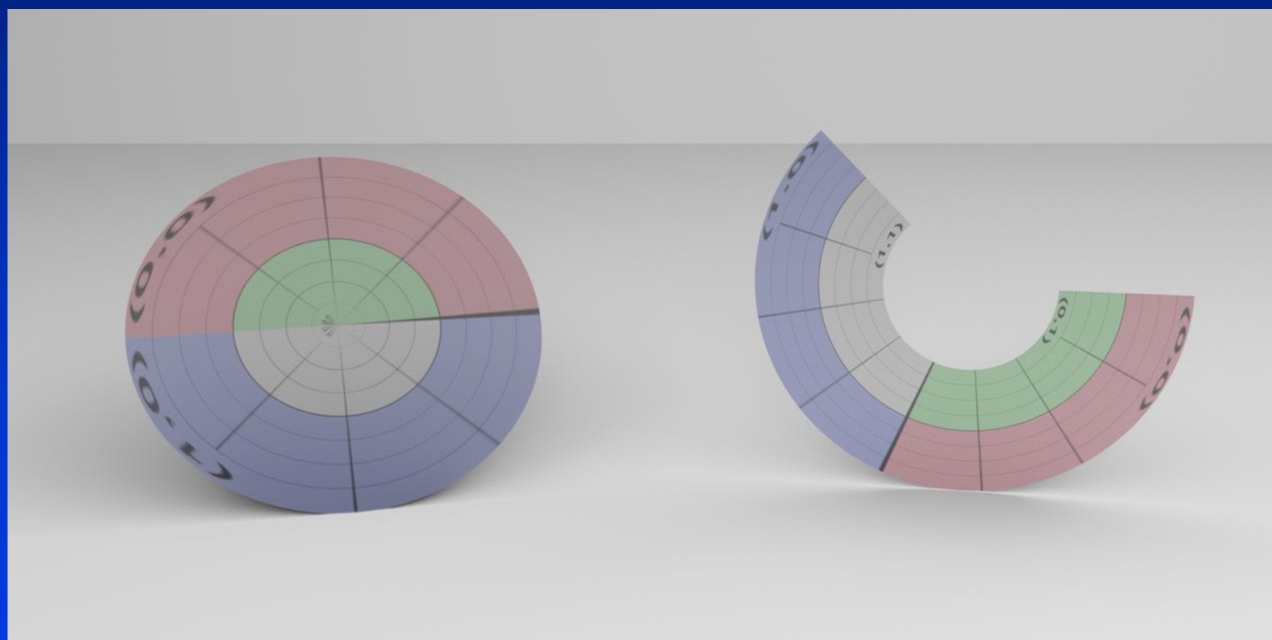
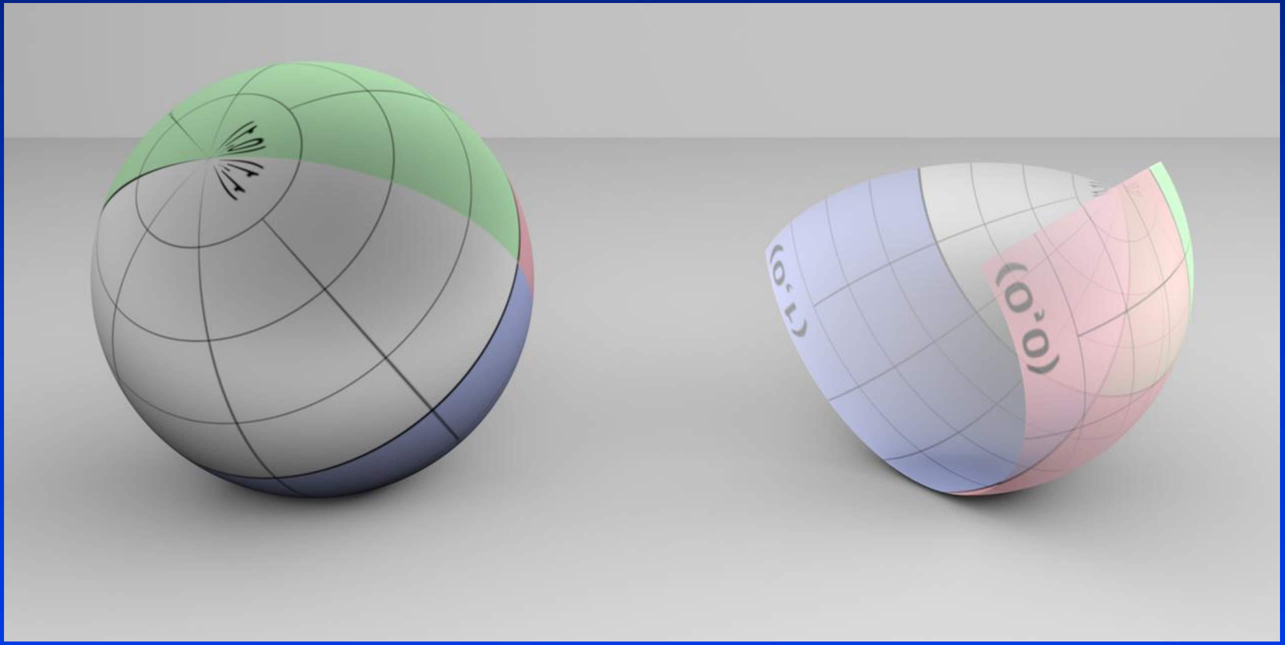
CSE328 Fundamentals of Computer Graphics: Concepts, Theory, Algorithms, and Applications

