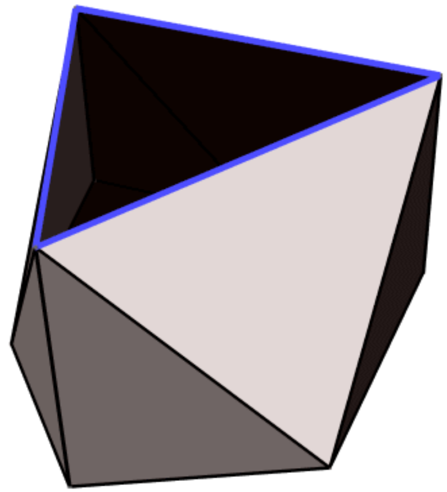
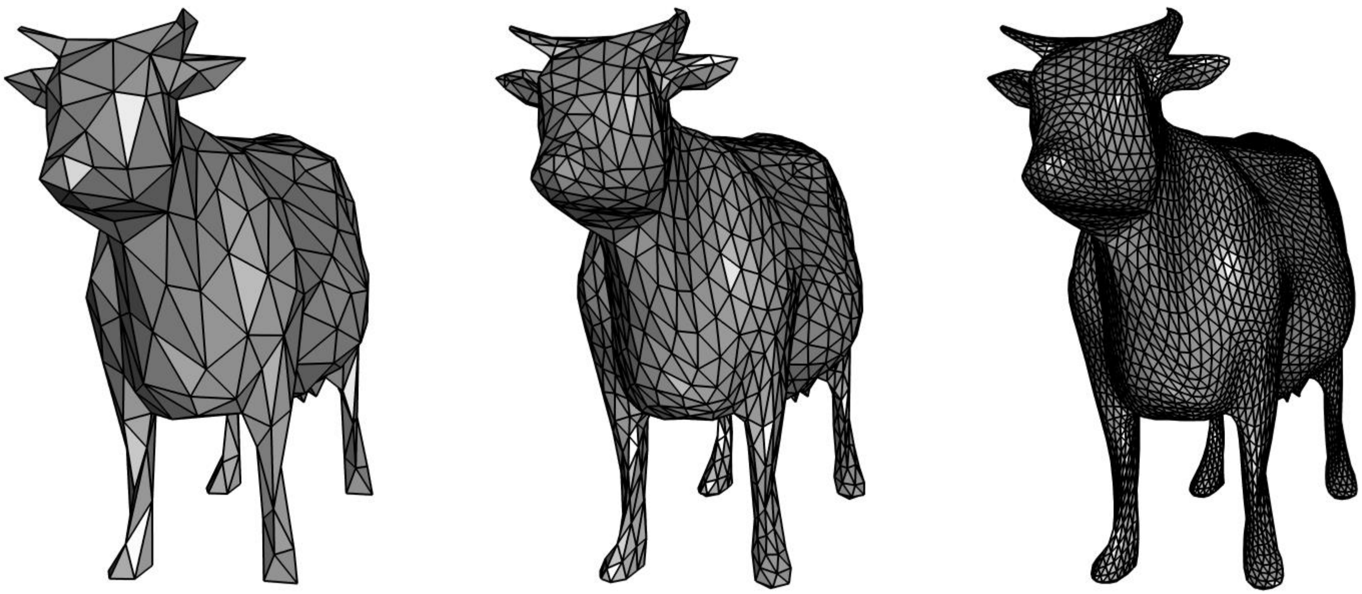
Subdivision Surfaces



Intro – Motivation

- Subdivide a coarse base polygonal mesh of arbitrary topology with rules that are as simple as possible



Intro – Advantages

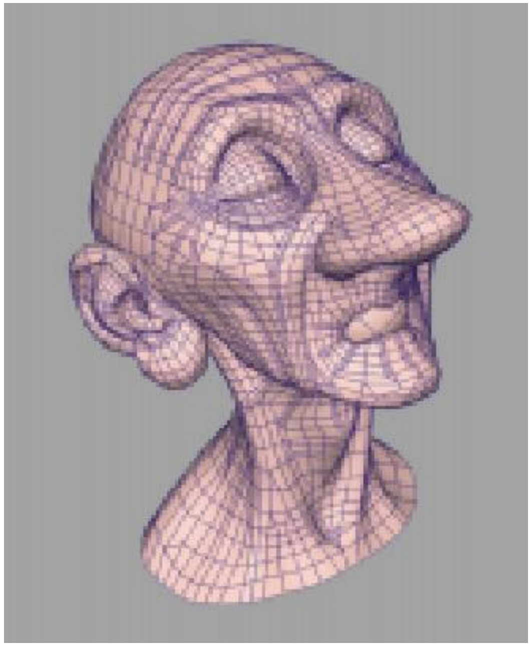
- **arbitrary topology:** (spherical, toroidal, double toroidal, ...)
- **scalability:** by definition multi-scale representation
- **numerical robust:** if you start with well behaved base meshes suitable for FEM solvers, the subdivided meshes inherit the good proberties
- **simple implementation:** code is quite simple even though mathematical analysis is rather involved
- **wavelets:** construction of basis functions and wavelets over surface possible and suitable for compression and multi-resolution representation of surface and functions
- **multi-scale-editing**

Intro – Design Goals

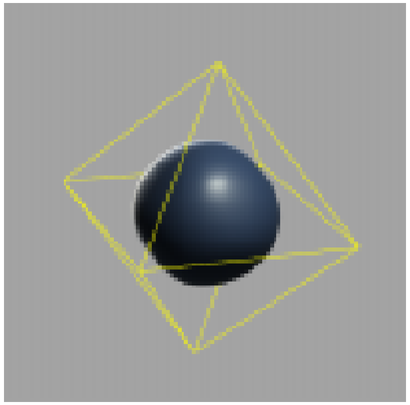
Computergraphik und Visualisierung

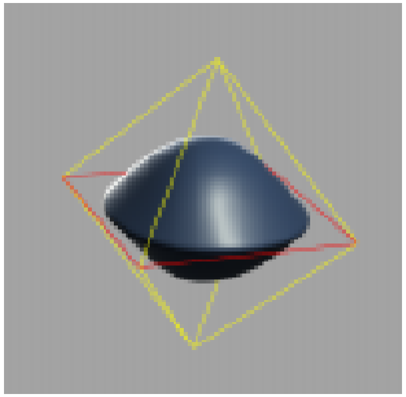
- **efficiency:** few computations
- **compact support:** influence of base vertex on shape should be limited to local neighborhood
- **local rules:** computational rules should not depend on distant points
- **affine invariance:** affine transformations and subdivision process should be commutative
- **simplicity:** rules should be as simple as possible
- **continuity:** limit curves and surfaces should be as smooth as possible

