

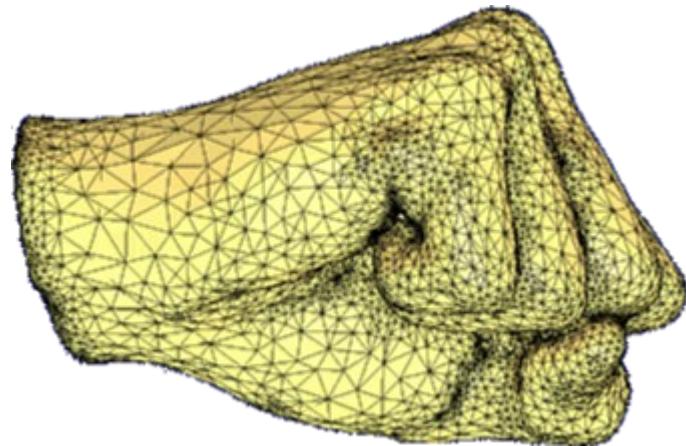
Ploskve

# Kakšna predstavitev ploskve je dobra

- Accurate
- Concise
- Intuitive specification
- Local support
- Affine invariant
- Arbitrary topology
- Guaranteed continuity
- Natural parameterization
- Efficient display
- Efficient intersections

# Najpogostejše predstavitev ploskev (surfaces)

- Poligonske mreže (polygon mesh)
- Parametrične ploskve
- Kvadrične ploskve



# Poligonske mreže

- Nabor povezanih ravninskih ploskev omejenih s poligoni
- Uporabno za škatle, modeliranje zunanjih objektov
- Računske napake so lahko poljubno majhne, vendar za ceno pomnilnika in računskega časa
- Povečave pokažejo geometrično zobčanje
- Primerni za predstavitev objektov z ravnimi površinami
- Redko uporabno za objekte s krivimi površinami
- Prostorsko potratno
- Enostavni algoritmi
- Hardwersko podprto

# Polygons - Pro

- Fast
- Arbitrary Topology
- Easy Stitching

# Polygons - Con

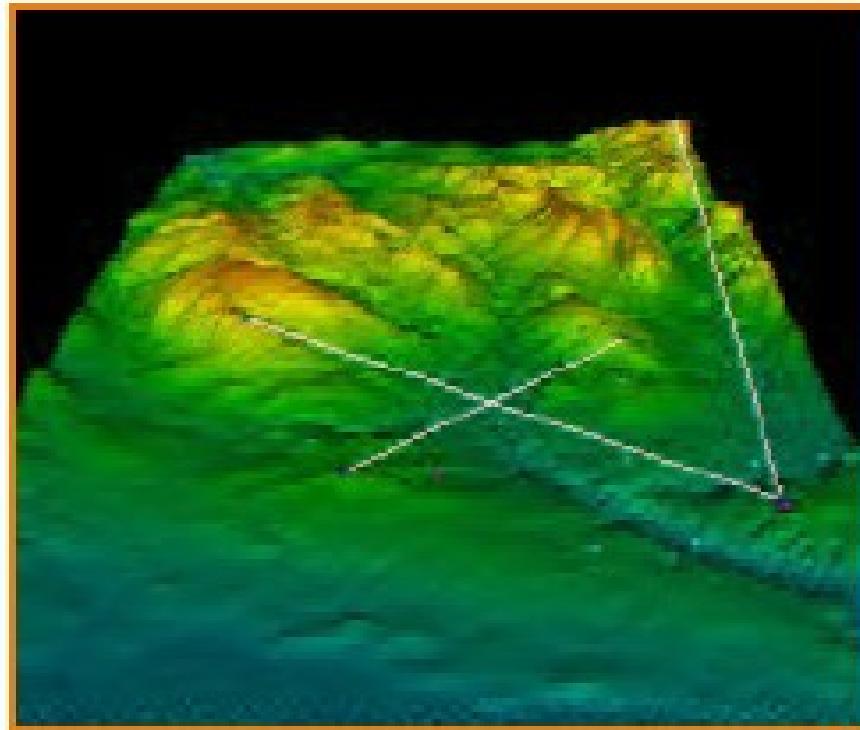
- C<sup>0</sup> Continuity
- High Counts for Complex Surfaces
- Edits Hard to do Globally

# Predstavitev ukrivljenih ploskev

- Eksplisitne (funkcije)
  - $z = f(x, y)$
- Implicit
  - $f(x, y, z) = 0$
- Parametric
  - $x = f(u, v)$
  - $y = g(u, v)$
  - $z = h(u, v)$
- Subdivision
  - $(x, y, z)$  defined by limit of recursive process

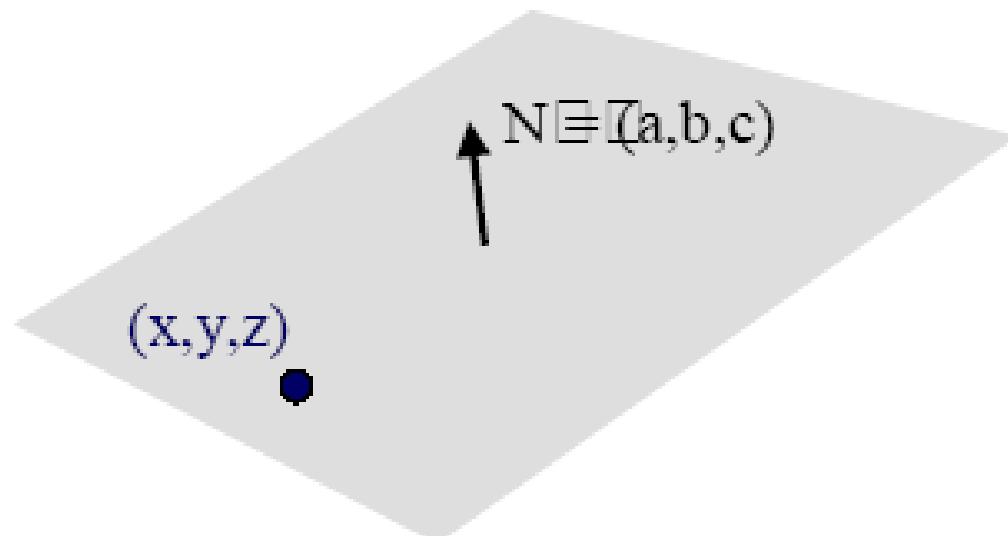
# Eksplicitne funkcije

- Boundary defined by explicit function:
  - $z = f(x, y)$



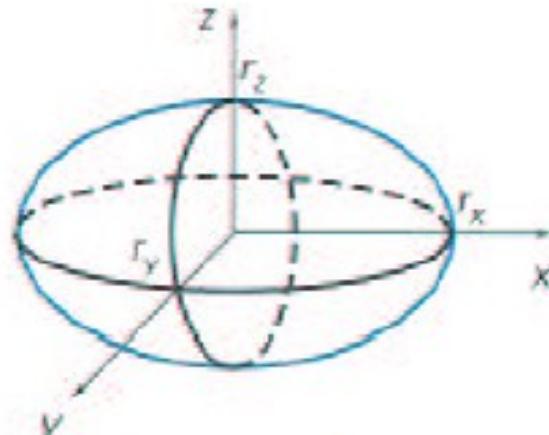
# Implicitne ploskve

- Boundary defined by implicit function:
  - $f(x, y, z) = 0$
- Example: linear (plane)
  - $ax + by + cz + d = 0$



# Implicitne ploskve

- Example: quadric
  - $f(x,y,z) = ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fxz + 2gx + 2hy + 2jz + k$
- Common quadric surfaces:
  - Sphere
  - Ellipsoid  $\rightarrow \left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 - 1 = 0$
  - Torus
  - Paraboloid
  - Hyperboloid



# Parametrične ploskve

- Boundary defined by parametric function:

$$x = f(u, v)$$

$$y = f(u, v)$$

$$z = f(u, v)$$

- Example (sphere):

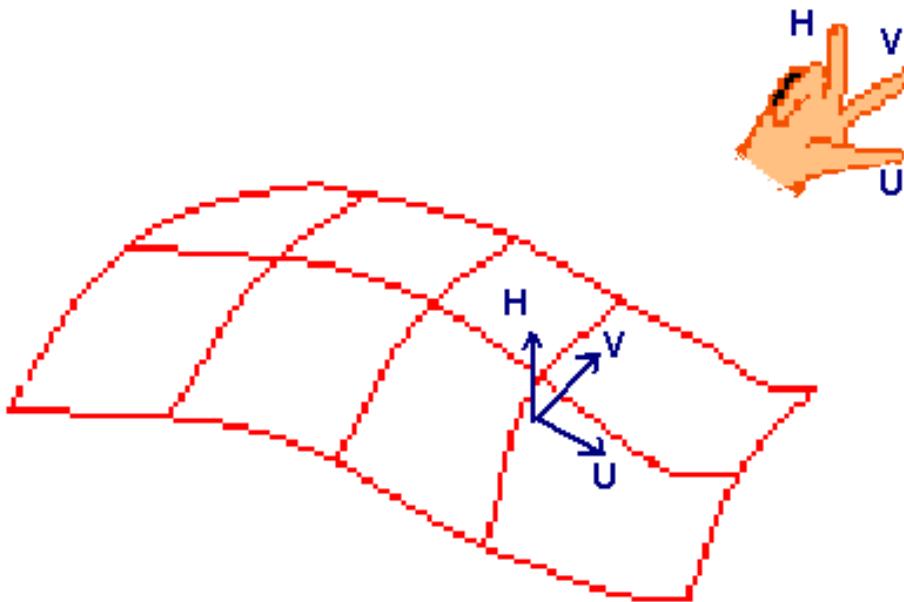
$$x = \cos(\theta)\cos(\phi)$$

$$y = \sin(\theta)\cos(\phi)$$

$$z = \sin(\phi)$$



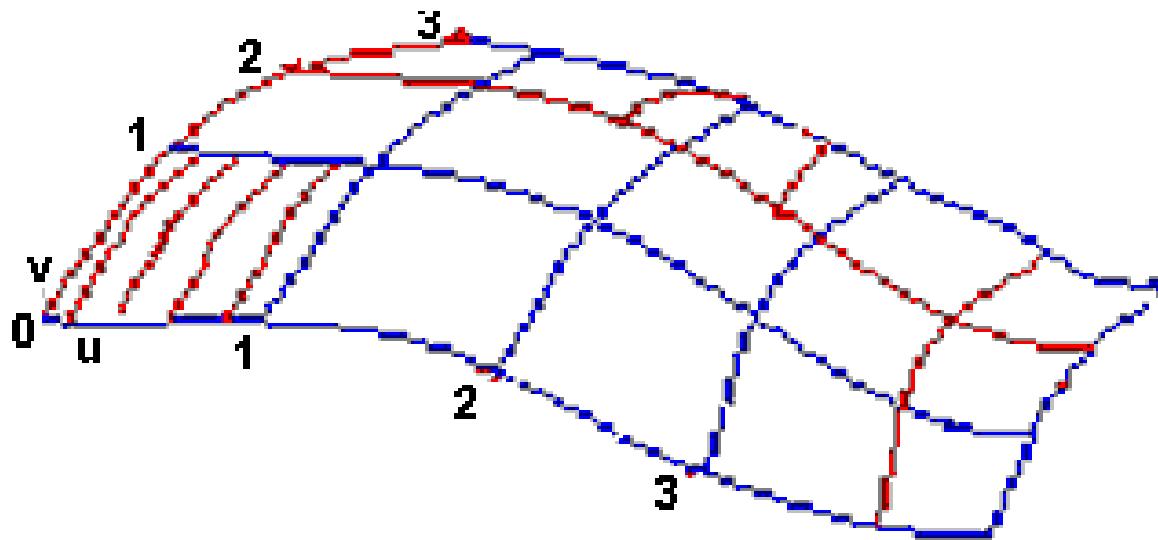
# Koordinatni sistem UV



koordinatni sistem UV je v razliko od kooordinatnega sistema XYZ uporabljiv za točke na ploskvi. Parametra u in v ustreza širini in dolžini. V primeru površine krogle je ta analogija jasna. Koncept prostora UV moramo razumeti, če želimo na površino ploskve narisati krivuljo ali če želimo nanjo nalepiti teksturo.

Usmerjenost u in v na ploskvi določa smer normale na površino. Pri tem pogosto uporabljamo "pravilo desne-roke:" desni palec kaže v smeri naraščajočega parametra u, desni kazalec kaže v smeri naraščajočega parametra v, sredinec pa kaže smer normale, pravokotne na površino, kot to prikazuje slika

# Izoparametrične krivulje

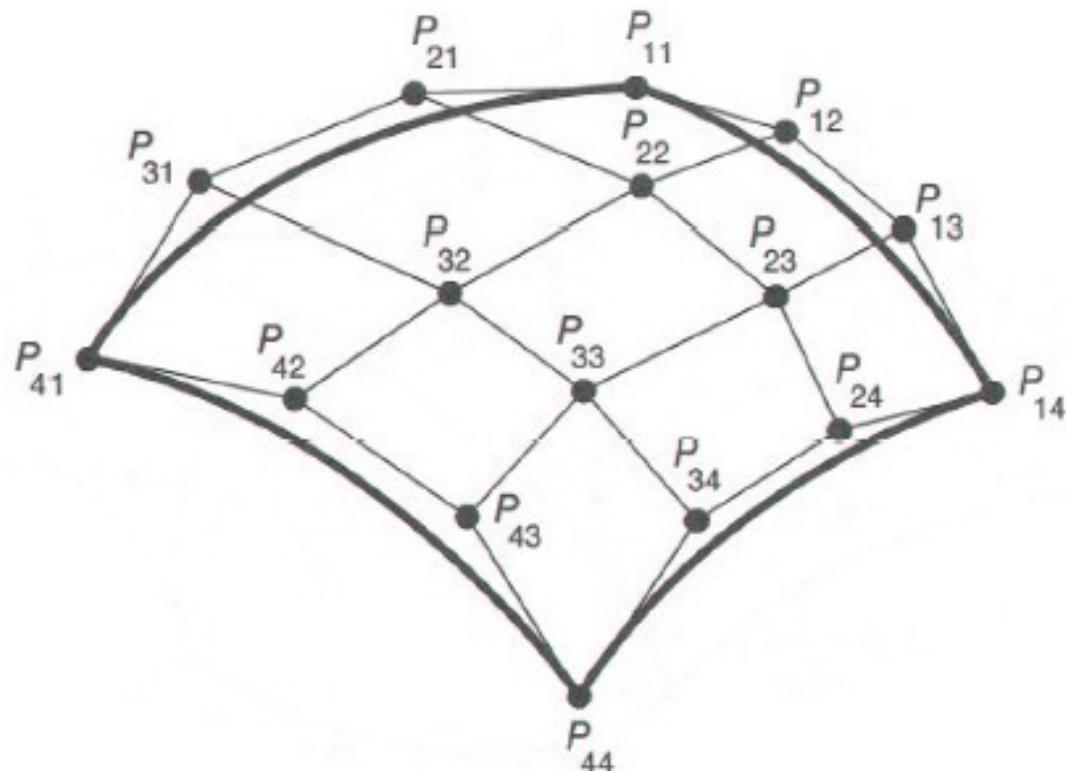


Parametrične ploskve pogosto ponazorujemo s takoimenovanimi izoparametričnimi krivuljami. Pri teh krivuljah je eden od obih parametrov konstanten, drugi pa se spominja.

