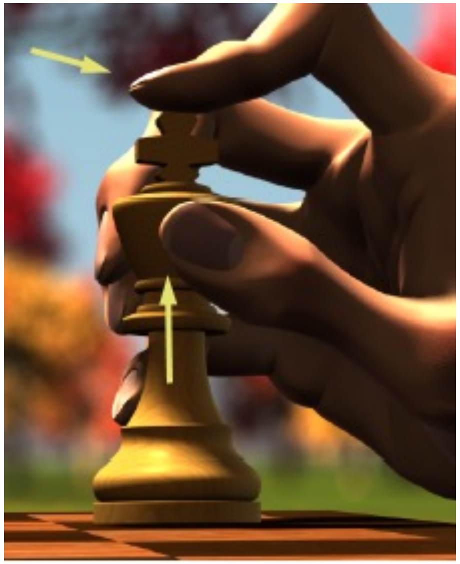
Computer Graphics

- Subdivision Surfaces -



Philipp Slusallek

Modeling

- **How do we ...**
  - Represent 3D objects in a computer?
  - Construct such representations quickly and/or automatically with a computer?
  - Manipulate 3D objects with a computer?
- **3D Representations provide the foundations for**
  - Computer Graphics
  - Computer-Aided Geometric Design
  - Visualization
  - Robotics, …
- **Different methods for different object representations**

3D Object Representations

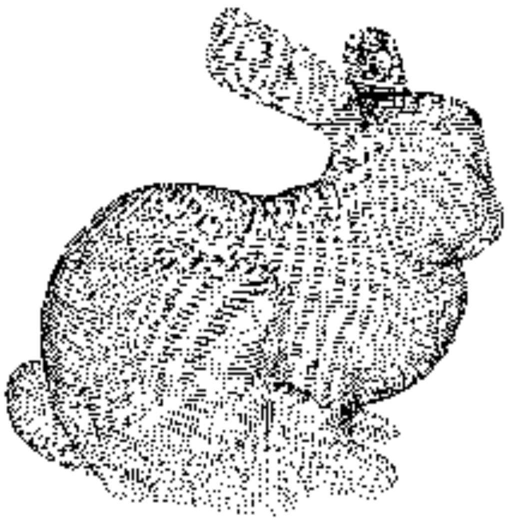
- **Raw data**
  - Range image
  - Point cloud
  - Polygon soup
- **Surfaces**
  - Mesh
  - Subdivision
  - Parametric
  - Implicit
- **Solids**
  - Voxels
  - BSP tree
  - CSG

Range Image

- **Range image**
  - Acquired from range scanner
    - E.g. laser range scanner, structured light, phase shift approach
  - Structured point cloud
    - Grid of depth values with calibrated camera
    - 2-1/2D: 2D plus depth

Point Cloud

- **Unstructured set of 3D point samples**
  - Often constructed from many range images
  - Or from direct image depth measurements
    - E.g., depth cameras (ToF/Time of Flight) or LIDAR sensors



Polygon Soup

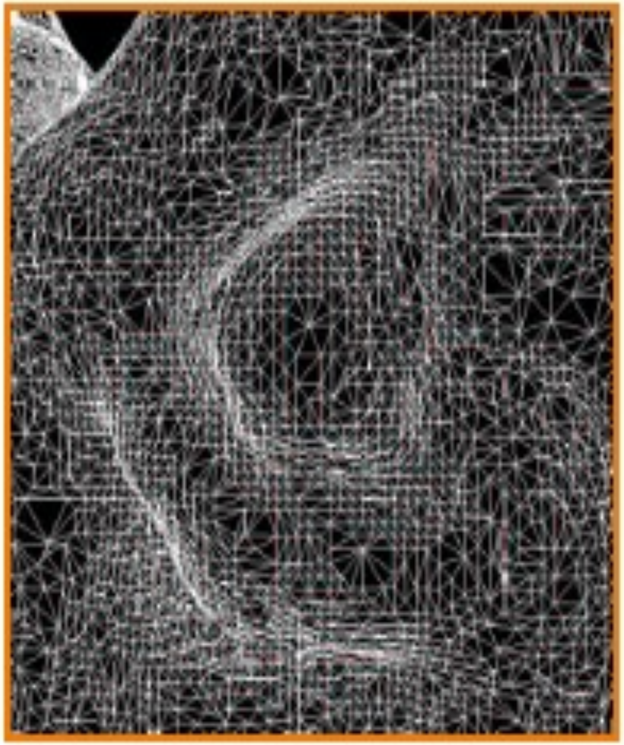
- **Unstructured set of polygons**

3D Object Representations

- **Raw data**
  - Point cloud
  - Range image
  - Polygon soup
- **Surfaces**
  - Mesh
  - Subdivision
  - Parametric
  - Implicit
- **Solids**
  - Voxels
  - BSP tree
  - CSG

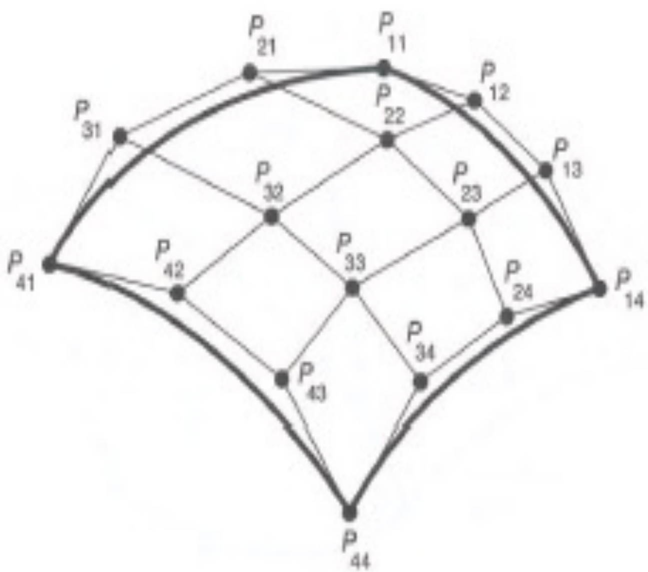
Mesh

- **Connected set of polygons (usually triangles)**
  - **Often arranged in some higher-level structures (strips, fans, meshes, …)**



Parametric Surface

- **Tensor product spline patches**
  - Careful constraints to maintain continuity



FvDFH Figure 11.44

Implicit Surface

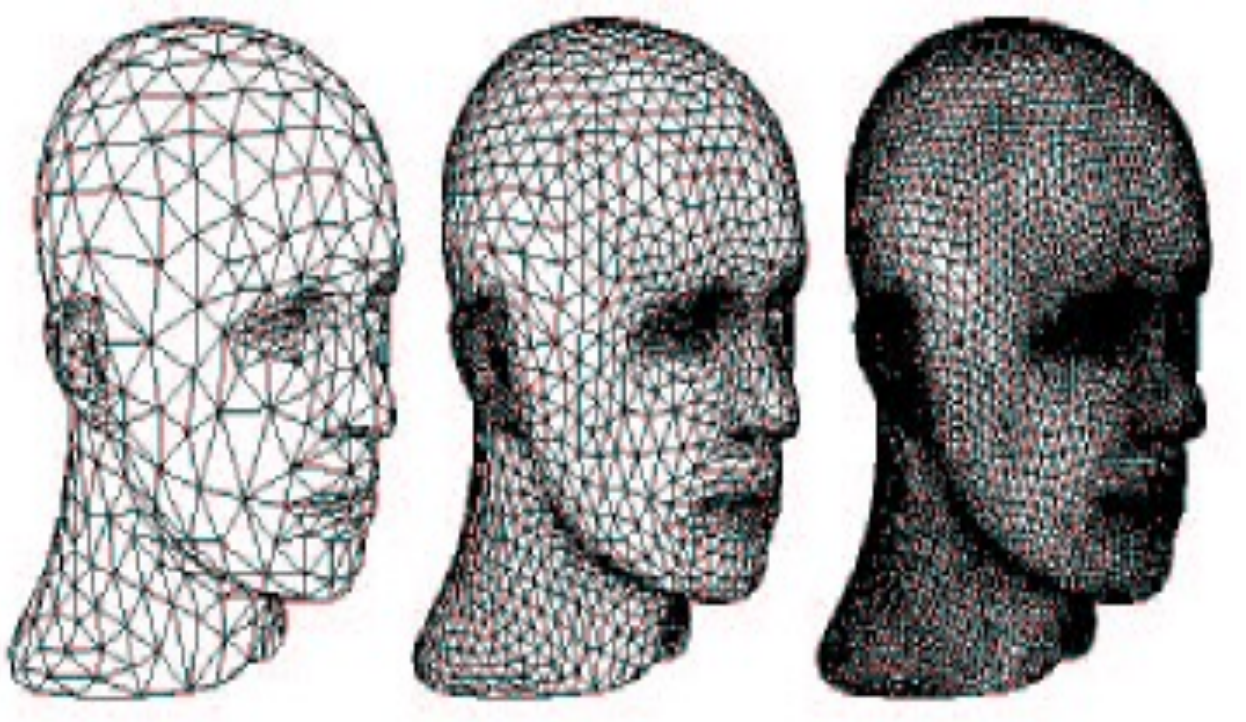
- **Points satisfying: F(x,y,z) = 0**

Polygonal Model

Implicit Model

Subdivision Surface

- **Coarse mesh & subdivision rule**
  - Define smooth surface as limit of sequence of refinements

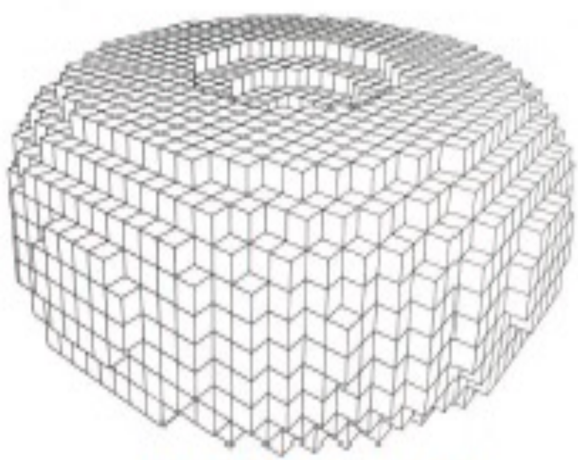


3D Object Representations

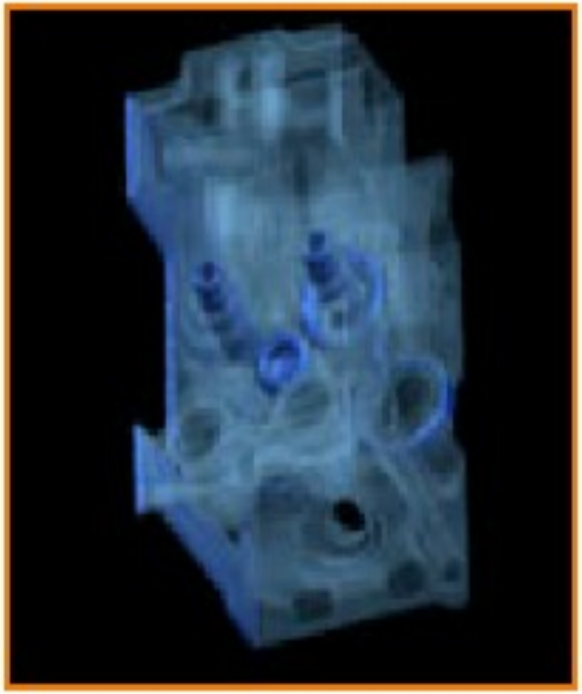
- **Raw data**
  - Point cloud
  - Range image
  - Polygon soup
- **Surfaces**
  - Mesh
  - Subdivision
  - Parametric
  - Implicit
- **Solids**
  - Voxels
  - BSP tree
  - CSG

Voxels

- **Uniform grid of volumetric samples**
  - Acquired from CAT, MRI, etc.



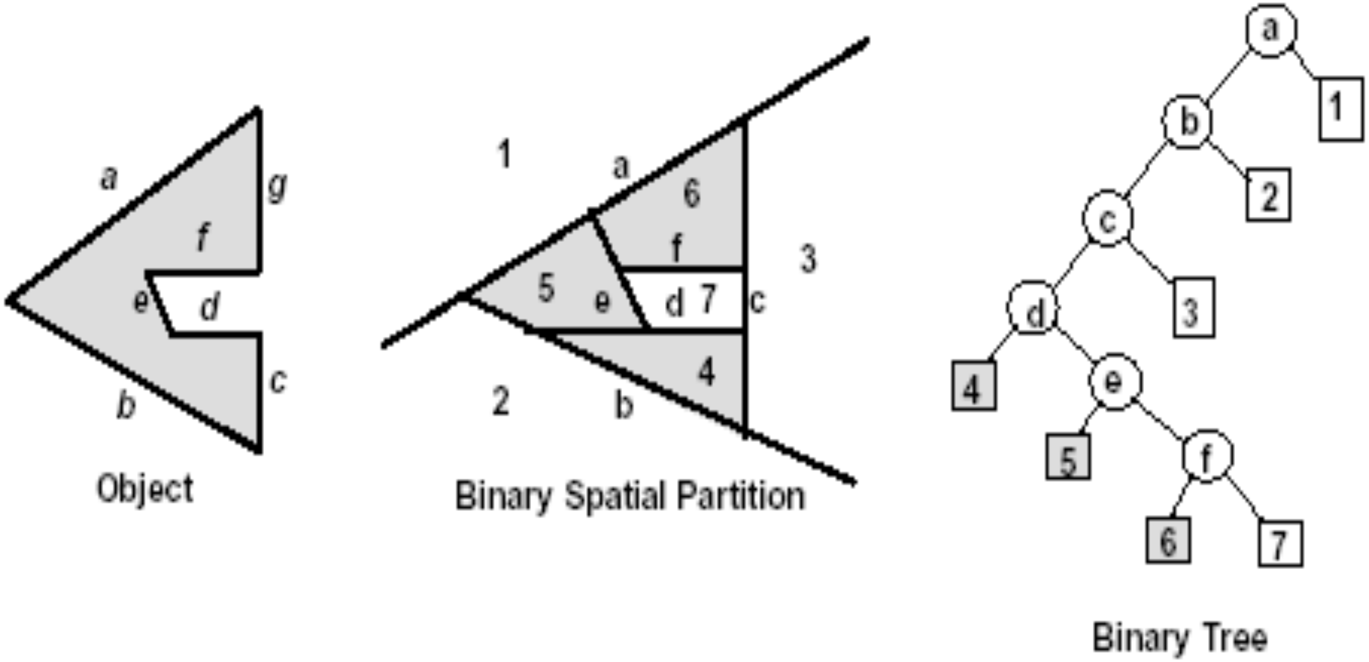
FvDFH Figure 12.20



Stanford Graphics Laboratory

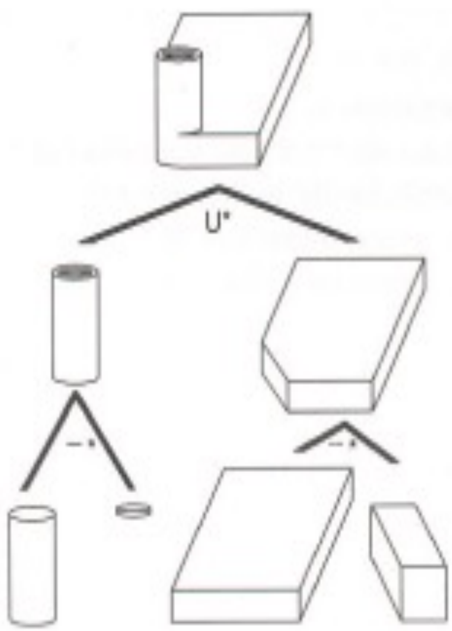
BSP Tree

- **Binary space partition with solid cells labeled**
  - Constructed from polygonal representations

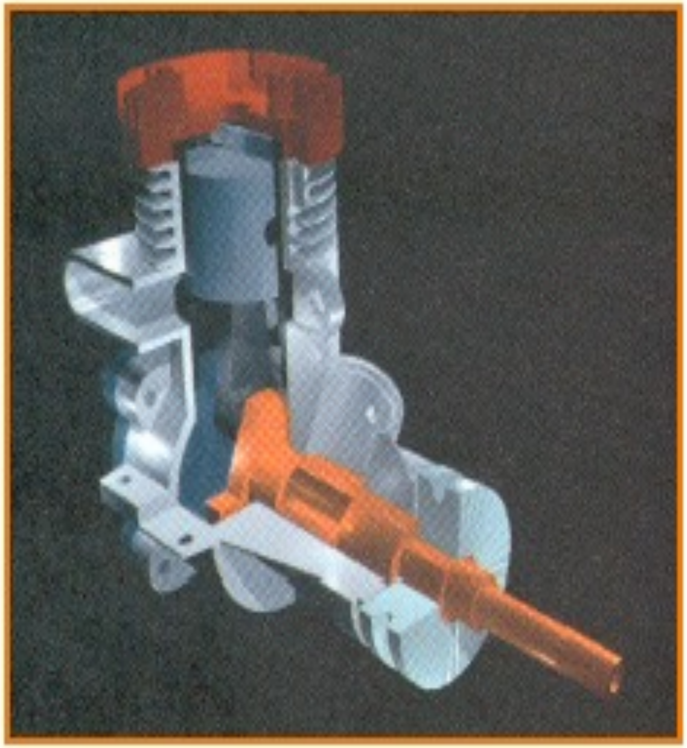


CSG

- **Hierarchy of boolean set operations (union, difference, intersect) applied to simple shapes**