Hong Qin
Department of Computer Science
Stony Brook University (SUNY at Stony Brook)
Stony Brook, New York 11794-2424
Tel: (631)632-8450; Fax: (631)632-8334
qin@cs.stonybrook.edu
http://www.cs.stonybrook.edu/~qin

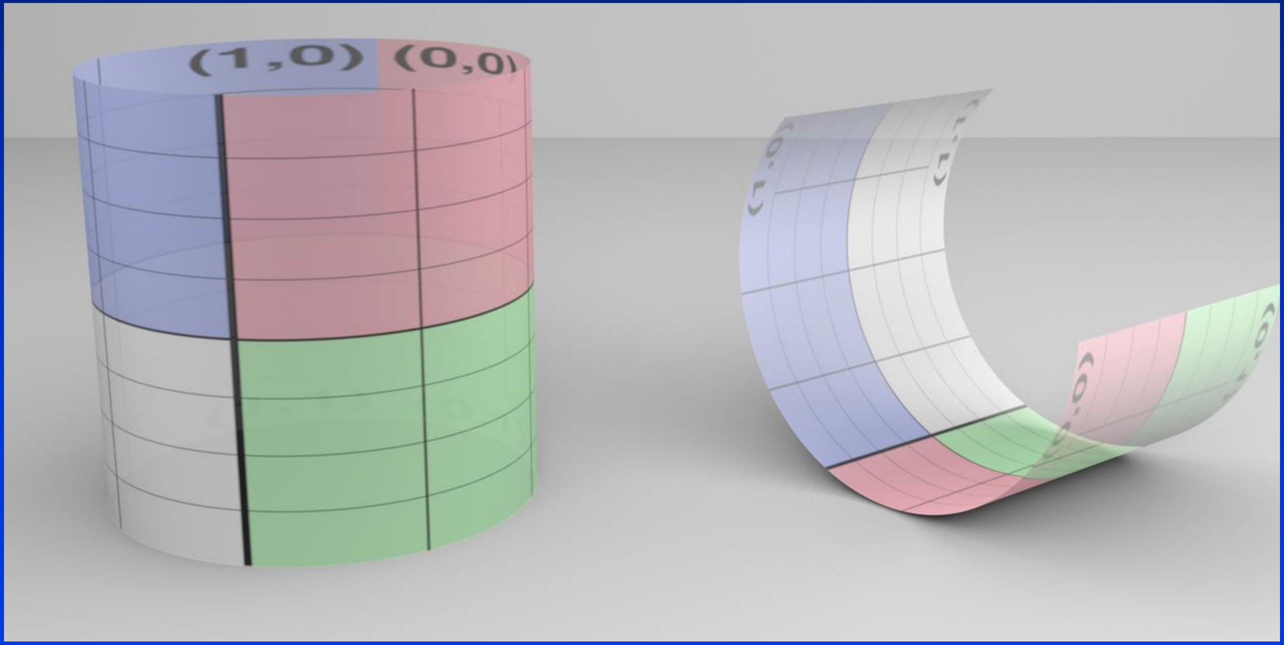
Disk



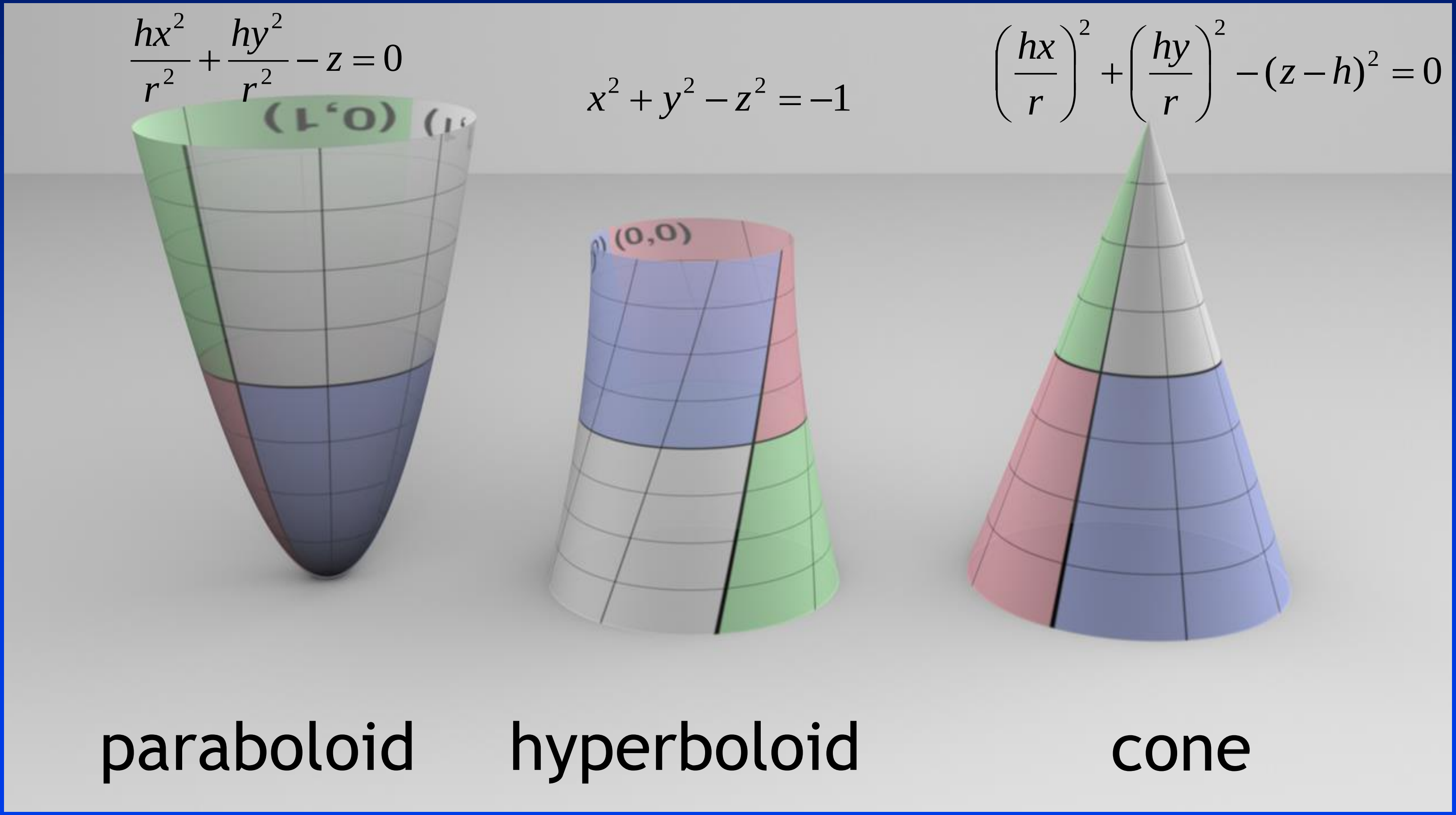
Sphere



Cylinder



Other Quadrics

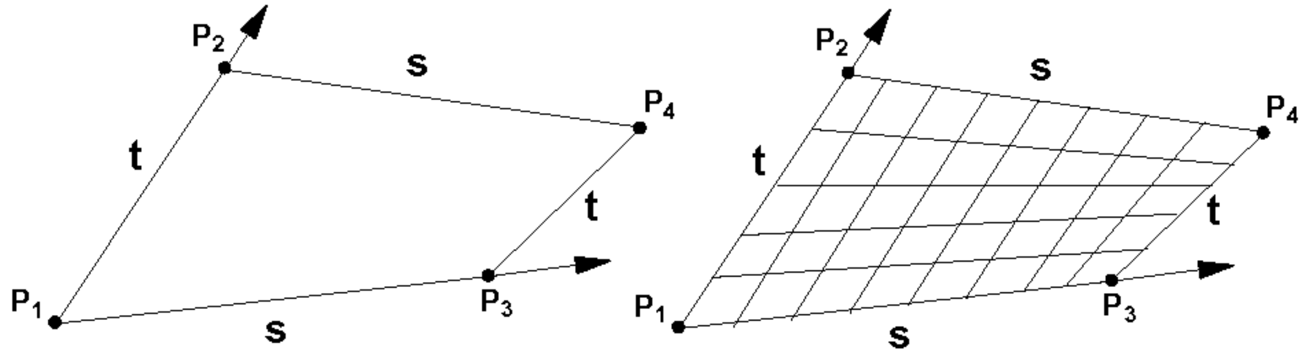


Parametric Surfaces

Bilinear Patch

- Perhaps the easiest example is bilinear interpolation