# Parametrične krpe

V primeru neplanarnih ploskev pogosto uporabljam takoimenovane "krpe" (patches). Število teh je enako produktu intervalov v smereh  $u$  in  $v$ .

- Each patch is defined by blending control points

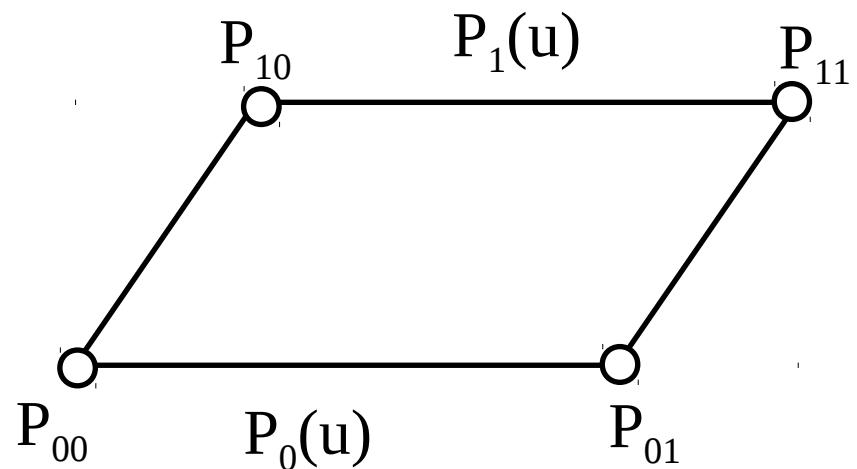


# Bilinearne krpe

- Linear in both  $\mathbf{u}$  and  $\mathbf{v}$
- $P_0(u)$  and  $P_1(u)$  are straight line segments

$$P_0(u) = (1-u) P_{00} + u P_{01}$$

$$P_1(u) = (1-u) P_{10} + u P_{11}$$



# Bilinearne krpe

Given:  $P_0(u) = (1-u) P_{00} + u P_{01}$

$$P_1(u) = (1-u) P_{10} + u P_{11}$$

- To get a parametric surface of the form:  $P(u,v) = (1-v) P_0(u) + v P_1(u)$

Results in:

$$P(u,v) = (1-v) [(1-u) P_{00} + u P_{01}] + v [(1-u) P_{10} + u P_{11}]$$

Or:

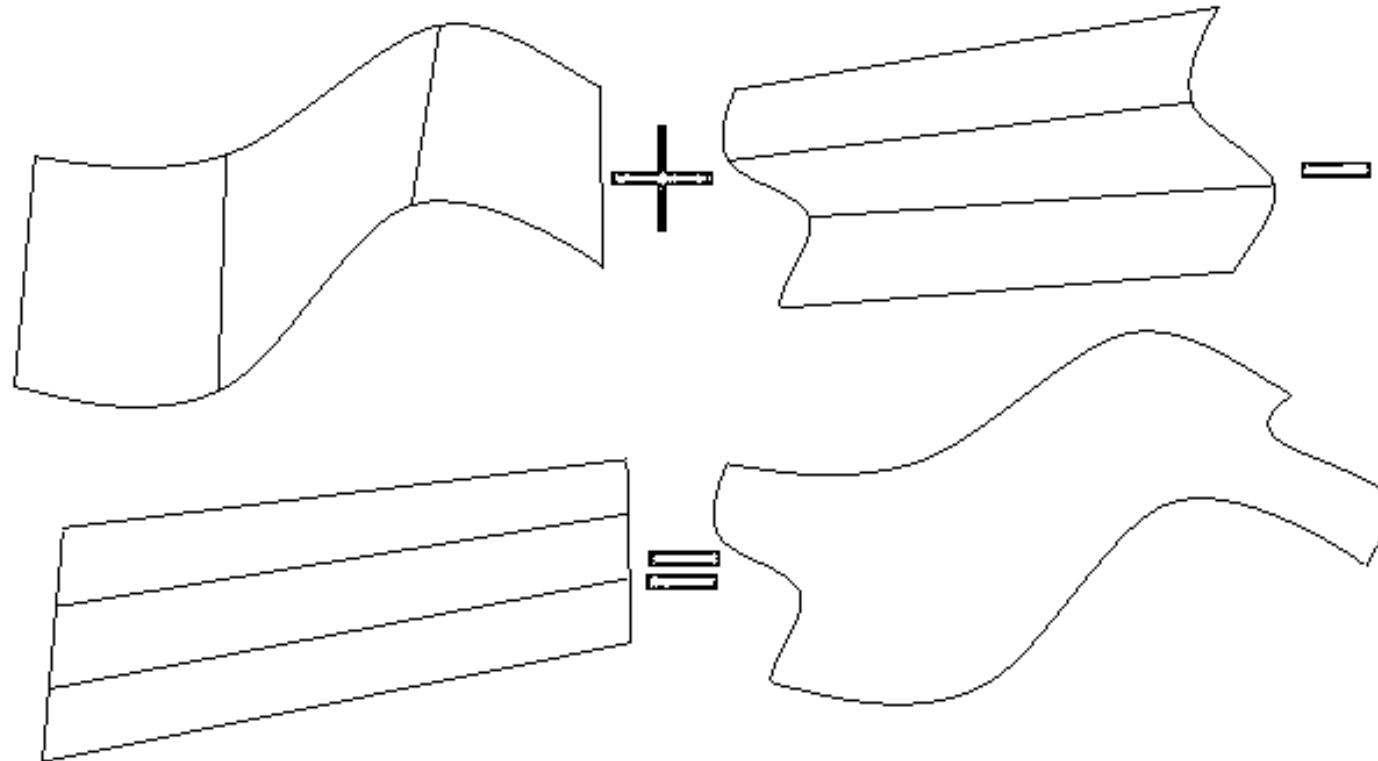
$$P(u,v) = (1-v)(1-u) P_{00} + (1-v)u P_{01} + v(1-u) P_{10} + uv P_{11}$$

# Coonsova krpa (Bilinear Blended Surfaces)

- Blended patch that interpolates between 4 boundary curves
- Combine
  - Ruled surface in **u**
  - plus**
  - Ruled surface in **v**
  - minus**
  - Bilinear surface

Coonsove krpe tvorijo pravokotno mrežo, ki jo interpoliramo med štirimi mejnimi krivuljami. Ta metoda nudi omejeno možnost zveznega prehoda med sosednjimi krpami

# Konstrukcija Coonsove krpe



# Coons Patch (Bilinear Blended Surfaces)

Ruled surface in  $u$

$$P(u,v) = (1-v) P_0(u) + v P_1(u)$$

Ruled surface in  $v$

$$P(u,v) = (1-u) P_0(v) + u P_1(v)$$

Bilinear surface

$$\begin{aligned} P(u,v) = & (1-v)(1-u) P_{00} + (1-v)u P_{01} + \\ & v(1-u) P_{10} + uv P_{11} \end{aligned}$$

# Coons Patch (Bilinear Blended Surfaces)

$$P(u,v) =$$

$$[(1-v) P_{u0}(u) + v P_{u1}(u)]$$

$$+ [ (1-u) P_{0v}(v) + u P_{1v}(v) ]$$

$$- [(1-u) (1-v) P_{00} + u(1-v) P_{01} u + P_{10} v (1-u) + P_{11} uv ]$$

Where  $P_{00} = P_{u0}(0)=P_{0v}(0)$

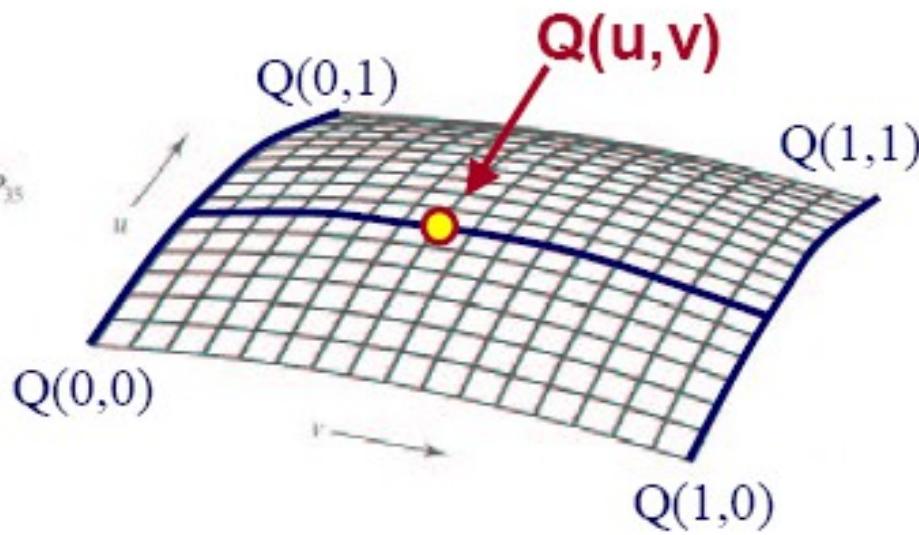
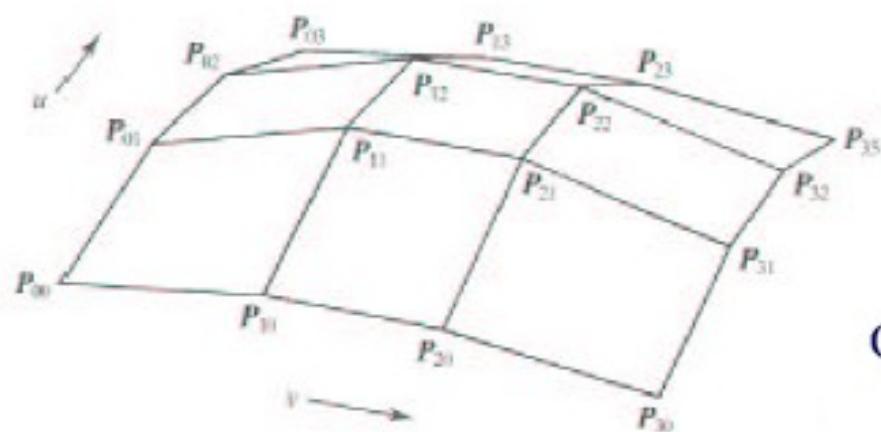
$$P_{01}= P_{u0}(1)=P_{1v}(0)$$

$$P_{10} = P_{u1}(0)=P_{0v}(1)$$

$$P_{11}= P_{u1}(1)=P_{1v}(1)$$

# Parametrične krpe

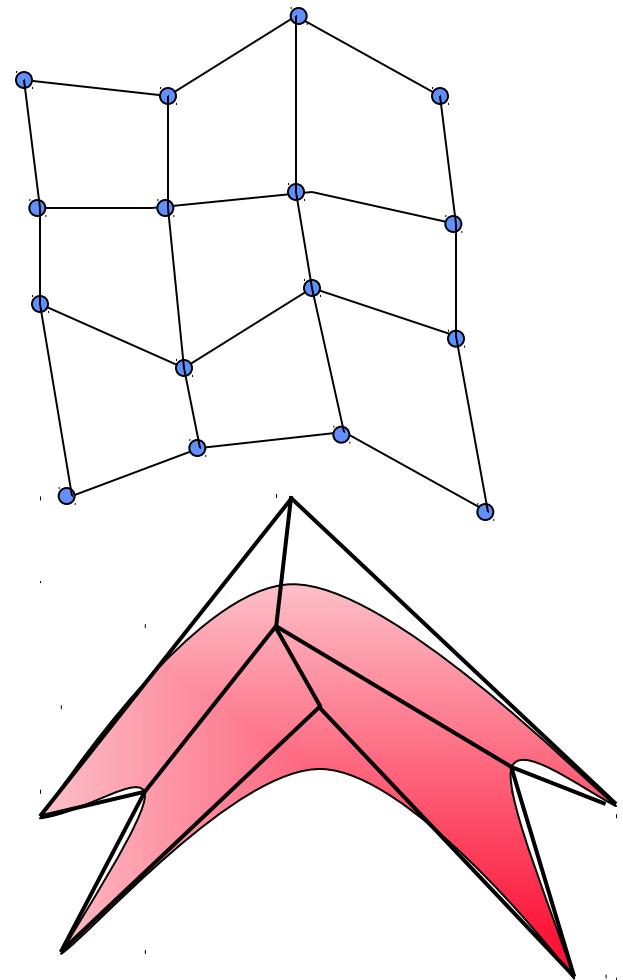
- Point  $Q(u,v)$  on the patch is the tensor product of parametric curves defined by the control points



# Tenzorski produkt

- Bezier patch
  - Tensor product of two Bezier curves
  - Product of Bernstein polynomials
$$p(s, t) = \sum_{j=1}^n \sum_{i=1}^n B_j^n(s) B_i^n(t) \mathbf{p}_{ij}$$
  - Bernstein interpolation of Bernstein polynomials
$$p(s, t) = \sum_{j=1}^n \sum_{i=1}^n (B_j^n(s) B_i^n(t)) \mathbf{p}_{ij}$$
- Works same way for B-splines

$$p(s, t) = \sum_{j=1}^n B_j^n(s) \left( \sum_{i=1}^n B_i^n(t) (\mathbf{p}_i)_j \right)$$



# Bikubične parametrične krpe

Point  $Q(u,v)$  on any patch is defined by combining control points with polynomial blending functions:

$$Q(u,v) = \mathbf{U} \mathbf{M} \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & P_{1,4} \\ P_{2,1} & P_{2,2} & P_{2,3} & P_{2,4} \\ P_{3,1} & P_{3,2} & P_{3,3} & P_{3,4} \\ P_{4,1} & P_{4,2} & P_{4,3} & P_{4,4} \end{bmatrix} \mathbf{M}^T \mathbf{V}^T$$
$$\mathbf{U} = [u^3 \quad u^2 \quad u \quad 1] \qquad \qquad \mathbf{V} = [v^3 \quad v^2 \quad v \quad 1]$$