FvDFH Figure 12.27



H&B Figure 9.9

Motivation

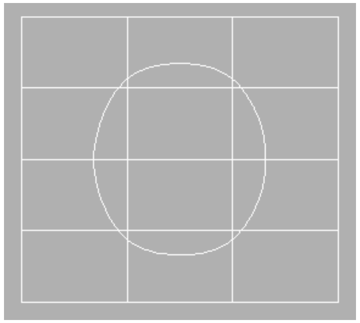
- **Splines**
  - Traditionally spline patches (NURBS) have been used in production for character animation.
- **Difficult to stitch together**
  - Maintaining continuity is hard
- **Difficult to model objects with complex topology**

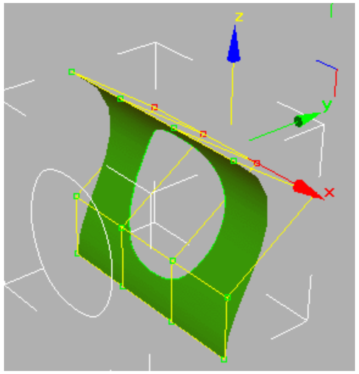
Subdivision in Character Animation
Tony Derose, Michael Kass, Tien Troung
(SIGGRAPH '98)

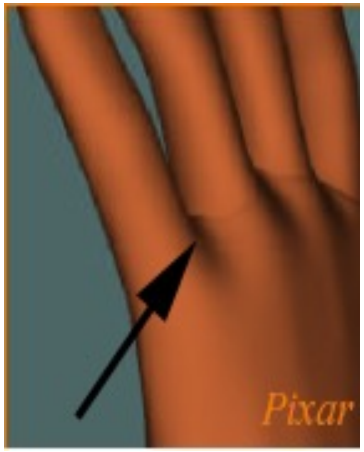
(Geri's Game, Pixar 1998)

Motivation

- **Splines (Bézier, NURBS, …)**
  - Easy and commonly used in CAD systems
  - Most surfaces are not made of quadrilateral patches
    - Need to trim surface: Cut off parts
  - Trimming NURBS is expensive and often has numerical errors
  - Difficult to stich together separate surfaces
  - Hard to hide seams

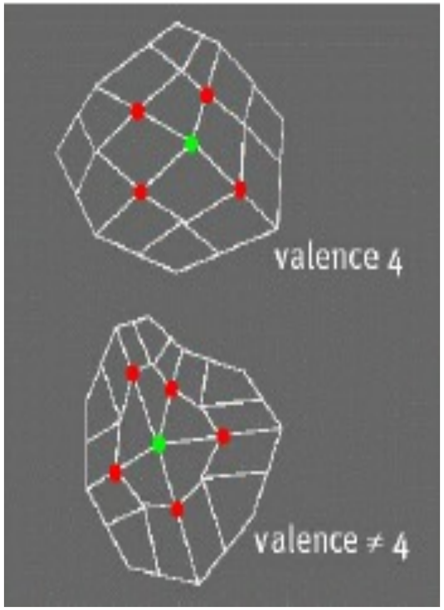


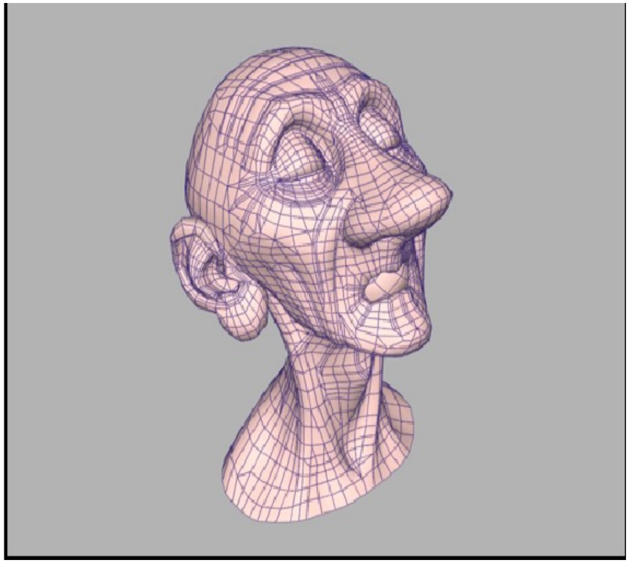




Why Subdivision Surfaces?

- **Subdivision methods have a series of interesting properties:**
  - Applicable to meshes of arbitrary topology (non-manifold meshes).
  - No trimming needed
  - Scalability, level-of-detail
  - Numerical stability
  - Fairly simple implementation
  - Compact support
  - Affine invariance
  - Automatic continuity (possibly with some isolated singular points)
  - Still somewhat less well supported by CAD tools





Types of Subdivision

- **Interpolating Schemes**
  - Limit Surfaces/Curve will pass through original set of data points.
- **Approximating Schemes**
  - Limit Surface will not necessarily pass through the original set of data points.

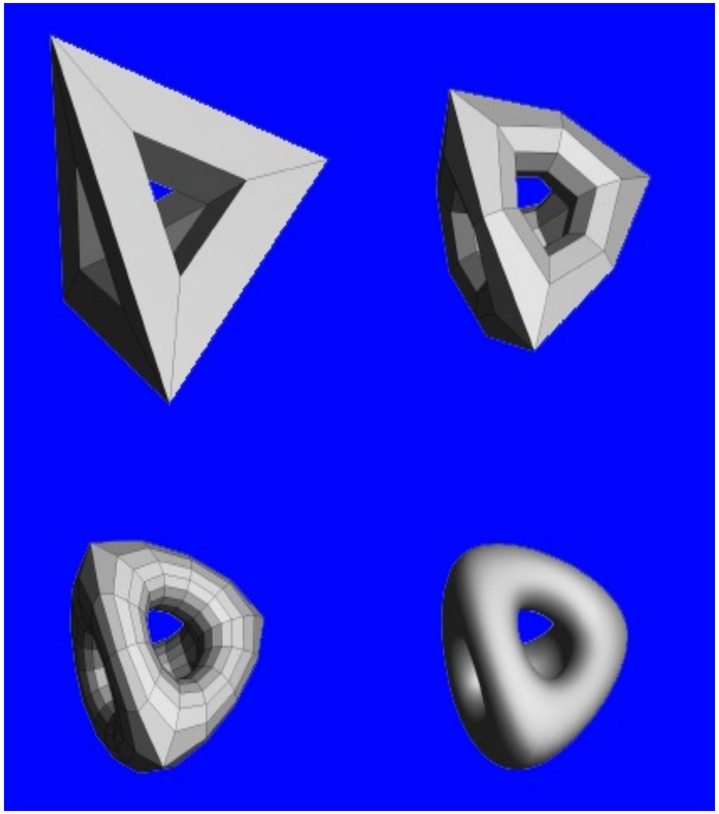
Example: Geri's Game

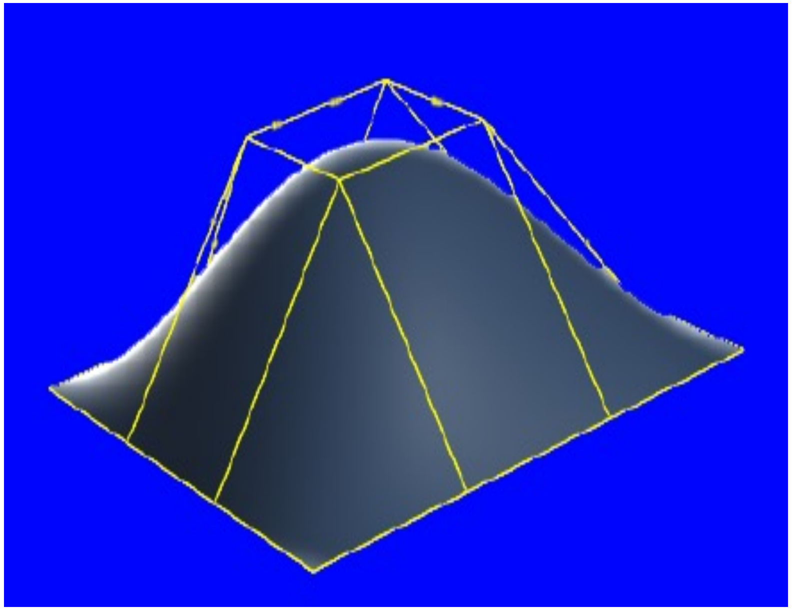
- **Subdivision surfaces are used for:**
  - Geri's hands and head
  - Clothes: Jacket, Pants, Shirt
  - Tie and Shoes

(Geri's Game, Pixar 1998)

Subdivision

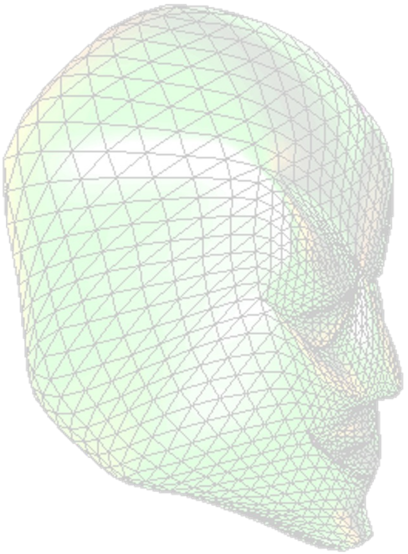
- **Construct a surface from an arbitrary polyhedron**
  - Subdivide each face of the polyhedron – and recurse
- **The limit will be a smooth surface**
  - Given the right subdivision rules are used





Subdivision Curves and Surfaces

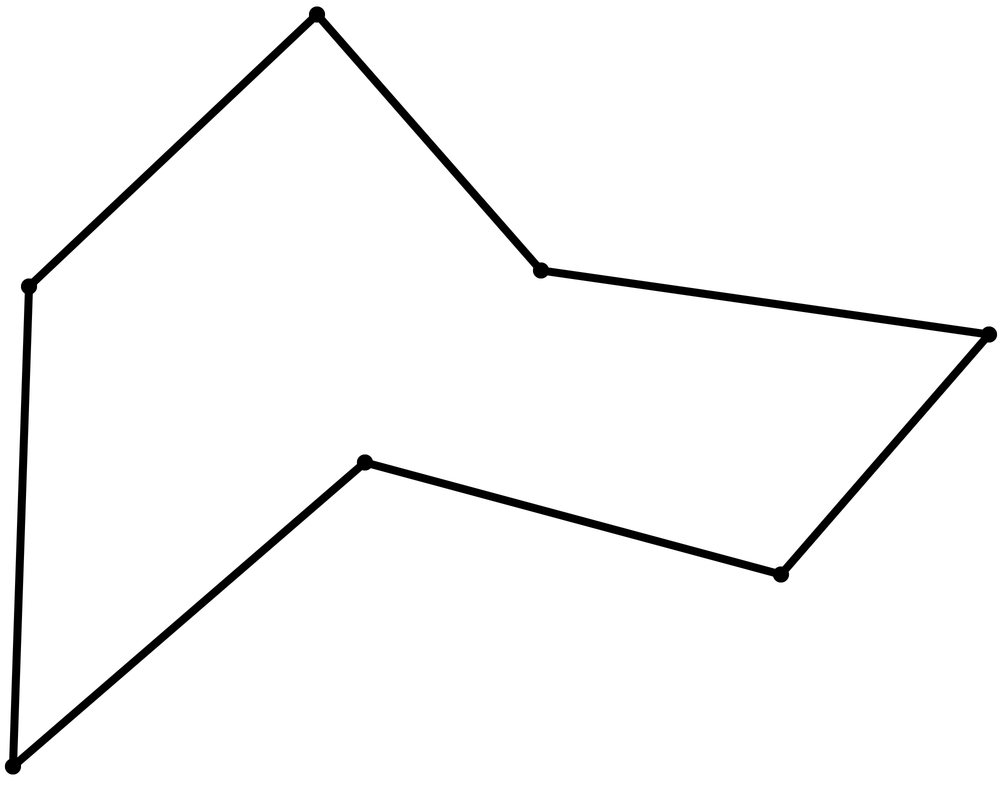
- **Subdivision curves**
  - The basic concepts of subdivision.
- **Subdivision surfaces**
  - Important known methods.
  - Discussion: subdivision vs. parametric surfaces .



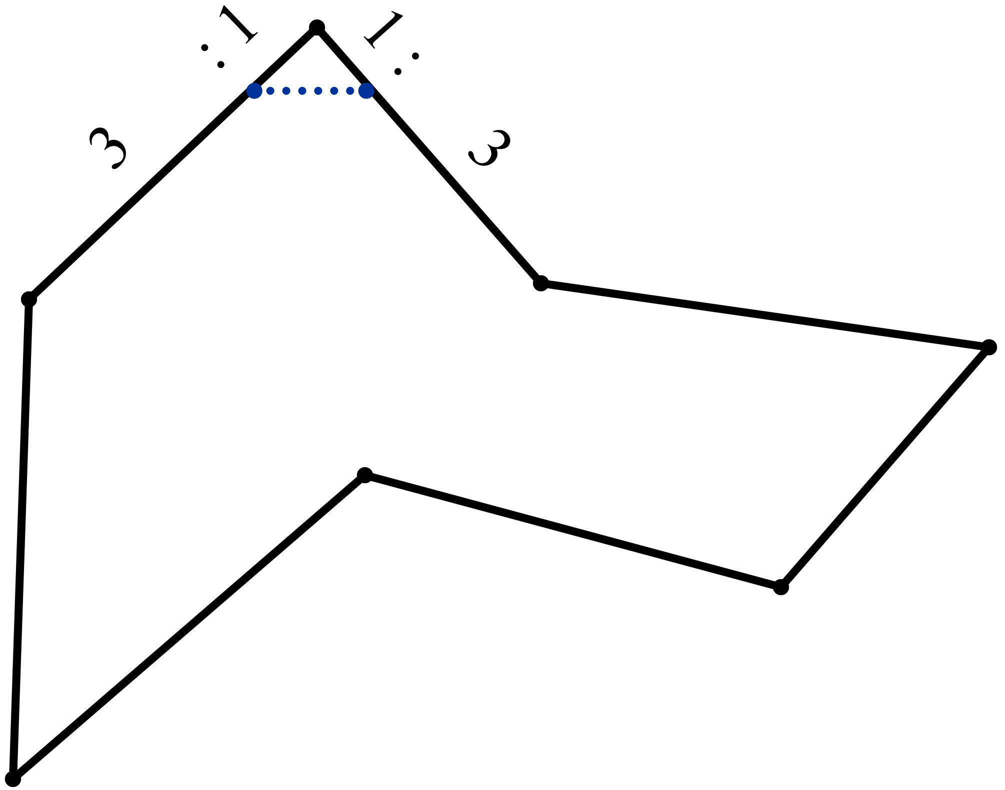
Based on slides Courtesy of Adi Levin, Tel-Aviv U.

Curves: Corner Cutting

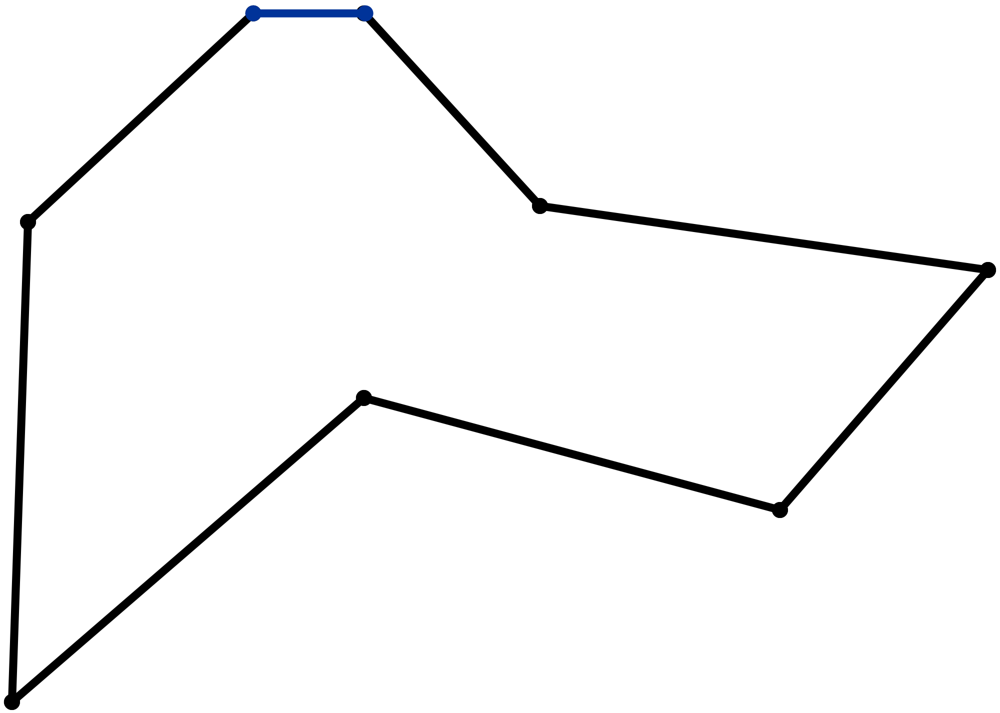
[George Chaikin, 1974]