Bi-lerp a (typically non-planar) quadrilateral

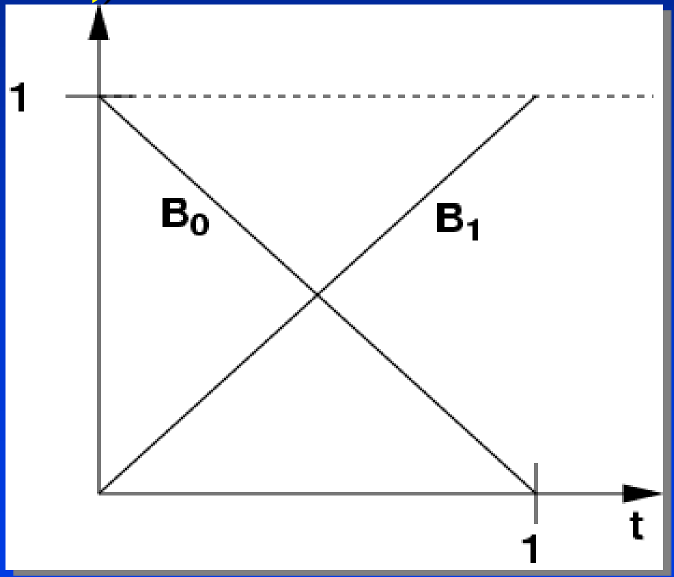


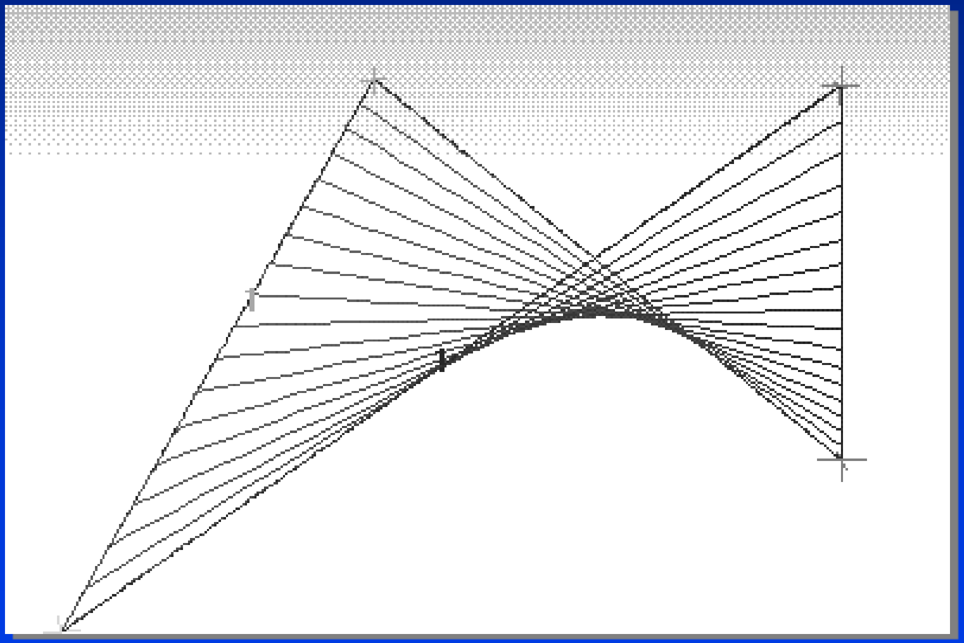
Notation: $\mathbf{L}(P_1, P_2, \alpha) \equiv (1-\alpha)P_1 + \alpha P_2$