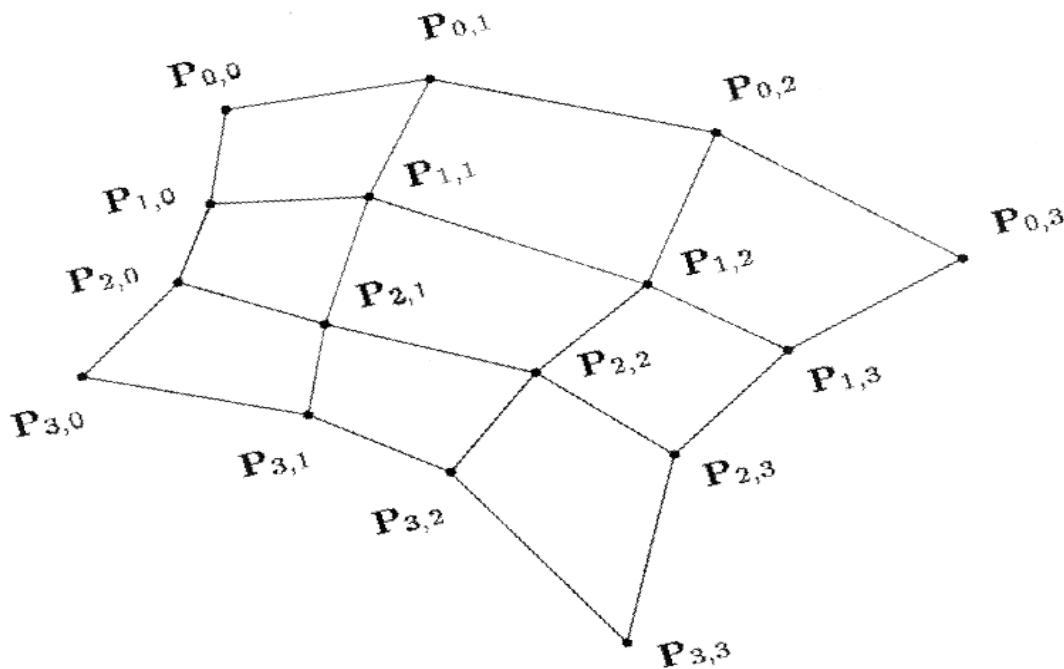
Where  $\mathbf{M}$  is a matrix describing the blending functions for a parametric cubic curve (e.g., Bezier, B-spline, etc.)

# Bézierove krpe



# Bézierove krpe

The following is an example of the control point lattice:



# Bézierove krpe

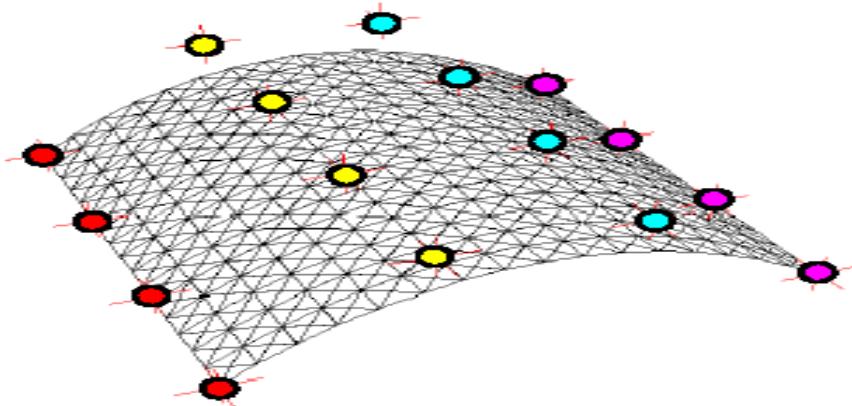
- These patches are based on Bézier curves
- The equation for a cubic patch is:

$$C(u, v) = \sum_{i=0..3} \sum_{j=0..3} P_{ij} B_i(u) B_j(v)$$

- That is, it is a lattice of Bézier curves
  - Specifically, 4 horizontal curves (u direction) and 4 vertical curves (v direction)
- These 8 curves share control points such that there are a total of 16 control points in the lattice

# Bézierove krpe

- The surface that is created is based on the control points:

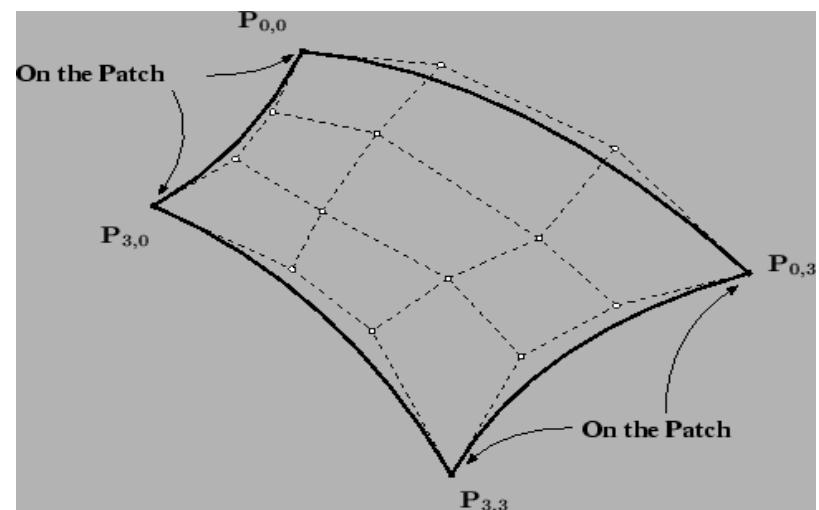
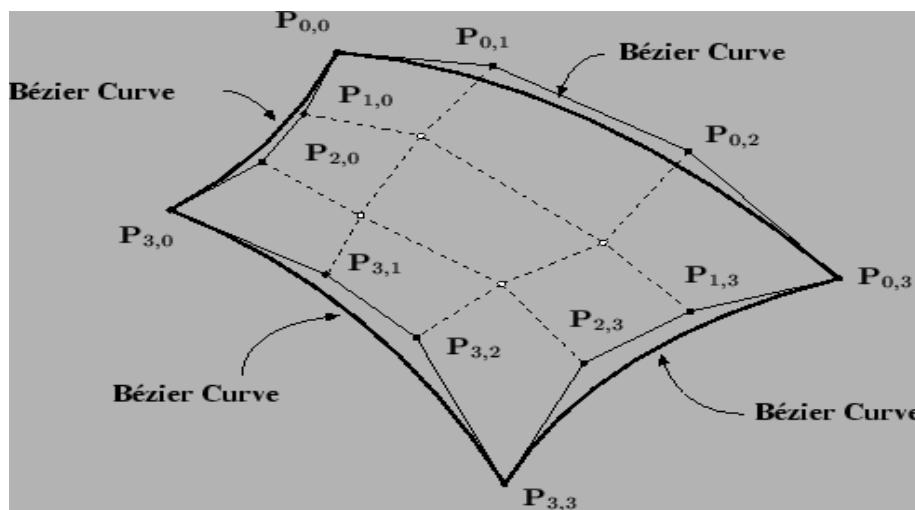


Uniformly polygonised patch  
and Bezier control points



# Bézierove krpe

- The surface is based on Bézier curves
- Thus, the surface only matches the control points at the 4 corners



# Bézierove krpe

- Adjusting the control points changes the shape of the surface:
- See the applet on:
  - [http://www.nbb.cornell.edu/neurobio/land/OldStudentProjects/cs490-96to97/anson/  
BezierPatchApplet/](http://www.nbb.cornell.edu/neurobio/land/OldStudentProjects/cs490-96to97/anson/BezierPatchApplet/)
- All the same issue occur with patches as with curves
  - Control points have global control
  - Increasing the number of control points will increase the degree of the patch

Demo

# Bézierove krpe

- Thus, we handle large curvy surfaces the same was as with curves: join patches together
- See the applet on:
  - <http://www1.ics.uci.edu/~frost/unex/JavaGraphics/course/Bezpatch.html>
- Recall the continuity constrains:
  - C<sub>0</sub> continuity: edges must match
  - C<sub>1</sub> continuity: colinear control points
- More difficult near joins of 4 patches

Demo

# Krpe z b-zlepki (B-Spline Patches)

- B-Spline patches are similar to Bézier patches except that they are based on B-Spline curves
- This implies that:
  - You can have **any number of control points** in either the u or v direction (4x4, 8x12, etc.) and still maintain piecewise C<sup>2</sup> cubic patches
  - With Uniform patches, the patch is will **no go through the control points**

# Krpe z b-zlepki

Creating a spline surface involves taking the product of the same spline blending functions used for spline curves as follows

$$P(u,v) = \sum_{i=0}^{n_i} \sum_{j=0}^{n_j} P_{i,j} N_{i,t}(u) N_{j,t}(v)$$

$u = 0 \rightarrow n_i - t_i + 2$   
 $v = 0 \rightarrow n_j - t_j + 2$   
 $t_i, t_j$  is the degree in each direction  
N are the blending functions as for  
spline curves

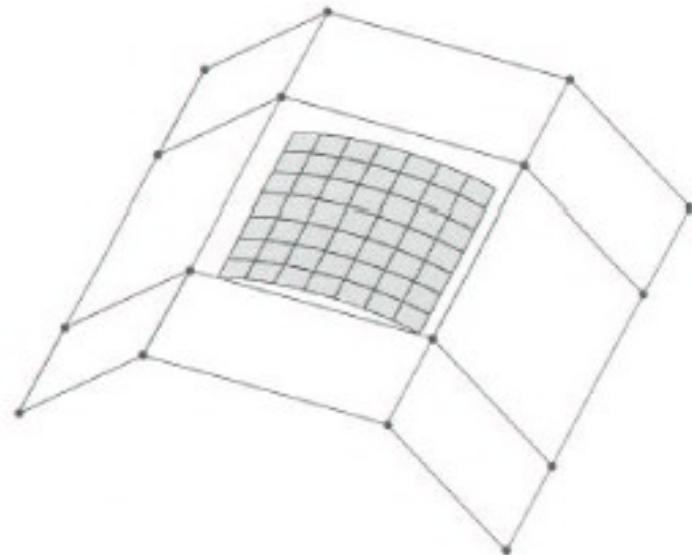
where the control points form a 2D array  $P_{ij}$ . Most of the properties of the spline curve also apply to spline surfaces. For example

- The surface passes through the end (corner) points
- The surface lies within the convex hull of the control points
- The smoothness of the surface can be controlled and this can be done independently in both directions.
- The resolution of the surface can be controlled and this can be different in each direction.

# Krpe z b-zlepki

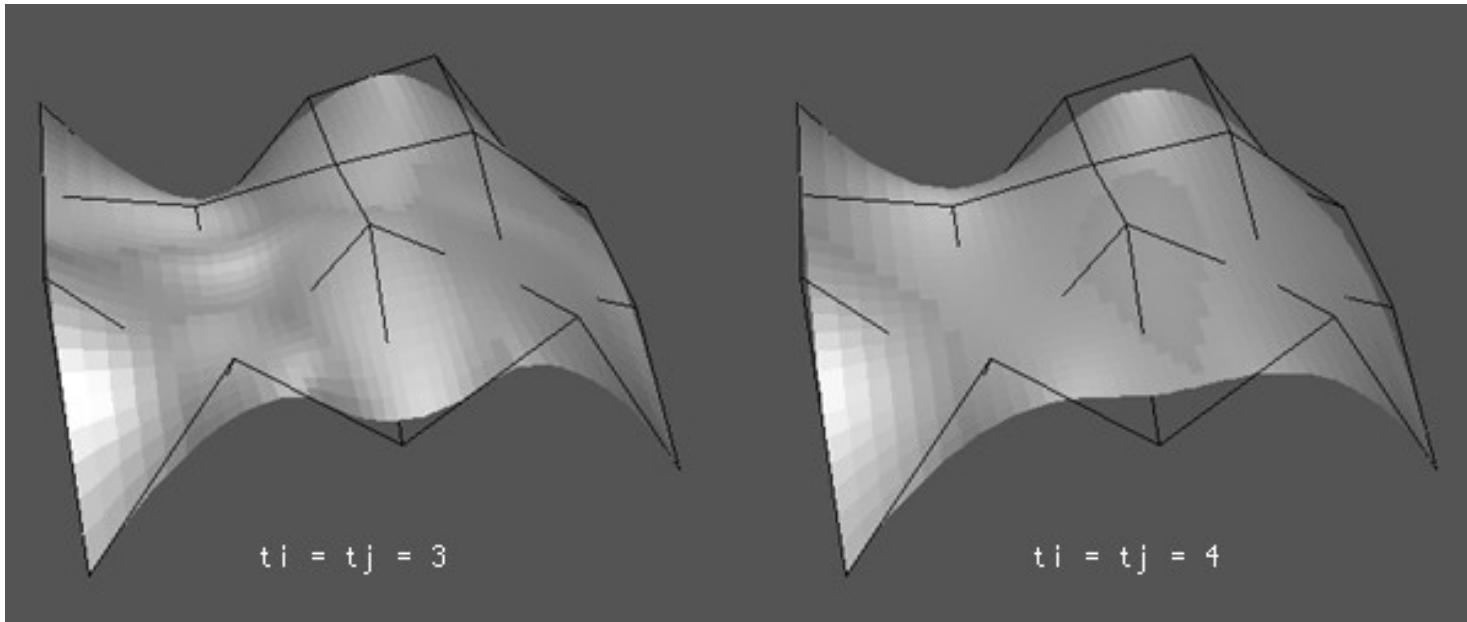
$$Q(u, v) = \mathbf{U} \mathbf{M}_{\text{B-Spline}} \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & P_{1,4} \\ P_{2,1} & P_{2,2} & P_{2,3} & P_{2,4} \\ P_{3,1} & P_{3,2} & P_{3,3} & P_{3,4} \\ P_{4,1} & P_{4,2} & P_{4,3} & P_{4,4} \end{bmatrix} \mathbf{M}_{\text{B-Spline}}^T \mathbf{V}$$

$$\mathbf{M}_{\text{B-Spline}} = \begin{bmatrix} -\frac{1}{6} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{6} \\ \frac{1}{2} & -1 & \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \end{bmatrix}$$