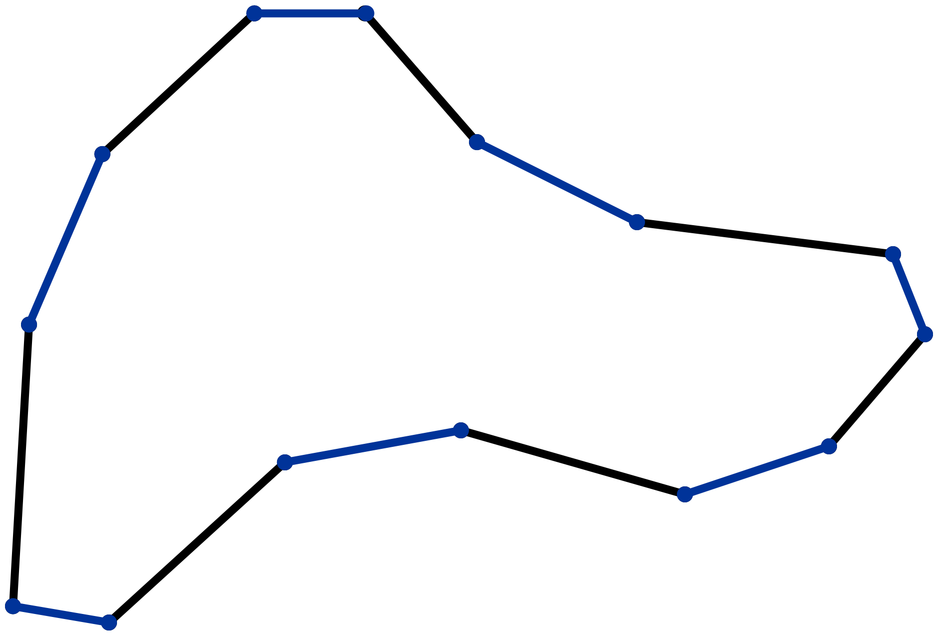
Corner Cutting



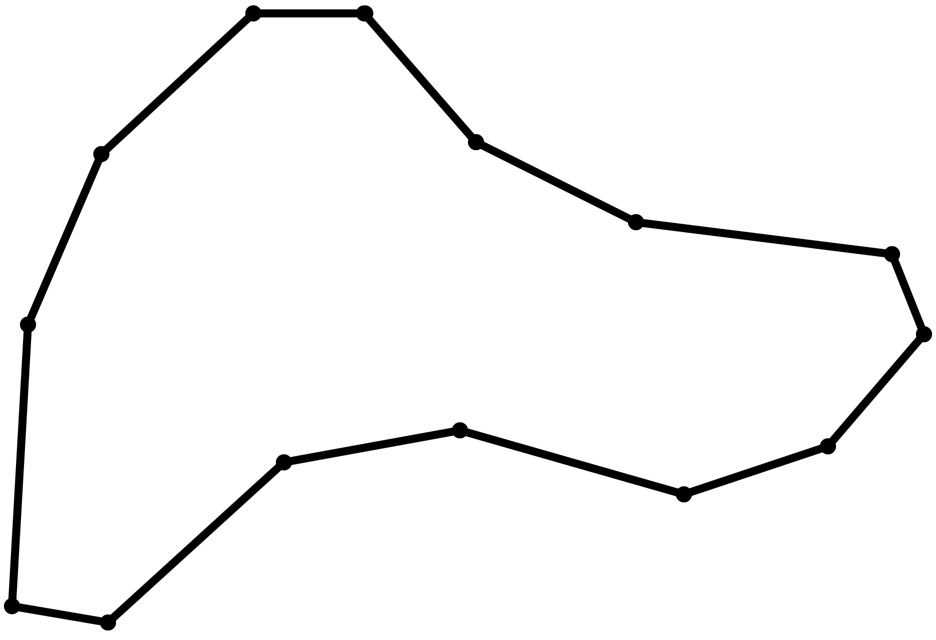
Corner Cutting



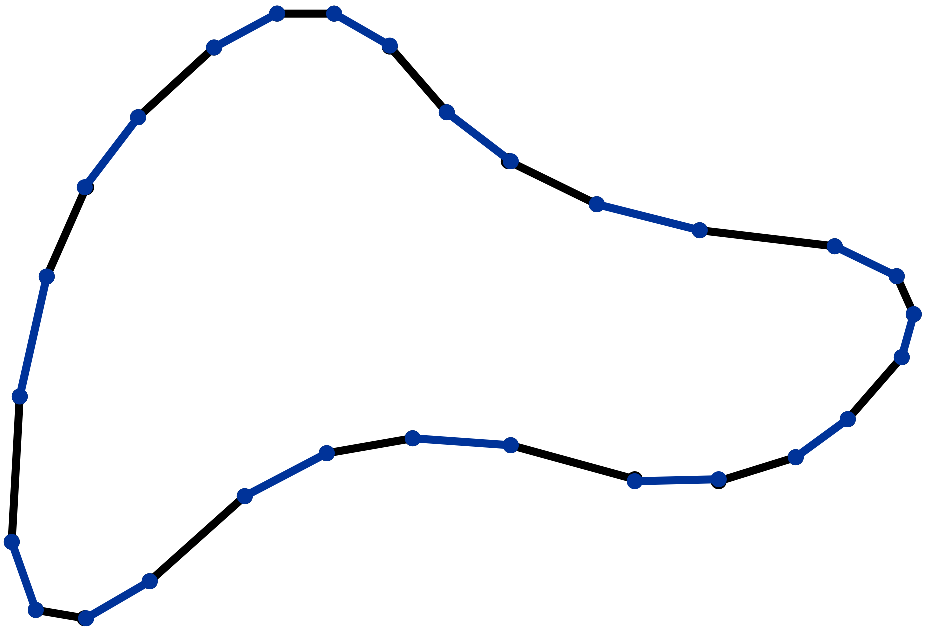
Corner Cutting



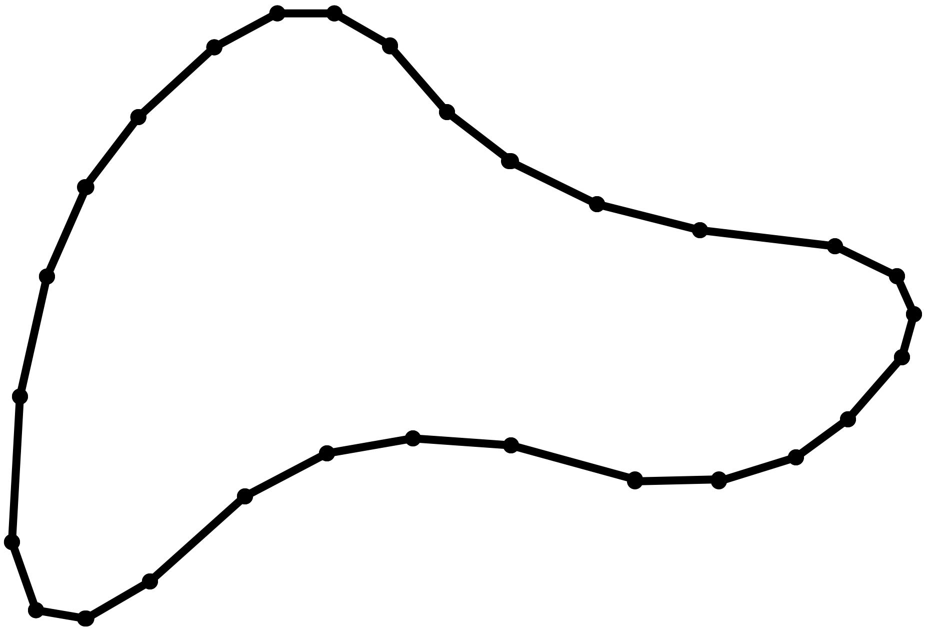
Corner Cutting



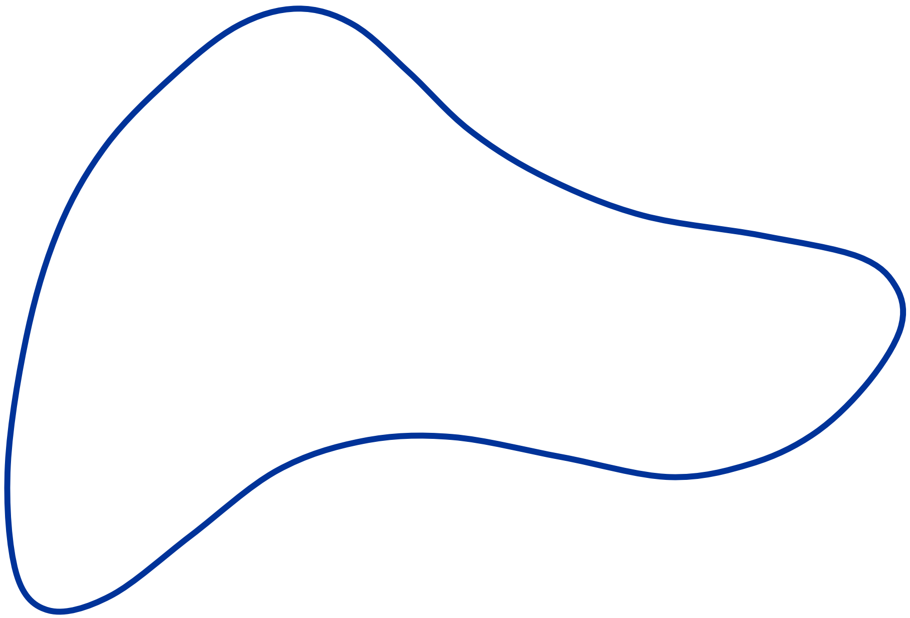
Corner Cutting



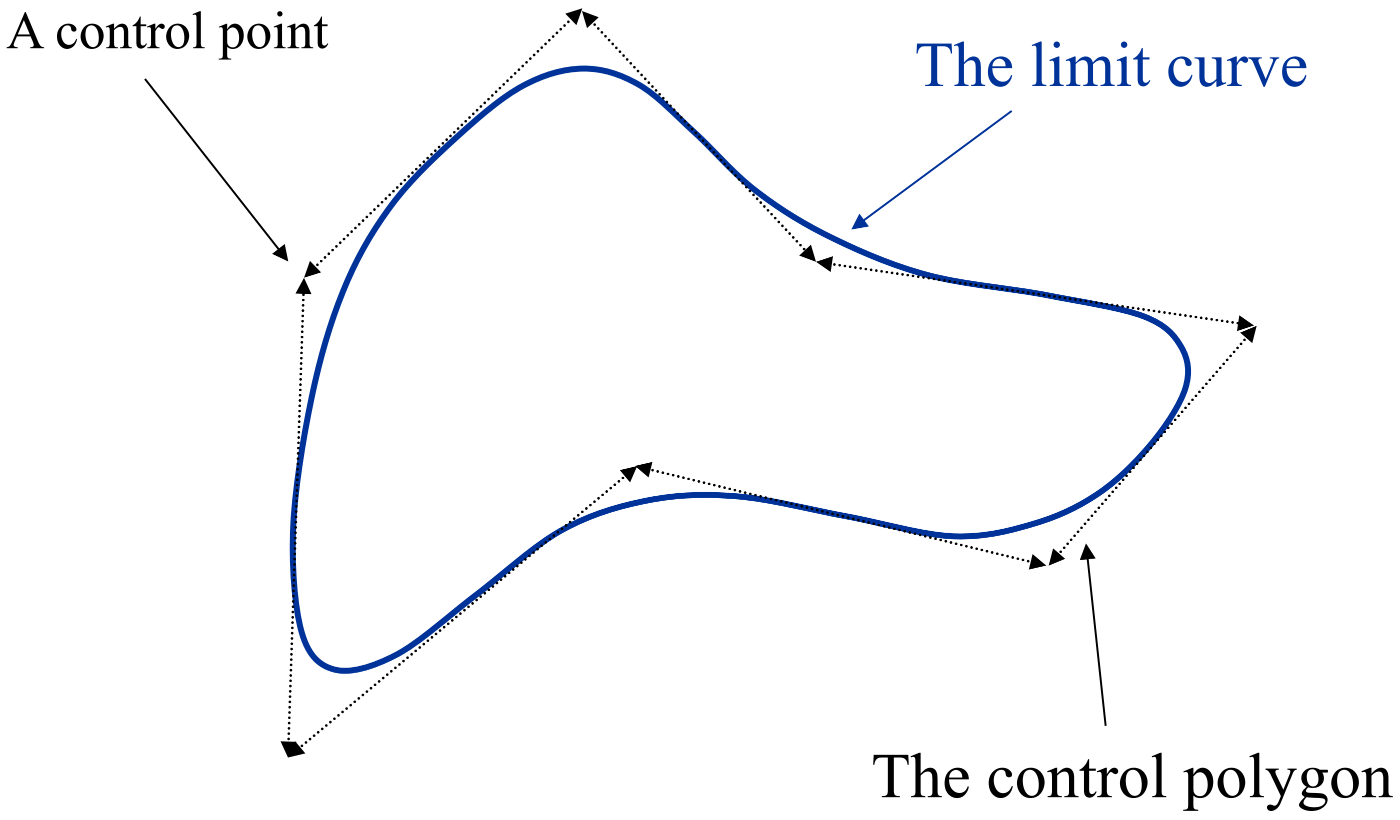
Corner Cutting



Corner Cutting

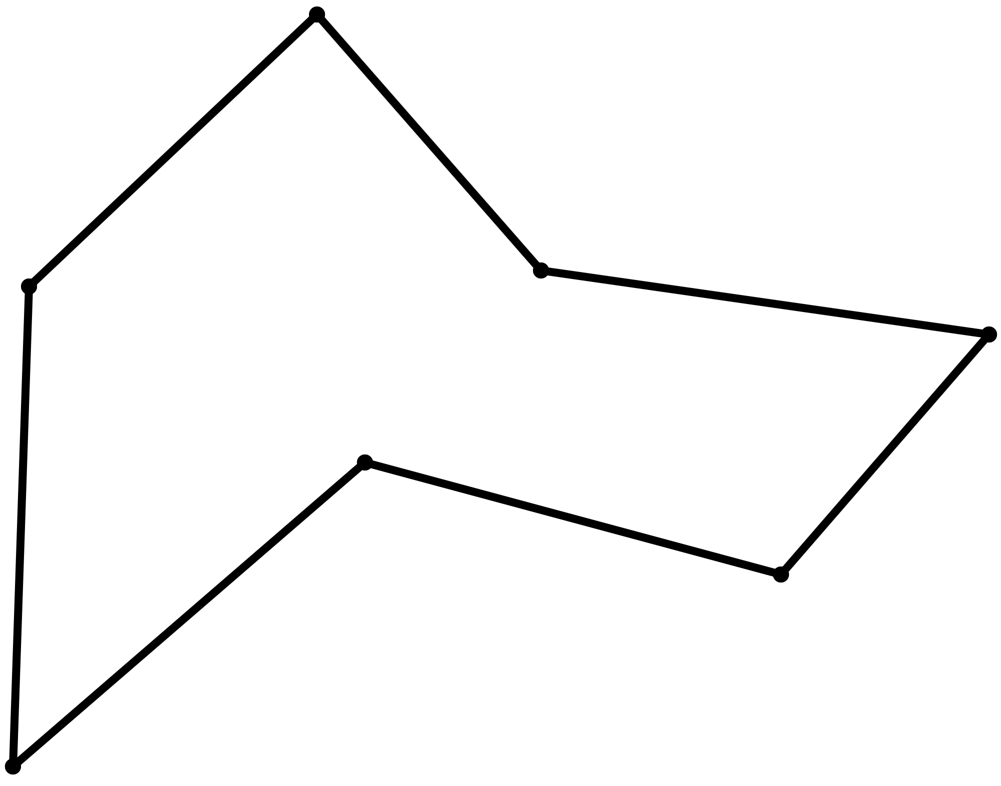


Corner Cutting

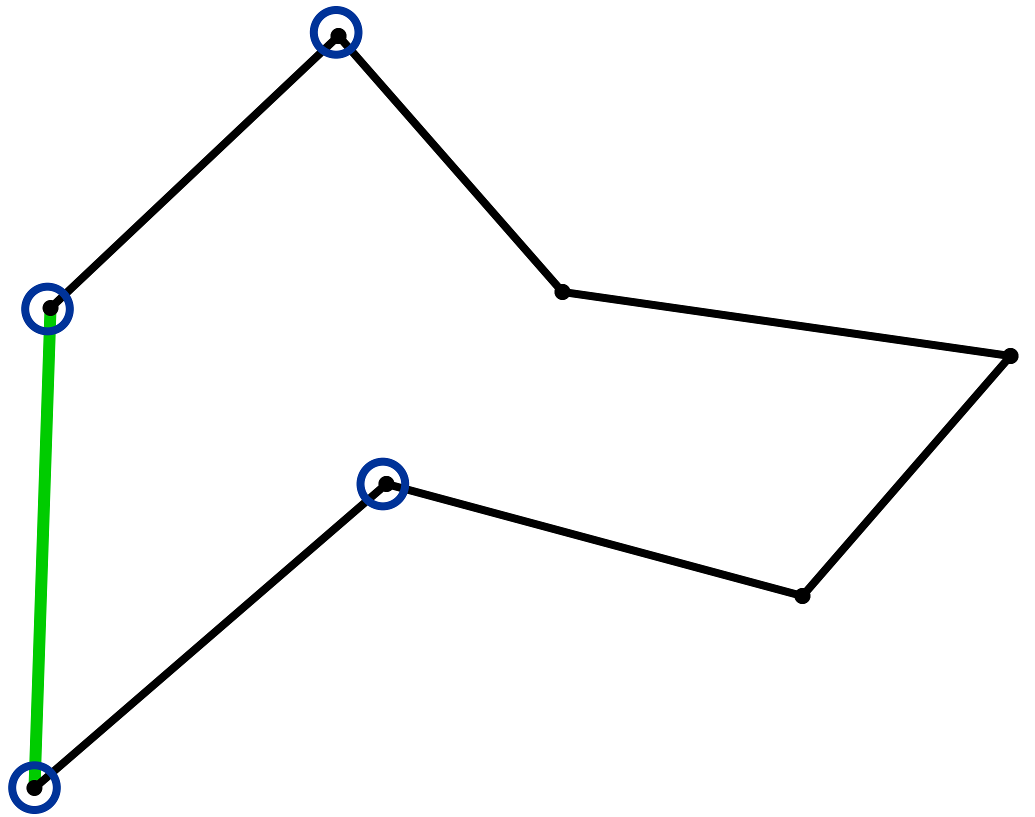


The 4-Point Scheme

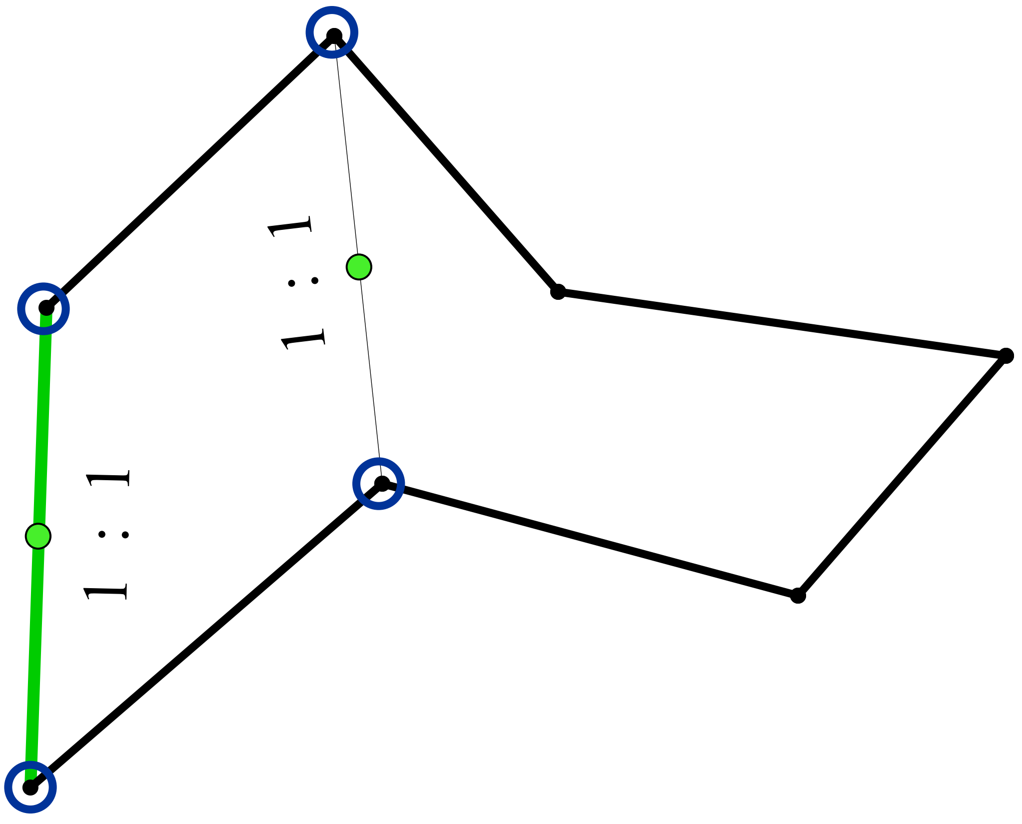
[Dyn, Levin, Gregory, 1987]