$$Q(s,t) = \mathbf{L}(\mathbf{L}(P_1, P_2, t), L(P_3, P_4, t), s)$$

Bilinear Patch

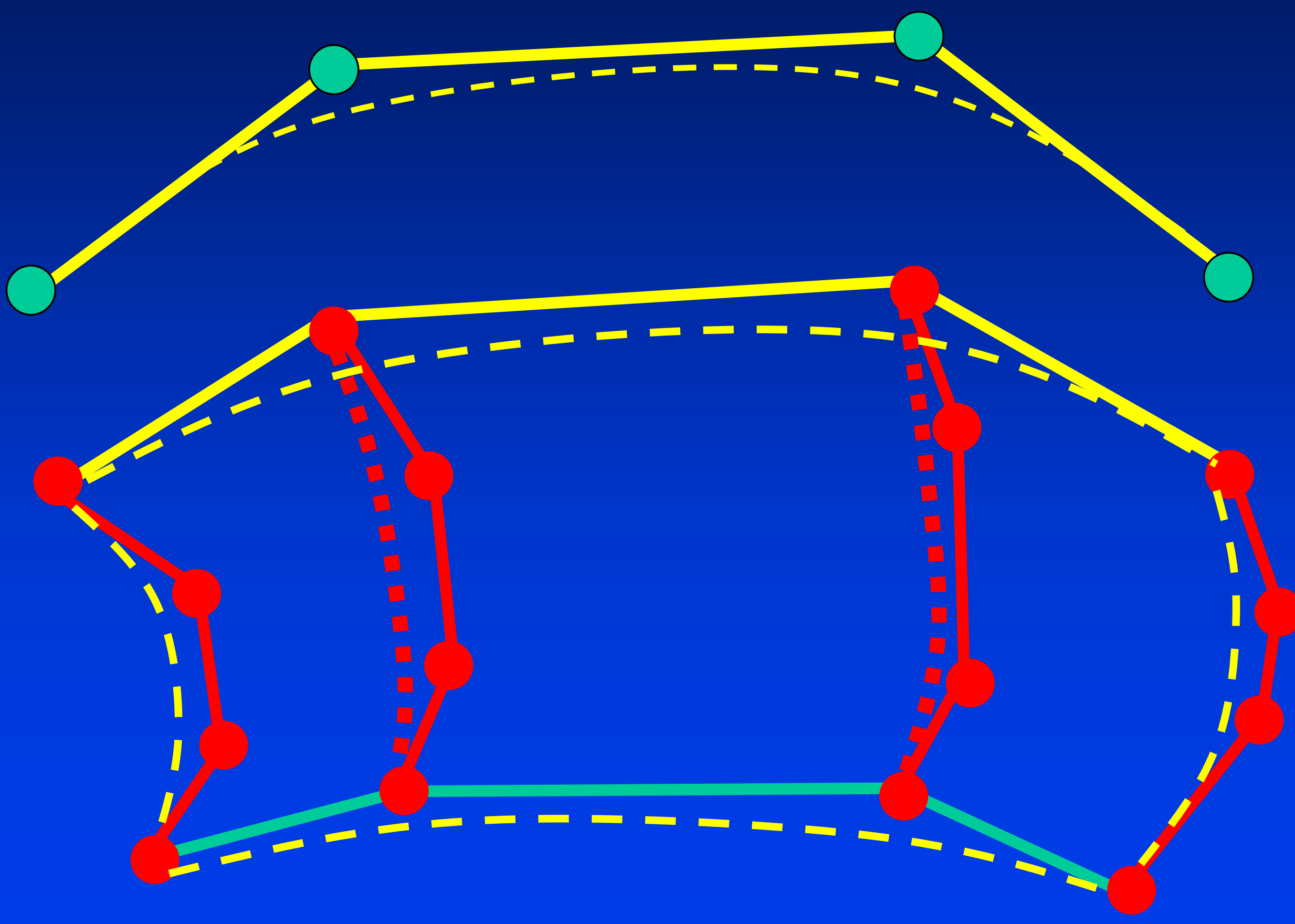
- Smooth version of quadrilateral with non-planar vertices... (four points are NOT on the same plane)





  – But will this help us model smooth surfaces?
  – Do we have control of the derivative at the edges?

From Curve to Surface



Parametric Representations

- Hermit curves and surfaces (S.A.Coons[63] and J.C.Ferguson[64])
- Bézier curves and surfaces (P.Bézier[66] and P.de Casteljau[59])
- B-Splines (W.J.Gordon and R.F.Riesenfeld 70s)
- NURBS (Versprille 75)
- Mathematical foundations (M.G.Cox[72], C.de Boor[72], et al)

