quad Edge



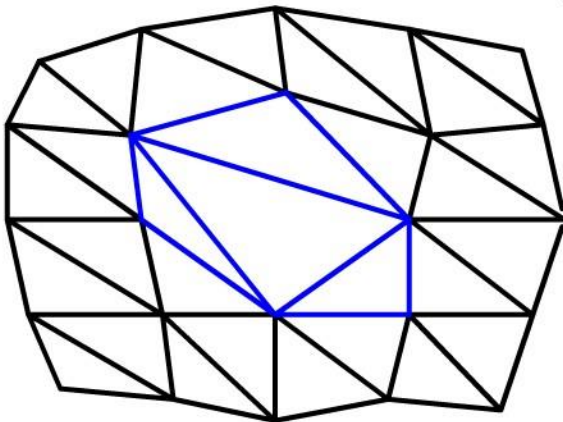
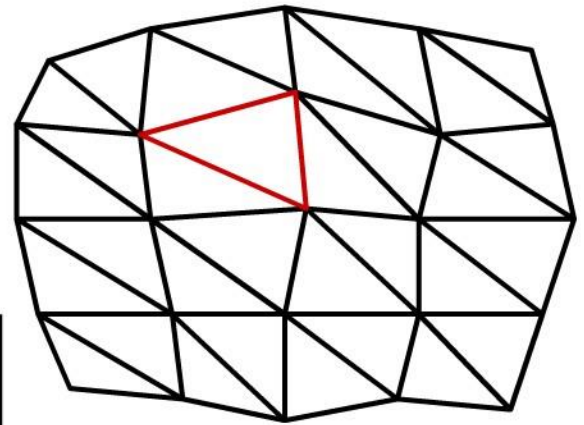
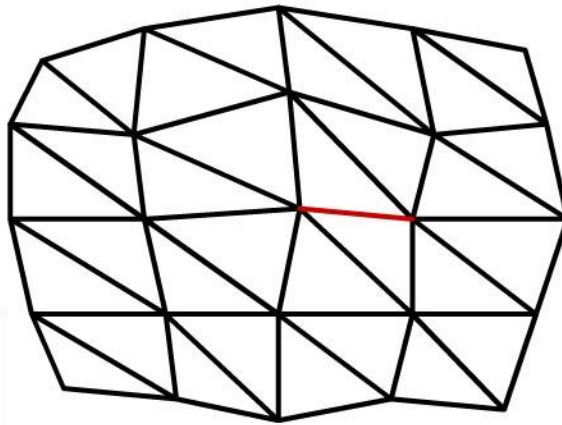
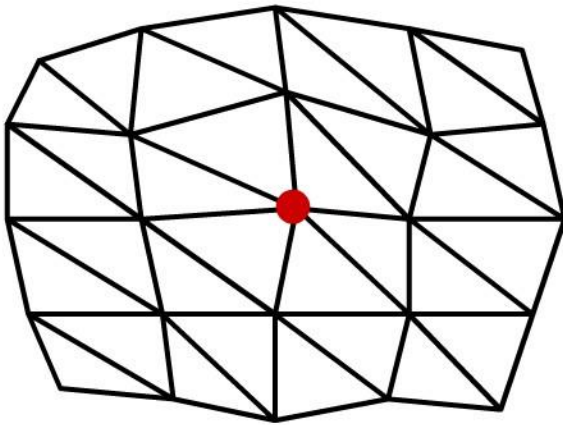
# Rotation Operation