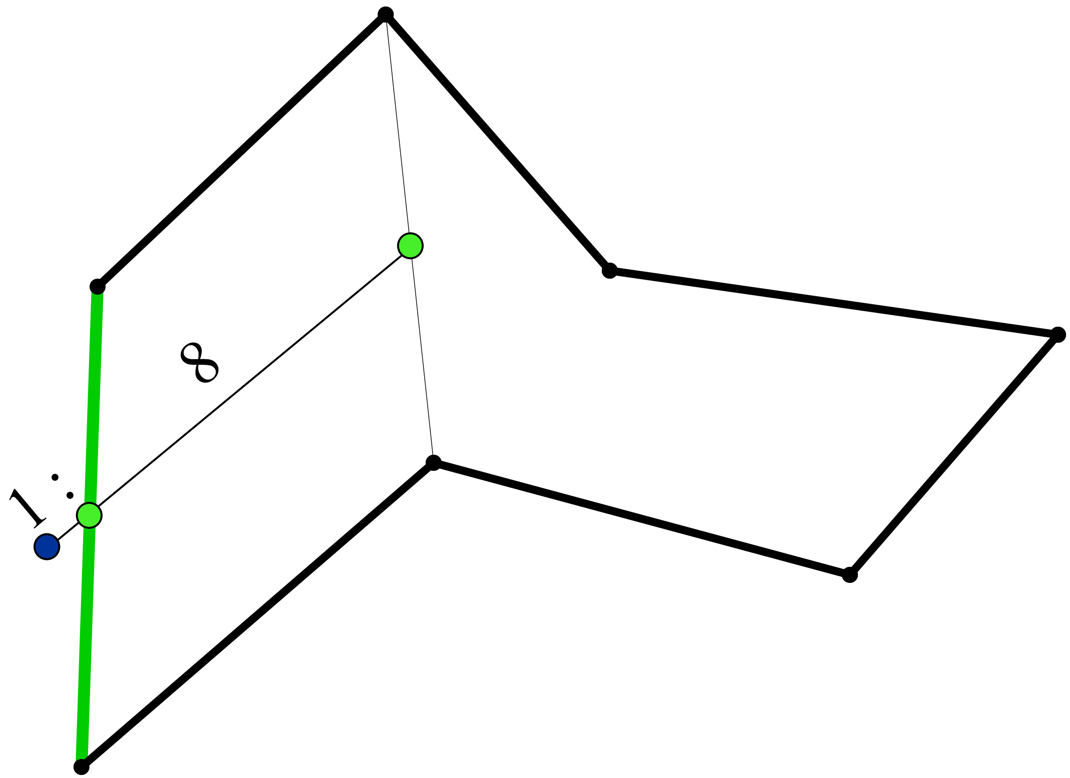
The 4-Point Scheme



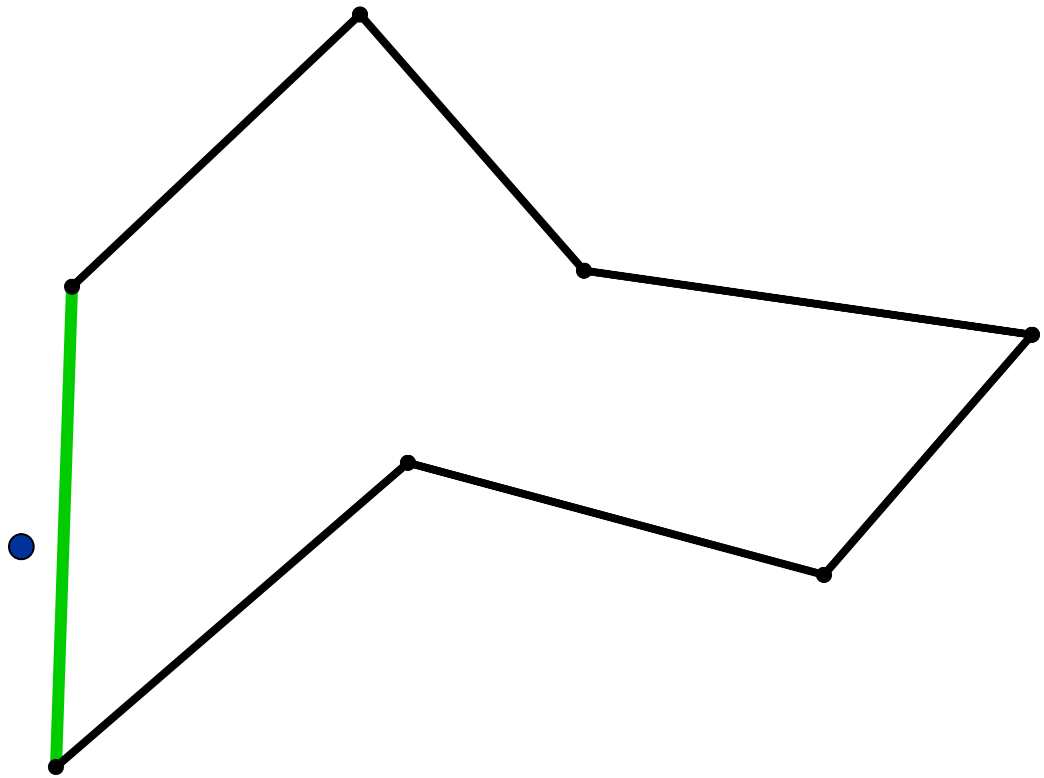
The 4-Point Scheme



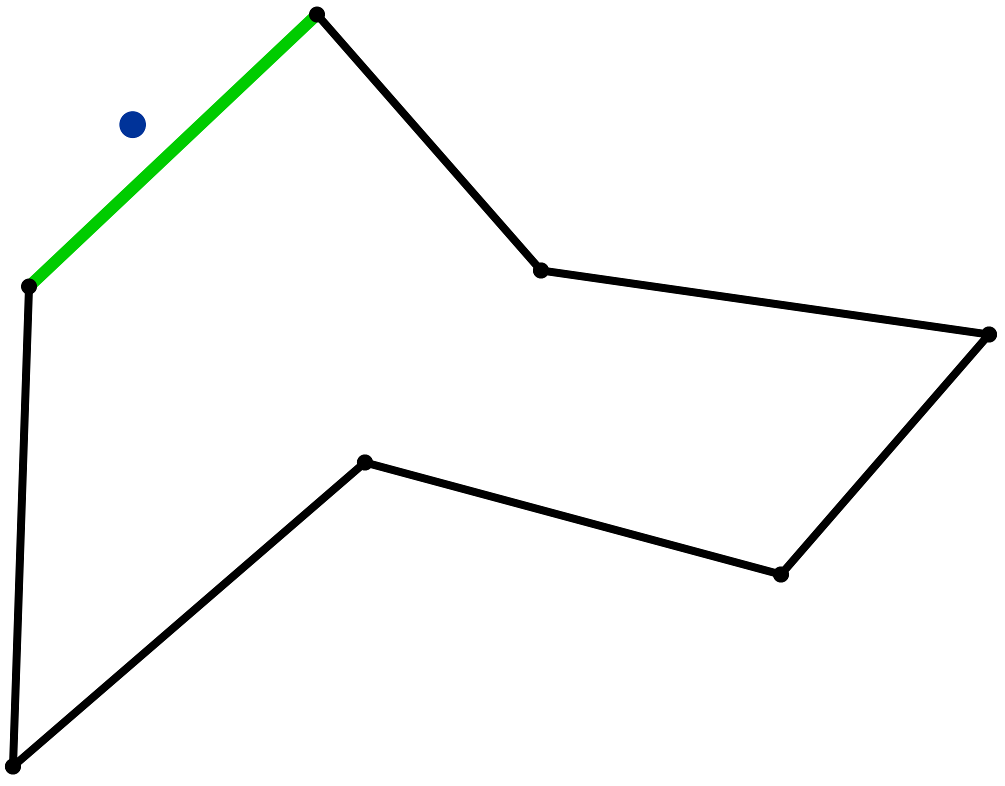
The 4-Point Scheme



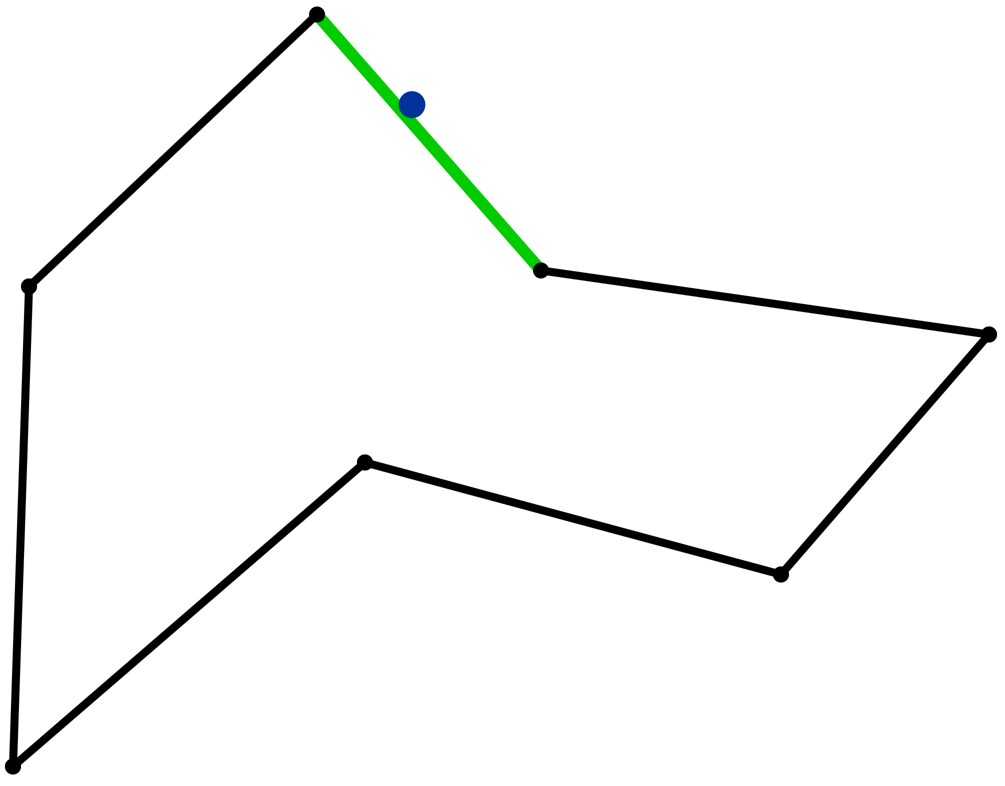
The 4-Point Scheme



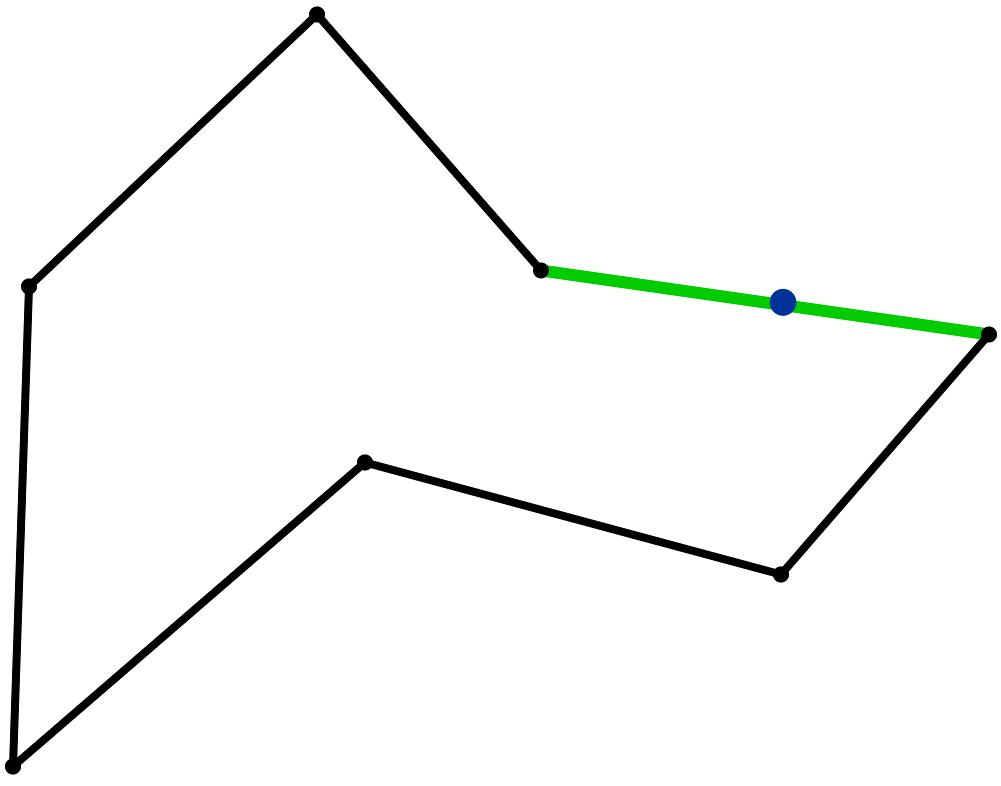
The 4-Point Scheme



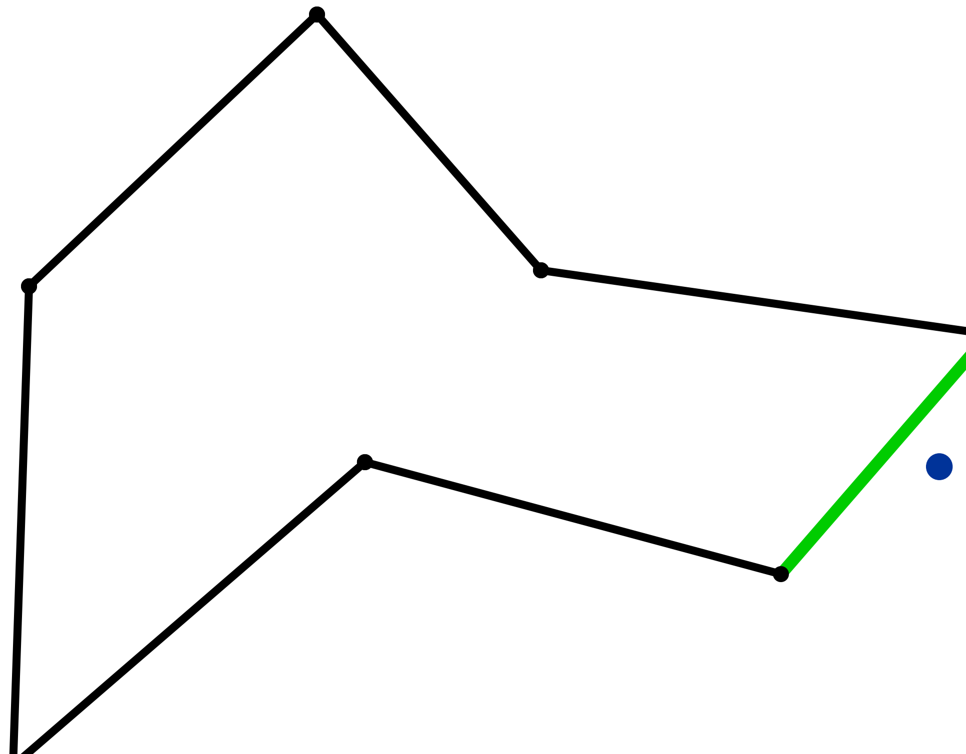
The 4-Point Scheme



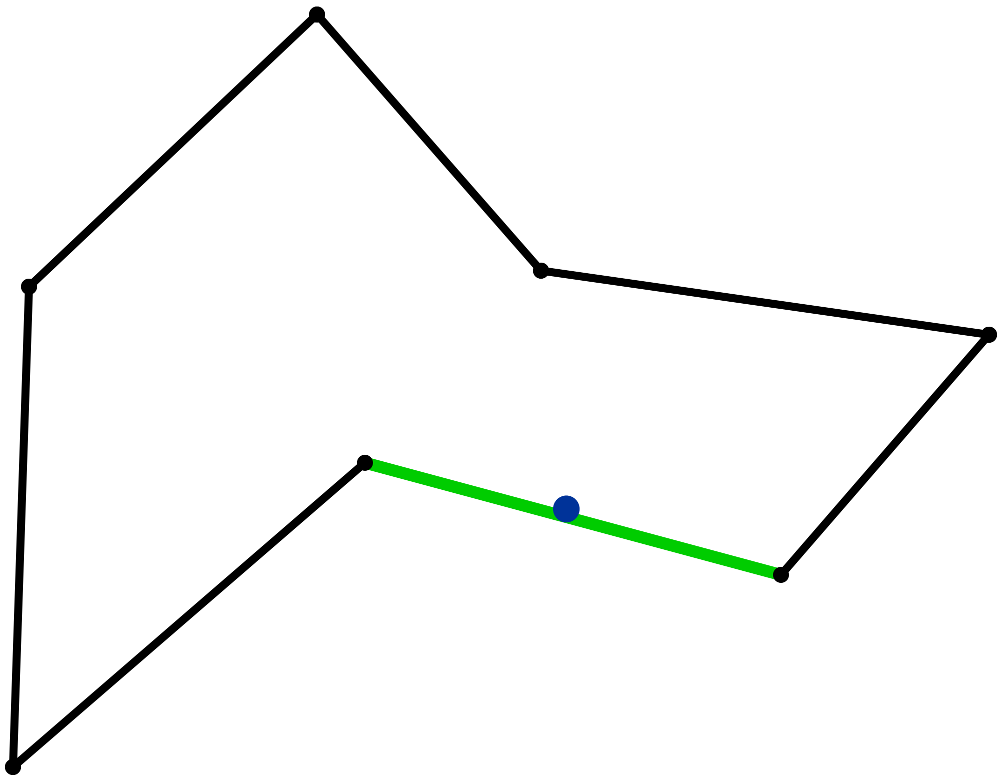
The 4-Point Scheme



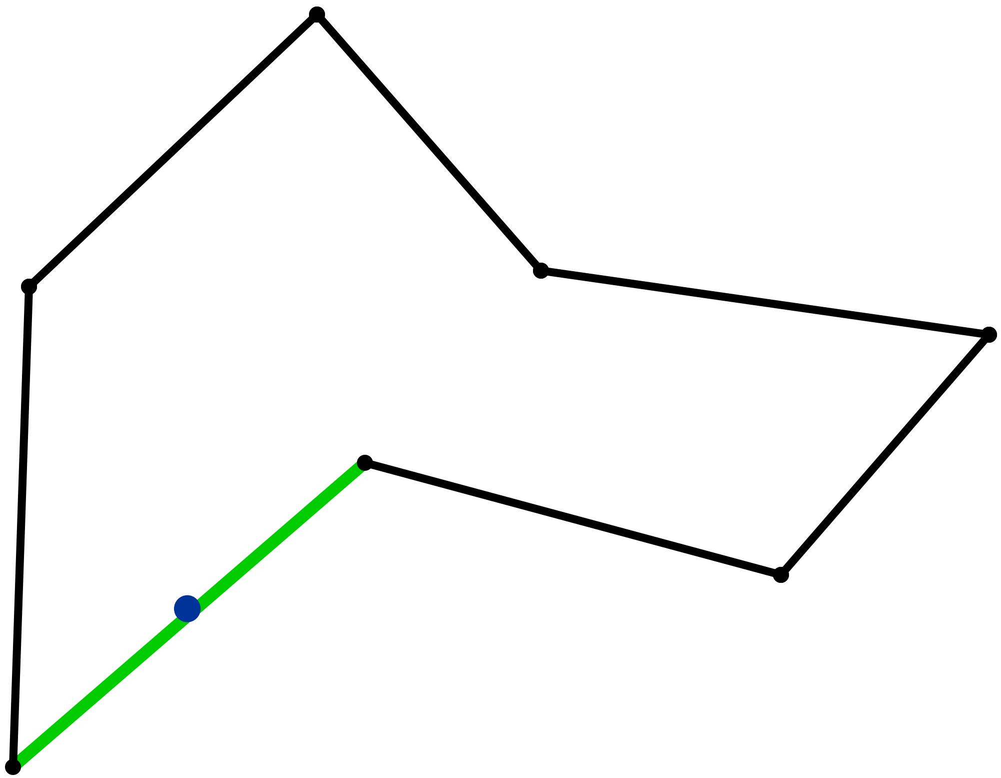
The 4-Point Scheme



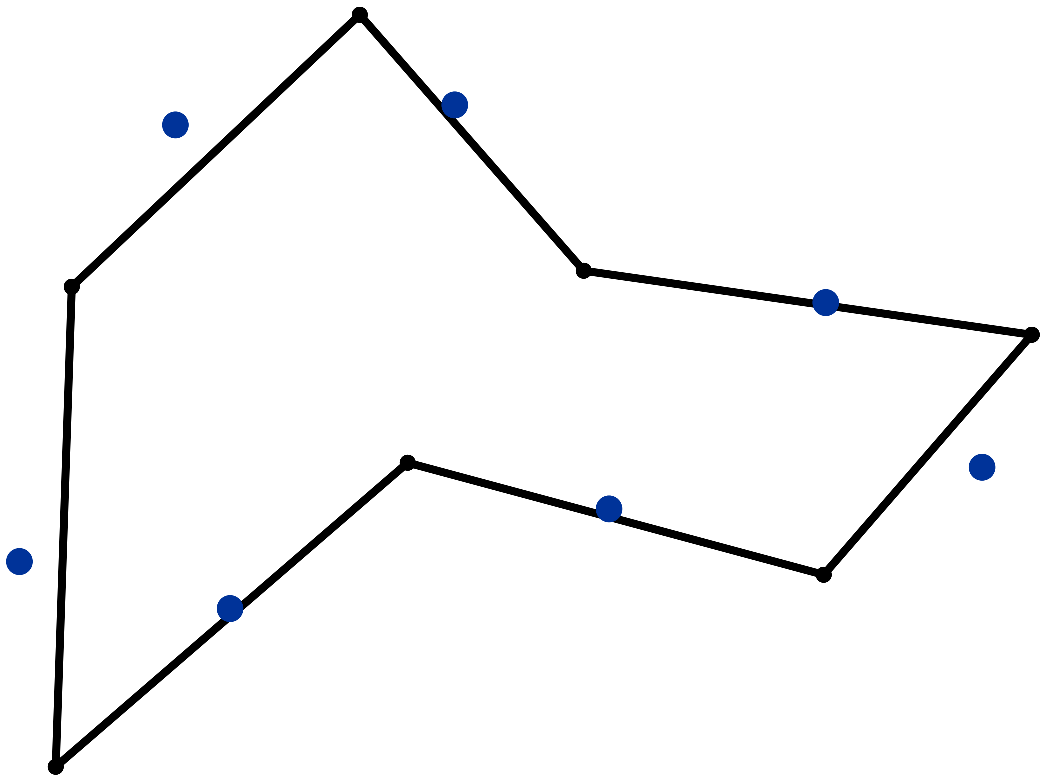
The 4-Point Scheme



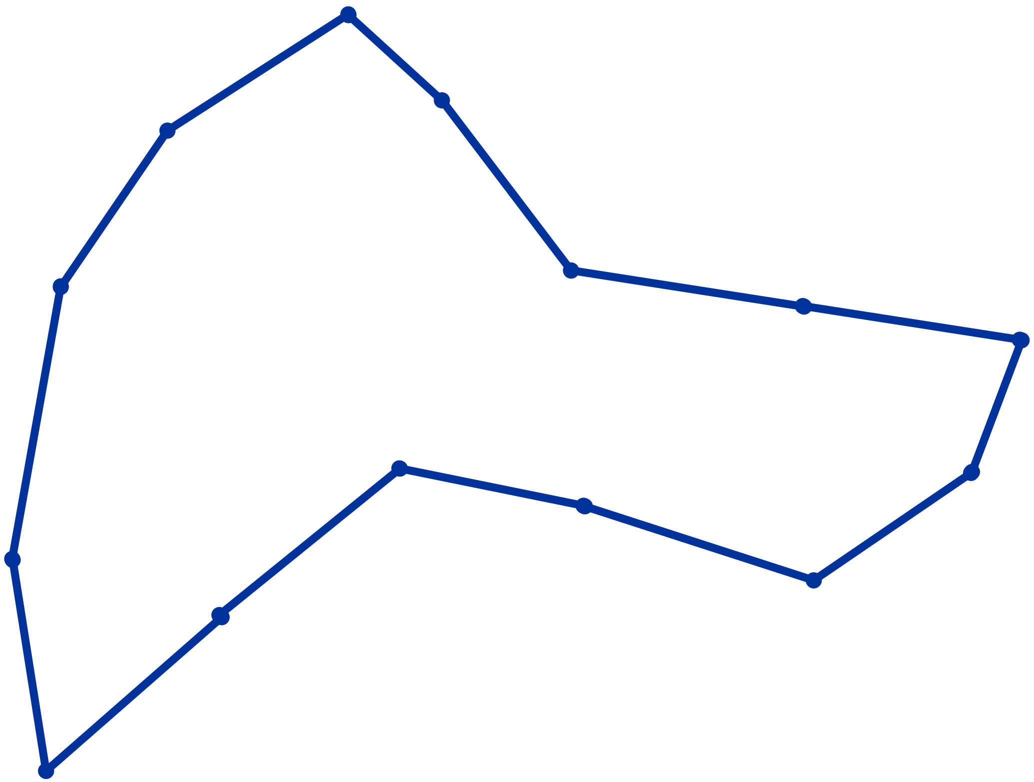
The 4-Point Scheme



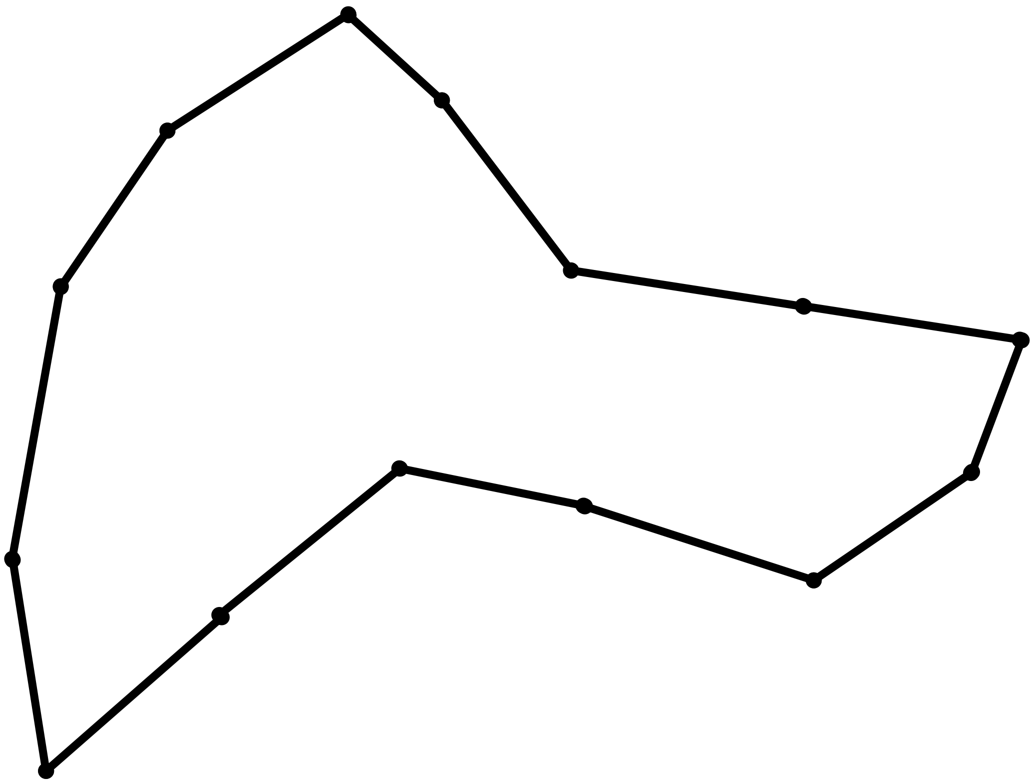
The 4-Point Scheme



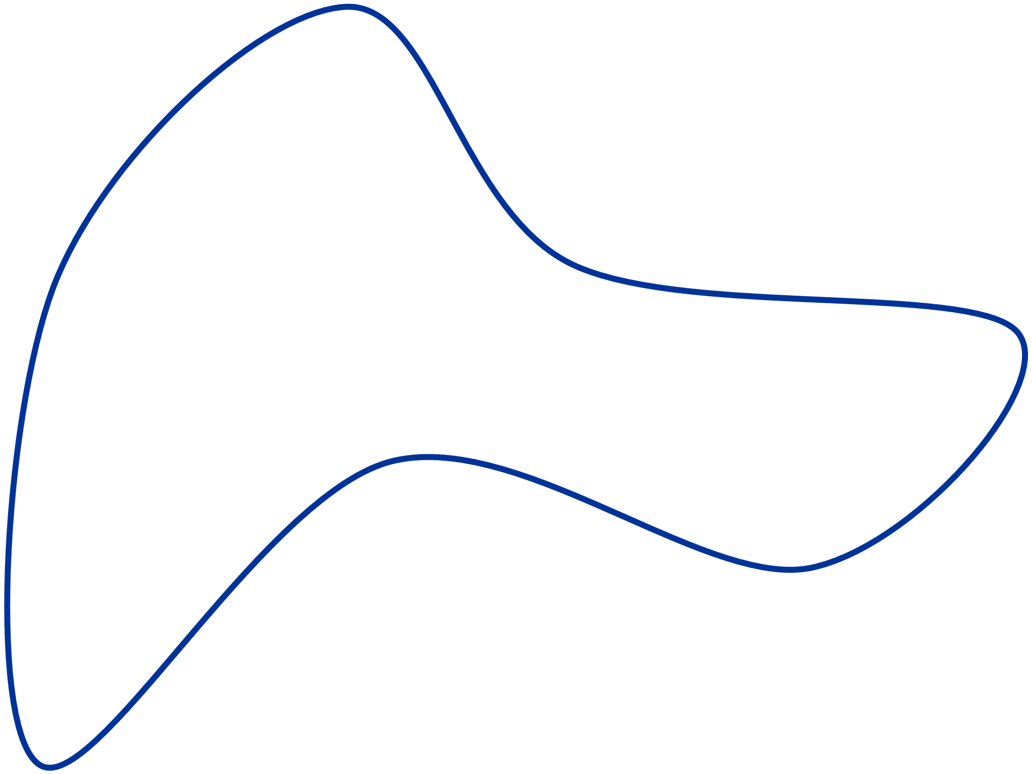
The 4-Point Scheme



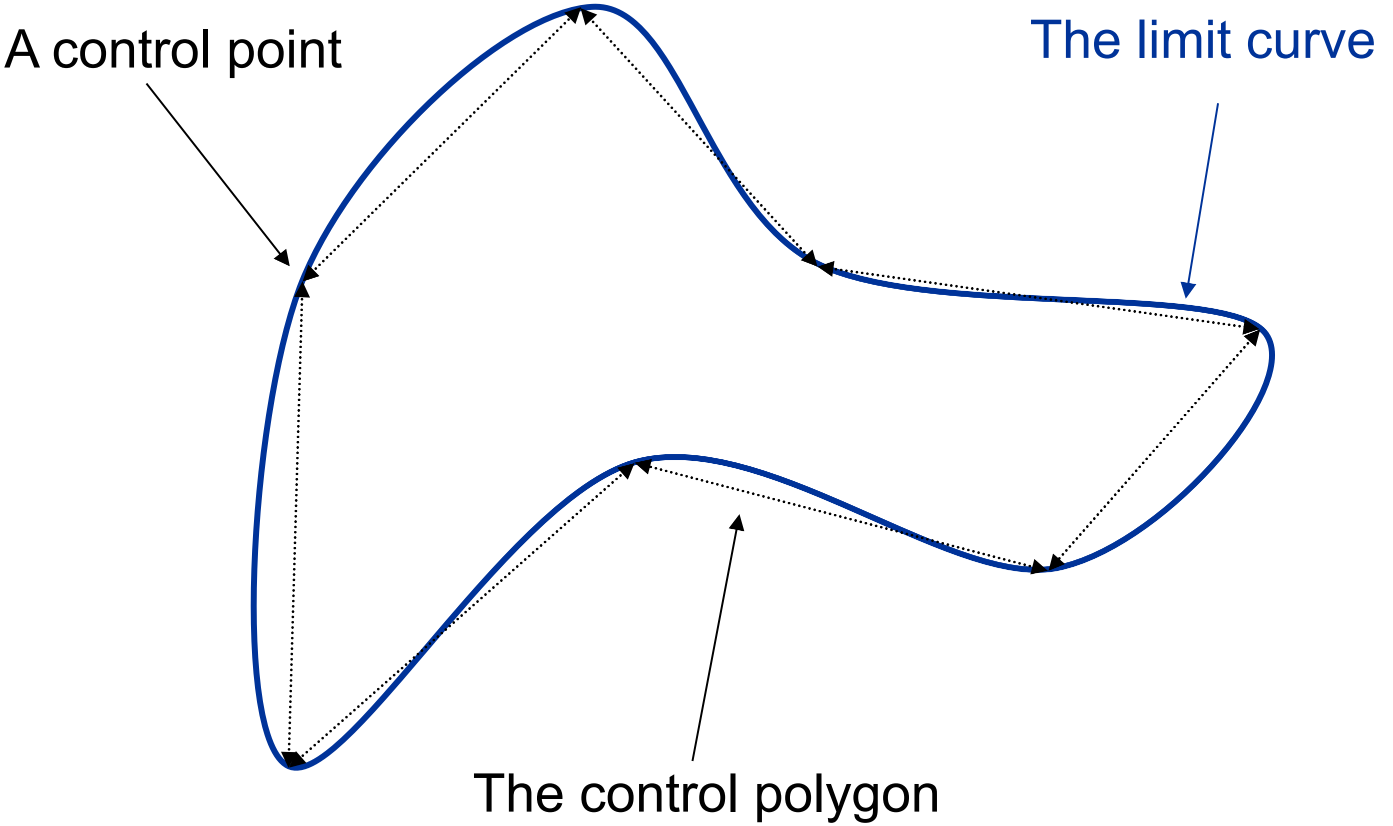
The 4-Point Scheme



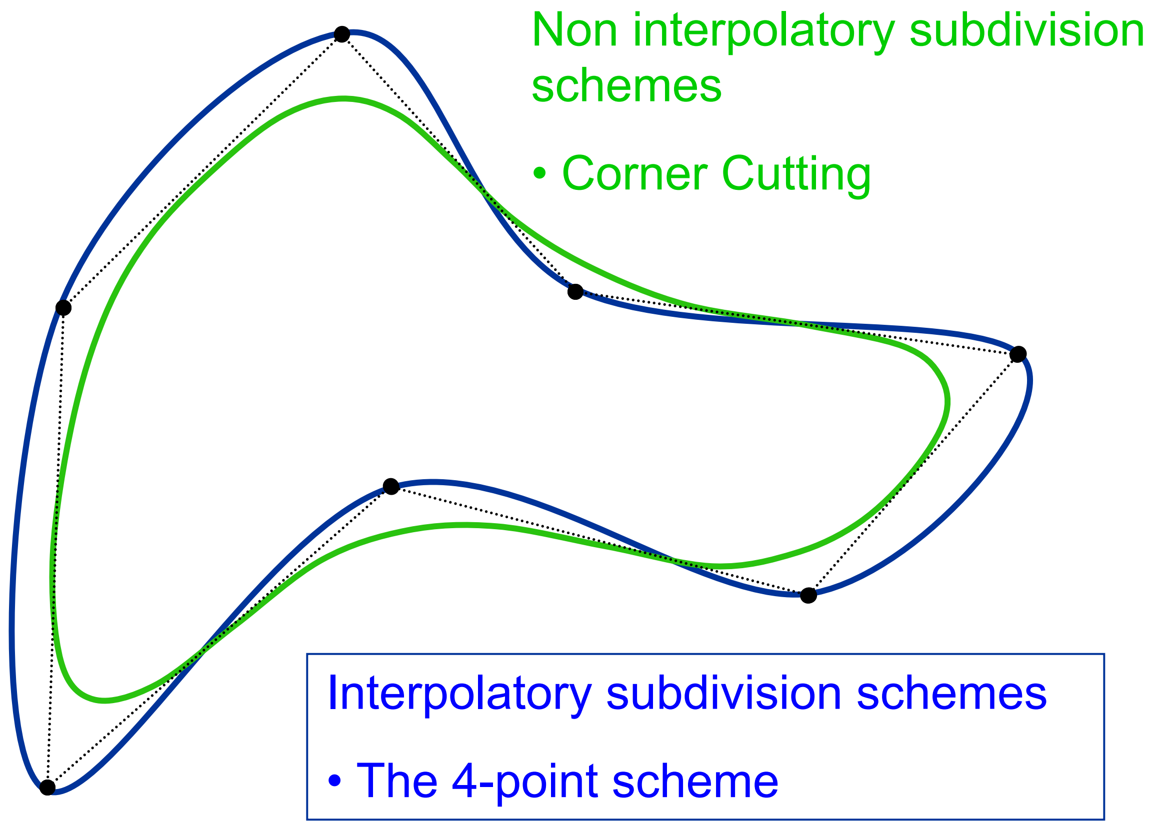
The 4-Point Scheme



The 4-Point Scheme



Subdivision Curves



Basic Concepts of Subdivision

- **Definition**
  - A subdivision curve is generated by repeatedly applying a subdivision operator to a given polygon (called the control polygon)

- **The central theoretical questions**
  - *Convergence:*
    Given a subdivision operator and a control polygon, does the subdivision process converge?
