

# **Parametric Surfaces**

**CSCI 4229/5229**

**Computer Graphics  
Fall 2010**

# Bézier Surfaces

- In one dimension

- $C_n(t) = \sum_{i=0}^n B_i^n(t) P_i, \quad t \in [0,1]$

- In two dimensions

- $S_{n,m}(t,r) = \sum_{i=0}^n B_i^n(t) \sum_{j=0}^m B_j^m(r) P_{ij}, \quad t,r \in [0,1]$

- $P_{ij}$  are points in 3D or 4D

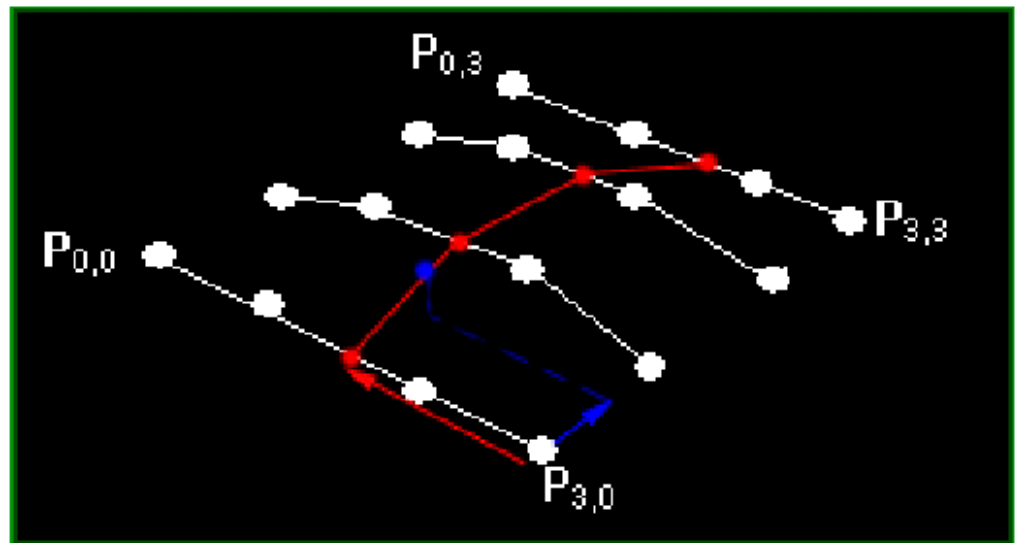
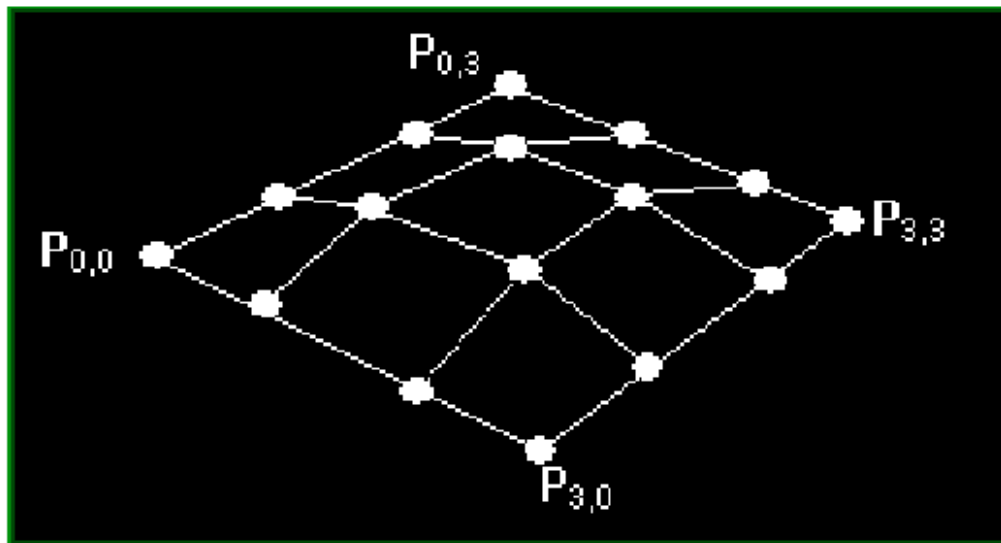
- Convex linear combination of points  $P_{ij}$

- Entire curve is in convex hull of points
  - Surface passes through 4 corner points

- Curve is smooth and differentiable

# 2D Cubic Bézier Surface

- 16 Control points
- Corner points set surface
- Interior points stretches surface



# Bicubic Bézier Patch

$$\begin{aligned} P &= (1-v)^3 \left( (1-u)^3 P_{00} + 3(1-u)^2 u P_{01} + 3(1-u)u^2 P_{02} + u^3 P_{03} \right) \\ &+ 3(1-v)^2 v \left( (1-u)^3 P_{10} + 3(1-u)^2 u P_{11} + 3(1-u)u^2 P_{12} + u^3 P_{13} \right) \\ &+ 3(1-v)v^2 \left( (1-u)^3 P_{20} + 3(1-u)^2 u P_{21} + 3(1-u)u^2 P_{22} + u^3 P_{23} \right) \\ &+ v^3 \left( (1-u)^3 P_{30} + 3(1-u)^2 u P_{31} + 3(1-u)u^2 P_{32} + u^3 P_{33} \right) \\ &= (1-u)^3 \left( (1-v)^3 P_{00} + 3(1-v)^2 v P_{10} + 3(1-v)v^2 P_{20} + v^3 P_{30} \right) \\ &+ 3(1-u)^2 u \left( (1-v)^3 P_{01} + 3(1-v)^2 v P_{11} + 3(1-v)v^2 P_{21} + v^3 P_{31} \right) \\ &+ 3(1-u)u^2 \left( (1-v)^3 P_{02} + 3(1-v)^2 v P_{12} + 3(1-v)v^2 P_{22} + v^3 P_{32} \right) \\ &+ u^3 \left( (1-v)^3 P_{03} + 3(1-v)^2 v P_{13} + 3(1-v)v^2 P_{23} + v^3 P_{33} \right) \end{aligned}$$