$e \rightarrow \text{Rot} \rightarrow \text{Rot} = e \rightarrow \text{Flip}$

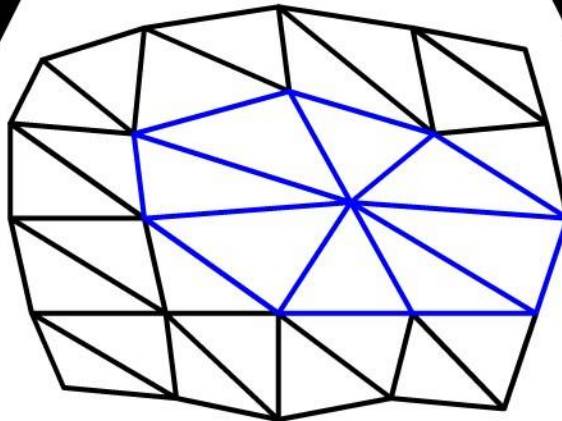


# Topological Operations

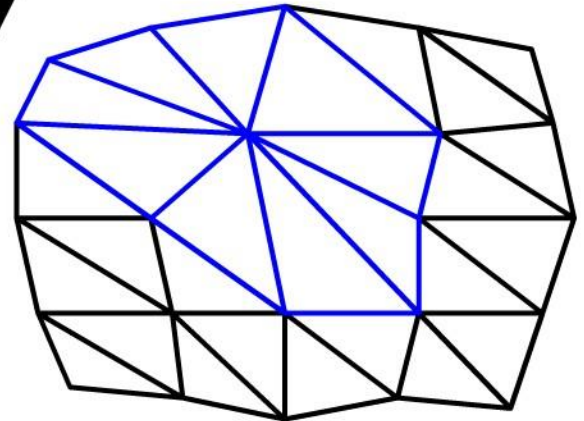
Original Mesh Segment



Vertex Removal



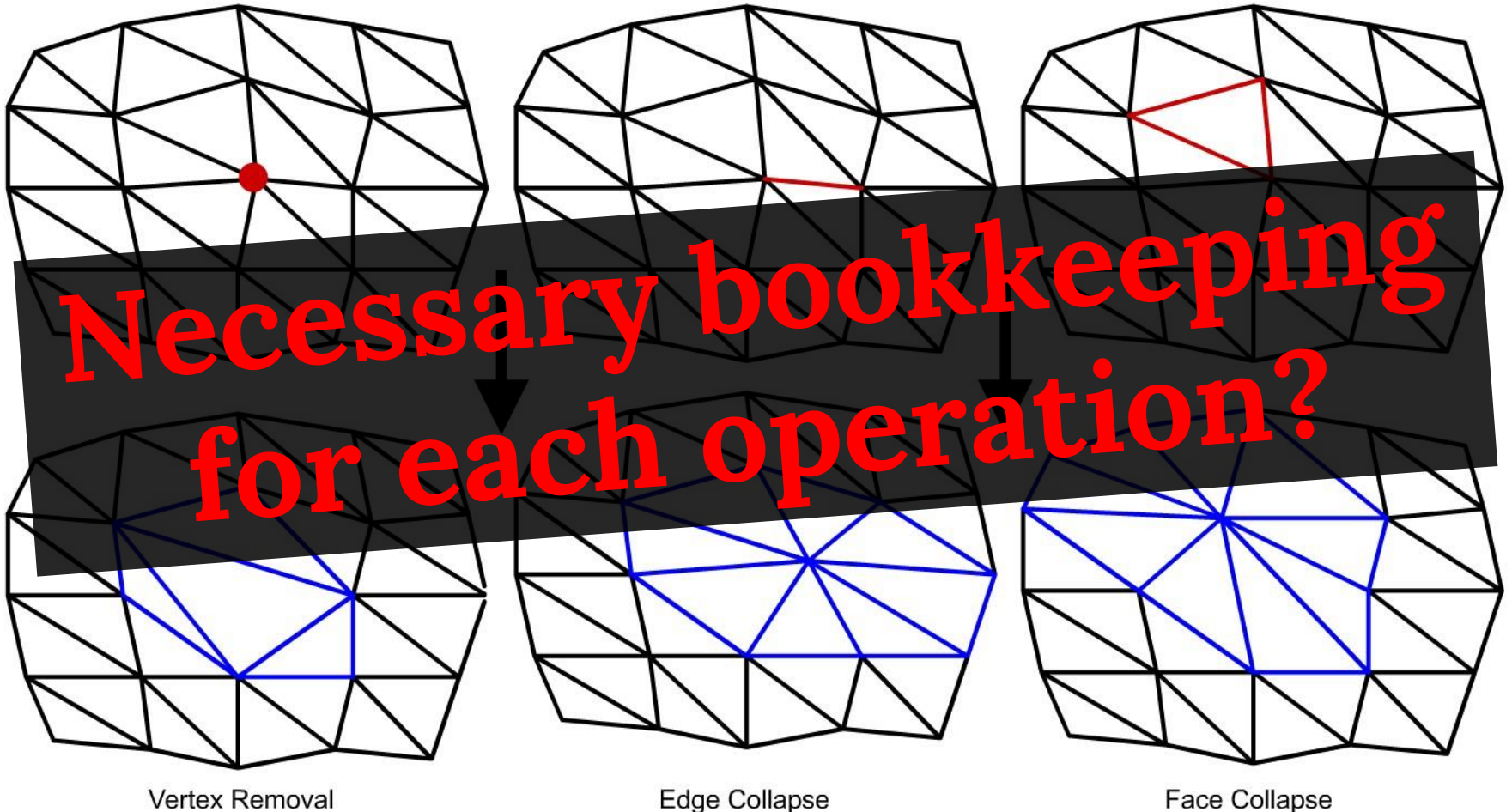
Edge Collapse



Face Collapse

# Topological Operations

Original Mesh Segment



# Take-Away

**Simple** Data  
Structures

**Complex** Data  
structures

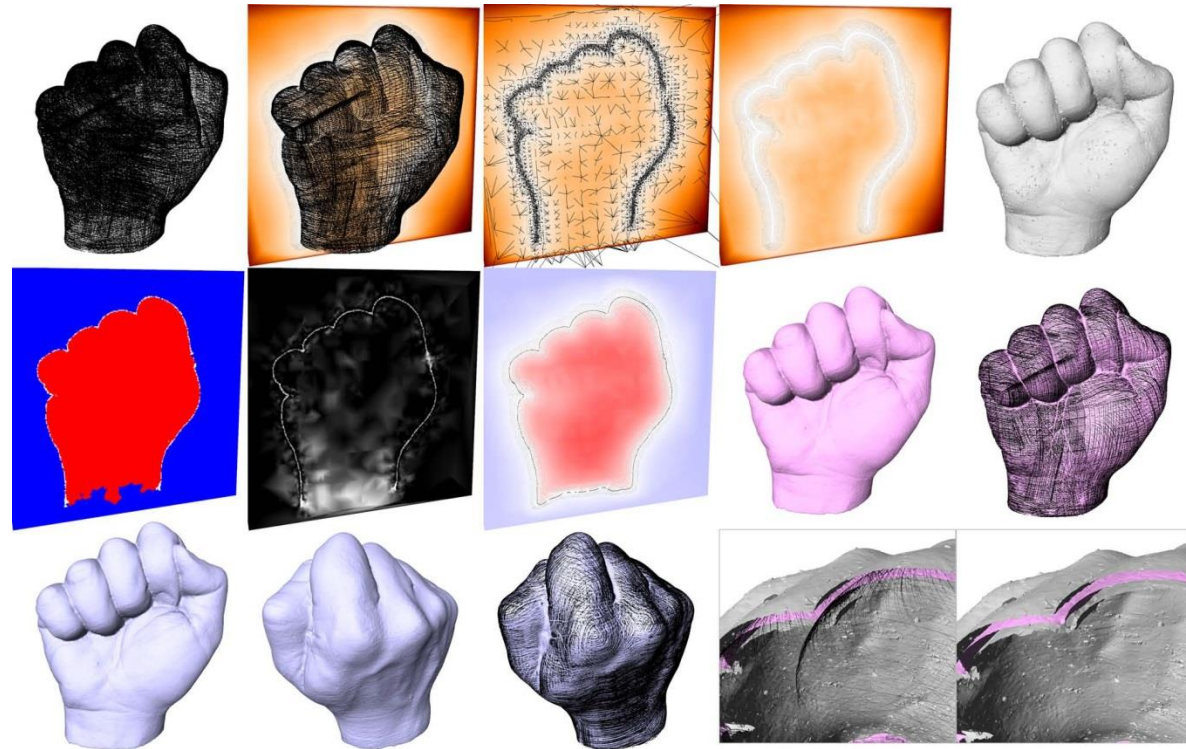


- Less memory
- Less efficient operations

- More memory
- More efficient operations



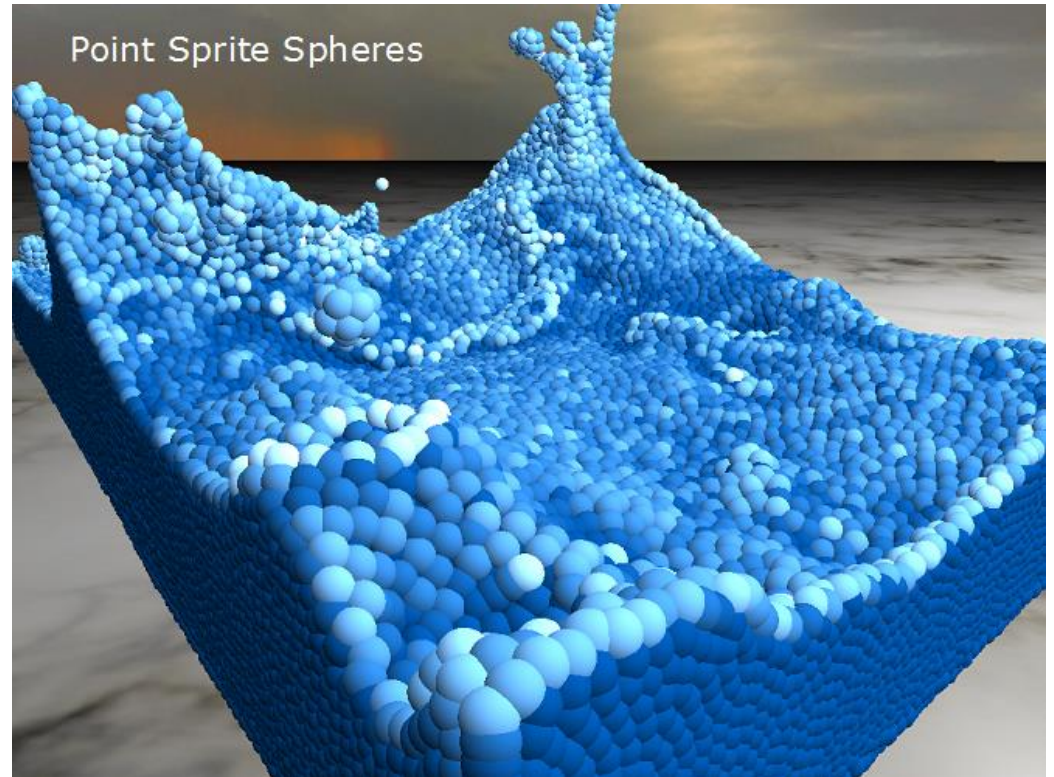
# Not the Only Model



[http://www.cs.umd.edu/class/spring2005/cmsc828v/papers/mpu\\_implicit.pdf](http://www.cs.umd.edu/class/spring2005/cmsc828v/papers/mpu_implicit.pdf) <ftp://ftp-sop.inria.fr/geometrica/alliez/signing.pdf>

## Implicit surfaces

# Not the Only Model



<http://www.itsartmag.com/features/cgfluids/>  
<https://developer.nvidia.com/content/fluid-simulation-alice-madness-returns>

# Smoothed-particle

# Not the Only Model

## AN IMPLICIT SURFACE TENSION MODEL

J. I. Hochstein\* and T. L. Williams\*\*

The University of Memphis  
Memphis, Tennessee

### ABSTRACT

A new implicit model for surface tension at a two-fluid interface is proposed for use in computational models of flows with free surfaces and its performance is compared to an existing explicit model. The new model is based on an evolution equation for surface curvature that includes the influence of advection as well as surface tension. A detailed development of the new model is presented as are the details of the computational implementation. The