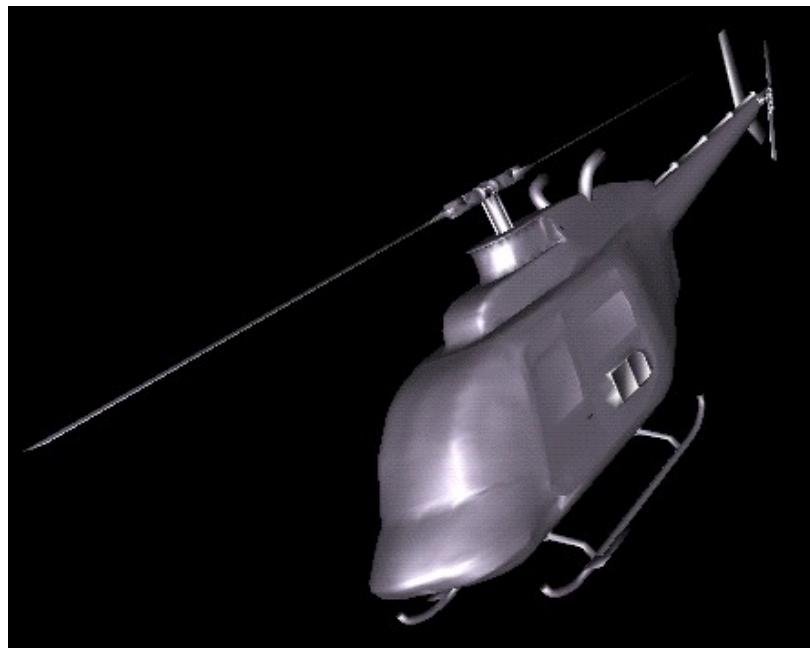
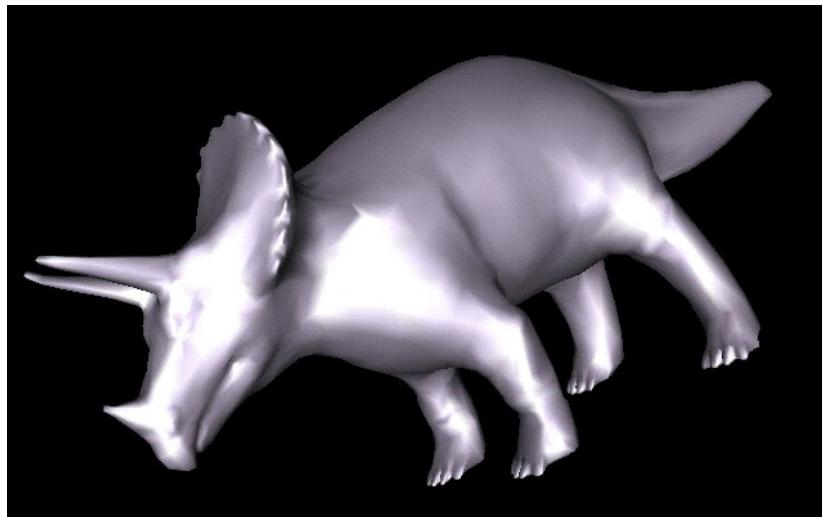
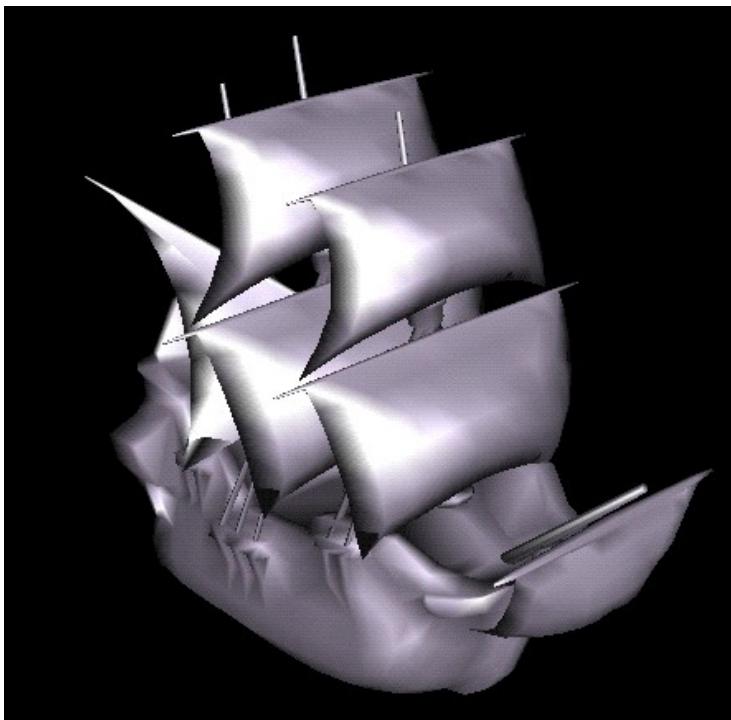


# Krpe z b-zlepki

As an example the following show the surface given a 4x5 matrix of control points, degree 3 in both directions, the surface is sampled on a 30x40 grid.



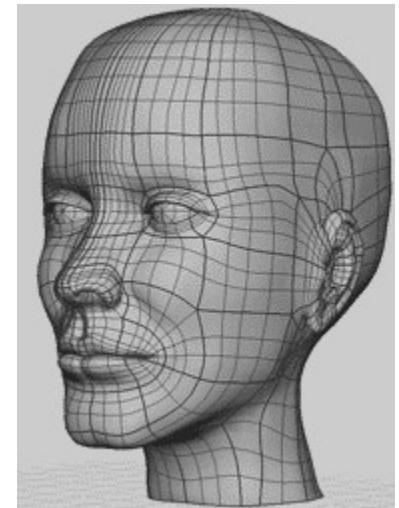
# NURBS



# Enačba za ploskev NURBS

Pri modeliranju s ploskvami NURBS moramo poznati pomen verteksov, stopnje, vozlov (knots) in uteži (weights).

$$G(s,t) = \frac{\sum_{i=0}^{k_1} \sum_{j=0}^{k_2} W(i,j) P(i,j) b_i(s) b_j(t)}{\sum_{i=0}^{k_1} \sum_{j=0}^{k_2} W(i,j) b_i(s) b_j(t)}$$



P(i,j) matrika verteksov:

število vrstic=(k1+1). število stolpcev=(k2+1)

W(i,j) matrika uteži verteksov: po ena za vsak verteks

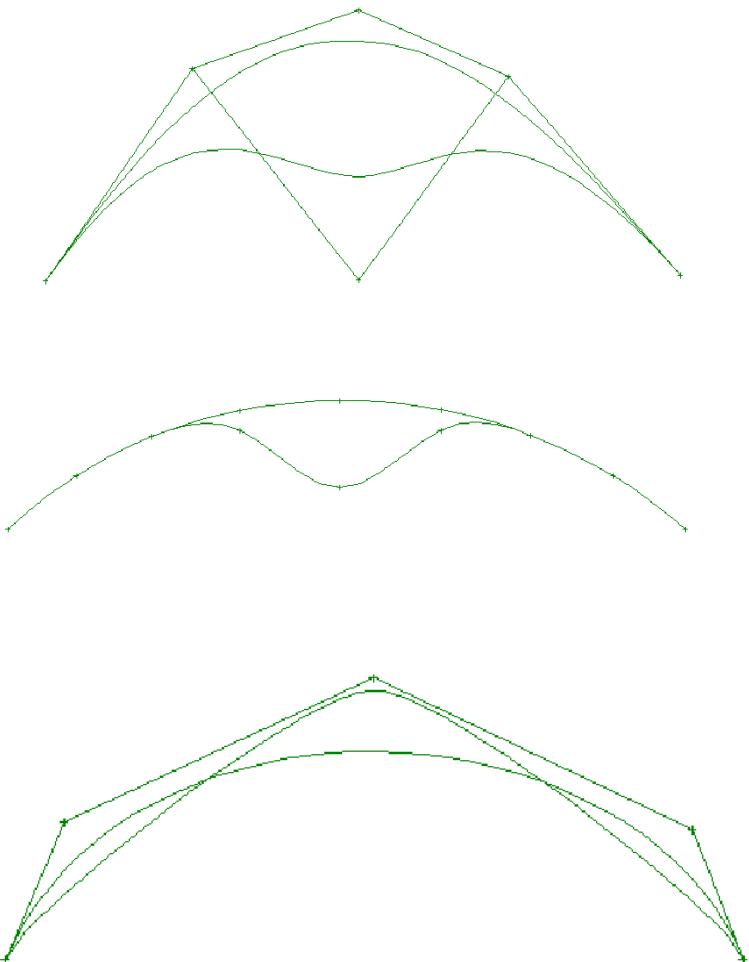
b<sub>i</sub>(s) vrstične mešalne (blending) polinomske funkcije stopnje M<sub>1</sub>

b<sub>j</sub>(t) stolpčne mešalne (blending) polinomske funkcije stopnje M<sub>2</sub>

s polje parametrov vrstičnih vozlov

t polje parametrov stolpčnih vozlov

# Modeliranje ploskev



Če premaknemo verteks, se v njegovi bližini premakne v smeri pomika tudi cela ploskev, kot bi bila na vzmeti.

Problem modeliranja s premiki verteksov:

- Ploskev težko naravnamo v točen položaj.
- Mreža verteksov zakriva obliko ploskve

Modeliranje s točkami na ploskvi: Vsakemu verteksu, ki "lebdi v zraku", ustreza ena točka (edit point) na ploskvi.

Nekateri modelirniki omogočajo premikanje teh točk.

# Parametrične ploskve

- Advantages:
  - Easy to enumerate points on surface
  - Possible to describe complex shapes
- Disadvantages:
  - Control mesh must be quadrilaterals
  - Continuity constraints difficult to maintain
  - Hard to find intersections

# Problemi z zlepki in krpami

- Difficult to stitch together
  - Maintaining continuity is hard
  - Trimming boundaries is hard
- Difficult to model objects with complex topology
  - OK for disk, cylinder, torus



# Subdivision in a production environment.

- Traditionally spline patches (NURBS) have been used in production for character animation.
- Difficult to control spline patch density in character modelling.



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

(Geri's Game, Pixar 1998)

# Deljenje ploskev

- Princip
  - Povečevanje števila kontrolnih točk
  - Boljši vpliv na obliko krivulje
- Raffinement successifs
  - Courbe résultante lisse
  - Nombreuses études mathématiques

# Zakaj delitev ploskev?

- Single Surface for All Modeling Operations
- Multi-Resolution Edits
- Extensions to Support Sharp Edges
- Arbitrary Topology
- No Cracks

# Sharp Edges...

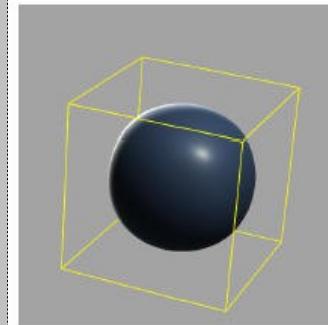
## 1. Tag Edges as “**sharp**” or “**not-sharp**”

- $n = 0$  – “**not sharp**”
- $n > 0$  – **sharp**

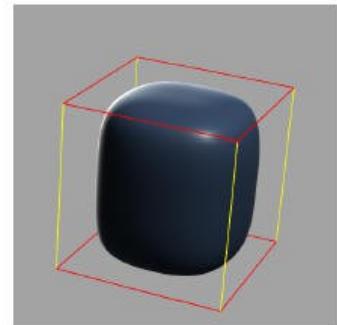
During Subdivision,

2. if an edge is “**sharp**”, use sharp subdivision rules. Newly created edges, are assigned a sharpness of  $n-1$ .
3. If an edge is “**not-sharp**”, use normal smooth subdivision rules.

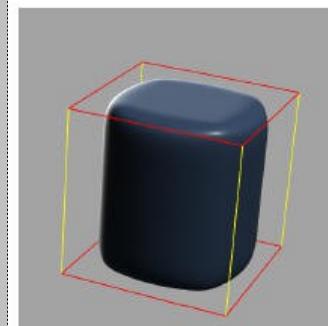
IDEA: Edges with a sharpness of “ $n$ ” do not get subdivided smoothly for “ $n$ ” iterations of the algorithm.



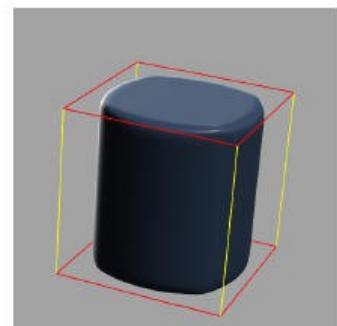
(a)



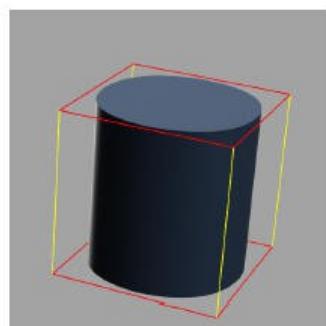
(b)



(c)



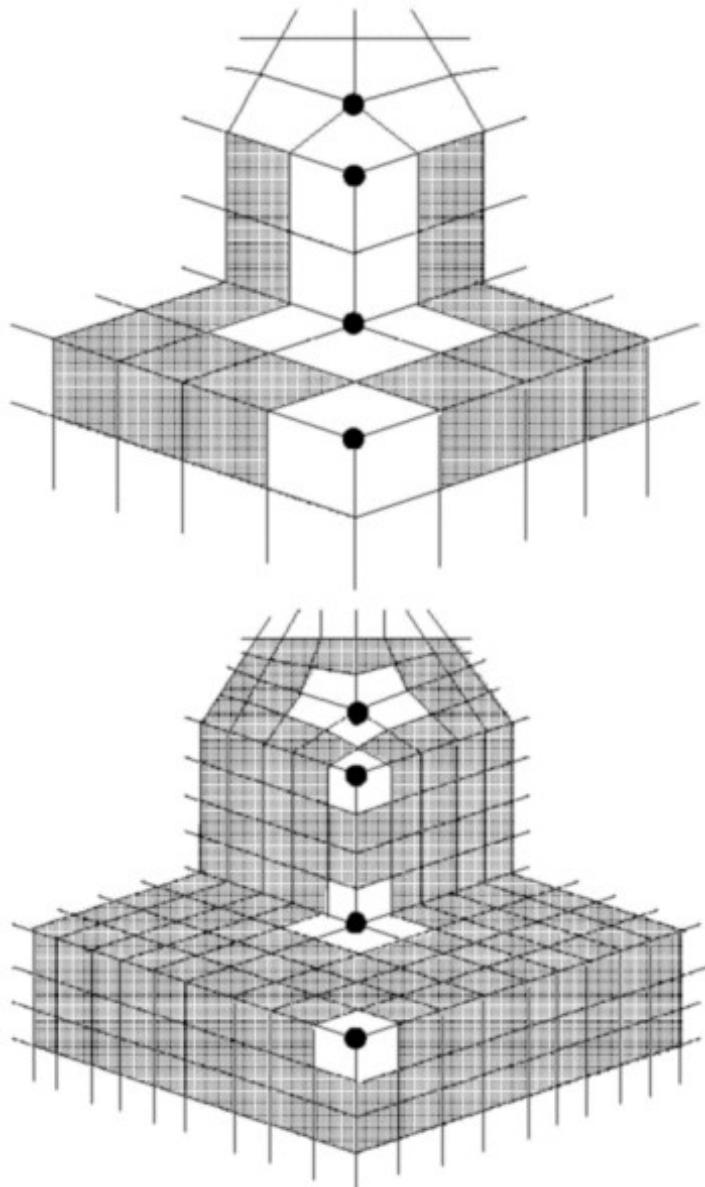
(d)