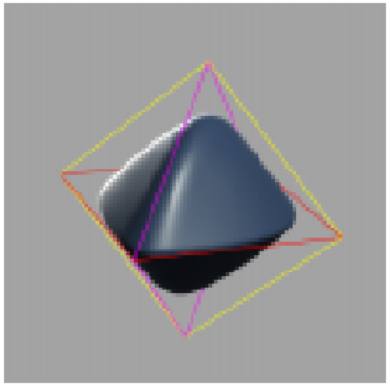
Intro – Examples

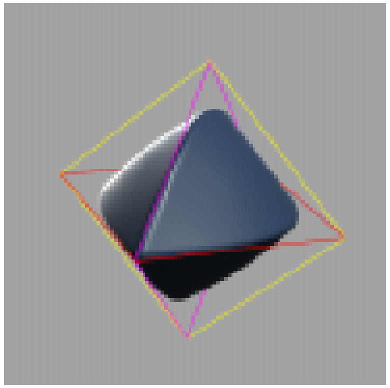


https://www.youtube.com/watch?v=uMVtpCPx8ow

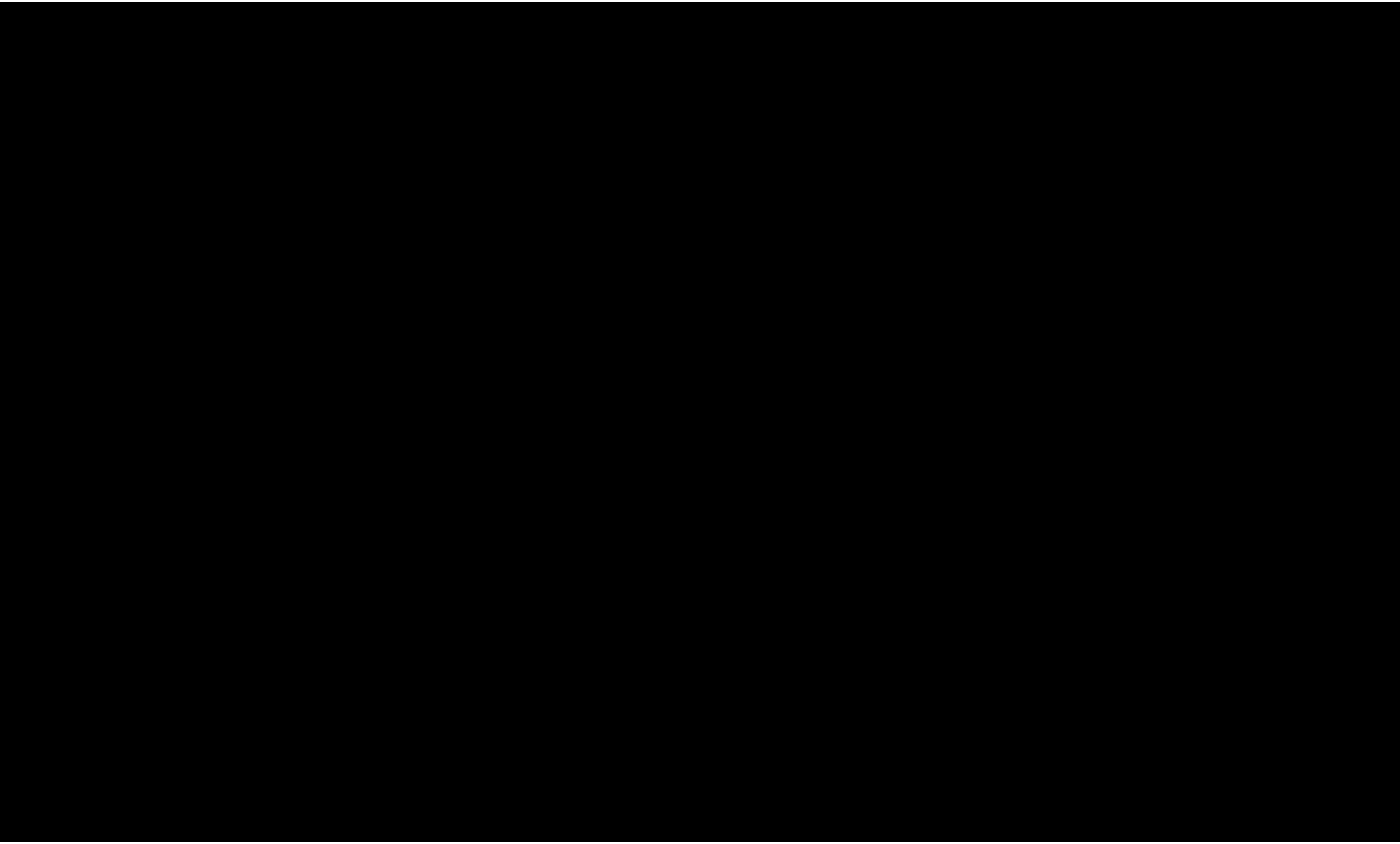




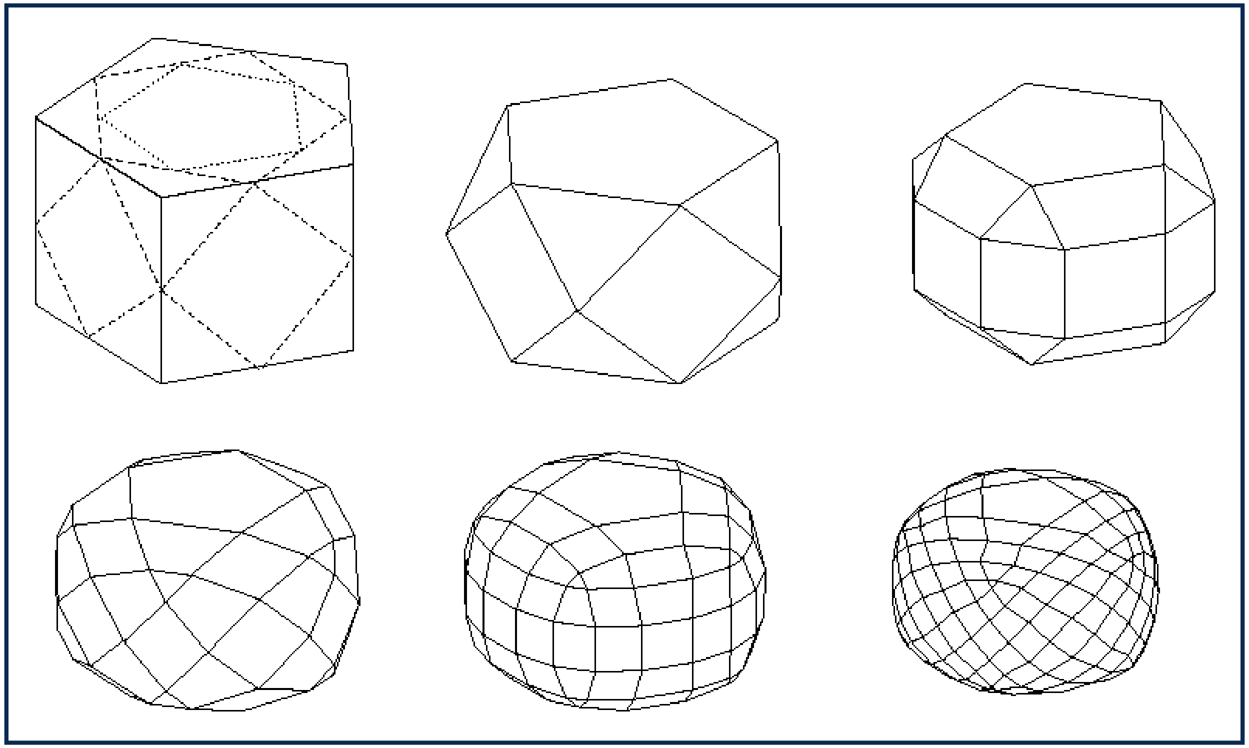




Geri's Game (Pixar)



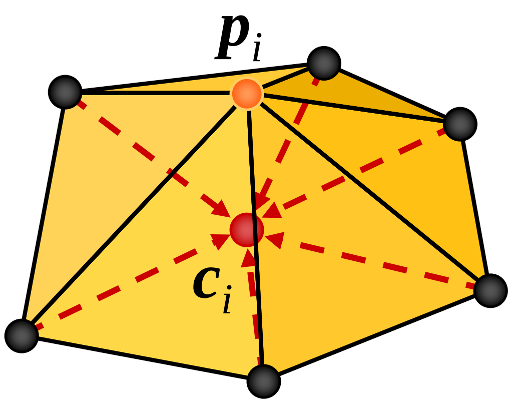
Motivation – Midpoint Subdivision

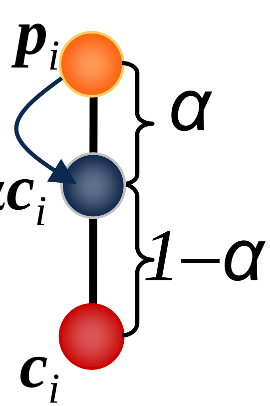


Peters, J. und Reif, U.: The Simplest Subdivision Scheme for Smoothing Polyhedra. *ACM Transactions on Graphics 16* (1997), S. 420-431.

Motivation – Smoothing

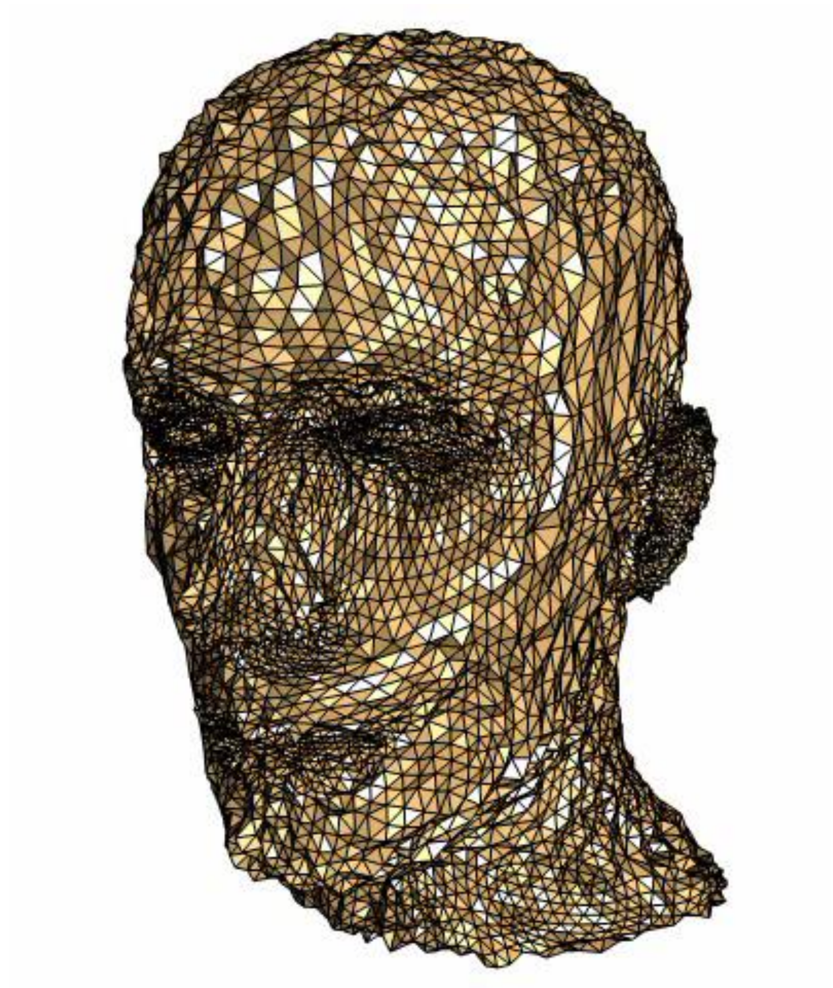
smoothing operator U_α^j:





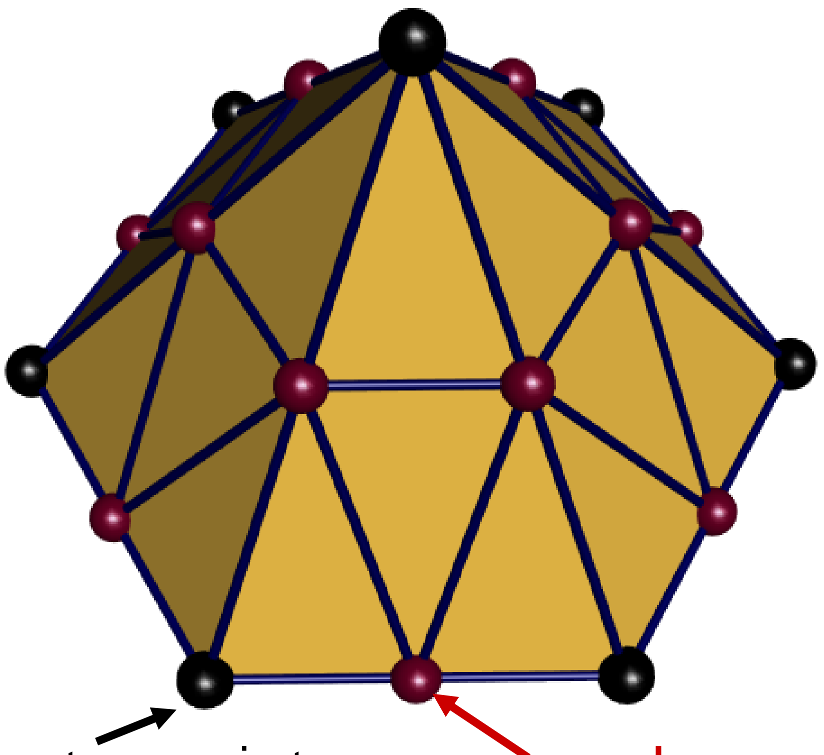
$$p_i' = (1-\alpha)p_i + \alpha c_i$$

$$P^{j+1} = U_\alpha^j P^j$$



Motivation – Interpretation

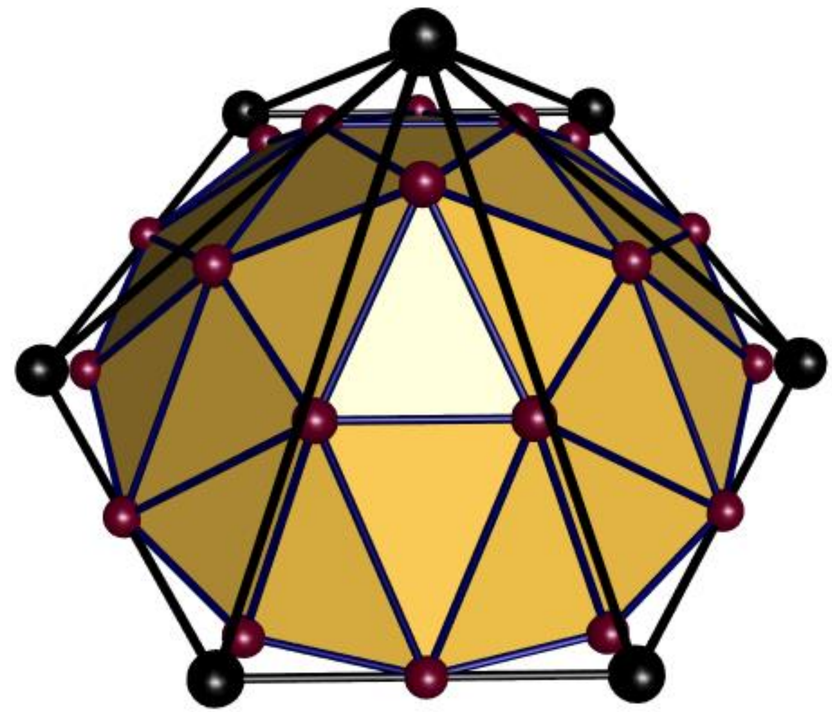
split process into subdivision and smoothing:



vertex points

edge points

subdivision

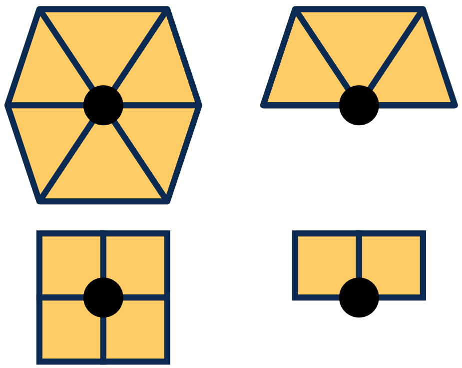


$$\boldsymbol{P}^{j+1} = \boldsymbol{U}_{\alpha} \boldsymbol{P}^{j+1}$$

smoothing

Terminology – Vocabulary

- ordinary vertex:
  in triangle meshes valence 6 and at boundary valence 4
  in quad meshes valence 4 and at boundary valence 3