performance of the new model is compared to an existing explicit model by using both models to predict the surface dynamics of several two-dimensional configurations. It is concluded that the new implicit surface tension model does perform better for configurations with a large surface tension coefficient. It is shown that, for several cases, the time step size is no longer limited by surface tension stability considerations (as it was using the explicit model), but rather by other limitations inherent in the existing volume advection algorithm.

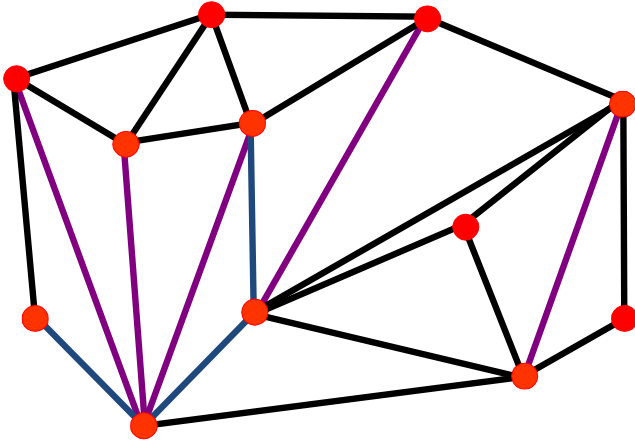
### INTRODUCTION

Incompressible flows with a free-surface exist in many industrial applications. Some examples include fuel atomization in internal combustion engines, droplet size control in ink-jet printers, formation of lead shot, control of liquid spacecraft propellant in low gravity, and the spinning of synthetic fibers. The technology for some of these applications has been developed by heavily relying on experimental study of the specific process involved. For others, such as spacecraft propellant management, experimental studies are prohibitively expensive and the ability to computationally model these process is essential for their development.

The modeling of flows with a free surface presents challenges unlike other types of flow problems in that a boundary condition must be applied at the free surface which is often in a transient state and irregularly shaped. This problem is exacerbated when the force due to surface tension

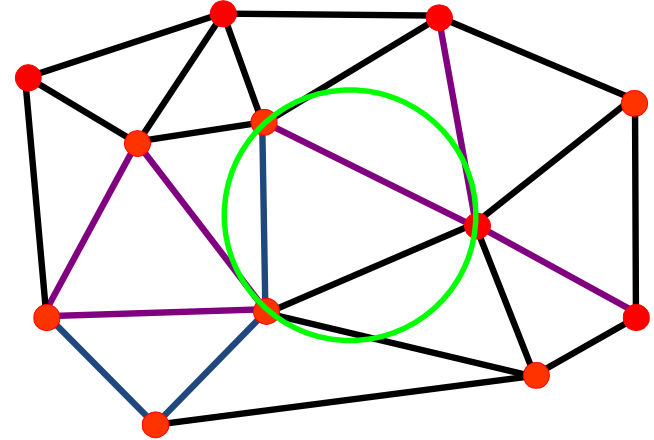
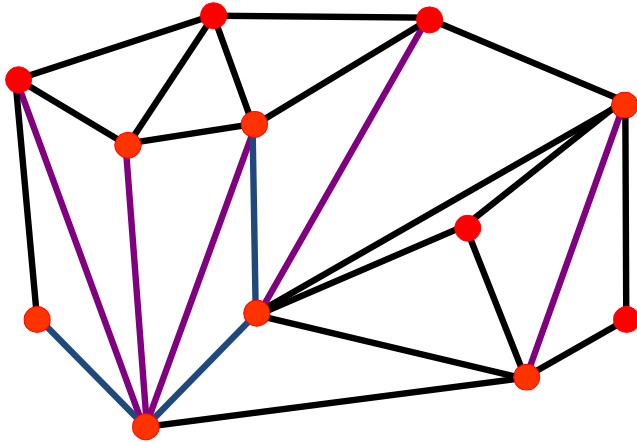