Parametric Representation

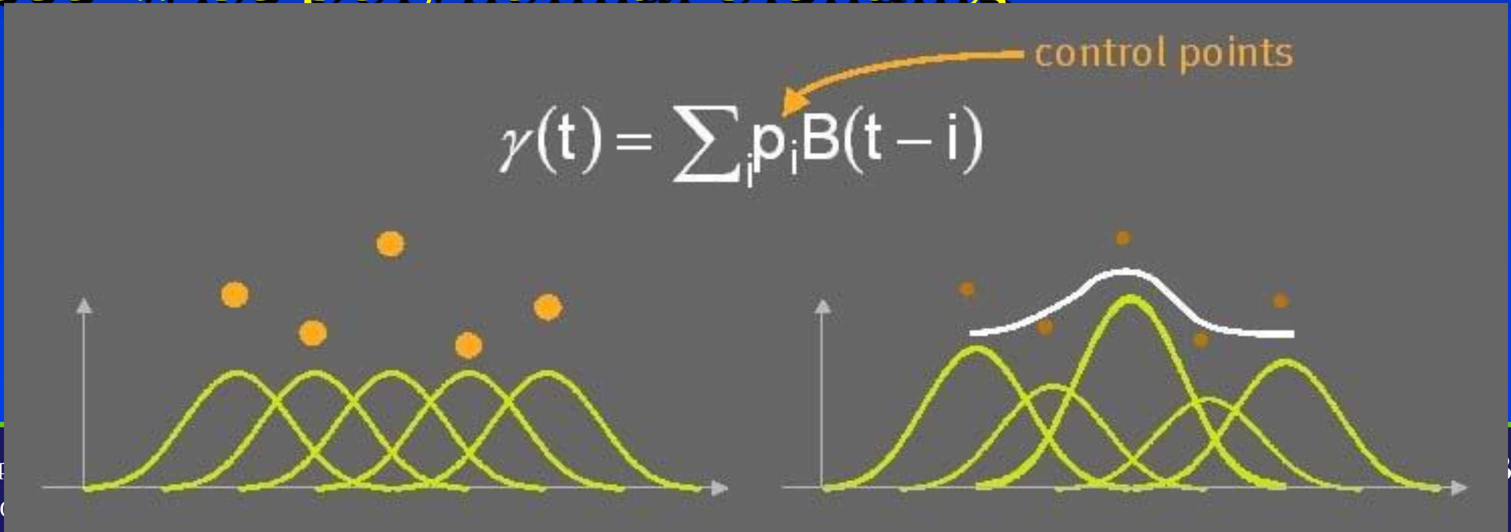
- Parametric curve functions

$$x = x(u), y = y(u), z = z(u)$$

- Parametric surface functions

$$x = x(u,v), y = y(u,v), z = z(u,v)$$

- Piece-wise polynomial blending



Surfaces

- From curves to surfaces
- A simple curve example (Bezier)

$$\mathbf{c}(u) = \sum_{i=0}^{3} \mathbf{p}_i B_i(u)$$
$$u \in [0,1]$$

- Consider each control point now becoming a Bezier curve

$$\mathbf{p}_i = \sum_{j=0}^{3} \mathbf{p}_{i,j} B_j(v)$$
$$v \in [0,1]$$

Surfaces

- Then, we have

$$\mathbf{s}(u,v)=\sum_{i=0}^{3}(\sum_{j=0}^{3}\mathbf{p}_{i,j}B_j(v))B(u)=\sum_{i=0}^{3}\sum_{j=0}^{3}\mathbf{p}_{i,j}B_i(u)B_j(v)$$

- Matrix form

$$\mathbf{s}(u,v)=\begin{bmatrix}B_0(u) & B_1(u) & B_2(u) & B_3(u)\end{bmatrix}\begin{bmatrix}\mathbf{p}_{0,0} & \mathbf{p}_{0,1} & \mathbf{p}_{0,2} & \mathbf{p}_{0,3}\\ \mathbf{p}_{1,0} & \mathbf{p}_{1,1} & \mathbf{p}_{1,2} & \mathbf{p}_{1,3}\\ \mathbf{p}_{2,0} & \mathbf{p}_{2,1} & \mathbf{p}_{2,2} & \mathbf{p}_{2,3}\\ \mathbf{p}_{3,0} & \mathbf{p}_{3,1} & \mathbf{p}_{3,2} & \mathbf{p}_{3,3}\end{bmatrix}\begin{bmatrix}B_0(v)\\ B_1(v)\\ B_2(v)\\ B_3(v)\end{bmatrix}$$

$$=UMPM^TV^T$$

Surfaces

- Further generalize to degree of n and m along two parametric directions

$$\mathbf{s}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{p}_{i,j} B_i^n(u) B_j^m(v)$$