- extraordinary vertex: all other vertices
- The first subdivision step separates all extraordinary vertices
- Therefore the limit surface only needs to be analized in neighborhood of ordinary and isolated extraordinary points

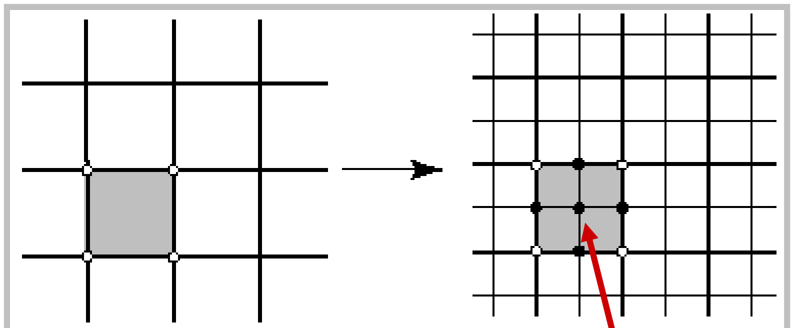
Terminology – Types of Schemata

Criteria for Cathegorization

- mesh type: triangle vs. quad mesh
- approximating vs. interpolating
- continuity of limit surfaces in nearly all points, where in general C^1*-continuity* is achievable and presumed
- subdivision step: face split vs. vertex split

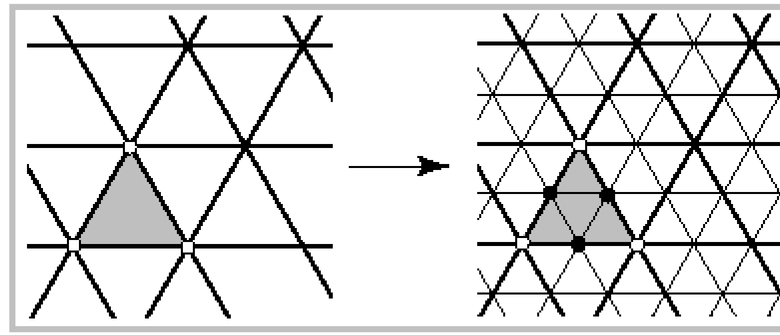
Computergraphik und Visualisierung

Face Split vs. Vertex Split

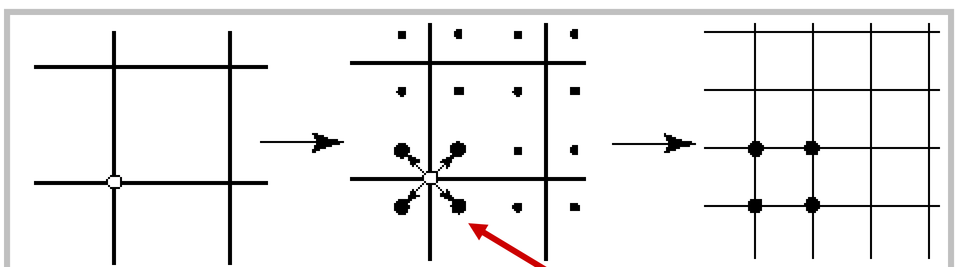


quad split

face points



triangle split



vertex split

corner points

Terminology – Types of Schemata

Cathegorization of Schemata

	face split	
	triangle mesh	quad mesh
approximating	Loop (C^2)	Catmull Clark (C^2)
interpolating	Butterfly (C^1)	Kobbelt (C^1)

	vertex split
	quad mesh
approximating	Doo Sabin, Midedge(C^1), Biquartic(C^2)

Quiz

Cathegorization

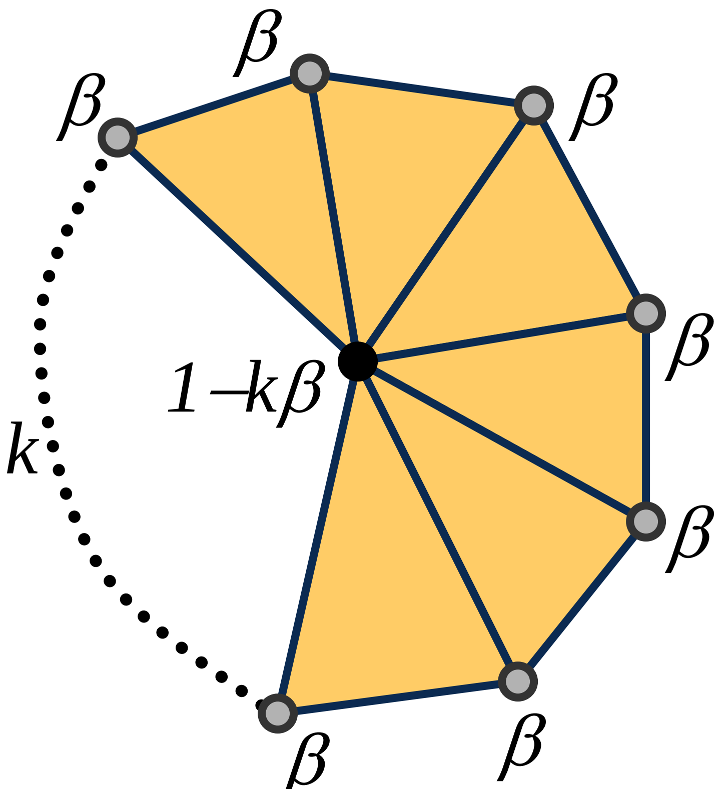
	face split	
	triangle mesh	quad mesh
approximating	Loop (C^2)	Catmull Clark (C^2)
interpolating	Butterfly (C^1)	Kobbelt (C^1)

	vertex split
	quad mesh
approximating	Doo Sabin, Midedge(C^1), Biquartic(C^2)

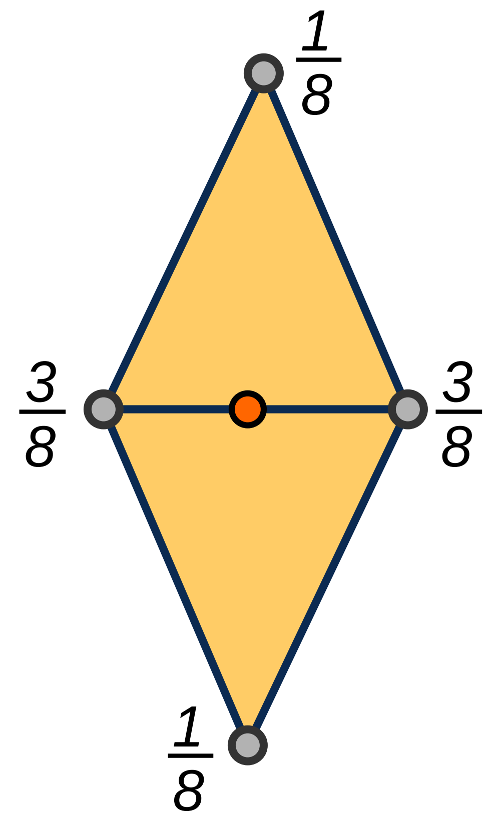
Loop Subdivision – Stencils

- Face split scheme with stencil for vertex and edge points:

$$\beta = \frac{1}{k}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{k}\right)^2\right)$$



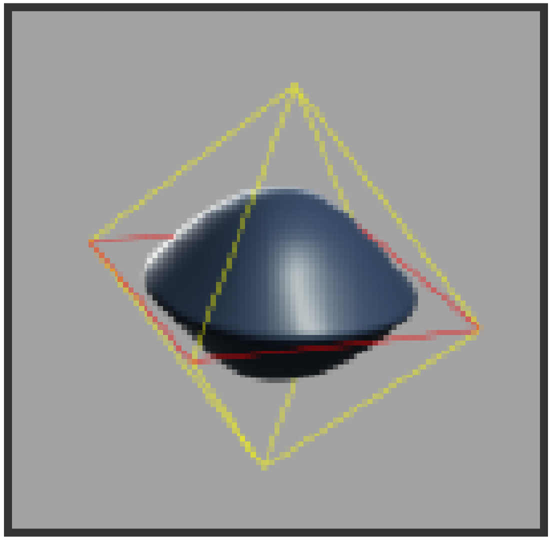
stencil for vertex (old) points

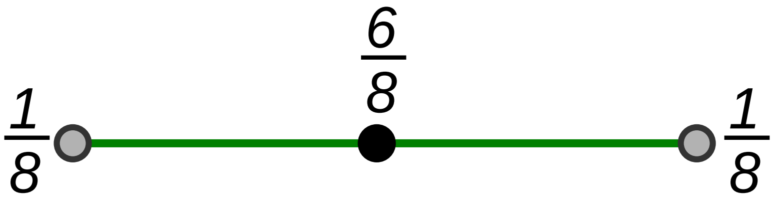


stencil for edge points

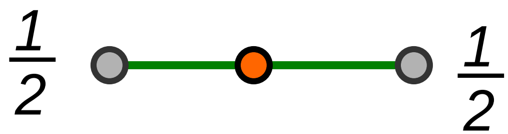
Loop Subdivision – Boundary

- Boundary stencils are used for vertices on the mesh boundary.
- The Loop scheme re-uses the cubic B-spline stencils
- sharp creases can be generated in the interior if one uses the boundary stencils