# TRIANGULATION



**Triangulation:** straight line plane graph with only triangle faces

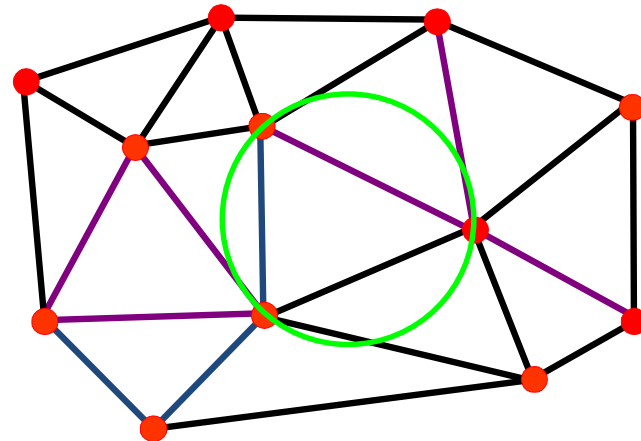
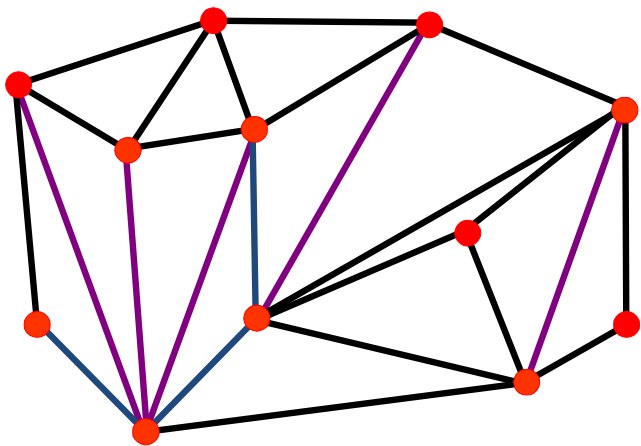
# TRIANGULATION



**Triangulation:** straight line plane graph with only triangle faces

**Delaunay triangulation** (DT) of a set of points:  
a triangulation s.t. the circumcircle of any triangle does not contain any other point.

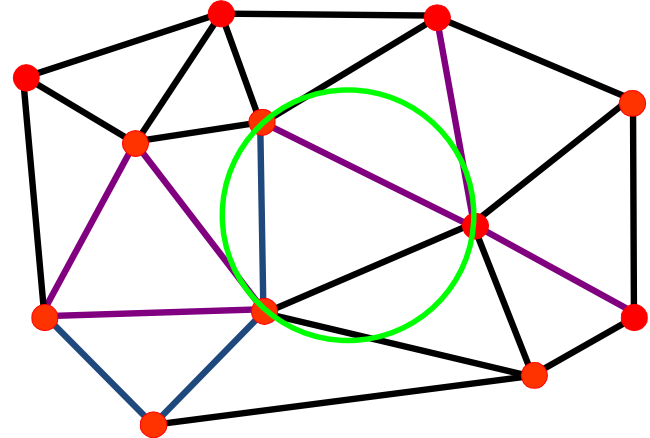
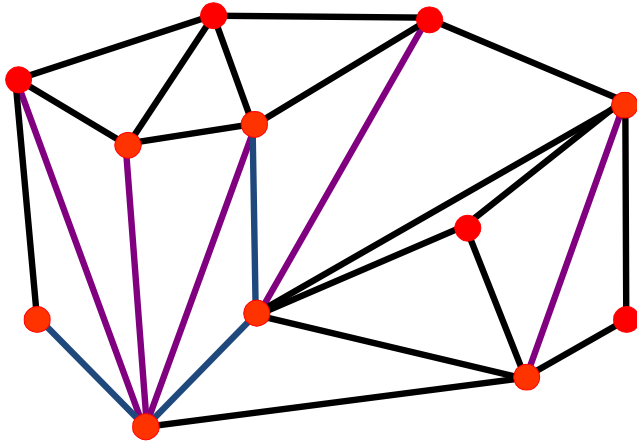
# FUN FACTS ABOUT DT



**Theorem.** *The exterior polygon of the DT is the convex hull.*

Also, DT is essentially the best triangulation you can get (produces no narrow triangles, if possible):

# FUN FACTS ABOUT DT



**Theorem.** *The exterior polygon of the DT is the convex hull.*

Also, DT is essentially the best triangulation you can get (produces no narrow triangles, if possible):

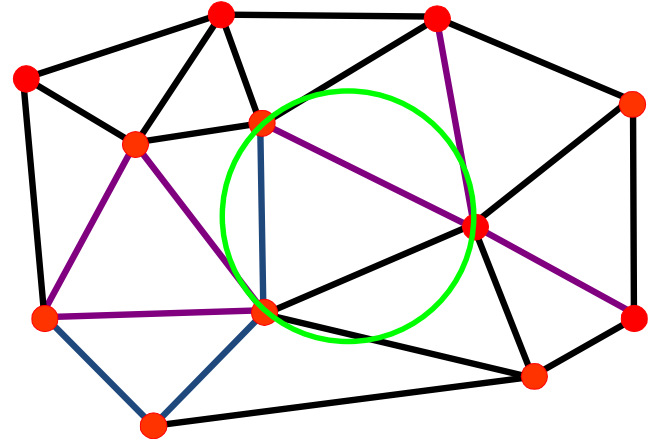
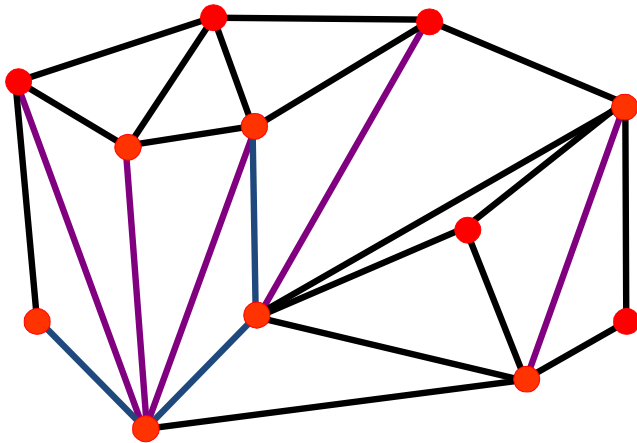
Set of points  $P = p_1, \dots, p_n \subset \mathbb{R}^2$

Triangulation  $T$  of  $P$

Define the angle vector of the triangulation,  $\alpha^T \in (0, \pi)^{3T}$ , to be the sorted angles of the triangles in the triangulation:  $\alpha_i^T \leq \alpha_{i+1}^T$ .