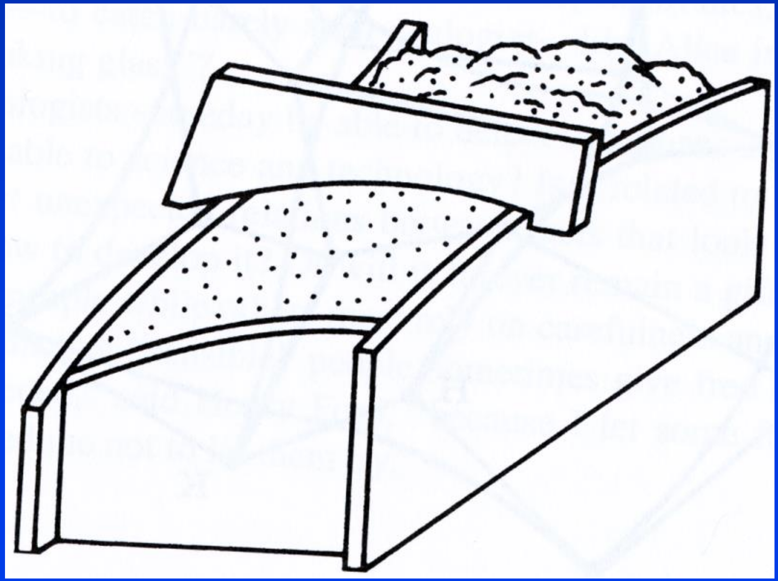
- Question: which control points are interpolated?
- How about B-spline surfaces???

Tensor-Product: Basic Concepts

- Direct generalization from two vectors:

$$\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \otimes \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \end{bmatrix} = \begin{bmatrix} a_1b_1 & a_2b_1 & a_3b_1 \\ a_1b_2 & a_2b_2 & a_3b_2 \\ a_1b_3 & a_2b_3 & a_3b_3 \\ a_1b_4 & a_2b_4 & a_3b_4 \end{bmatrix}$$

- Similarly, we can define a surface as the tensor product of two curves....



Tensor Product Surfaces

- Where are they from?
- Monomial form

$$\mathbf{s}(u,v)=\sum_i\sum_j \mathbf{a}_{i,j}u^i v^j$$

- Bezier surface

$$\mathbf{s}(u,v)=\sum_i\sum_j \mathbf{p}_{i,j}B_i^m(u)B_j^n(v)$$

- B-spline surface

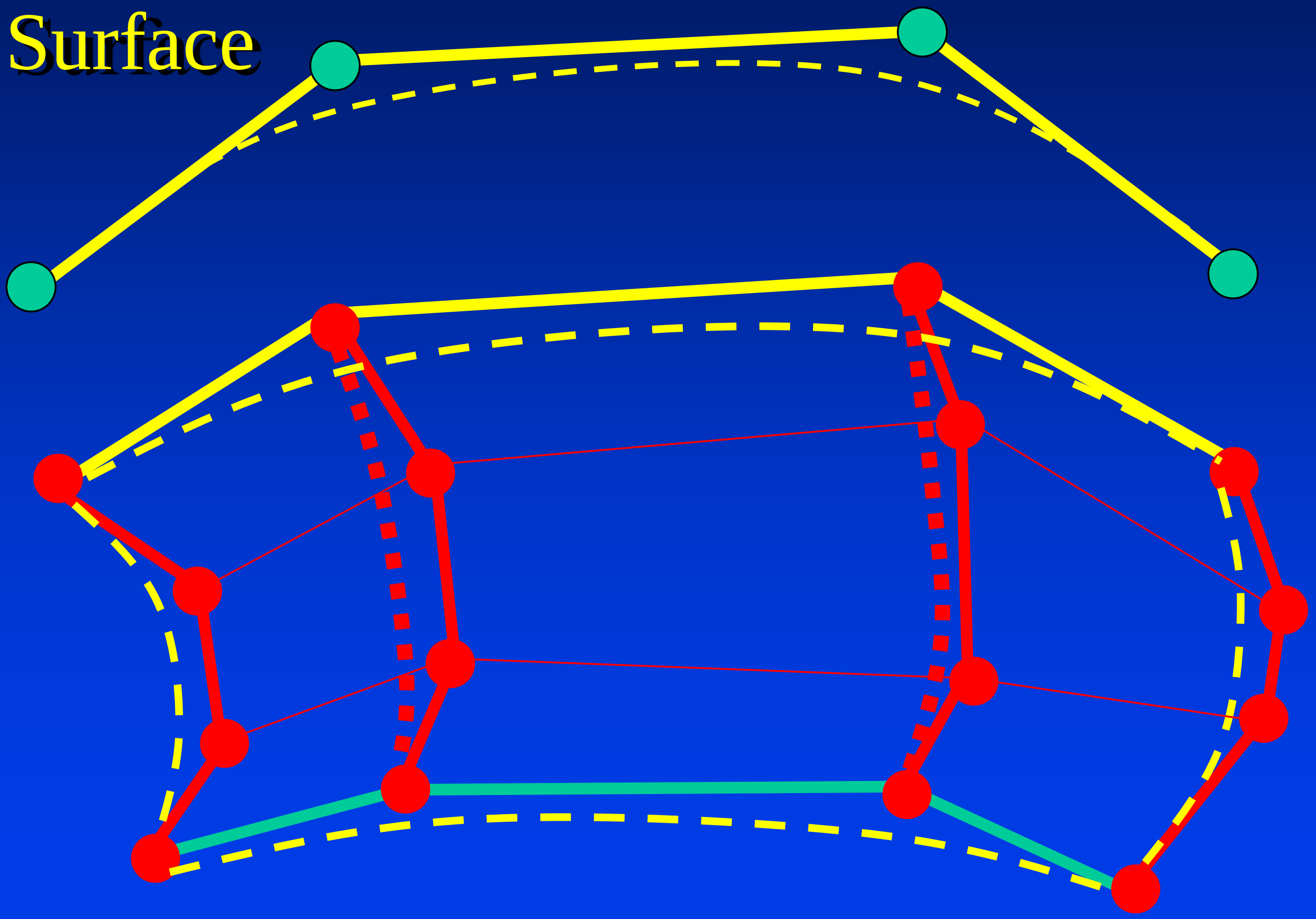
$$\mathbf{s}(u,v)=\sum_{i=0}^{m}\sum_{j=0}^{n} \mathbf{p}_{i,j}B_{i,k}(u)B_{j,l}(v)$$