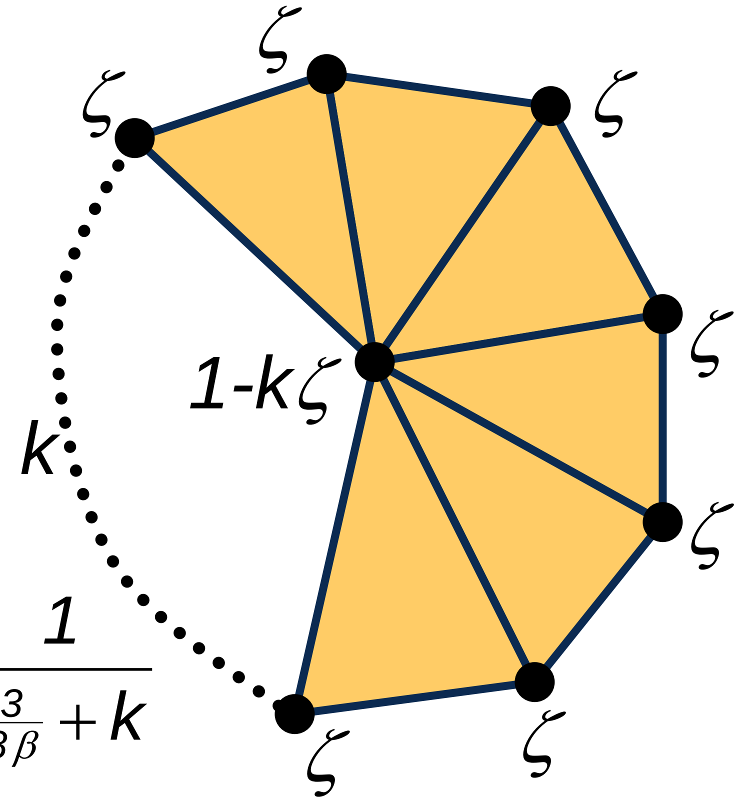
stencil for old boundary points



stencil for new boundary points

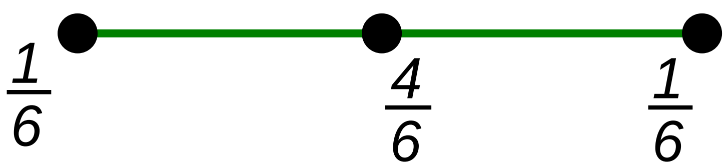
Loop Subdivision – Limit Position

- Generalization of the Eigenvalue analysis technique for curves yields also for the surface case limit stencils to compute the limit position without further subdivision



$$\zeta = \frac{1}{\frac{3}{8\beta} + k}$$

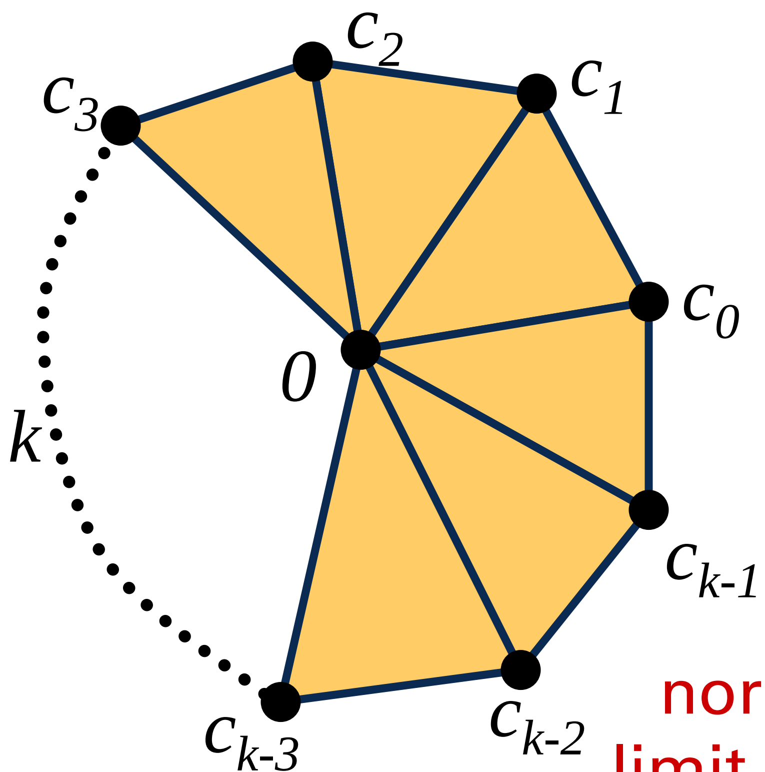
limit stencil for interior points

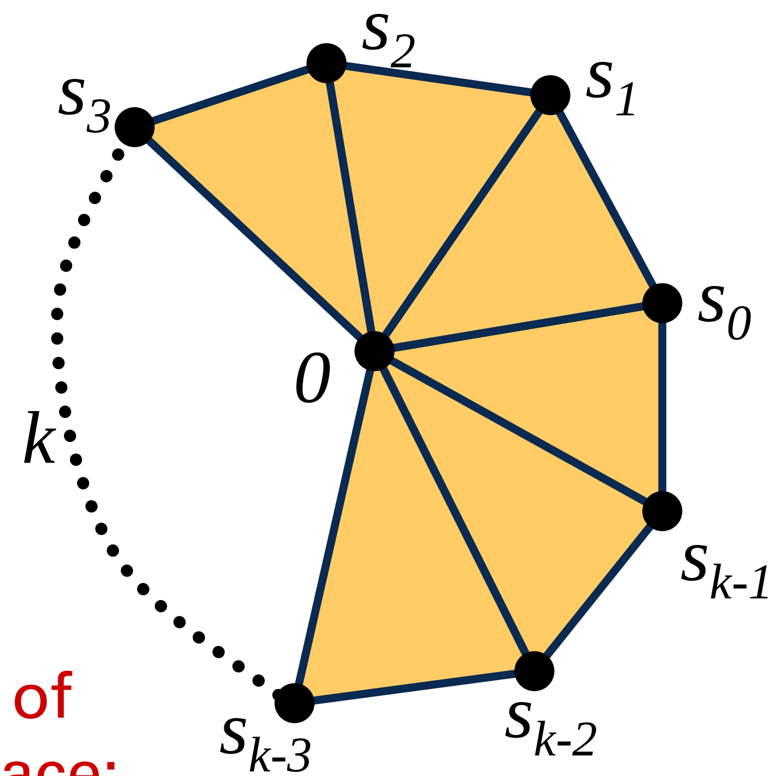


limit stencil for boundary points

Loop Subdivision – Limit Tangents

Limit stencils for tangent vectors:





normal of
limit surface:

$$c_i = \cos\frac{2\pi i}{k} \quad \Rightarrow \vec{\boldsymbol{t}}_c$$

$$\vec{\boldsymbol{n}} = \vec{\boldsymbol{t}}_c \times \vec{\boldsymbol{t}}_s$$

$$\vec{\boldsymbol{t}}_s \Leftarrow \quad s_i = \sin\frac{2\pi i}{k}$$

Loop Subdivision – Limit Tangents

Limit tangent stencil at boundary:

- along boundary:

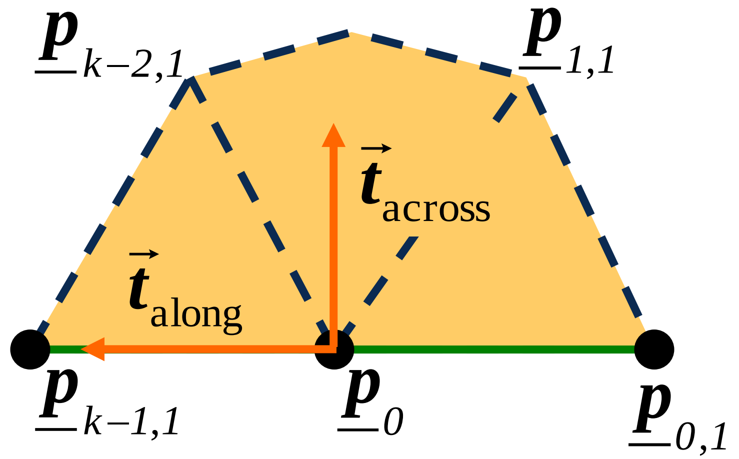
$$\vec{t}_{\text{along}} = \underline{p}_{k-1,1} - \underline{p}_{0,1}$$

- orthogonal to boundary:

$$k = 2: \vec{t}_{\text{across}} = \underline{p}_{0,1} + \underline{p}_{1,1} - 2\underline{p}_{0}$$

$$k = 3: \vec{t}_{\text{across}} = \underline{p}_{1,1} - \underline{p}_{0}$$

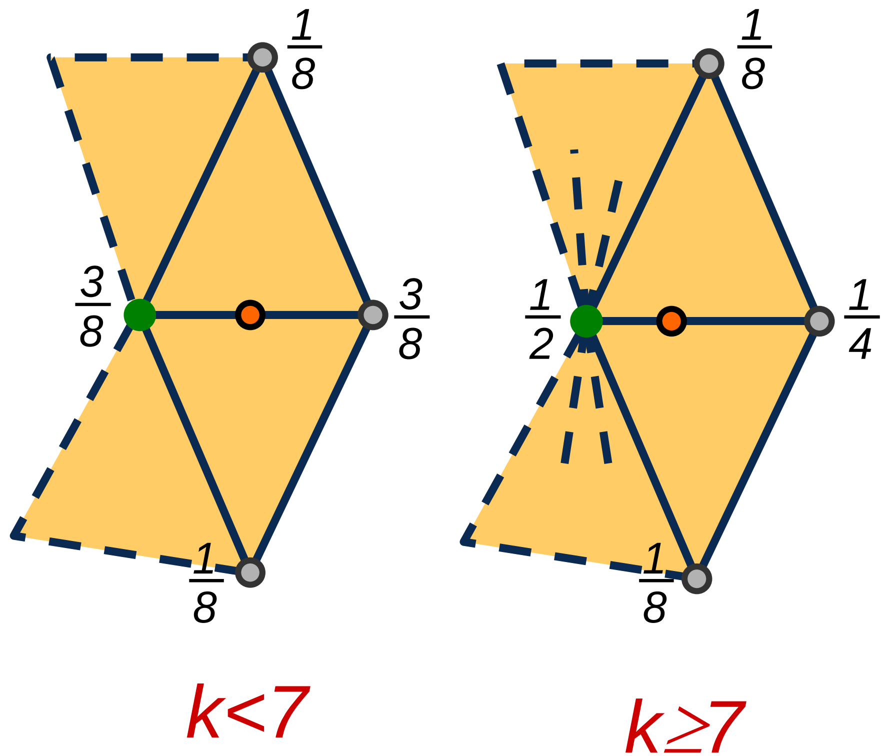
$$k > 3: \vec{t}_{\text{across}} = 2(1-\cos\theta_k)\sum_{i=1}^{k-2}\sin(i\theta_k)\underline{p}_{i,1} - \sin\theta_k\left(\underline{p}_{0,1} + \underline{p}_{k-1,1}\right) \qquad \theta_k = \frac{\pi}{k-1}$$



$$\vec{n} = \vec{t}_{\text{across}} \times \vec{t}_{\text{along}}$$

Modified Loop Subdivision

- At extraordinary boundary vertices, the limit tangent plane is not even C^1 continuous.
- As the boundary curve should not depend on the valence of the vertex, the rules for vertices on interior edges incident to extraordinary boundary vertices are modified instead.