Define  $S > T$  if  $S$  is larger than  $T$ , lexicographically.

# FUN FACTS ABOUT DT



Also, DT is essentially the best triangulation you can get (produces no narrow triangles, if possible):

Set of points  $P = p_1, \dots, p_n \subset \mathbb{R}^2$

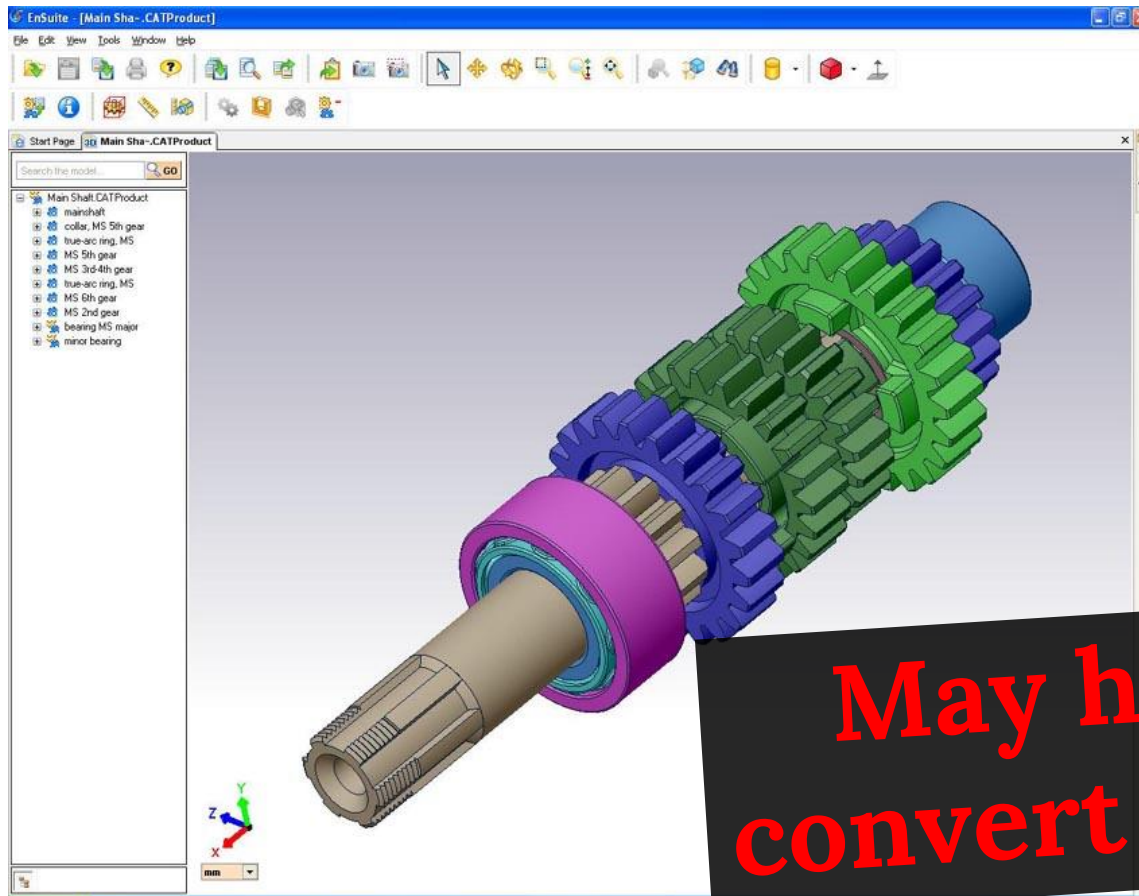
Triangulation  $T$  of  $P$

Define the angle vector of the triangulation,  $\alpha^T \in (0, \pi)^{3T}$ , to be the sorted angles of the triangles in the triangulation:  $\alpha_i^T \leq \alpha_{i+1}^T$ .

Define  $\mathcal{S} > \mathcal{T}$  if  $\mathcal{S}$  is larger than  $\mathcal{T}$ , lexicographically.