(e)

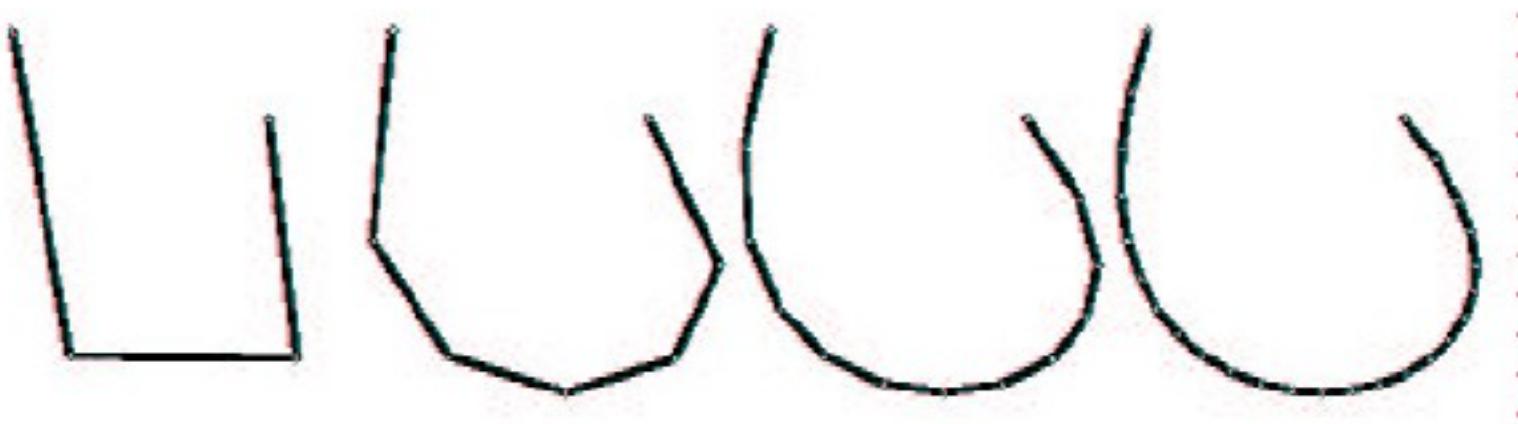
# Subdivision Terminology

- Vertices:
  - Regular/Extraordinary
  - Odd/Even
  - Face/Edge
- Edges:
  - Boundaries and Creases
- Control Mesh



# Delitev (subdivision)

- How do you make a smooth curve?



# Delitev krivulj (Curve Subdivision)

- Edge points

$$\mathbf{p}_{2i}^1 = \frac{1}{2} \mathbf{p}_i + \frac{1}{2} \mathbf{p}_{i+1}$$

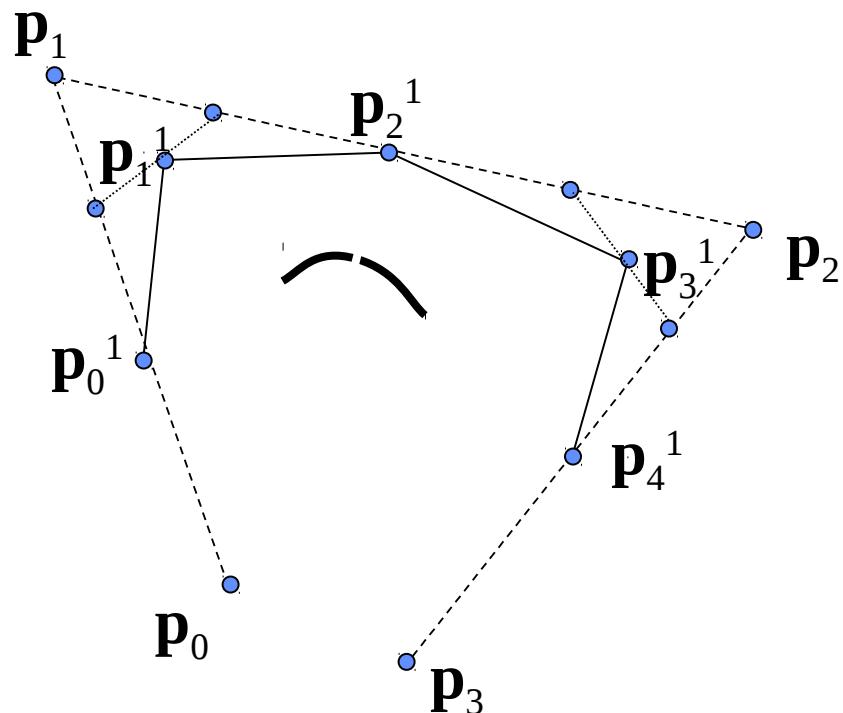
- edge points at midpoint between vertices

- Vertex points

$$\mathbf{p}_{2i+1}^1 = \frac{1}{8} \mathbf{p}_i + \frac{3}{4} \mathbf{p}_{i+1} + \frac{1}{8} \mathbf{p}_{i+2}$$

- midpoint between midpoints between old vertices and new edge points

$$= \frac{1}{2} (\frac{1}{2} \mathbf{p}_{2i}^1 + \frac{1}{2} \mathbf{p}_{i+1}) + \\ \frac{1}{2} (\frac{1}{2} \mathbf{p}_{i+1} + \frac{1}{2} \mathbf{p}_{2i+1}^1)$$



$$\begin{bmatrix} \mathbf{p}_0^1 \\ \mathbf{p}_1^1 \\ \mathbf{p}_2^1 \\ \mathbf{p}_3^1 \\ \mathbf{p}_4^1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$$

# Delitev ploskev (subdivision surfaces)

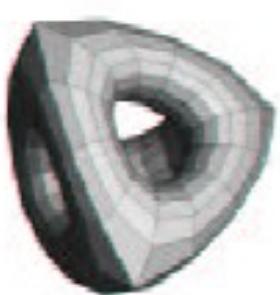
- Coarse mesh & subdivision rule
  - Define smooth surface as limit of sequence of refinements



(a)



(b)



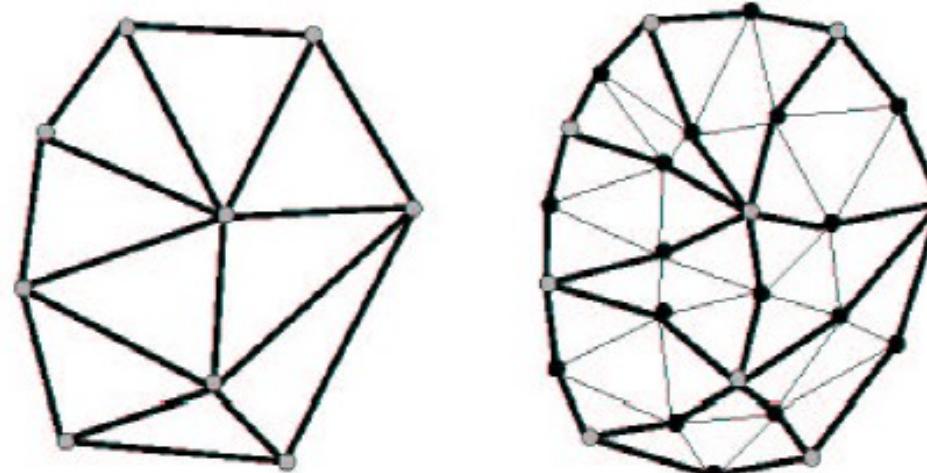
(c)



(d)

# Ključna vprašanja

- How to refine mesh?
  - Topology changes
- Where to place new vertices
  - Provable properties about limit surface

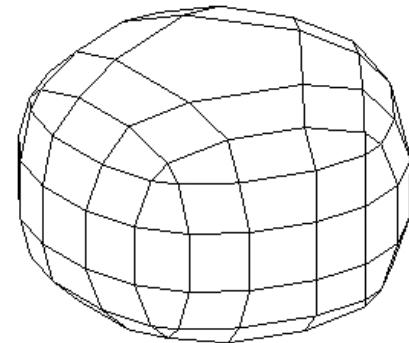
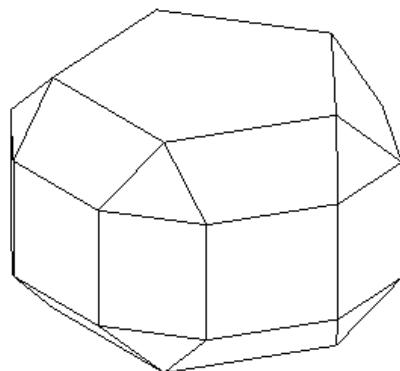
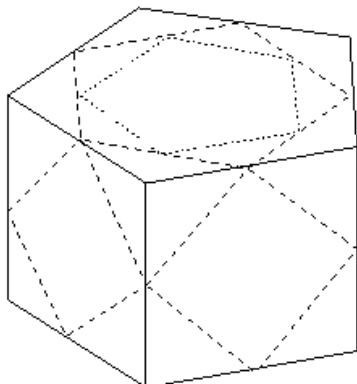


# Pregled shem deljenja

- Rezanje vogalov
- Dodajanje verteksov
  - Interpolacija
  - Aproksimacija

# Rezanje vogalov

- Old Vertices are Discarded
- Common Schemes:
  - Doo-Sabin
  - Mid-Edge



# Interpolacija

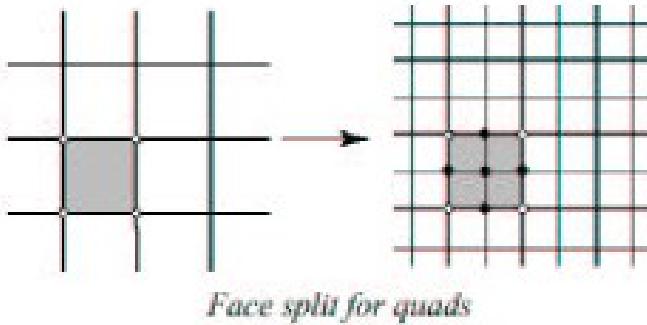
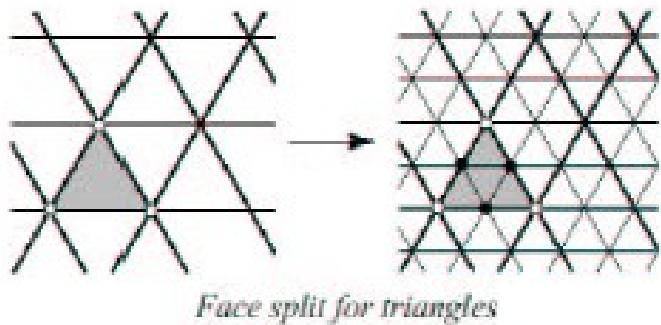
- Even Vertices Remain Stationary
- ‘Inflates’ Out to Limit Surface (bad)
- Common Schemes:
  - Modified Butterfly (triangle based)
  - Kabbelt (quad based)

# Aproksimacija

- Even Vertices are Moved
- Converges in to Limit Surface
- Convergence Faster than Interpolating Schemes
- Better Mathematical Properties
- Common Schemes:
  - Loop (triangle based)
  - Catmull-Clark (quad based)