modified stencils close to boundary

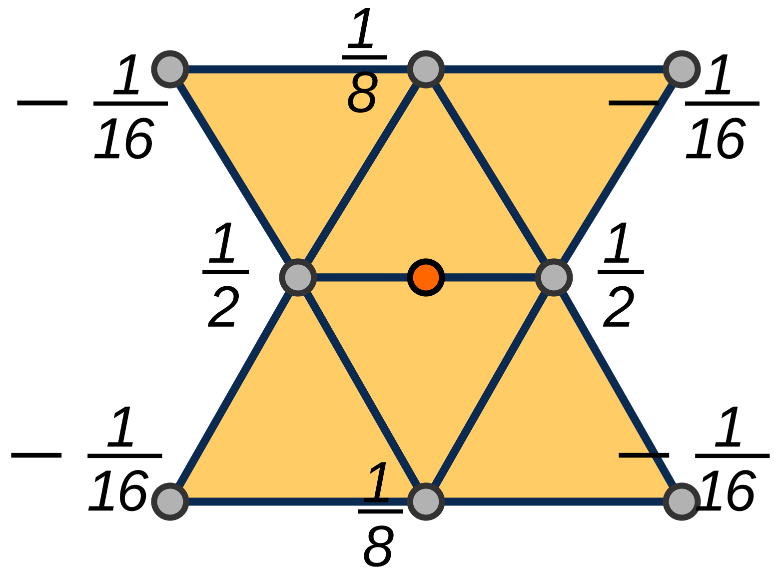
Cathegorization

	face split	
	triangle mesh	quad mesh
approximating	Loop (C^2)	Catmull Clark (C^2)
interpolating	Butterfly (C^1)	Kobbelt (C^1)

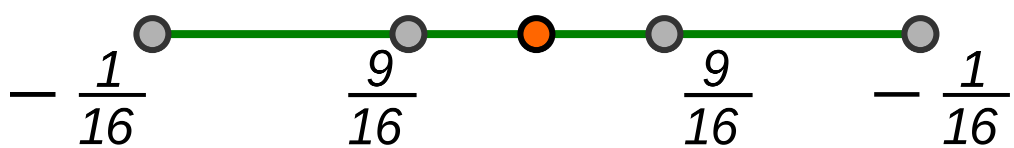
	vertex split
	quad mesh
approximating	Doo Sabin, Midedge(C^1), Biquartic(C^2)

Butterfly Subdivision

- Also a face split scheme on triangle meshes
- Positions of old vertices are preserved, such that we get an interpolating scheme
- Butterfly-stencil is used for edge points incident to two regular vertices
- On boundary edges the 4-point-rule is applied



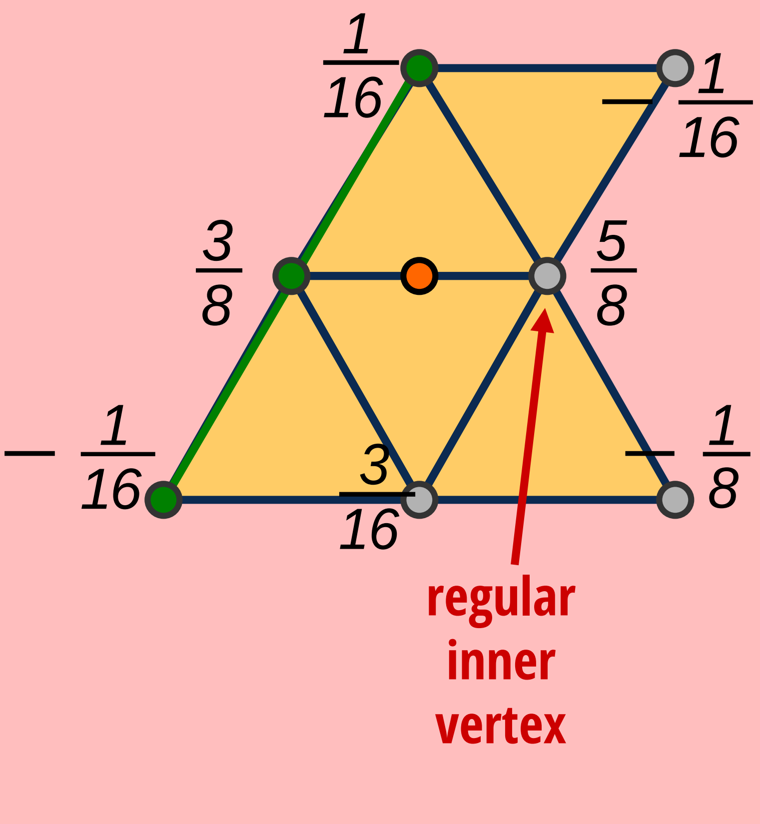
Butterfly-rule is stencil for regular edge points



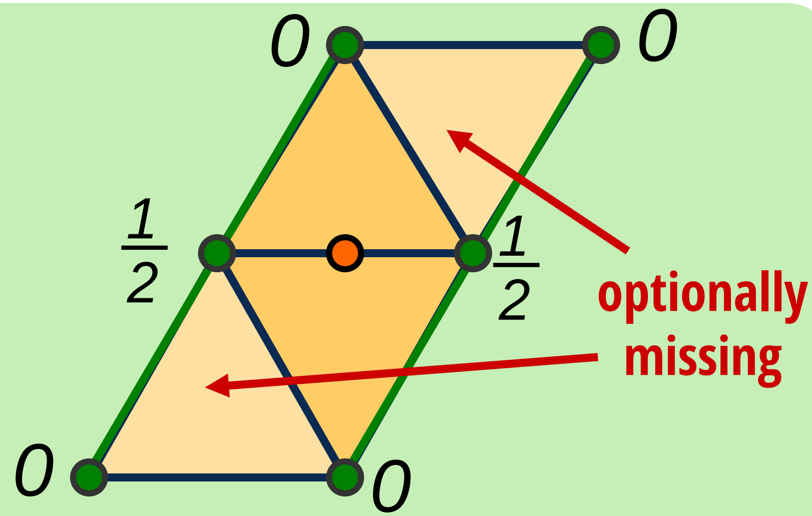
stencil for boundary edge points

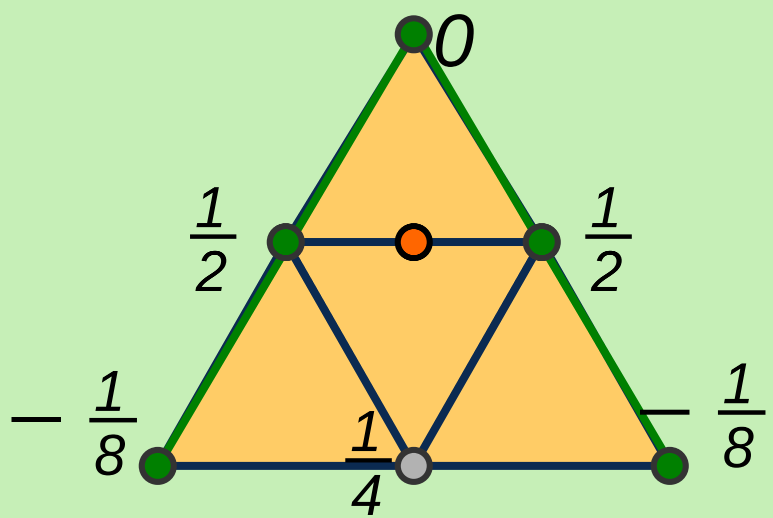
Butterfly Subdivision

regular boundary stencil:



a boundary vertex incident to edge

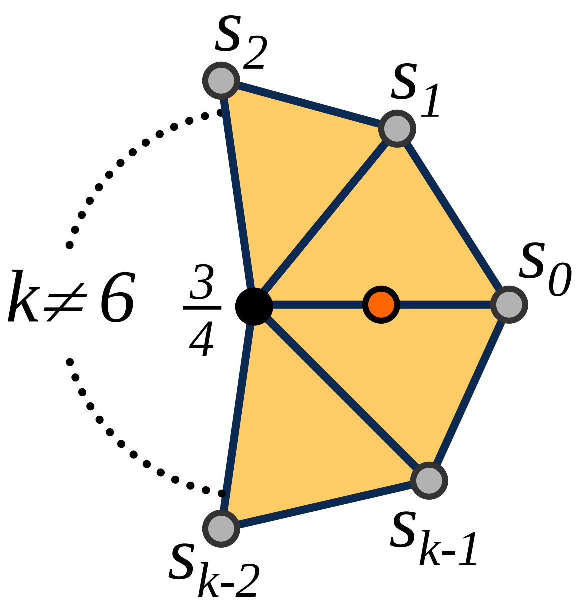




2 boundary vertices incident to edge

Butterfly Subdivision

- One incident extraordinary vertex:



$$k = 3: s_0 = \tfrac{5}{12}, s_{1/2} = -\tfrac{1}{12}$$

$$k = 4: s_0 = \tfrac{3}{8}, s_{1/3} = 0, s_2 = -\tfrac{1}{8}$$

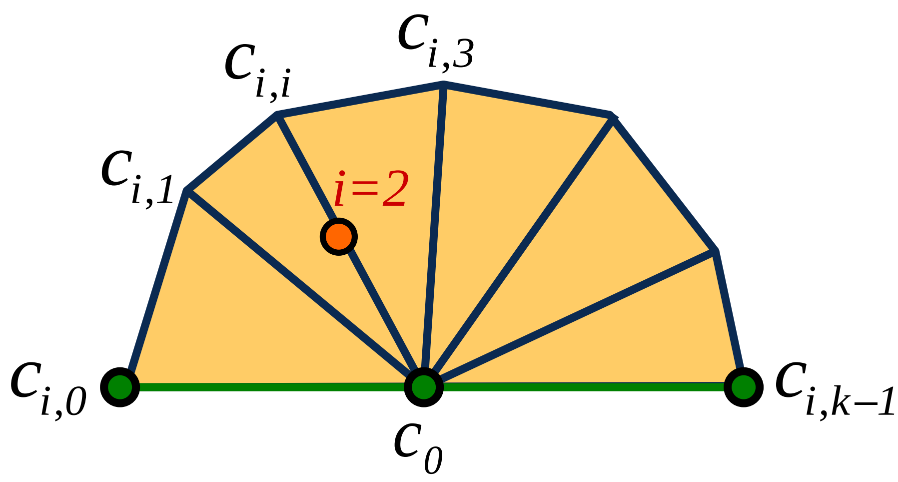
$$k > 4: s_i = \tfrac{1}{k}\left(\tfrac{1}{4} + \cos\tfrac{2\pi i}{k} + \tfrac{1}{2}\cos\tfrac{4\pi i}{k}\right)$$

- In case of two incident extraordinary vertices average result of applying rule twice for each vertex

Butterfly Subdivision

Extraordinary boundary rule:

- use always if boundary vertex has valence unequal to 4



- $$c_0(i,k) = 1 - \frac{\sin(\theta_k)\sin(i\theta_k)}{(k-1)(1-\cos(\theta_k))}$$

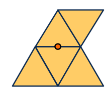
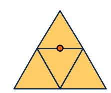
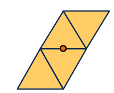
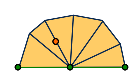
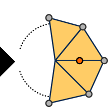
$$c_{i,0}(i,k) = c_{i,k-1}(i,k) = \frac{1}{4}\cos(i\theta_k) - \frac{1}{2}\frac{\sin(2\theta_k)\sin(2i\theta_k)}{(k-1)(\cos(\theta_k)-\cos(2\theta_k))}$$

$$c_{i,j}(i,j,k) = \frac{\sin(i\theta_k)\sin(j\theta_k) + \frac{1}{2}\sin(2i\theta_k)\sin(2j\theta_k)}{k-1} \qquad \theta_k = \frac{\pi}{k-1}$$