Then DT is the best possible triangulation! I.e.  $DT \geq T \forall T$

# Obtaining Geometry

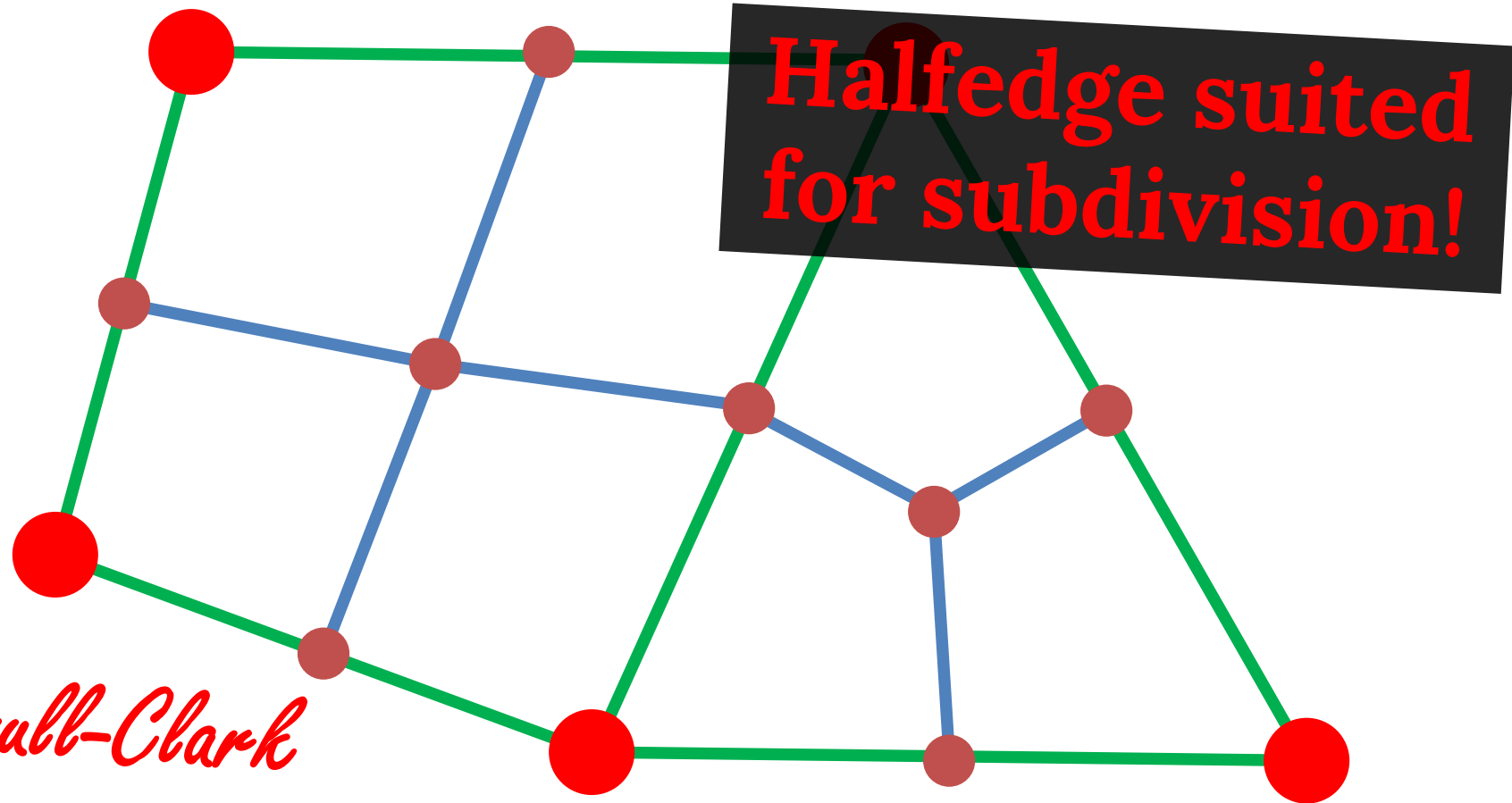


**May have to  
convert to mesh**

<http://www.cad-sourcing.com/wp-content/uploads/2011/12/free-cad-software.jpg>

**Cleanest: Design software**

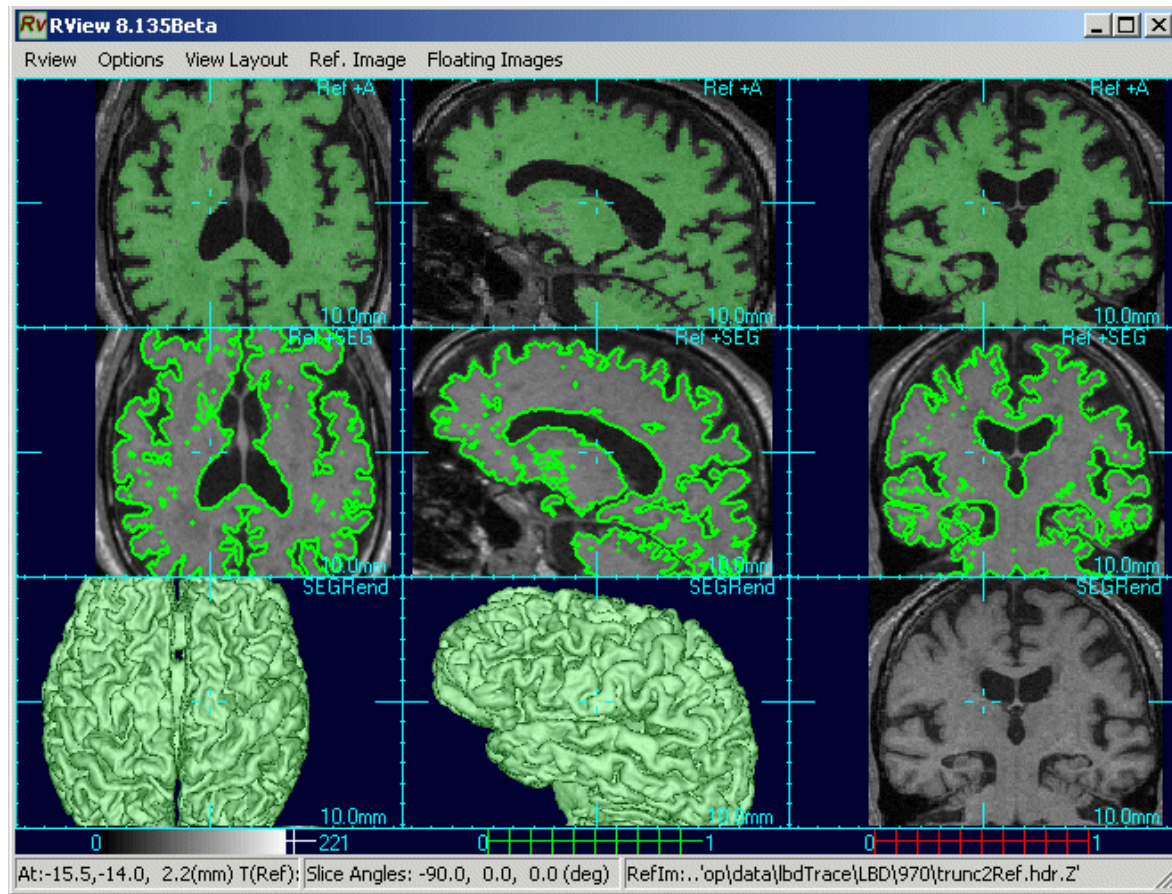
# Obtaining Geometry



*Catmull-Clark*

**Cleanest: Design software**

# Obtaining Geometry

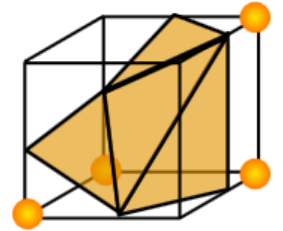
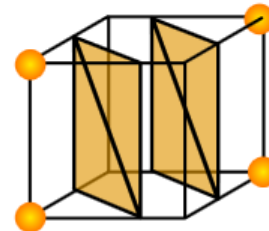
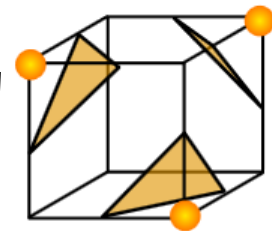
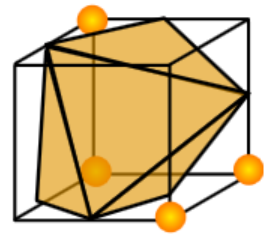
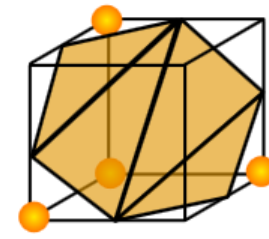
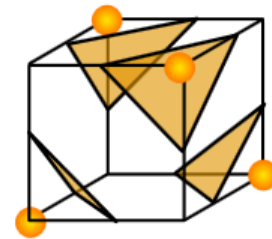
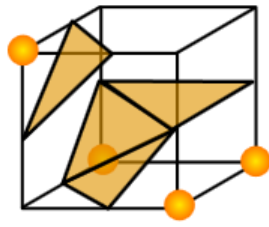
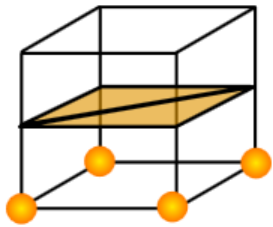
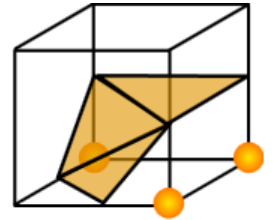
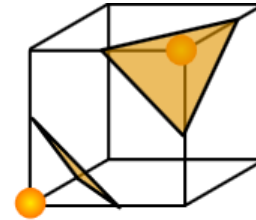
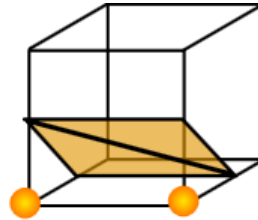
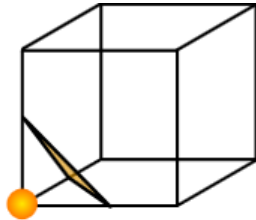
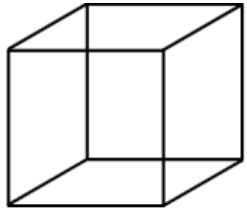