  - *Smoothness:*
    Does the subdivision process converge to a smooth curve? How smooth is it?

Surfaces Subdivision Schemes

- **A control net consists of vertices, edges, and face**
- **Refinement**
  - In each iteration, the subdivision operator refines the control net, increasing the number of vertices (approximately) by a factor of 4
- **Limit Surface**
  - In the limit the vertices of the control net converge to a limit surface
- **Topology and Geometry**
  - Every subdivision method has a method to generate the topology of the refined net, and rules to calculate the location of the new vertices

Surfaces Subdivision Schemes

- **A control net consists of vertices, edges, and face**
- **Refinement**
  - In each iteration, the subdivision operator refines the control net, increasing the number of vertices (approximately) by a factor of 4
- **Limit Surface**
  - In the limit the vertices of the control net converge to a limit surface
- **Topology and Geometry**
  - Every subdivision method has a method to generate the topology of the refined net, and rules to calculate the location of the new vertices

Subdivision Schemes

- **There are different subdivision schemes/rules**
  - Different methods for refining topology
- **Different rules for positioning vertices**
  - Interpolating versus approximating

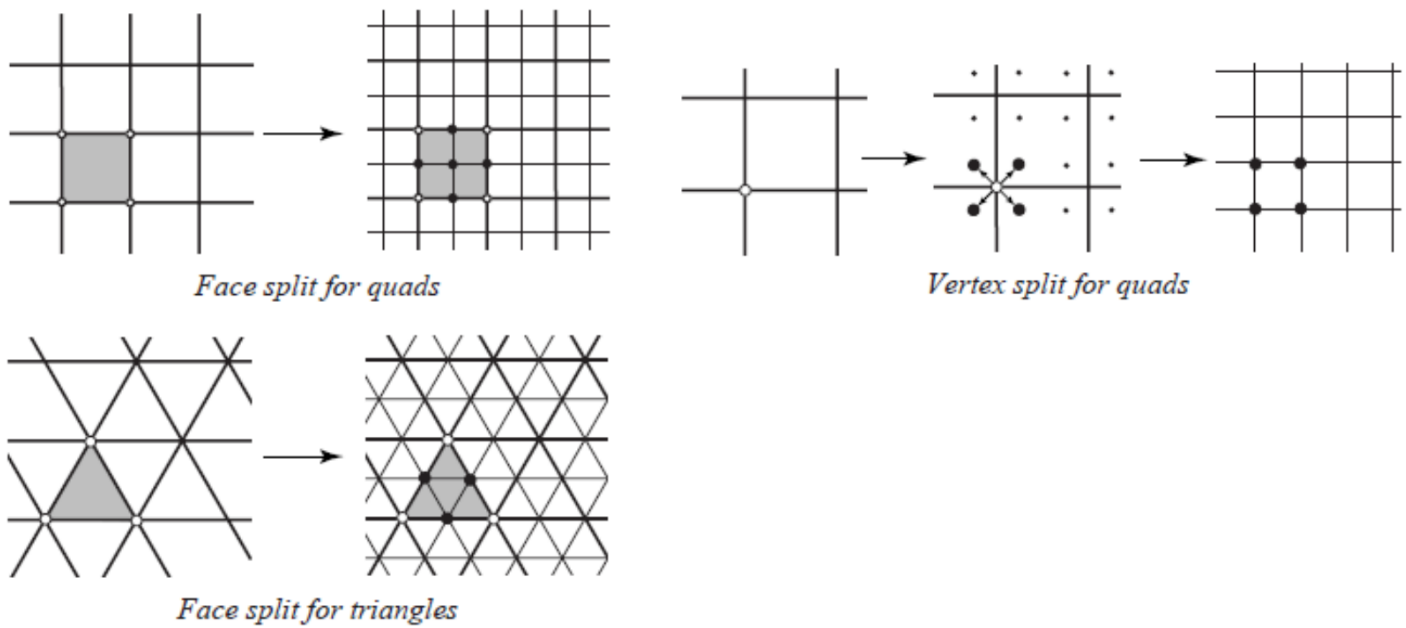
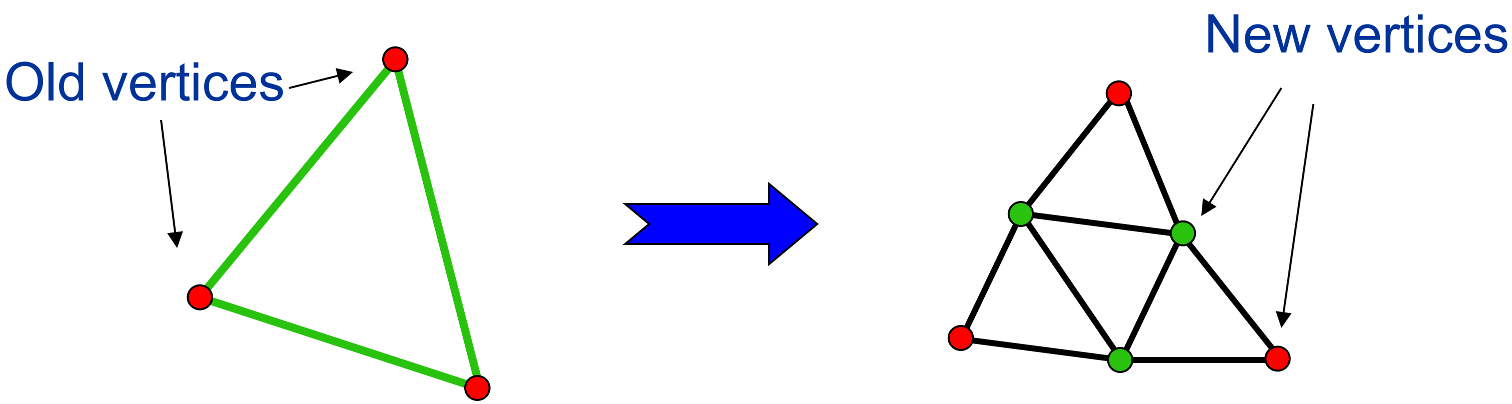


Figure 4.1: *Different refinement rules.*

Triangular Subdivision

- **For control nets whose faces are triangular**



- Every face is replaced by 4 new triangular faces.
- The are two kinds of new vertices
  - Green vertices are associated with old edges
  - Red vertices are associated with old vertices

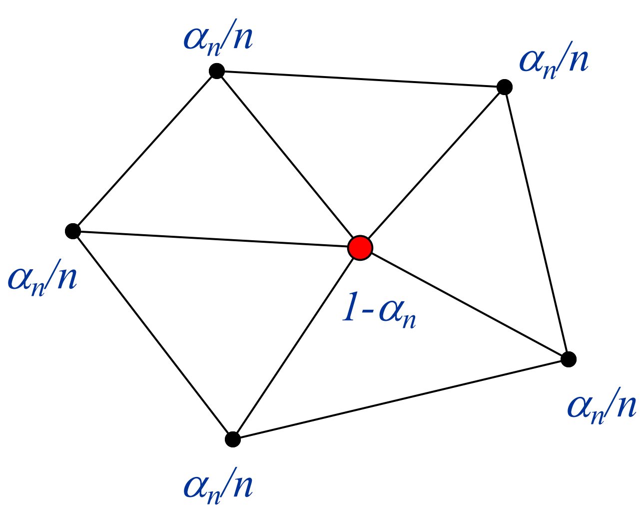
Loop Subdivision Scheme

- **Works on triangular meshes**
- **Is an Approximating Scheme**
- **Guaranteed to be smooth everywhere except at** *extraordinary* **vertices.**

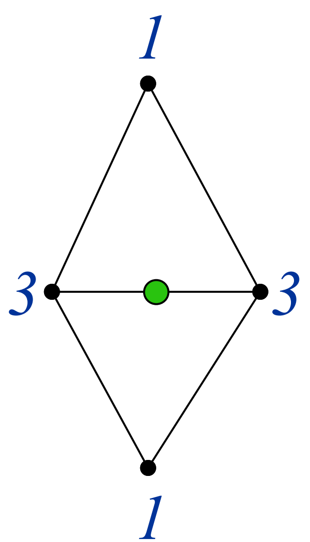
Loop's Scheme

- **Location of New Vertices**
  - Every new vertex is a weighted average of the old vertices. The list of weights is called the subdivision mask or the *stencil*

A rule for new red vertices



A rule for new green vertices



$$\alpha_n = \frac{1}{64}\left(40 - \left(3 + 2\cos\left(\frac{2\pi}{n}\right)\right)^2\right)$$

Original

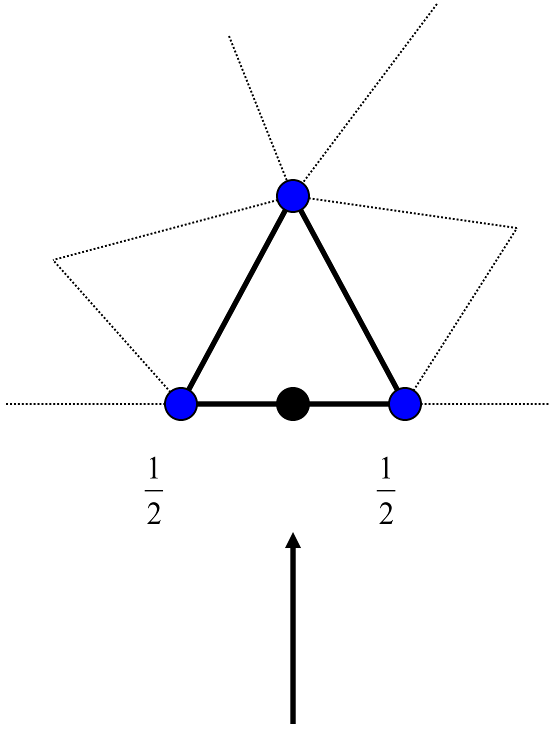
$$\alpha_n = \begin{cases} \frac{3}{8} & n > 3 \\ \frac{3}{16} & n = 3 \end{cases}$$