- In case of several extraordinary vertices average results of rules for each extraordinary vertex

Butterfly Subdivision

Decision tree

- Edge is boundary ➔ boundary rule
- both incident vertices are regular inner or boundary vertices
  - two inner vertices ➔ Butterfly-Regel
  - one inner vertex ➔ 
  - two boundary vertices ➔  or 
- one extraordinary vertex
  - extraordinary boundary vertex ➔ 
  - extraordinary inner vertex ➔ 
- two extraordinary vertices ➔ average rules for one extraordinary vertex

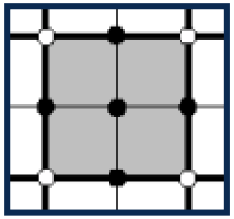
Cathegorization

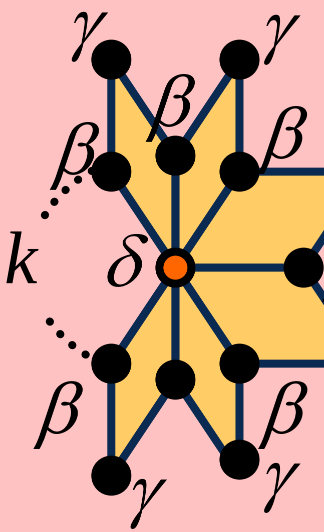
	face split	
	triangle mesh	quad mesh
approximating	Loop (C^2)	Catmull Clark (C^2)
interpolating	Butterfly (C^1)	Kobbelt (C^1)

	vertex split
	quad mesh
approximating	Doo Sabin, Midedge(C^1), Biquartic(C^2)

Catmull-Clark Subdivision

- generalizes cubic tensor product B-spline surfaces. It is based on face split of quad meshes.



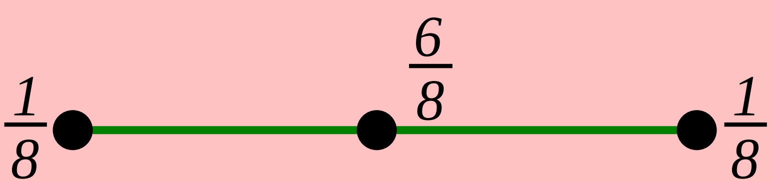


$$\beta = \frac{3}{2k^2}$$

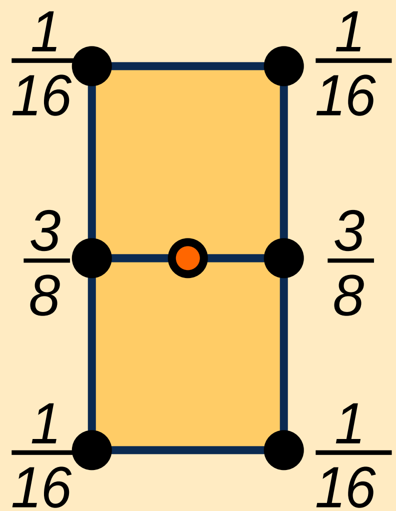
$$\gamma = \frac{1}{4k^2}$$

$$\delta = 1 - k\cdot(\beta + \gamma)$$

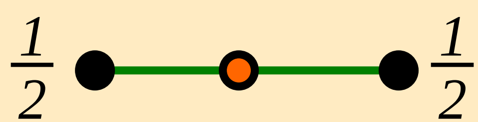
stencil for vertex points



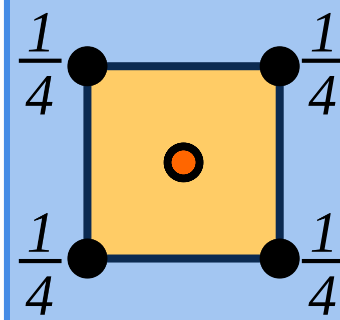
stencil for boundary vertex points



stencil for edge points



stencil for boundary edge points



stencil for face points

Catmull-Clark Subdivision

- Generalization to polygonal meshes possible.
- To achieve tangent plane continuity along boundary at extraordinary vertices again a modified rule is necessary.

- Catmull-Clark subdivision surfaces are popular in feature film production

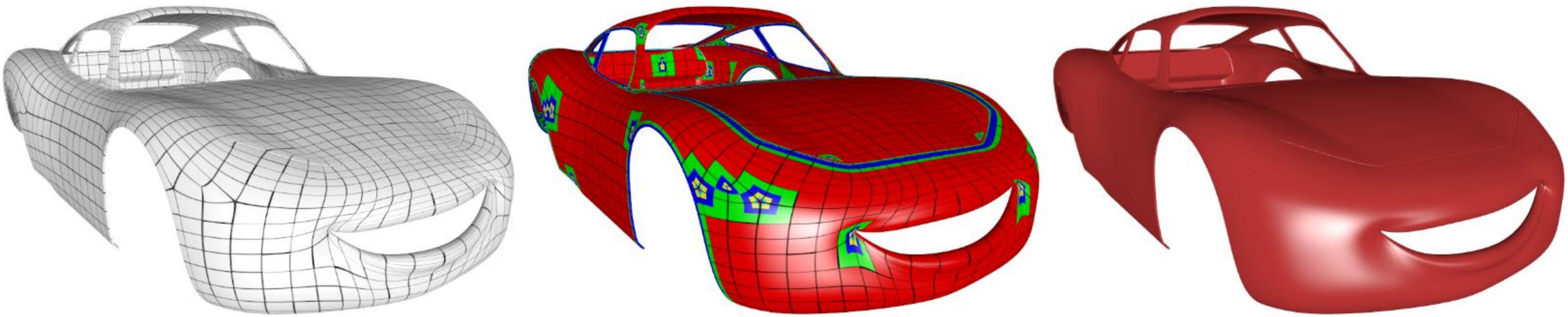


Fig. 1. Input base mesh (left), subdivision patch structure (center), and final model rendered with our method (right) © Disney/Pixar

Nießner, M., Loop, C., Meyer, M., & Derose, T. (2012). Feature-adaptive GPU rendering of Catmull-Clark subdivision surfaces. ***ACM Transactions on Graphics (TOG), 31*****(1), 6.**

Cathegorization

Computergraphik und Visualisierung

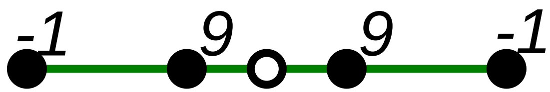
	face split	
	triangle mesh	quad mesh
approximating	Loop (C^2)	Catmull Clark (C^2)
interpolating	Butterfly (C^1)	Kobbelt (C^1)

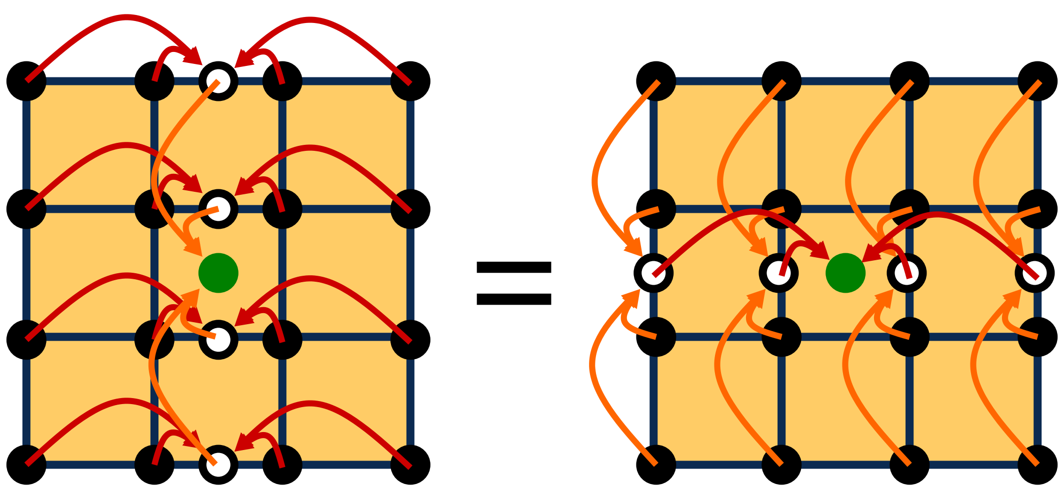
	vertex split
	quad mesh
approximating	Doo Sabin, Midedge(C^1), Biquartic(C^2)

Kobbelt's Subdivision Scheme

- Interpolating scheme on quad meshes
- tensor product of 4-point scheme in regular case

edge stencil for boundary edge or sharp crease



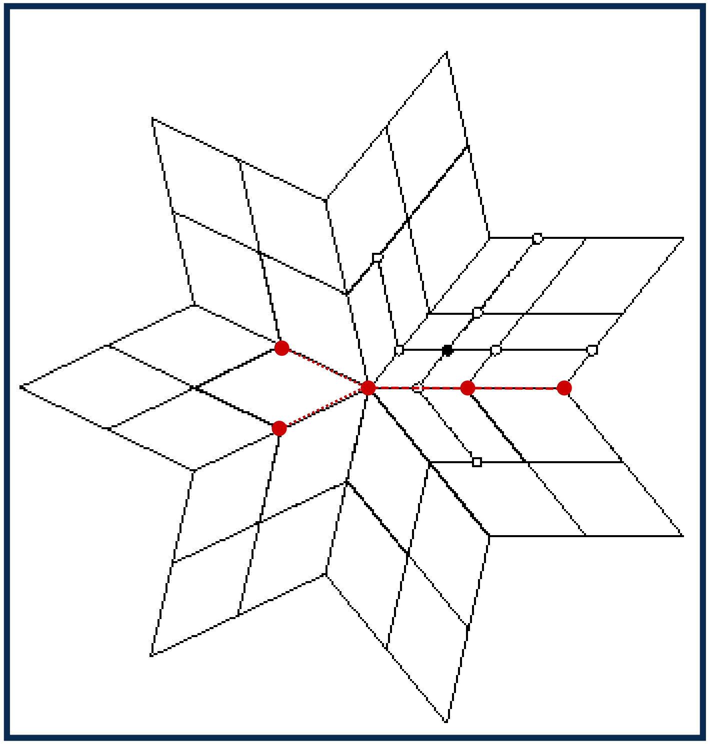


$$= \begin{bmatrix} 1 & -9 & -9 & 1 \\ -9 & 81 & 81 & -9 \\ -9 & 81 & 81 & -9 \\ 1 & -9 & -9 & 1 \end{bmatrix}$$

Tensor product for face points

Kobbelt's Subdivision Scheme

- Positions of vertex points are preserved for interpolation
- In the regular case the order of the tensor product does not matter.
- face points can be computed from edge points also if face is incident to extraordinary vertex in two different ways
- the stencil for edge point computation is adapted such that both ways to compute the face point yield same result



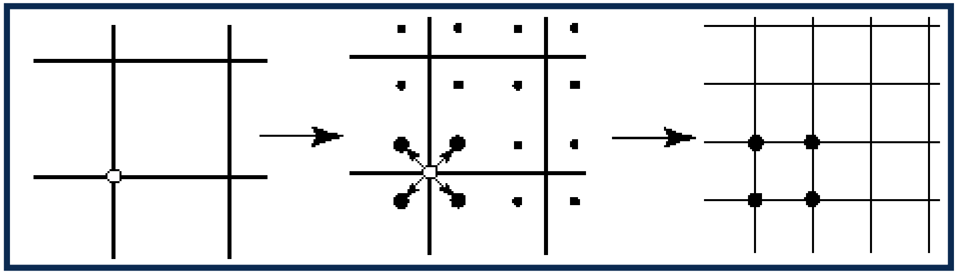
Kobbelt, L. (1996, August). Interpolatory subdivision on open quadrilateral nets with arbitrary topology. In *Computer Graphics Forum* (Vol. 15, No. 3, pp. 409-420). Edinburgh, UK: Blackwell Science Ltd.