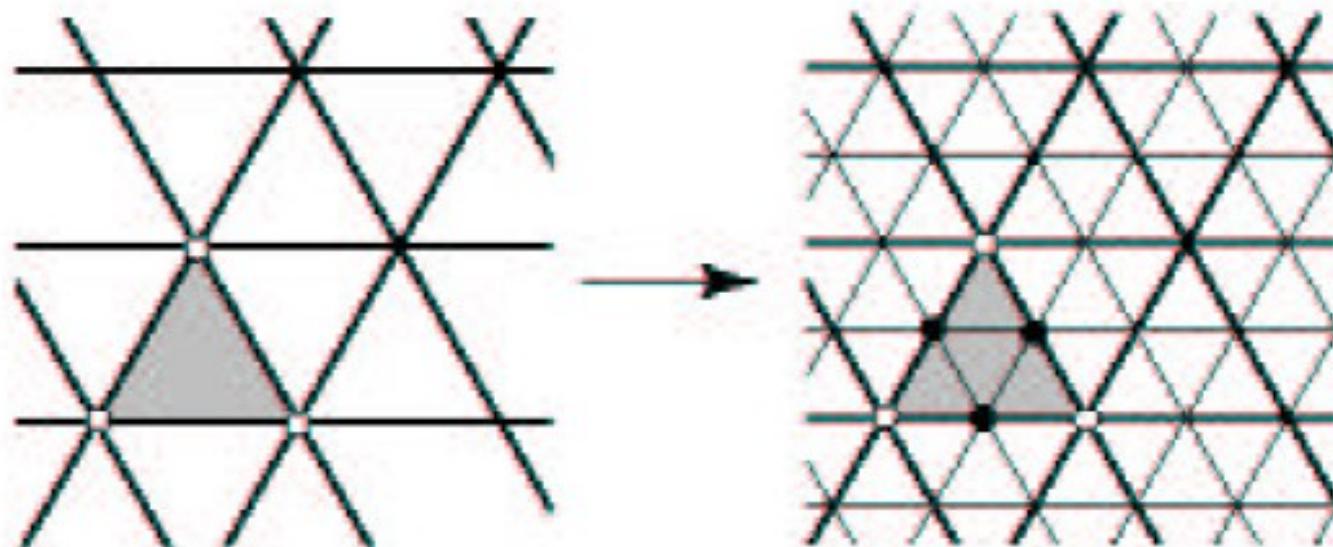
# Različne sheme delitve

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

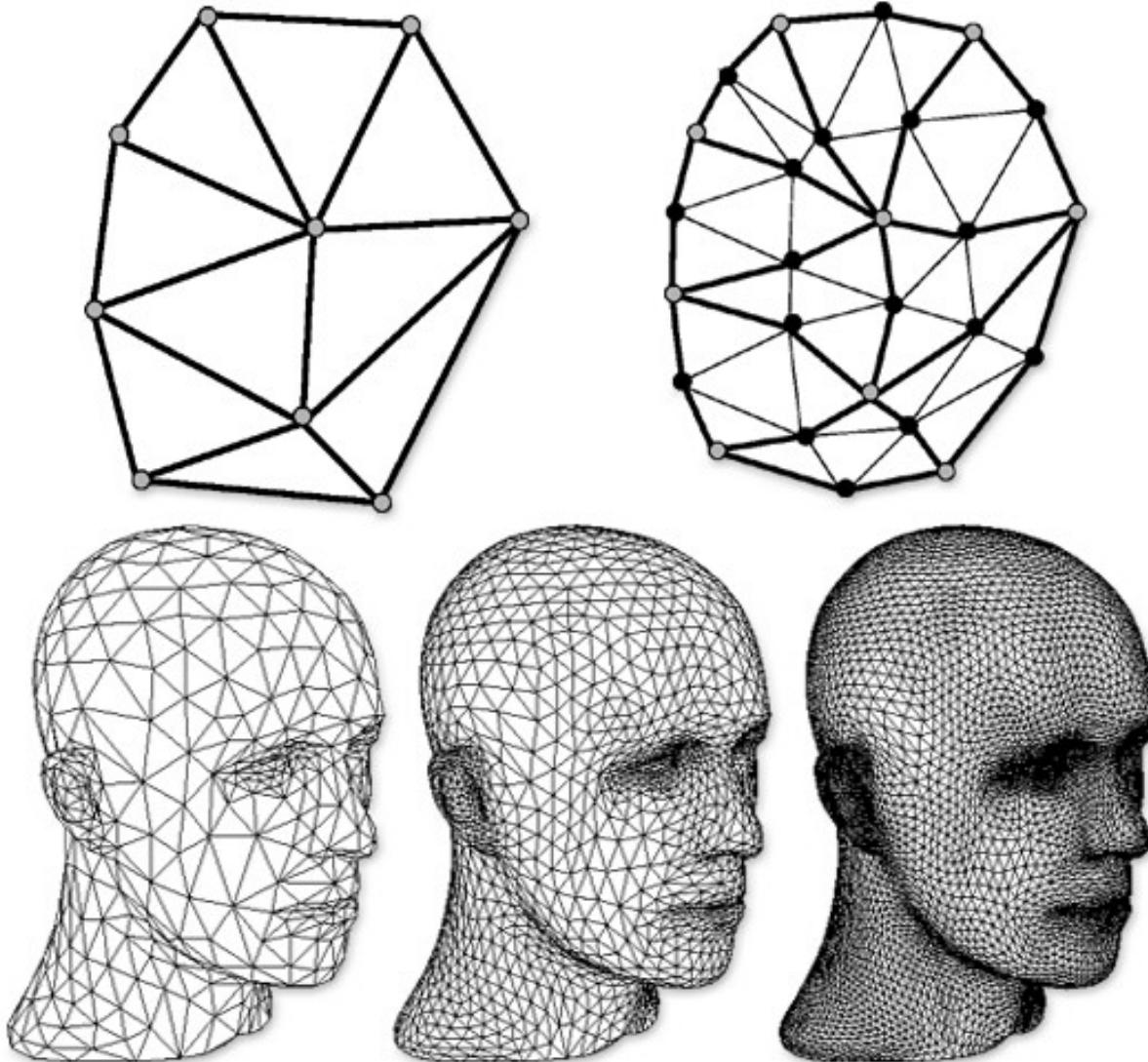


# Delitvena shema v zanki (Loop subdivision scheme)

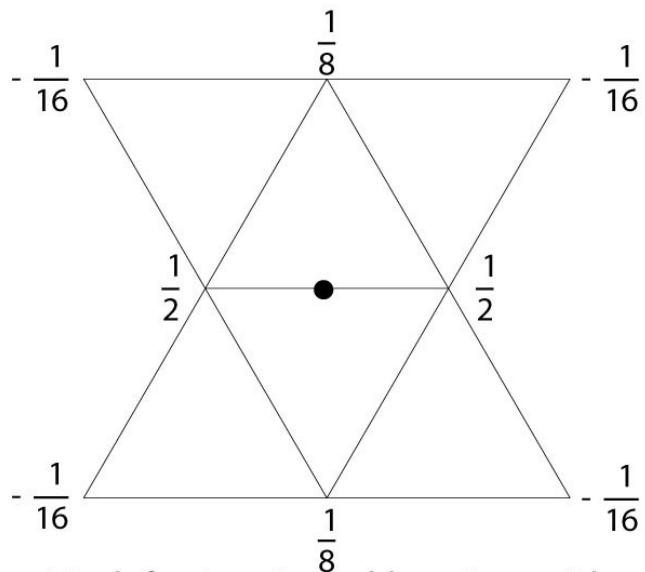
- How refine mesh?
  - Refine each triangle into 4 triangles by splitting each edge and connecting new vertices



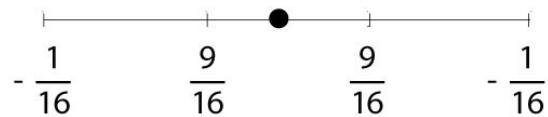
# Delitvena shema v zanki



# Metuljčkasta delitev (Butterfly subdivision)

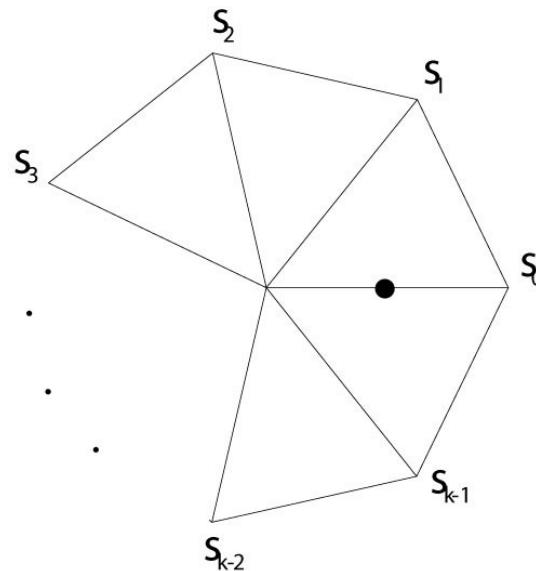


Mask for interior odd vertices with regular neighbors



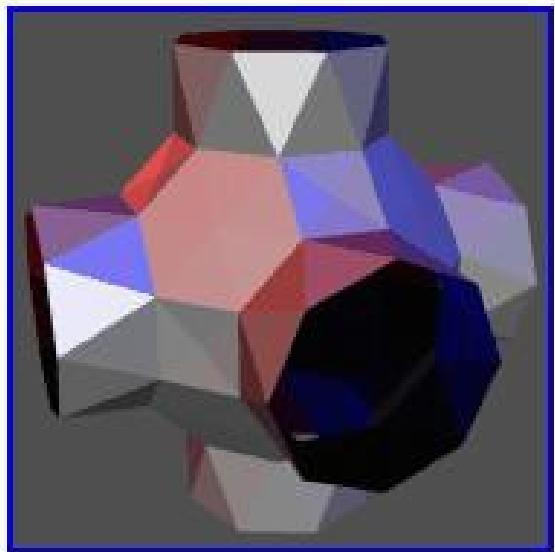
Mask for crease and boundary vertices

a. Masks for odd vertices

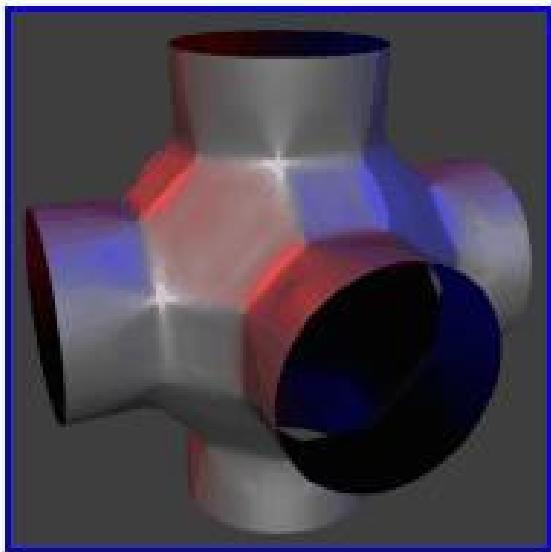


b. Mask for odd vertices adjacent to an extraordinary vertex

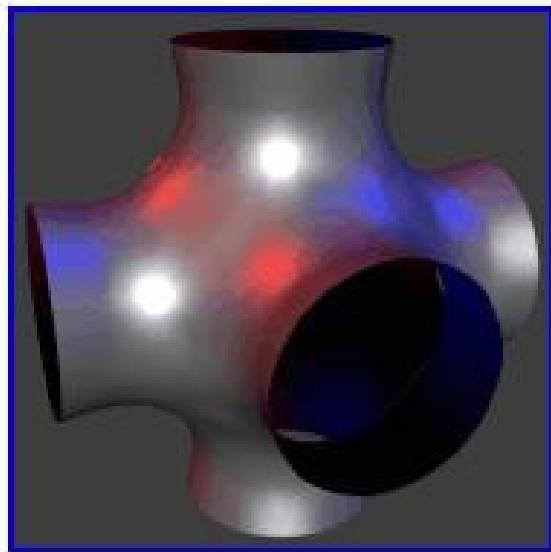
# Butterfly subdivision



initial mesh



Butterfly subdivision



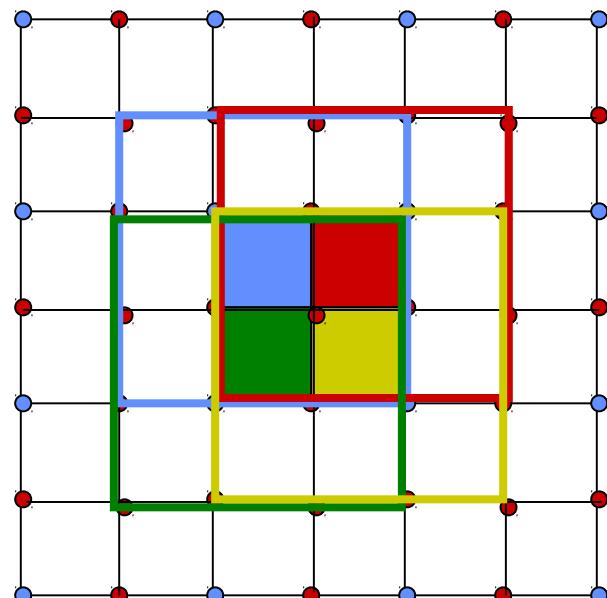
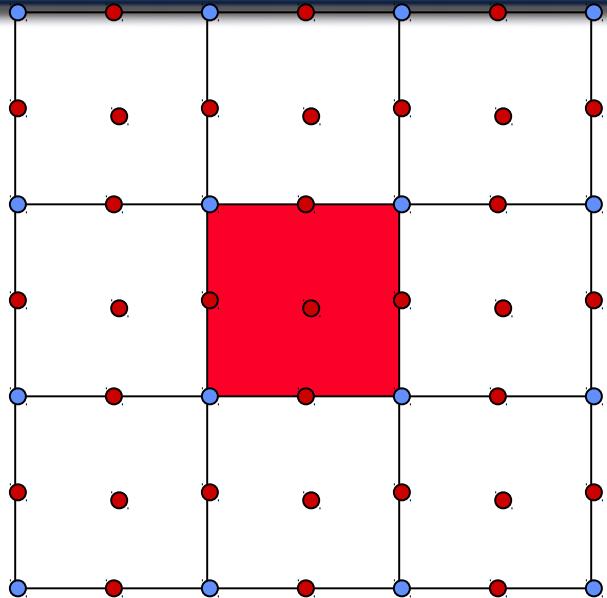
Modified Butterfly subdivision

# Krpe z b-zlepki (B-Spline Patches)

- Tensor product of two curves

$$\mathbf{p}(s, t) = \sum_{j=0}^n \sum_{i=0}^n N_j^n(s) N_i^n(t) \mathbf{p}_{ij}$$

- Need to subdivide control points to create four sub-patches
- Need to generate new control points
  - vertex points (replacing control points)
  - edge points
  - face points



# Face Points

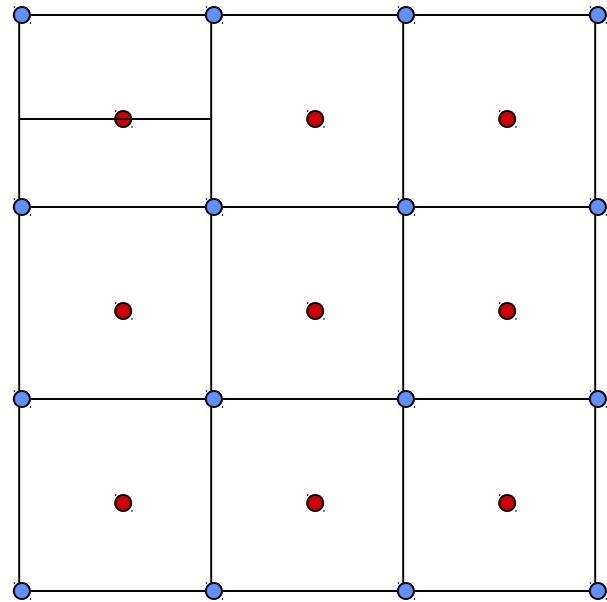
- Approximate edge points as midpoint of control points

$$E = \frac{1}{2} p + \frac{1}{2} p$$

- Face point is midpoint of approximate edge points

$$F = \frac{1}{2} E + \frac{1}{2} E$$

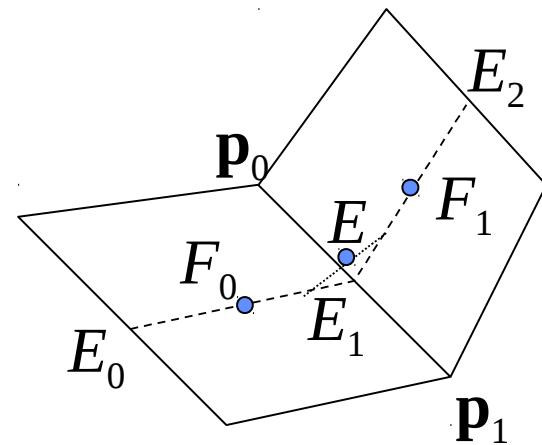
$$= \frac{1}{4} p + \frac{1}{4} p + \frac{1}{4} p + \frac{1}{4} p$$