<http://www.colin-studholme.net/software/rview/rvmanual/morphtool5.gif>

# Volumetric extraction



# Obtaining Geometry

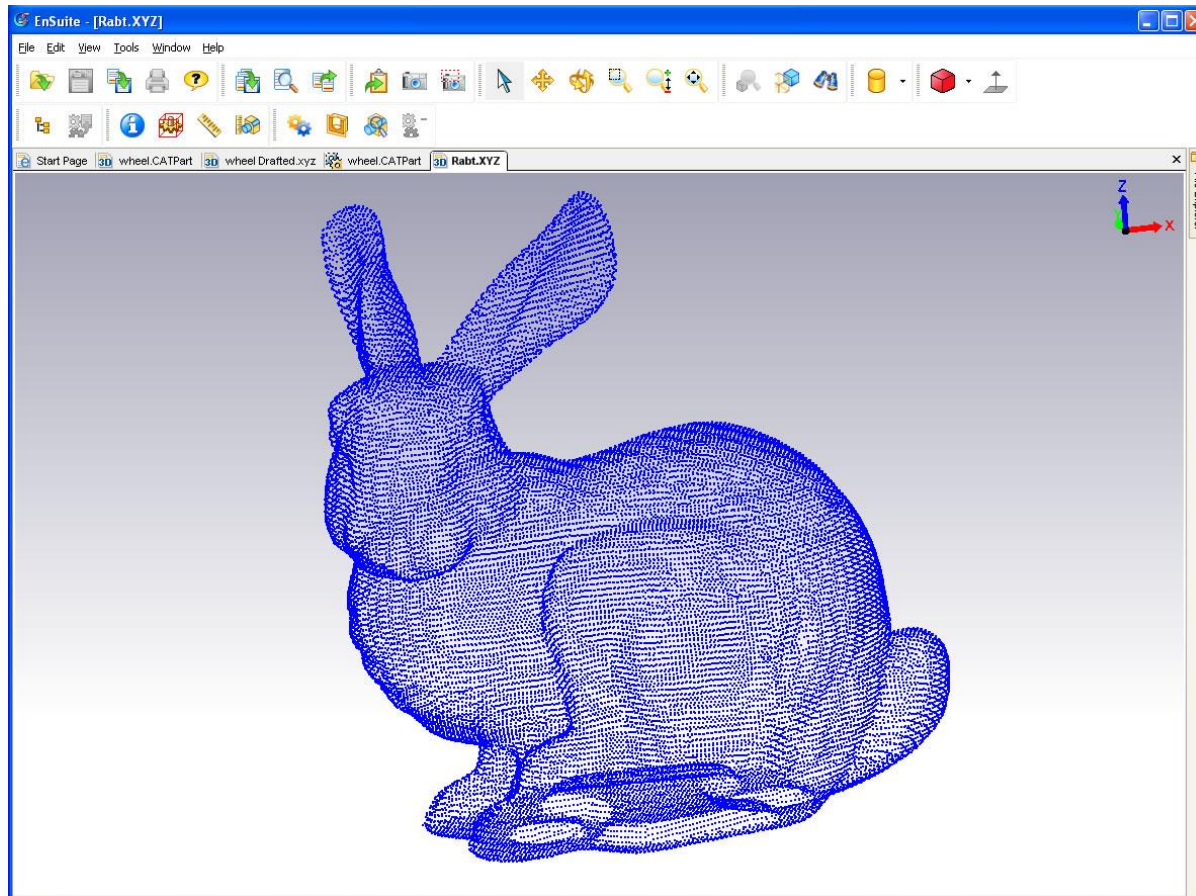


Marching cubes:  
Isosurface

extraction

Volumetric extraction

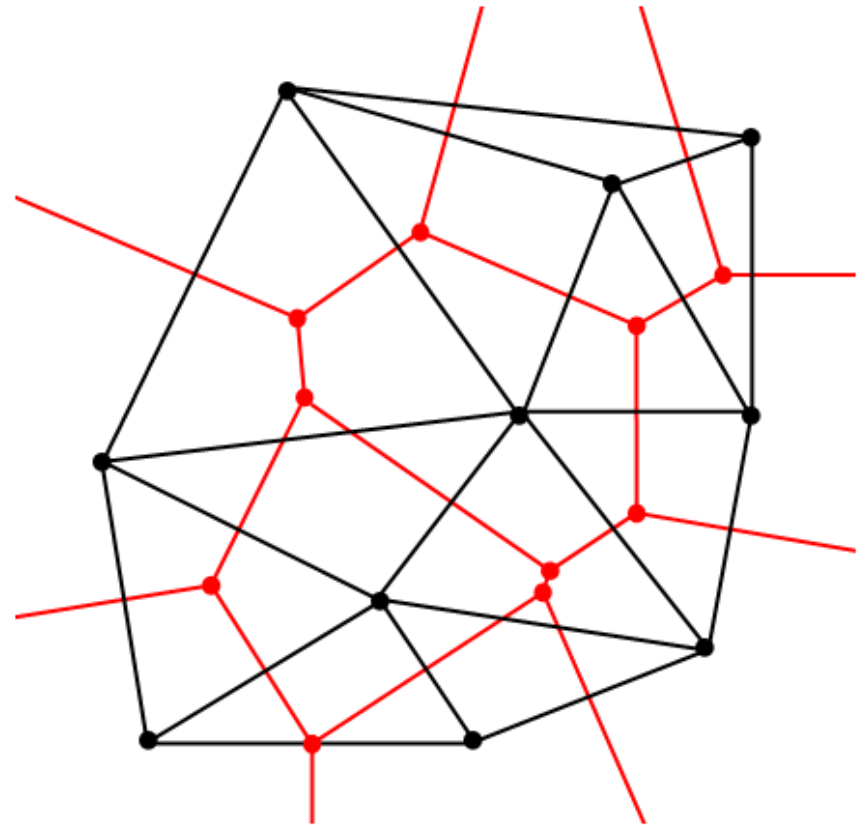
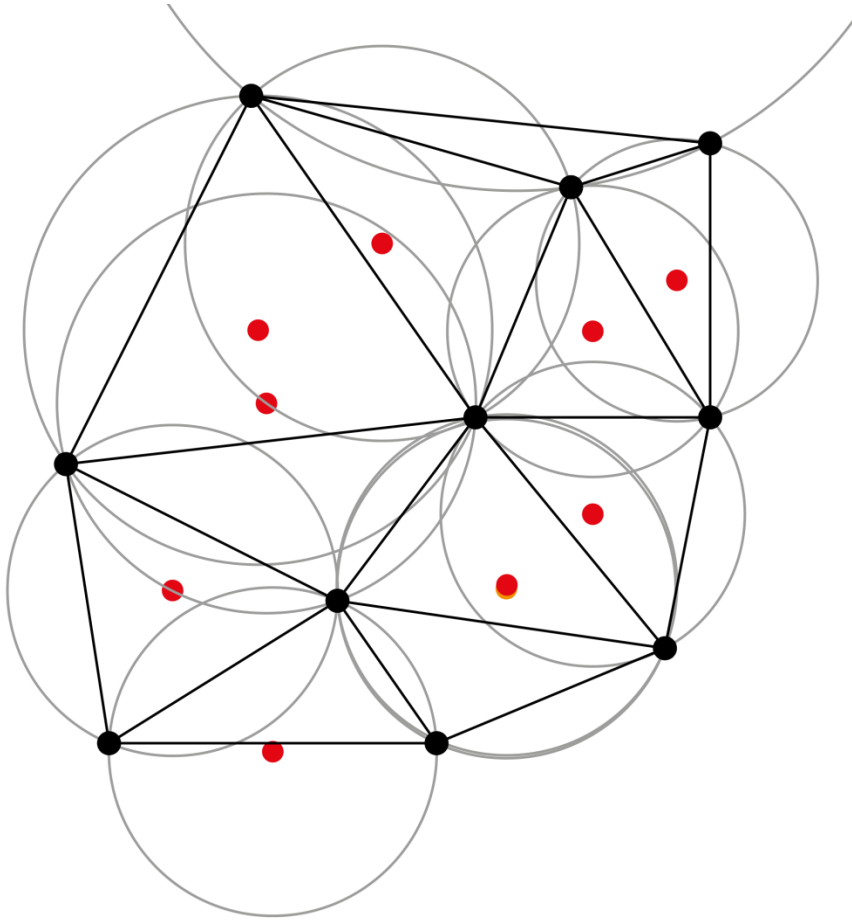
# Obtaining Geometry



<http://www.engineeringspecifier.com/public/primages/pr1200.jpg>

## Point clouds

# Delaunay Triangulation



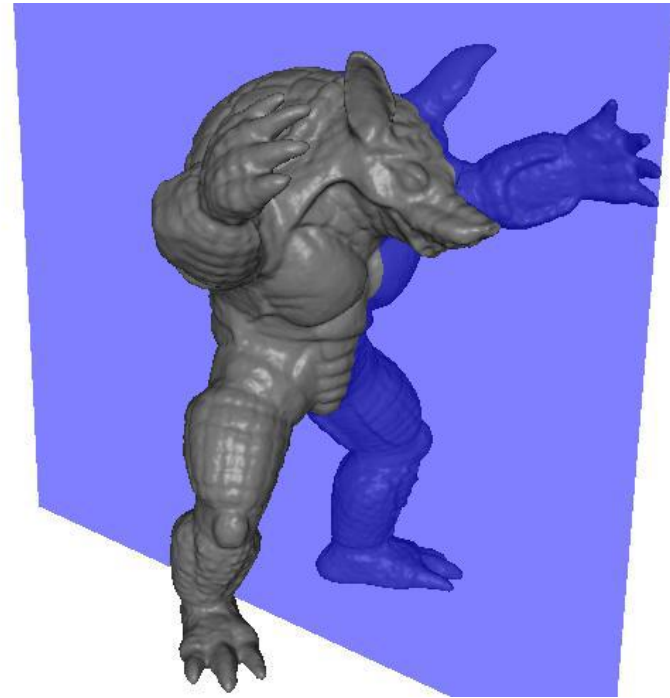
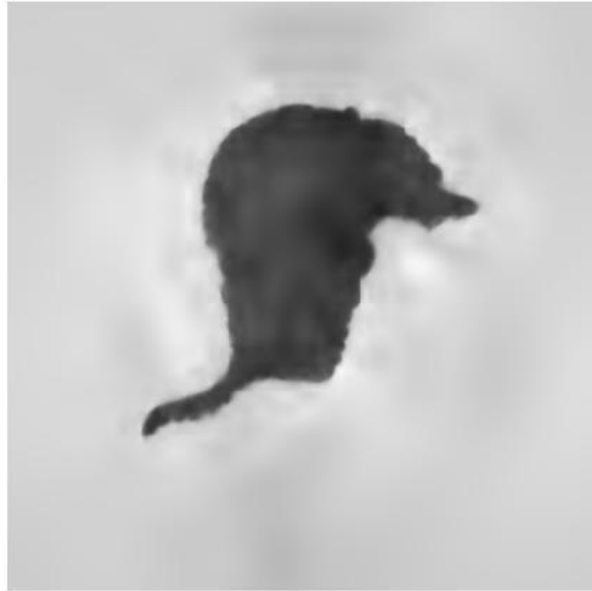
[http://en.wikipedia.org/wiki/Delaunay\\_triangulation](http://en.wikipedia.org/wiki/Delaunay_triangulation)

**Well-behaved dual mesh**

# Strategies for Surface Delaunay

- **Tangent plane**  
Derive local triangulation from tangent projection
- **Restricted Delaunay**  
Usual Delaunay strategy but in smaller part of  $R^3$
- **Inside/outside labeling**  
Find inside/outside labels for tetrahedra
- **Empty balls**  
Require existence of sphere around triangle with no other point

# Poisson Reconstruction



**Poisson Surface Reconstruction**  
**Kazhdan, Bolitho and Hoppe (SGP 2006)**