Cathegorization

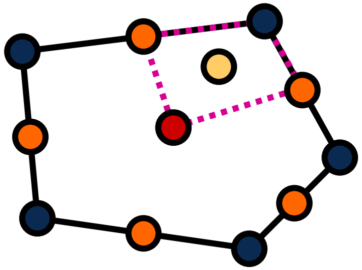
	face split	
	triangle mesh	quad mesh
approximating	Loop (C^2)	Catmull Clark (C^2)
interpolating	Butterfly (C^1)	Kobbelt (C^1)

	vertex split
	quad mesh
approximating	Doo Sabin, Midedge(C^1), Biquartic(C^2)

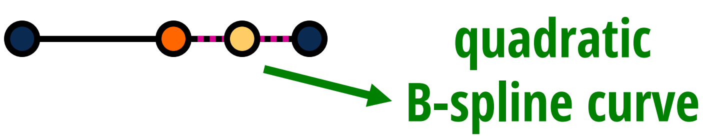
Doo-Sabin Subdivision



- Vertex split scheme for polygonal meshes.
- Simple evaluation with intermediate points:
  - edge point: center of edge
  - face point: center of face
- For each polygon corner the new point computes to the average of the following four points around the corner:
  - 1 vertex point
  - 2 edge points
  - 1 face point



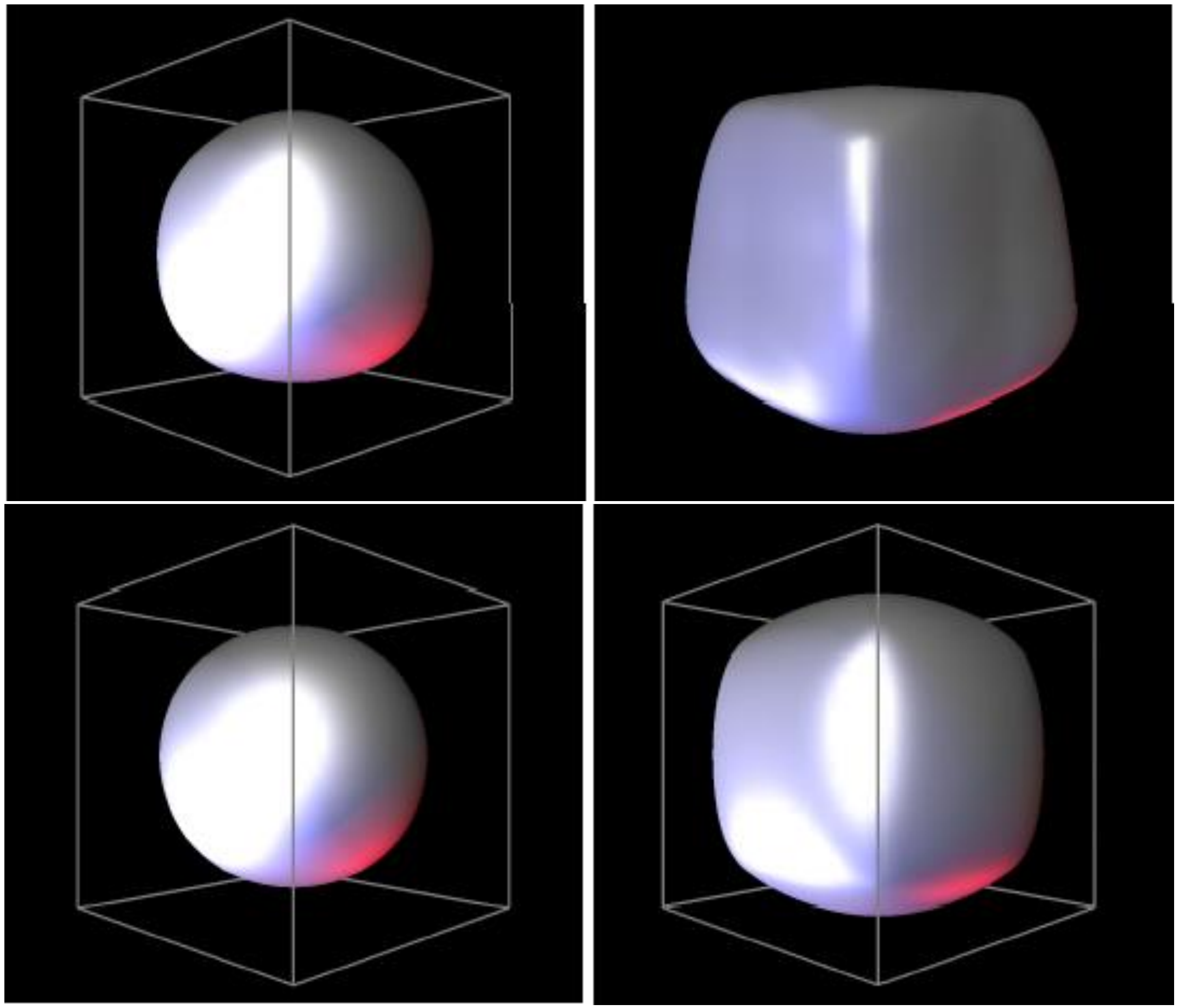
- At the boundary one only averages the following 2 points:
  - 1 vertex point
  - 1 edge point