Warren

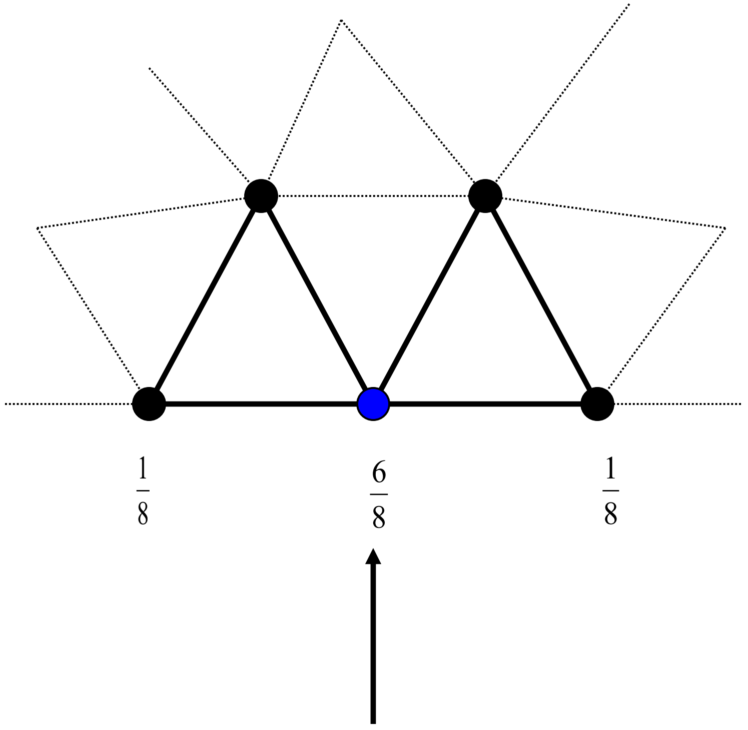
n -the vertex valence

Loop Subdivision Boundaries

- **Subdivision Mask for Boundary Conditions**

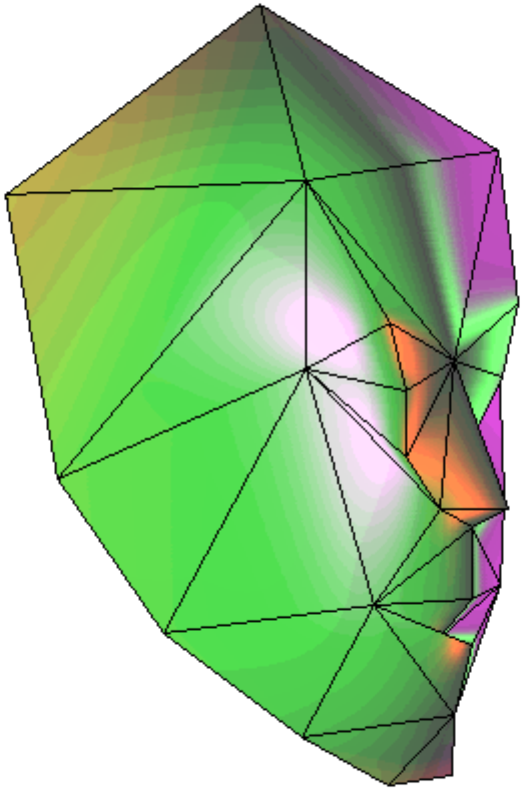


Edge Rule

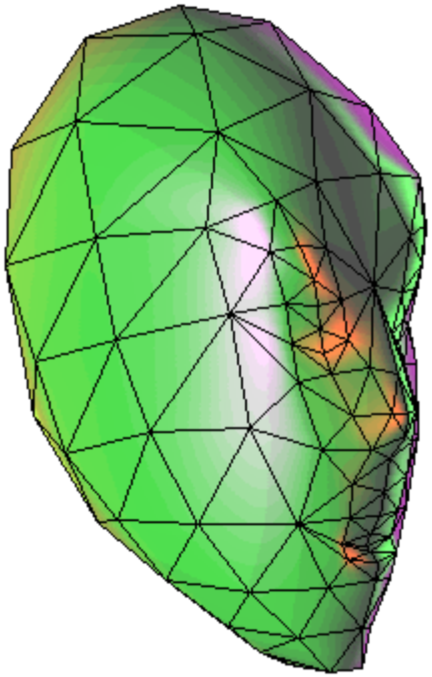


Vertex Rule

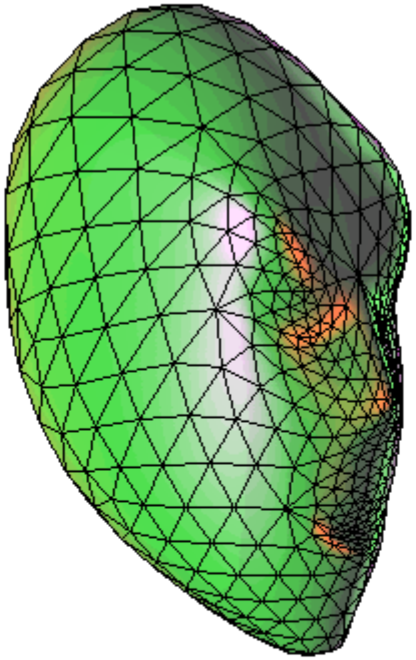
The Original Control Net



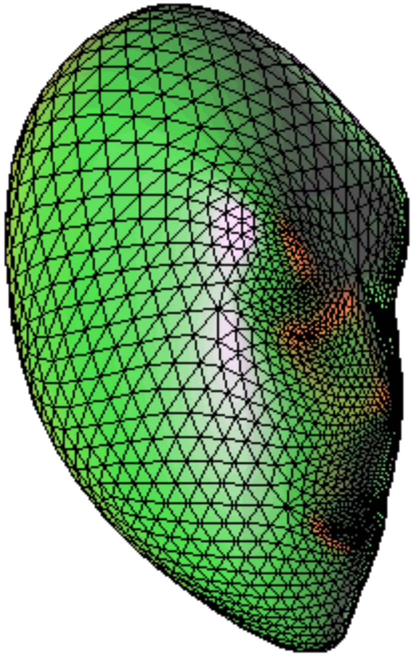
After 1st Iteration



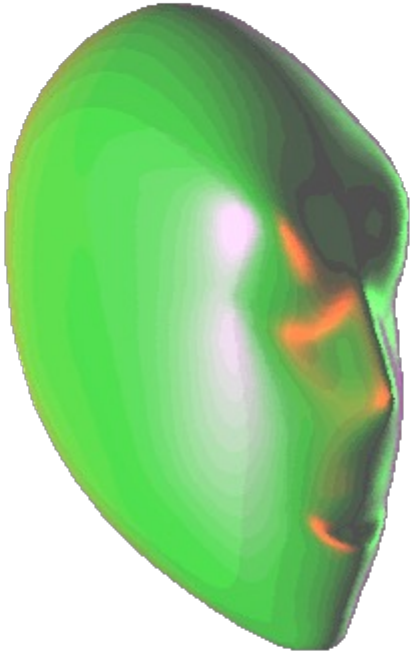
After 2nd Iteration



After 3rd Iteration



The Limit Surface



The limit surfaces of Loop's subdivision have continuous curvature almost everywhere

The (Modified) Butterfly Scheme

- **(Modified) Butterfly Scheme**
  - This is an interpolatory scheme
  - The new red vertices inherit the location of the old vertices
  - The new green vertices are calculated by the following stencil

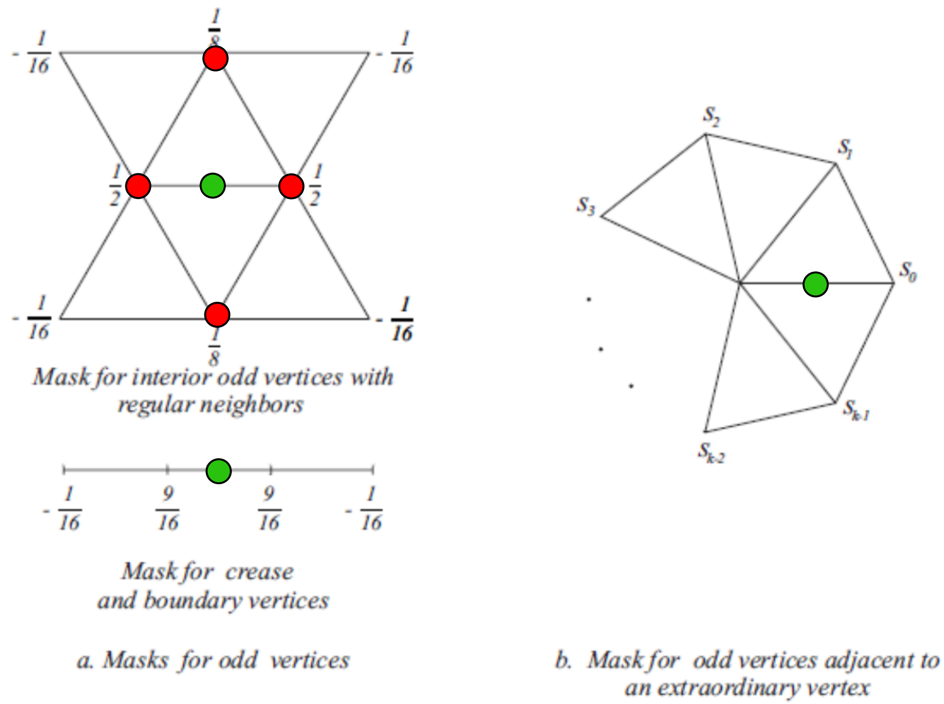
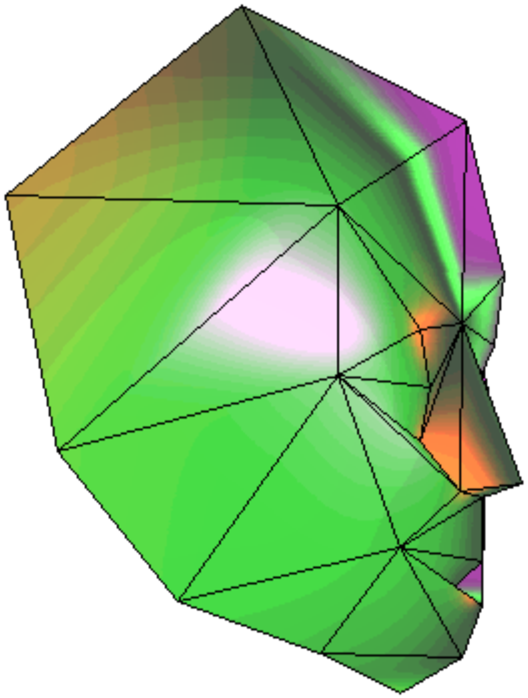
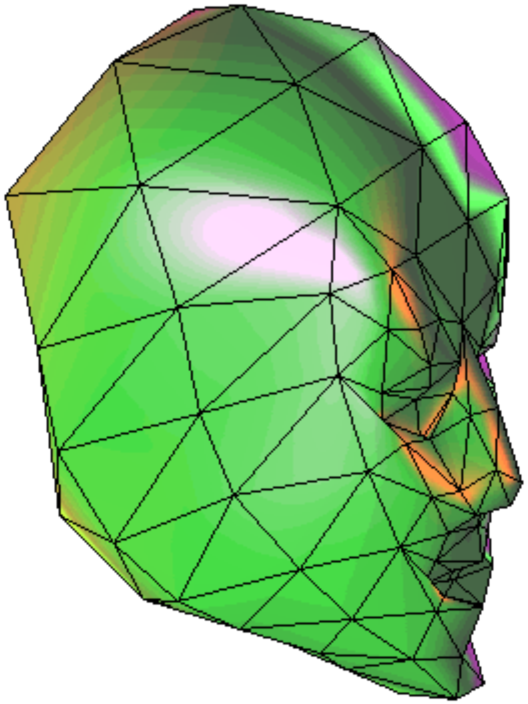


Figure 4.5: *Modified Butterfly subdivision. The coefficients s_i are* $\frac{1}{k}\left(\frac{1}{4}+\cos\frac{2i\pi}{k}+\frac{1}{2}\cos\frac{4i\pi}{k}\right)$ *for* $k > 5$. *For* $k = 3$, $s_0 = \frac{5}{12}$, $s_{1,2} = -\frac{1}{12}$; *for* $k = 4$, $s_0 = \frac{3}{8}$, $s_2 = -\frac{1}{8}$, $s_{1,3} = 0$.

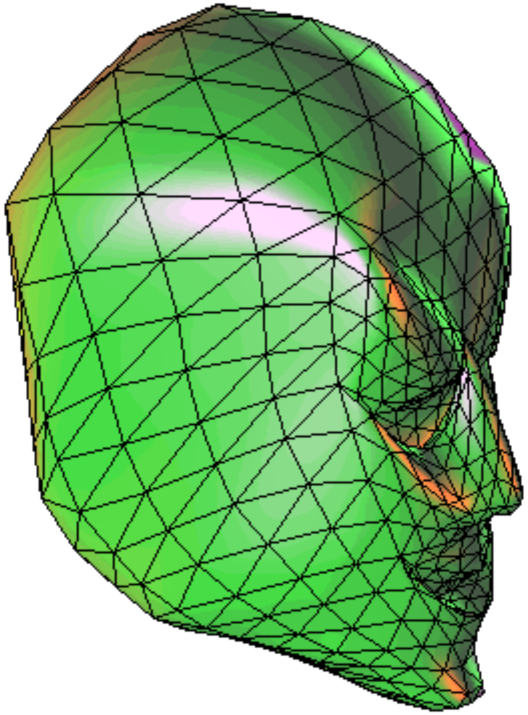
The Original Control Net



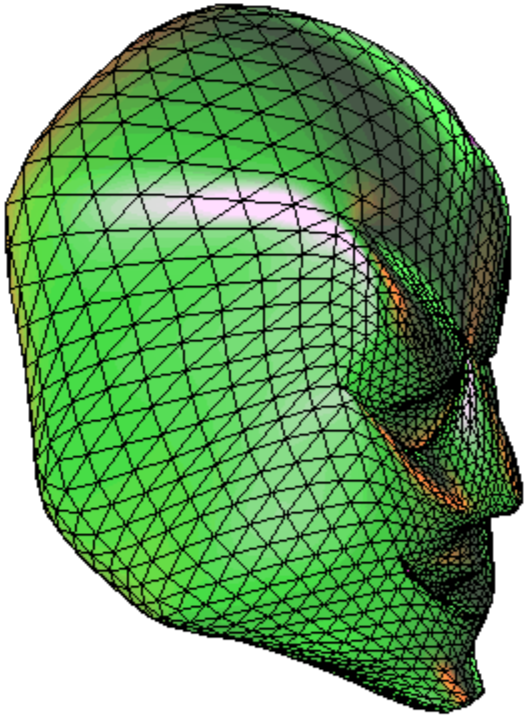
After 1st Iteration



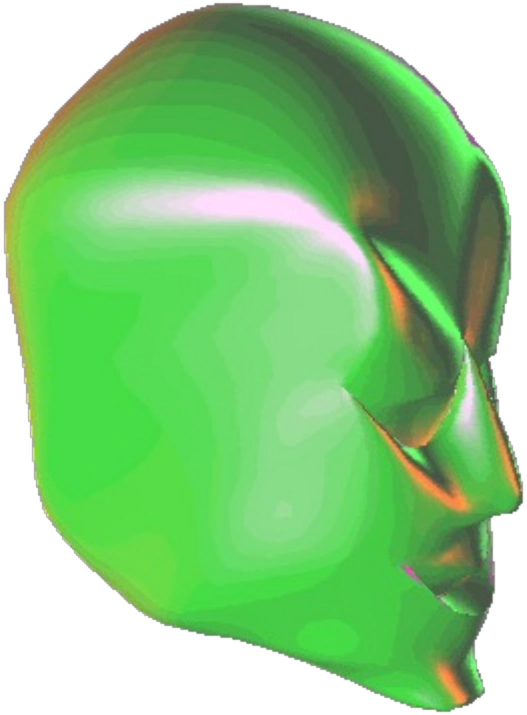
After 2nd Iteration



After 3rd Iteration



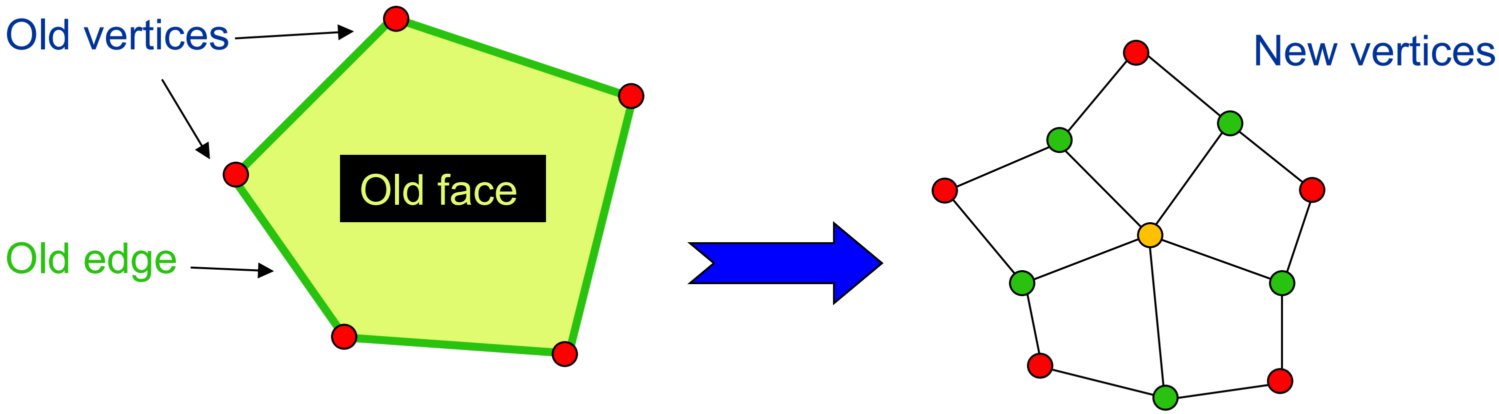
The Limit Surface



The limit surfaces of the Butterfly subdivision are smooth but are nowhere twice differentiable.

Quadrilateral Subdivision

- **Works for control nets of arbitrary topology**
  - After one iteration, all the faces are quadrilateral.



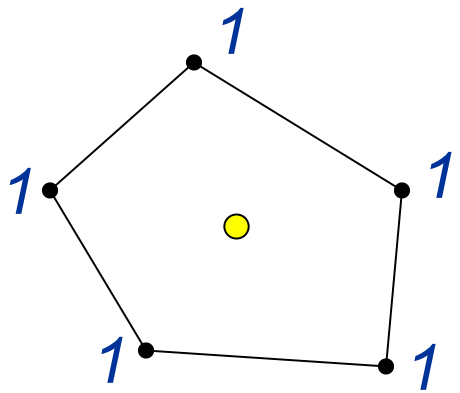
Every face is replaced by quadrilateral faces.
The are three kinds of new vertices:

- Yellow vertices are associated with old faces
- Green vertices are associated with old edges
- Red vertices are associated with old vertices.