# Edge Points

- Face points are midpoints between approx. edge points
- Approx. edge point is midpoint between control points
- Actual edge point is midpoint between midpoints between approx edge point and face points

$$\begin{aligned} E &= 1/2 (1/2 (1/2 E_0 + 1/2 E_1) + 1/2 E_1) + \\ &\quad 1/2 (1/2 E_1 + 1/2 (1/2 E_1 + 1/2 E_2)) \\ &= 1/2 (1/2 F_0 + 1/2 (1/2 \mathbf{p}_0 + 1/2 \mathbf{p}_1)) + \\ &\quad 1/2 (1/2 (1/2 \mathbf{p}_0 + 1/2 \mathbf{p}_1) + 1/2 F_1) \\ &= 1/4 (F_0 + \mathbf{p}_0 + \mathbf{p}_1 + F_1) \end{aligned}$$



# Vertex Points

$$V_0 = 1/4 E_0 + 1/2 \mathbf{p}_0 + 1/4 E_1$$

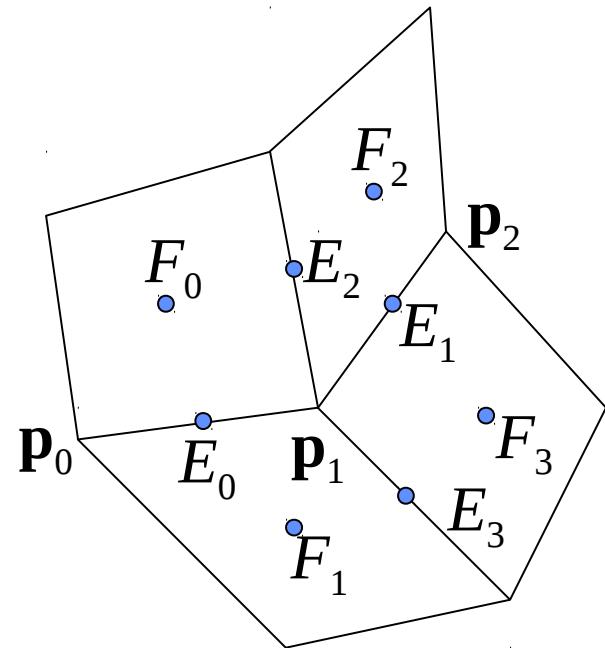
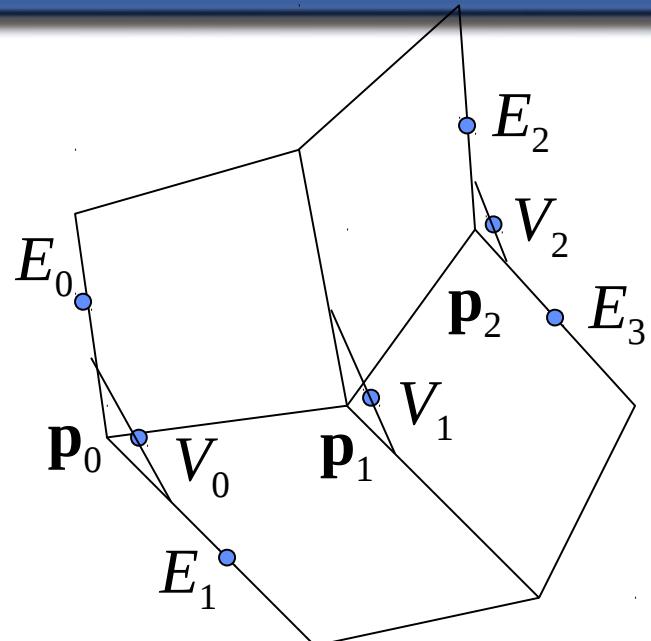
$$V_2 = 1/4 E_2 + 1/2 \mathbf{p}_2 + 1/4 E_3$$

$$V = 1/2 (1/2 (1/2 V_0 + 1/2 V_1) + 1/2 V_1) + \\ 1/2 (1/2 V_1 + 1/2 (1/2 V_1 + 1/2 V_2))$$

$$= 1/4 (1/4 (F_0 + F_1 + \mathbf{p}_0 + \mathbf{p}_1) + \\ 1/4 (F_2 + F_3 + \mathbf{p}_1 + \mathbf{p}_2) + 2 V_1)$$

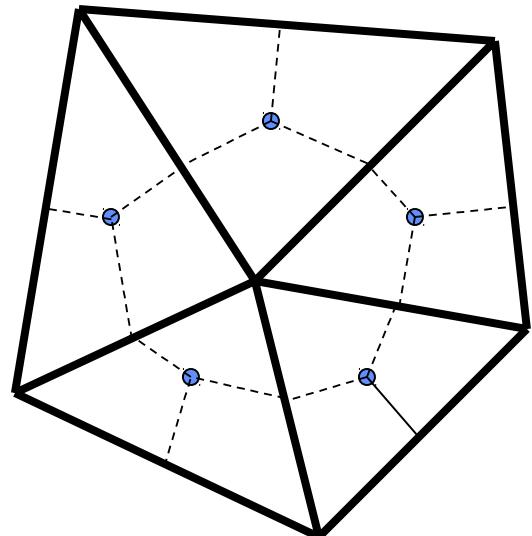
$$= 1/4 (1/4 (F_0 + F_1 + F_2 + F_3) + \\ 1/4 (\mathbf{p}_0 + 2 \mathbf{p}_1 + \mathbf{p}_2) + \\ 2/4 (E_2 + E_3 + 2 \mathbf{p}_1))$$

$$= 1/16(F_0 + F_1 + F_2 + F_3 + \\ 2E_0 + 2E_1 + 2E_2 + 2E_3 + 4\mathbf{p}_1)$$

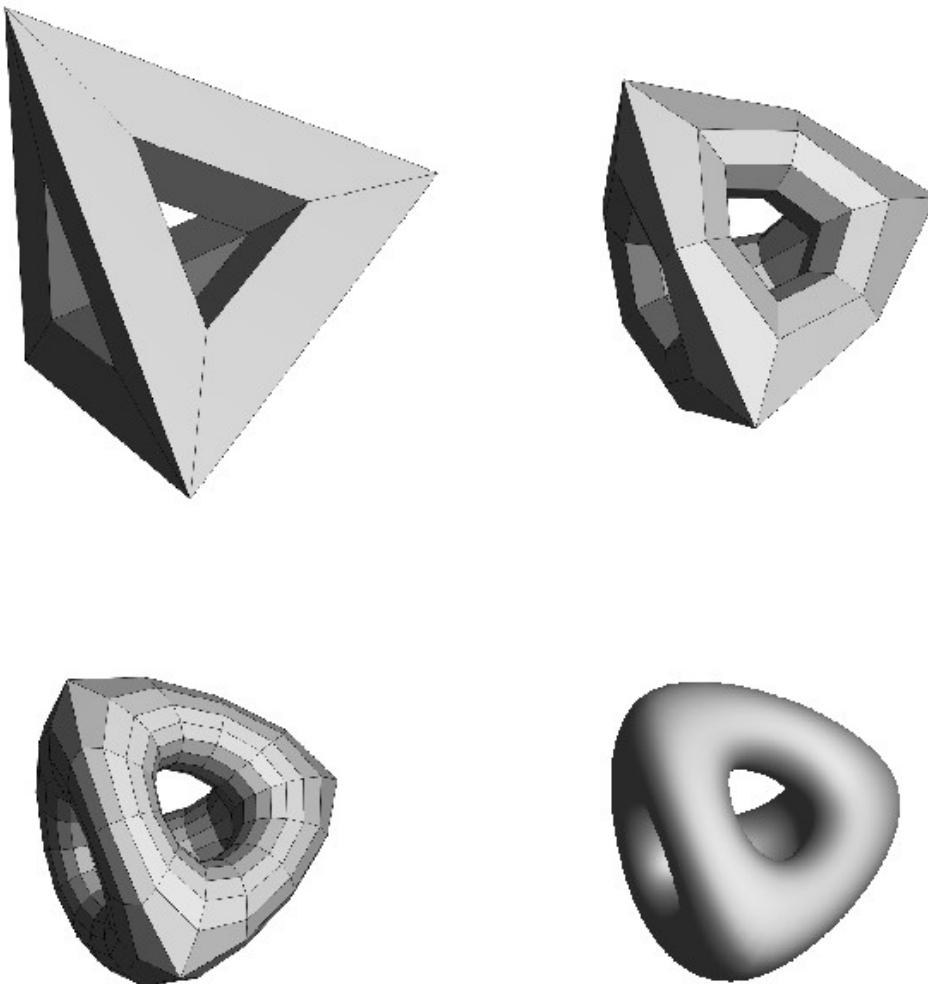


# Delitev Catmull-Clark

- Face points = average of ( $n$ ) control points
- Edge points = average of two control points and two face points
- Vertex points = average of...
  - average of adjacent face points
  - twice the average of midpoints of adjacent edges
  - $(n - 3)$  terms of the control point