Comparison – Cube

Loop

Butterfly



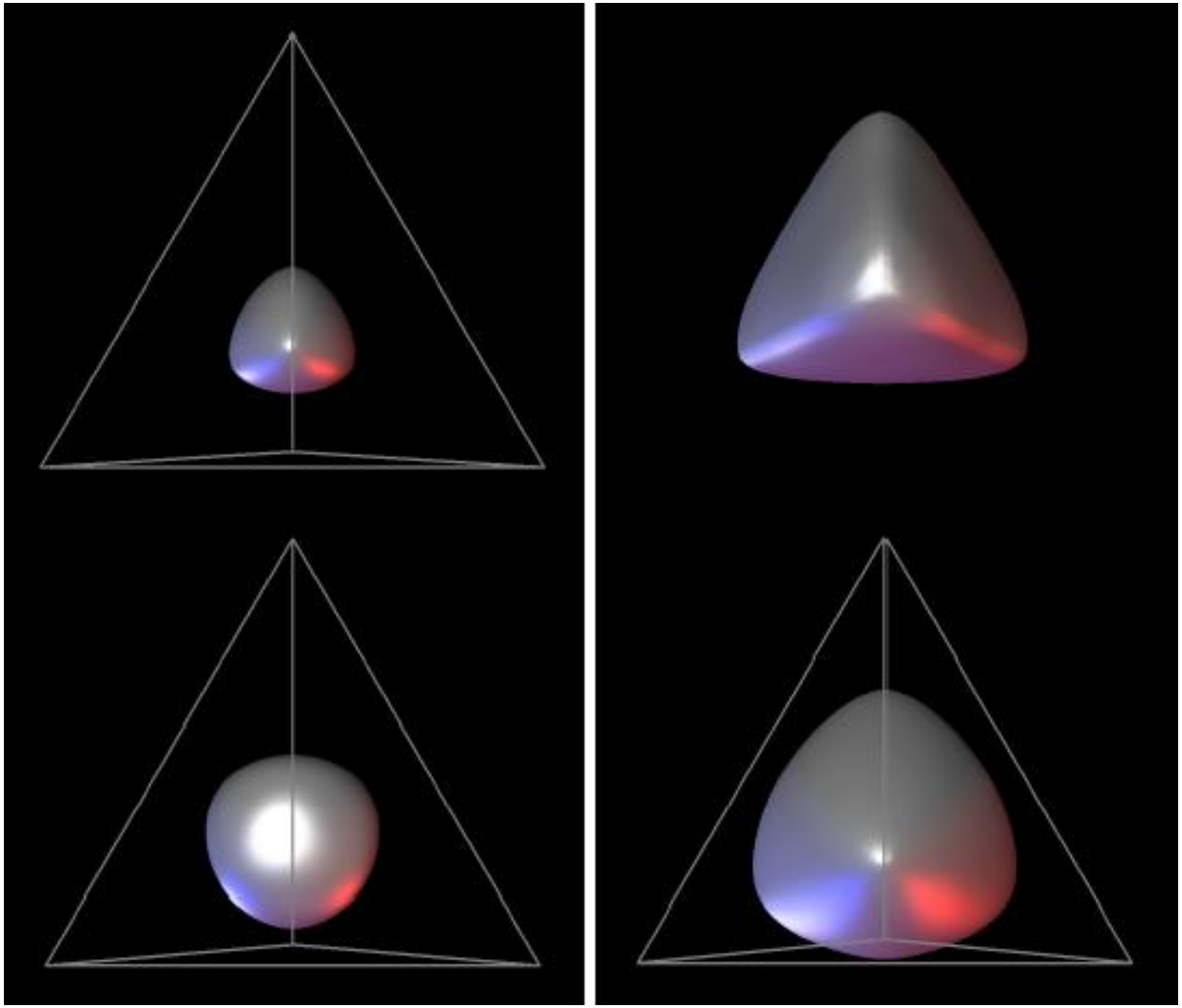
Catmull-Clark

Doo-Sabin

Comparison – Tetrahedron

Loop

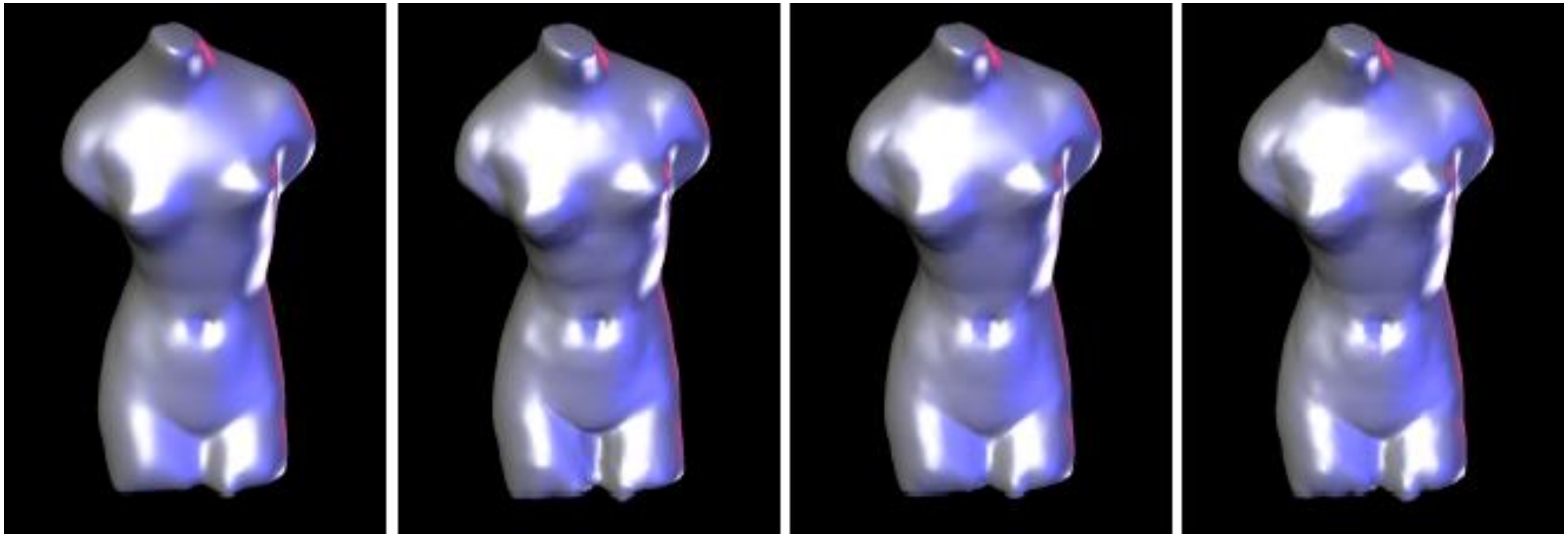
Butterfly



Catmull-Clark

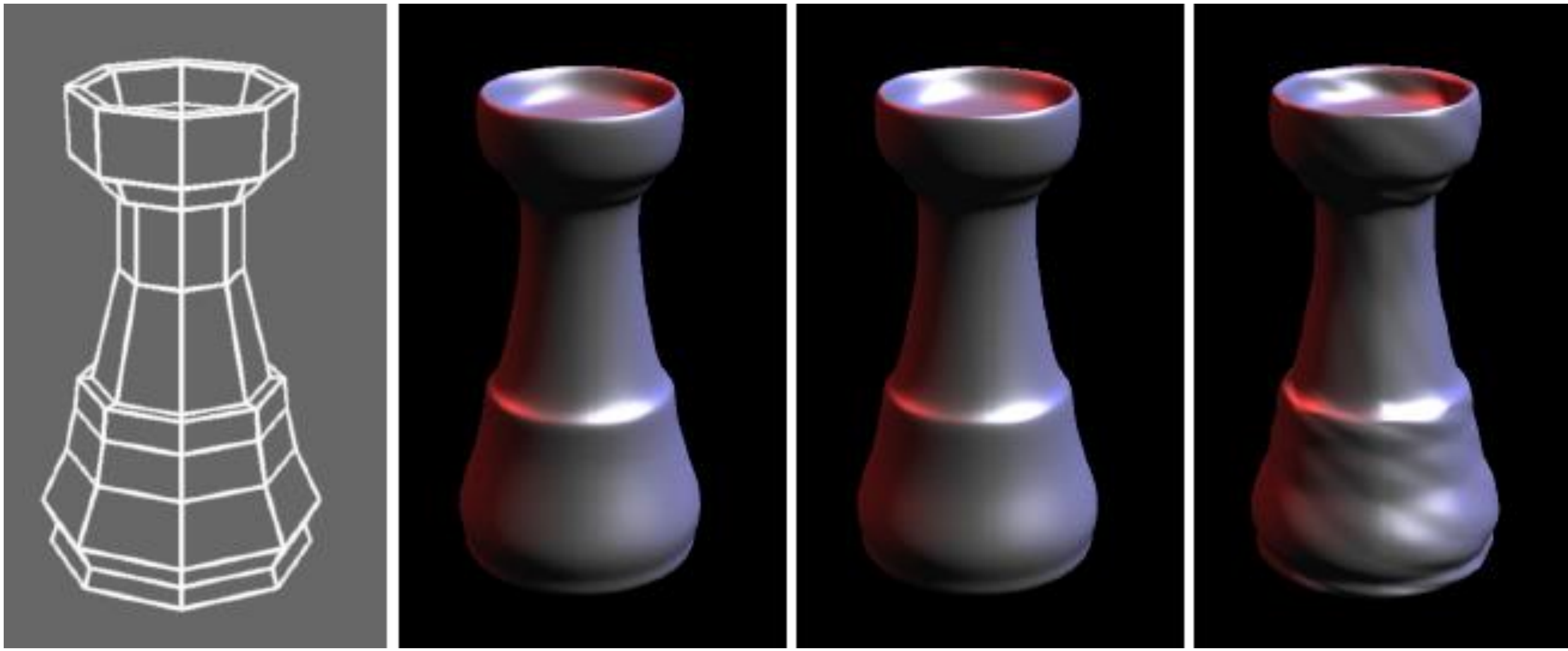
Doo-Sabin

Comparison – Torso



Loop Butterfly Catmull-Clark Doo-Sabin

Comparison – Rook



Loop

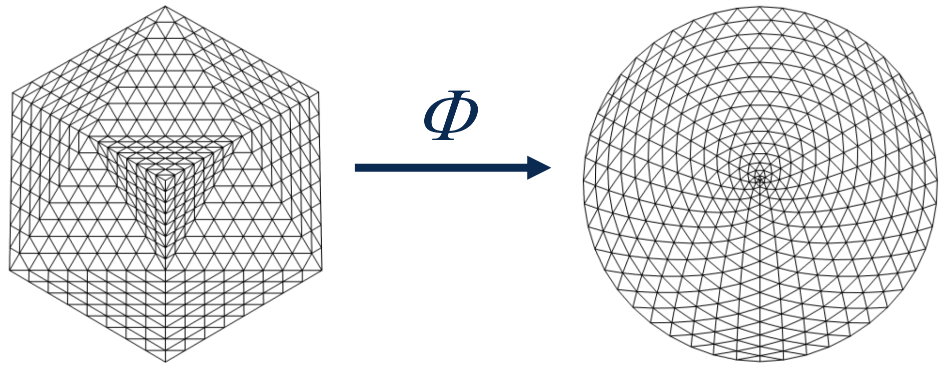
Catmull-Clark

Catmull-Clark after triangulation

Diskussion – Continuity

- For surfaces one has to also prove that the charakteristic mapping Φ of a regular 2D base mesh to the plane is bijective

subdivision

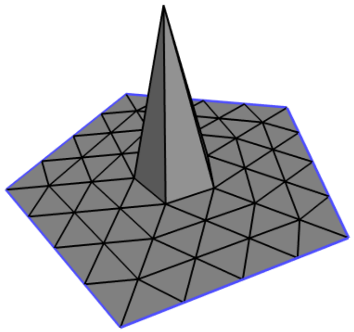


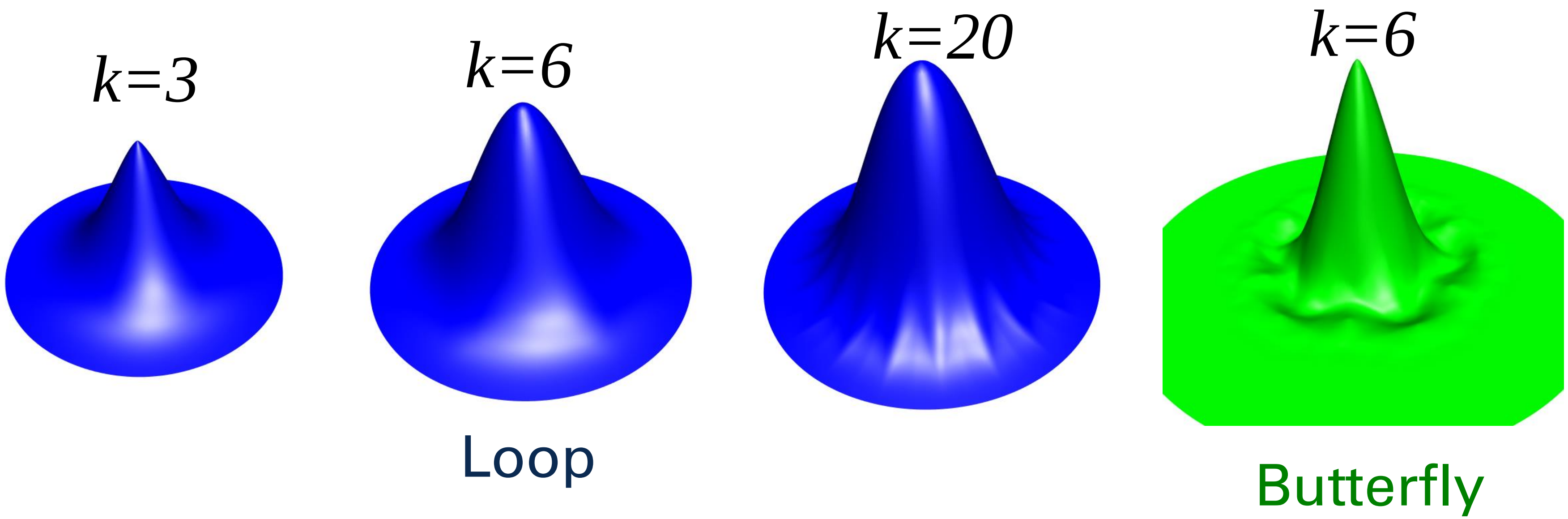
smoothing

- There does not exist a surface subdivision scheme that is C^2-continuous also at extraordinary points
- Sometimes the sufficient conditions of the z-transform do not prove the actual continuity. Then it can help to analyze sequences of pairs of subdivision steps.

Diskussion – Basis

- Subdivision of a regular mesh around an isolated regular or extraordinary point with the central vertex displaced along z by 1 yields a visualization of the influence or basis function corresponding to one vertex





- All subdivision schemes have different artefacts at extraordinary vertices

Diskussion – Interpolation

- Interpolation of the positions of a base mesh is also possible for approximating schemes if limit stencil is known
  - Let $\boldsymbol{L}$ be the square matrix, which maps the matrix of original positions $\boldsymbol{P}^0$ to the vextor of limit positions $\boldsymbol{P}^\infty$: $\boldsymbol{P}^\infty = \boldsymbol{L} \cdot \boldsymbol{P}^0$
  - Let $\boldsymbol{P}$ be the matrix with the to be interpolated points
  - Solving $\boldsymbol{P}^\infty = \boldsymbol{P}$ for $\boldsymbol{P}^0 = \boldsymbol{L}^{-1} \cdot \boldsymbol{P}$ yields base mesh locations in a way that limit subdivision surface interpolates $\boldsymbol{P}$.
- As $\boldsymbol{L}$ is sparse, $\boldsymbol{L}^{-1} \cdot \boldsymbol{P}$ can be computed iteratively in linear time
- $\boldsymbol{L}^{-1}$ is not sparse und therefore does not yield a local subdivision scheme.

demo