Catmull Clark's Scheme

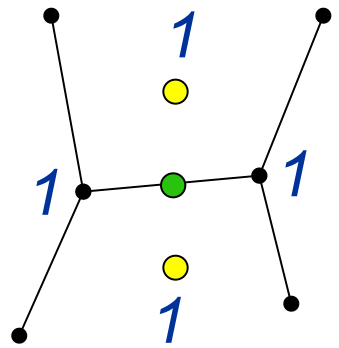
Step 1

First, all the yellow vertices are calculated



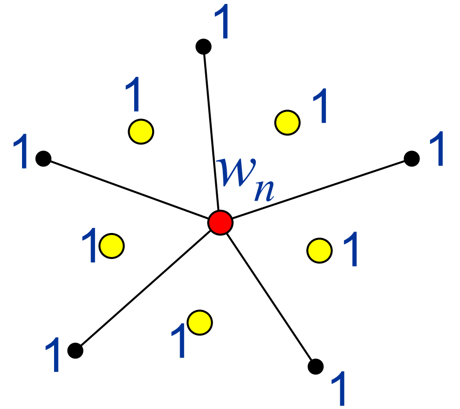
Step 2

Then the green vertices are calculated using the values of the yellow vertices



Step 3

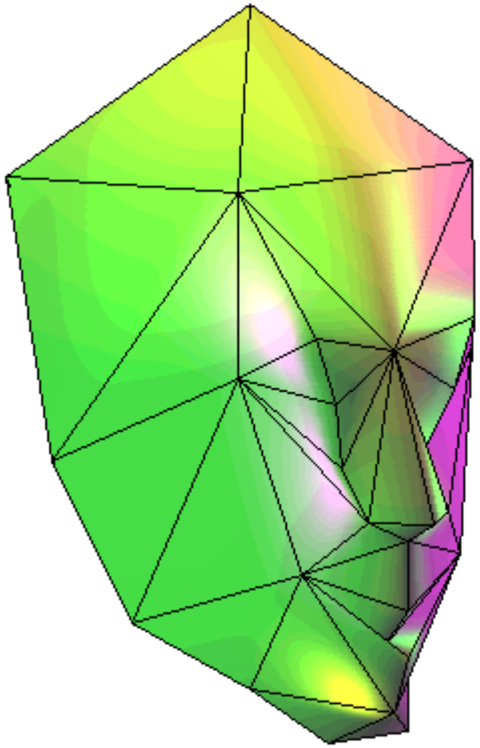
Finally, the red vertices are calculated using the values of the yellow vertices



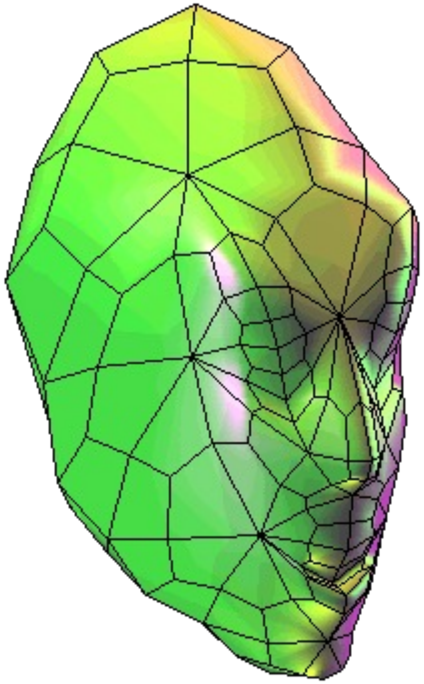
n -the vertex valence

$$w_n = n(n-2)$$

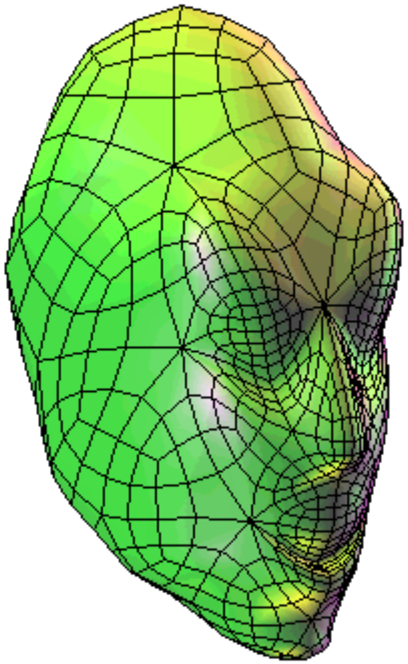
The Original Control Net



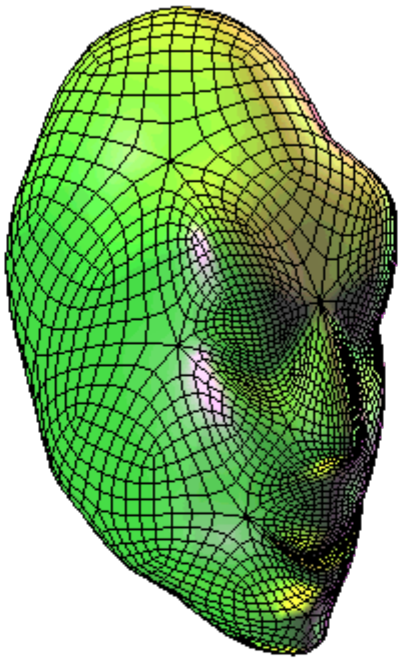
After 1st Iteration



After 2nd Iteration



After 3rd Iteration

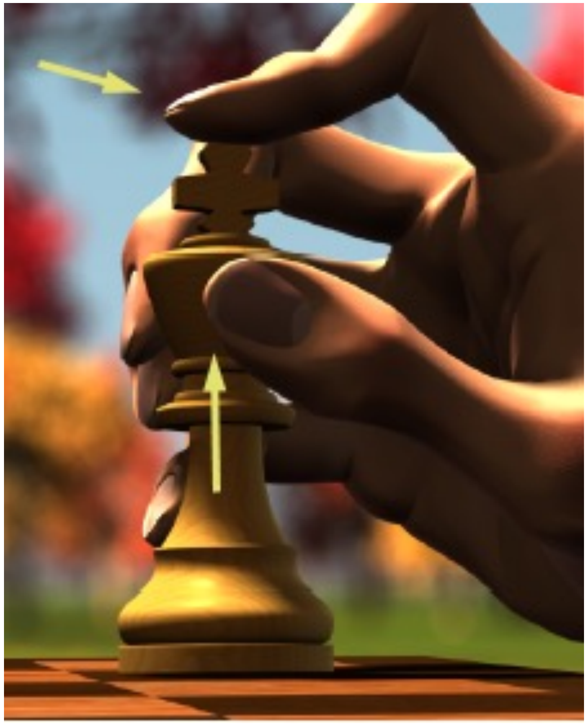


The Limit Surface

The limit surfaces of Catmull-Clarks's subdivision have continuous curvature almost everywhere

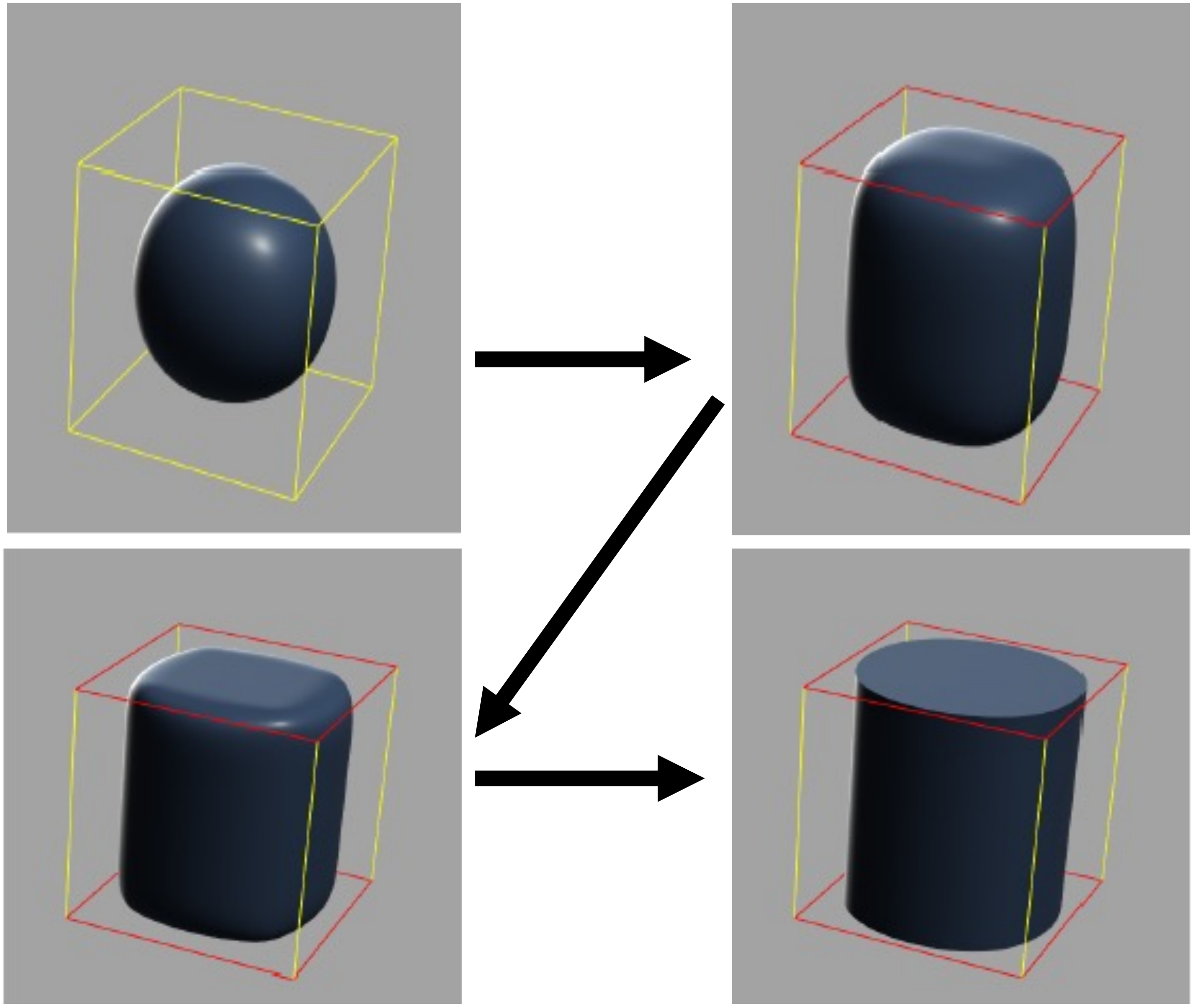
Edges and Creases

- **Most surface are not smooth everywhere**
  - Edges & creases
  - Can be marked in model
    - Weighting is changed to preserve edge or crease
- **Generalization to semi-sharp creases (Pixar)**
  - Controllable sharpness
  - Sharpness (s) = 0, smooth
  - Sharpness (s) = inf, sharp
  - Achievable through hybrid subdivision step
    - Subdivision iff s==0
    - Otherwise, parameter is decremented



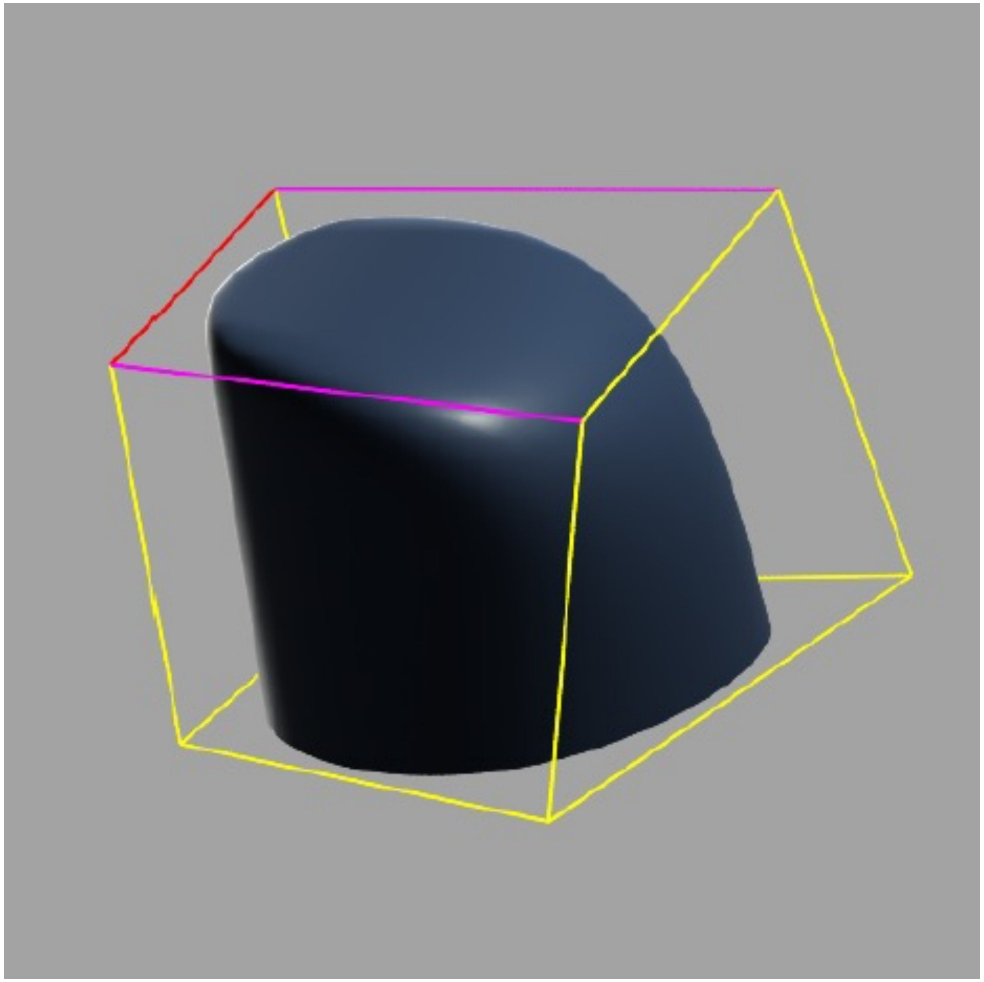
Edges and Creases

- **Increasing sharpness of edges**



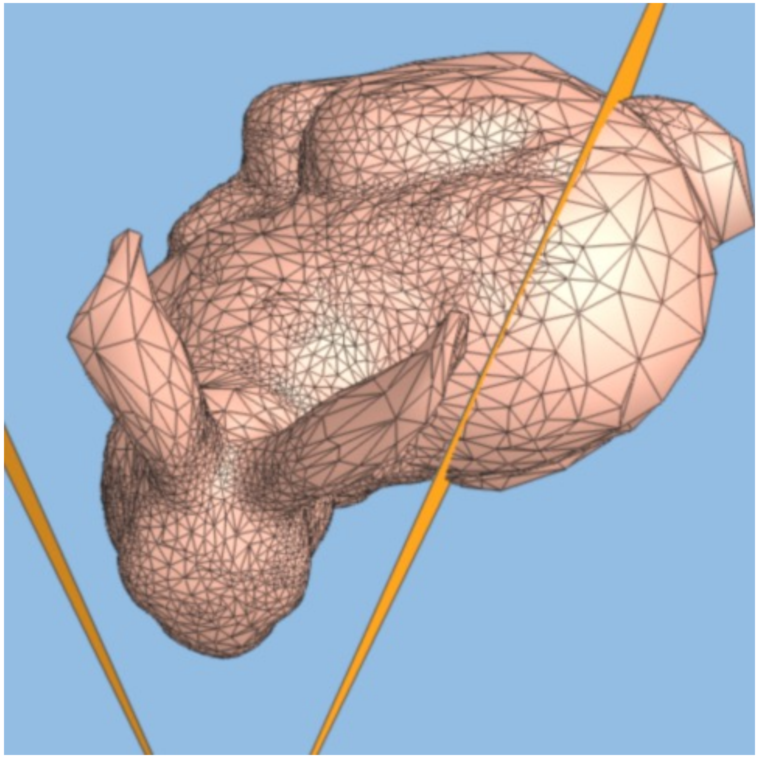
Edges and Creases

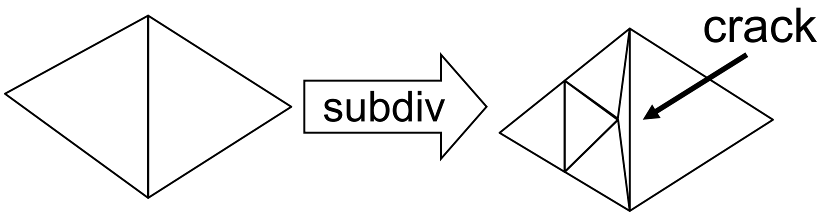
- **Can be changed on a edge by edge basis**



Adaptive Subdivision

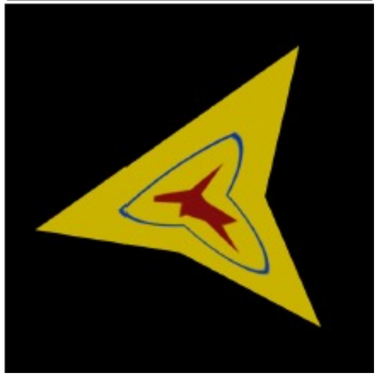
- **Not all regions of a model need to be subdivided.**
- **Idea: Use some criteria and adaptively subdivide mesh where needed.**
  - Curvature
  - Screen size
    - Make triangles < size of pixel
  - View dependence
    - Distance from viewer
    - Silhouettes
    - In view frustum
  - Careful!
    - Must avoid "cracks"

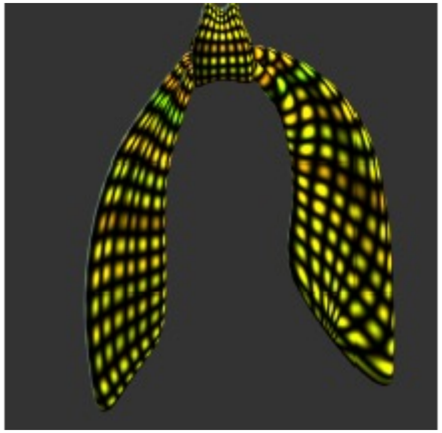




Texture mapping

- **Solid color painting is easy, already defined**
- **Texturing is not so easy**
  - Using polygonal methods can result in distortion
- **Solution**
  - Assign texture coordinates to each original vertex
  - Subdivide them just like geometric coordinates
- **Introduces a smooth scalar field**
  - Used for texturing in Geri's jacket, ears, nostrils





Advanced Topics

- **Hierarchical Modeling**
  - Store offsets to vertices at different levels
  - Offsets performed in normal direction
  - Can change shape at different resolutions while rest stays the same
- **Surface Smoothing**
  - Can perform filtering operations on meshes
    - E.g. (weighted) averaging of neighbors
- **Level-of-Detail**
  - Can easily adjust maximum depth for rendering