# Bicubic Bézier Patch Normal

$$\begin{aligned}
 \frac{\partial P}{\partial u} &= -3(1-u)^2 \left( (1-v)^3 P_{00} + 3(1-v)^2 v P_{10} + 3(1-v)v^2 P_{20} + v^3 P_{30} \right) \\
 &+ 3(1-3u)(1-u) \left( (1-v)^3 P_{01} + 3(1-v)^2 v P_{11} + 3(1-v)v^2 P_{21} + v^3 P_{31} \right) \\
 &+ 3u(2-3u) \left( (1-v)^3 P_{02} + 3(1-v)^2 v P_{12} + 3(1-v)v^2 P_{22} + v^3 P_{32} \right) \\
 &+ 3u^2 \left( (1-v)^3 P_{03} + 3(1-v)^2 v P_{13} + 3(1-v)v^2 P_{23} + v^3 P_{33} \right) \\
 \frac{\partial P}{\partial v} &= -3(1-v)^2 \left( (1-u)^3 P_{00} + 3(1-u)^2 u P_{01} + 3(1-u)u^2 P_{02} + u^3 P_{03} \right) \\
 &+ 3(1-3v)(1-v) \left( (1-u)^3 P_{10} + 3(1-u)^2 u P_{11} + 3(1-u)u^2 P_{12} + u^3 P_{13} \right) \\
 &+ 3v(2-3v) \left( (1-u)^3 P_{20} + 3(1-u)^2 u P_{21} + 3(1-u)u^2 P_{22} + u^3 P_{23} \right) \\
 &+ 3v^2 \left( (1-u)^3 P_{30} + 3(1-u)^2 u P_{31} + 3(1-u)u^2 P_{32} + u^3 P_{33} \right)
 \end{aligned}$$

$$N = \frac{\partial P}{\partial u} \times \frac{\partial P}{\partial v}$$

# Surfaces in OpenGL

- Two-dimensional Evaluators
- Can be used to generate vertexes, normals, colors and textures
- Curve defined analytically using Bezier surfaces
- Evaluated at discrete points and rendered using polygons

# Surfaces in OpenGL

- `glEnable()`
  - Enables types of data to generate
  - `GL_AUTO_NORMAL` generates normals for you
- `glMap2d()`
  - Defines control points and domain
- `glEvalCoord2d()`
  - Generates a data point
- `glMapGrid2d()` & `glEvalMesh2()`
  - Generates a series of data points

`glMap2d(type,Umin,Umax,Ustride,Uorder,  
Vmin,Vmax,Vstride,Vorder,points)`

- *type* of data to generate
  - `GL_MAP1_VERTEX_3`
  - `GL_MAP1_VERTEX_4`
  - `GL_MAP1_NORMAL`
  - `GL_MAP1_COLOR_4`
  - `GL_MAP1_TEXTURE_COORD_1`
  - `GL_MAP1_TEXTURE_COORD_2`
  - `GL_MAP1_TEXTURE_COORD_3`
  - `GL_MAP1_TEXTURE_COORD_4`
- *Umin*&*Umax* and *Vmin*&*Vmax* are limits (often 0&1)
- *Ustride* is the number of values in data (3,4)
- *Vstride* is the number of values in a row of data
- *Uorder* & *Vorder* is the order of the curve (4=cubic)
- *points* is the array of data points (16 for bi-cubic)
- **Remember to also call `glEnable()`**



# glEvalCoord2d(u,v)

- Generate one vertex for each glMap2d() type currently active (e.g. texture, normal, vertex)
- To generate the whole surface, loop over quads and call glEvalCoord2d() once for each vertex
- Exercise entire parameter space
  - u from Umin to Umax (0 to 1)
  - v from Vmin to Vmax (0 to 1)

# Generating a complete surface

- `glMapGrid2d(N , U1 , U2 , M , V1 , V2)`
- `glEvalMesh2(mode , N1 , N2 , M1 , M2)`
- This is equivalent to

```
for (j=M1;j<M2;j++)
{
    glBegin(GL_QUAD_STRIP);
    for (i=N1;i<=N2;i++)
    {
        glEvalCoord1(U1+i*(U2-U1)/N , V1+j*(V2-V1)/M);
        glEvalCoord1(U1+i*(U2-U1)/N , V1+(j+1)*(V2-V1)/M);
    }
    glEnd();
}
```

# The Utah Teapot

- Generated by Martin Newell in 1975
  - 32 Patches specified as Bezier surfaces
  - 10 Base patches with reflections
  - 126 control points
- Complex shape
  - Hole in handle
  - Hollow spout
- Non-convex
  - Can cast shadows on itself



# The Utah Teapot: Then and Now