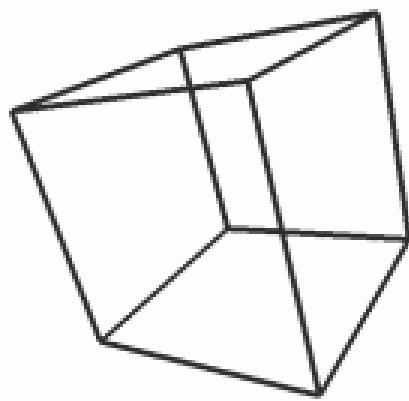
# Primer delitve Catmull-Clark



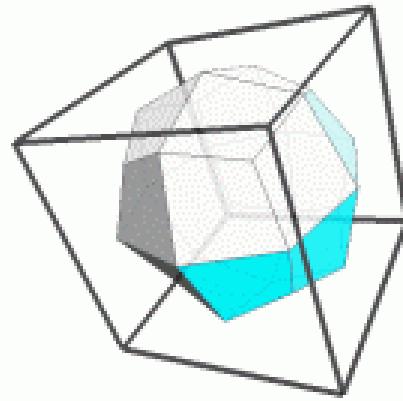
# Še en primer



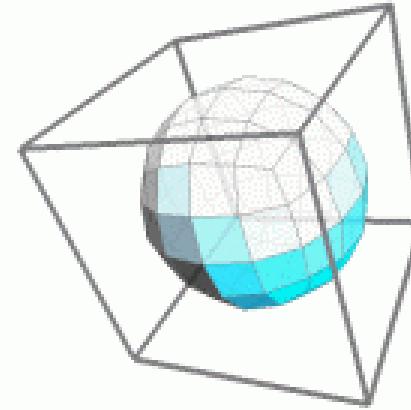
# Catmull-Clarkova delitev ploskev



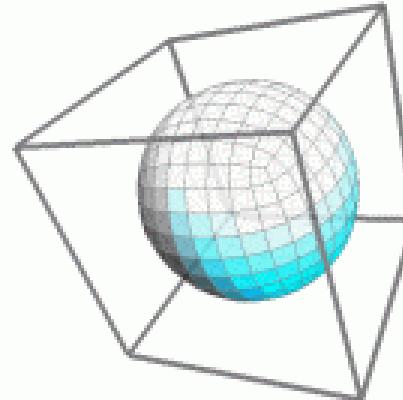
**Original Cube**



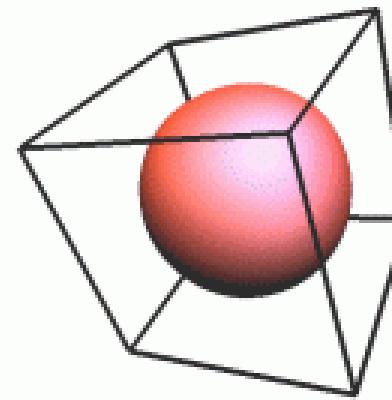
**The 1st subdivision**



**The 2nd subdivision**

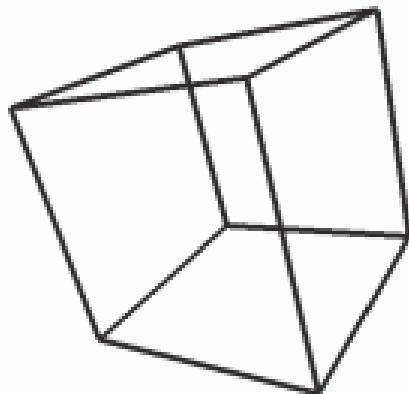


**The 3rd subdivision**

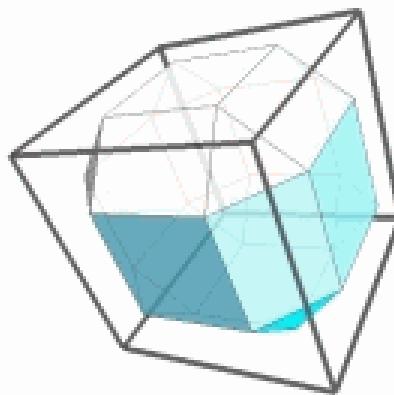


**The 5th subdivision**

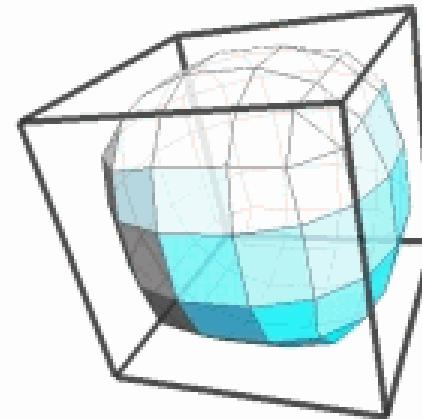
# Doo- Sabinova delitev ploskev



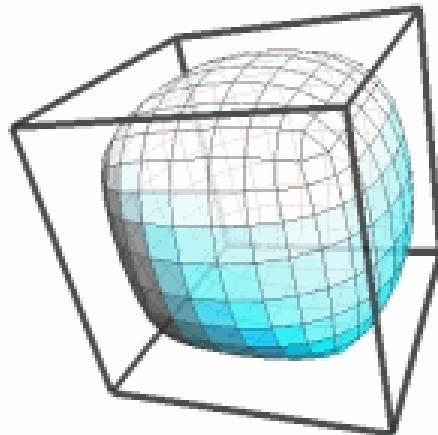
Original Cube



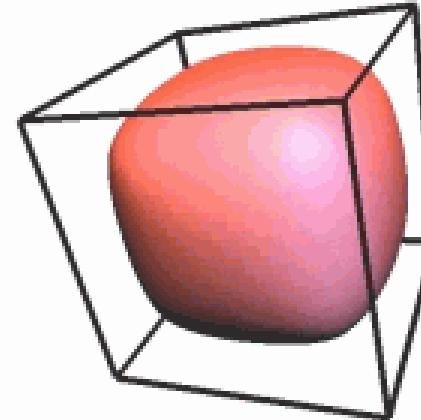
The 1st subdivision



The 2nd subdivision

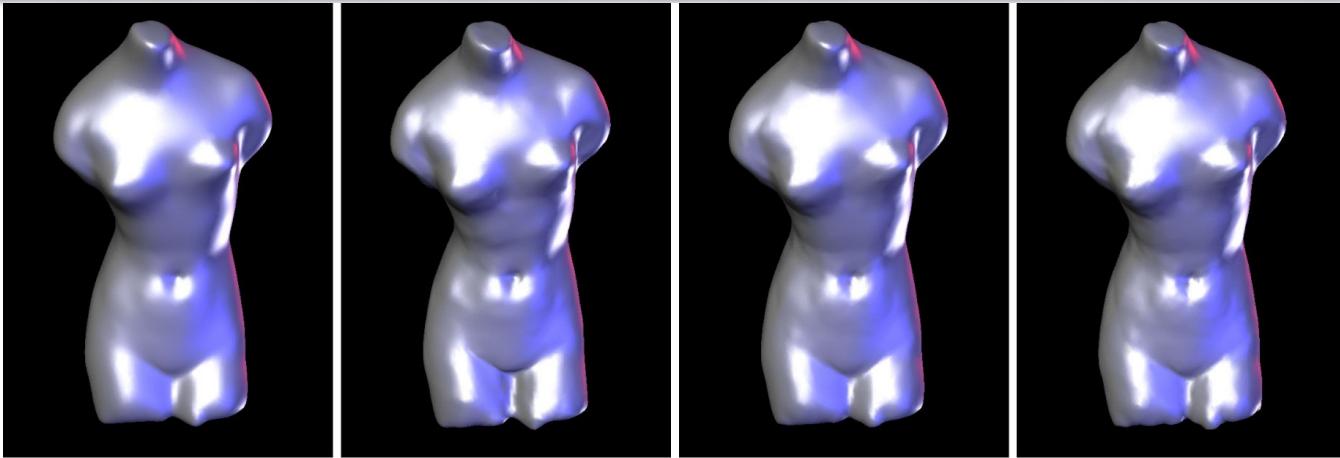


The 3rd subdivision



The 5th subdivision

# Primeri



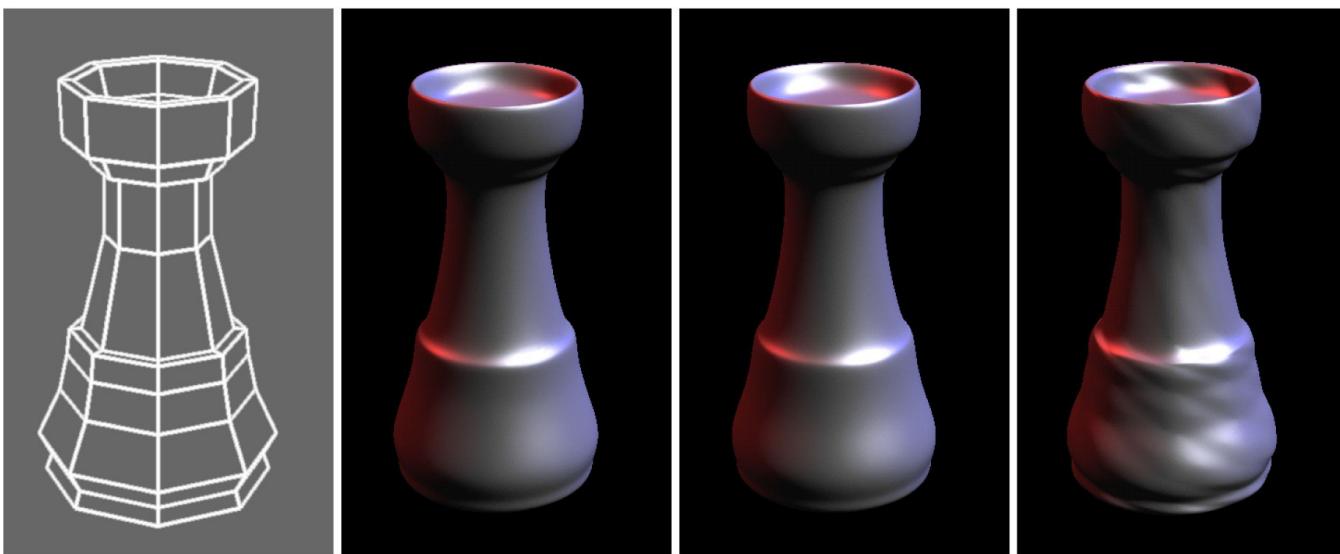
*Loop*

*Butter*

*Catmull-Clark*

*Doo-Sabin*

Figure 4.20: *Different subdivision schemes produce similar results for smooth meshes.*

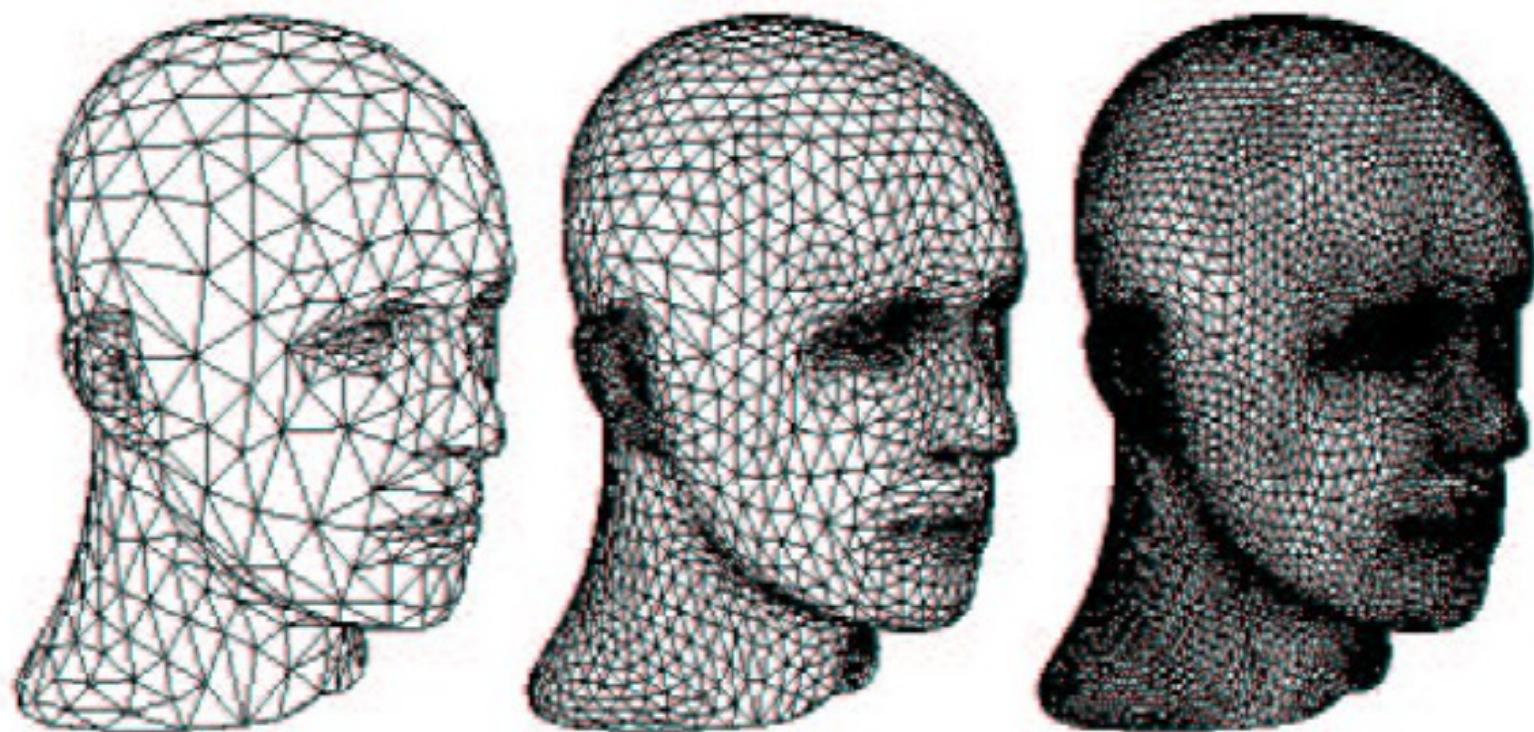


*Initial mesh*

*Loop*

*Catmull-Clark*

*Catmull-Clark, after  
triangulation*



V limiti imamo zagotovljeno gladkost ploskve

# Deljenje površin – kako je z zveznostjo

## Deljenje površine

*Trikotne mreže*

*Aproksimacija*

Zanka ( $C^2$ )

*Interpolacija*

Metuljčna ( $C^1$ )

*Kvadratne mreže*

Catmull-Clark ( $C^2$ )

Kobbelt ( $C^1$ )

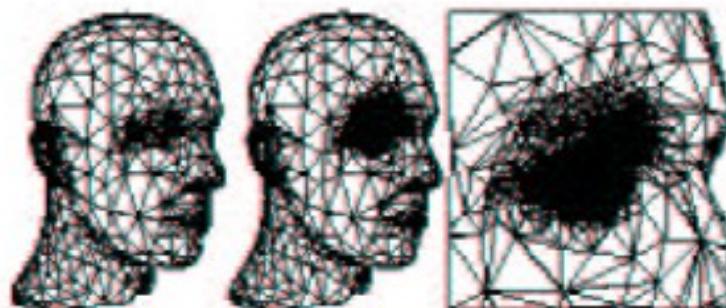
## Deljenje ogljišč

Doo-Sabin, Midedge ( $C^1$ )

Bikvadrično ( $C^2$ )

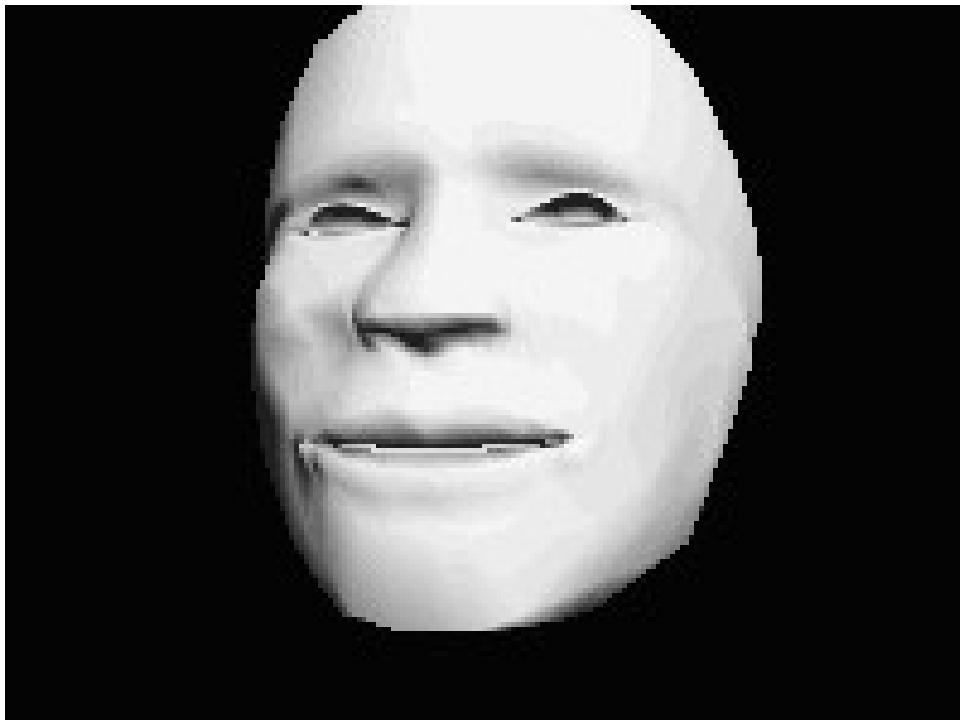
# Delitev ploskev, povzetek

- Advantages:
  - Simple method for describing complex surfaces
  - Relatively easy to implement
  - Arbitrary topology
  - Local support
  - Guaranteed continuity
  - Multiresolution
- Difficulties:
  - Intuitive specification
  - Parameterization
  - Intersections



# In kaj pridobimo?

- Ni špranj



# Značilnosti poligonov

Dobro:

- Hitri
- Poljubna topologija
- Lahko spajanje v večje strukture

Slabo:

- $C^0$  zveznost
- Veliko število poligonov pri kompleksnih površinah
- Težko globalno urejanje

# Značilnosti B-zlepkov

## Dobro:

- Matematično elegantne formule
- Dobro poznana tehika
- $C^2$  povsod

## Slabo:

- Zahtevajo odpravljanje špranj ('Socking' to Prevent Cracks)
- Težko modeliranje ostrih in zavitih robov



# Značilnosti NURBS

## Dobro:

- CV uteži => Ostri robovi, dobra kontrola
- Trim Curves
- Dobro podprt v knjižnicah

## Slabo:

- Trim Curves podvrženo numeričnim napakam
- Very Hard to Sock with Trim Curves
- Težka implementacija
- Orodja so zaenkrat premalo napredna

# Značilnosti implicitnih ploskev

## Dobro:

- Lahko modeliramo “gumijaste” objekte (‘Blobby’ Objects)
- Matematično trivialno

## Slabo:

- Slab nadzor
- Težko modeliranje podrobnosti
- Težko doseganje pravilne poligonacije

# Zakaj delitev površin?

- Raba operacij nad eno samo površino
- Editiranje na več nivojih resolucije
- Omogočajo ostre robeve
- Poljubna topologija
- Ni špranj