- General case

$$\mathbf{s}(u,v)=\sum_i\sum_j \mathbf{v}_{i,j}F_i(u)G_j(v)$$

Tensor Product Surface

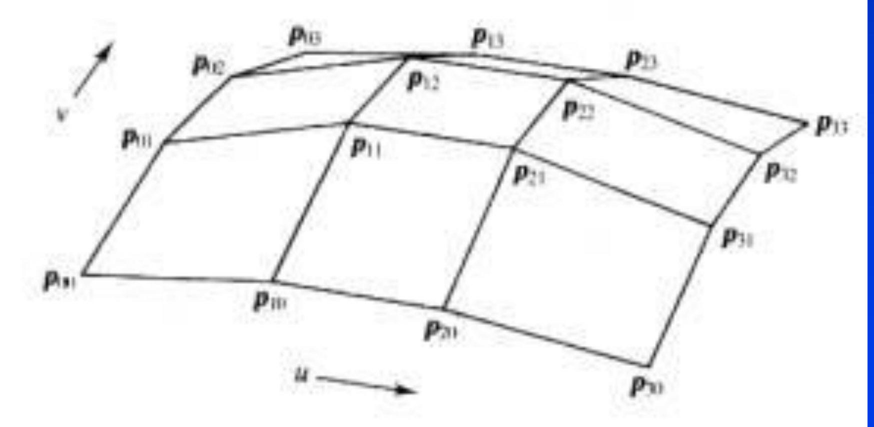
- Bezier Surface



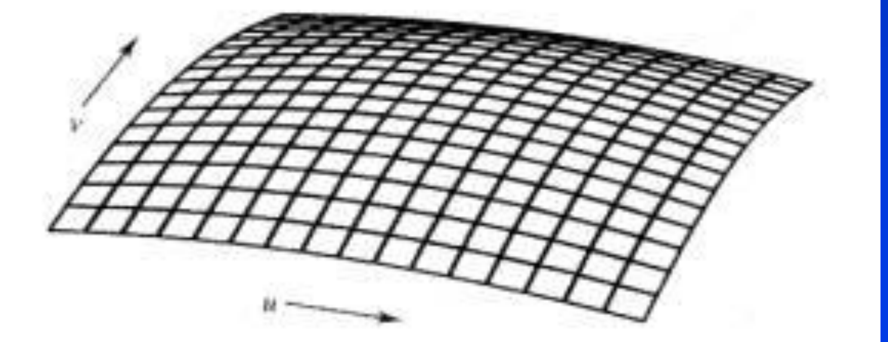
Bicubic Bezier Patch

- How do we define a tensor-product bicubic Bezier surface?

$$
\begin{aligned}
Q(s,t) = \mathbf{CB}(\quad & \mathbf{CB}(P_{00}, P_{01}, P_{02}, P_{03}, t),\\
& \mathbf{CB}(P_{10}, P_{11}, P_{12}, P_{13}, t),\\
& \mathbf{CB}(P_{20}, P_{21}, P_{22}, P_{23}, t),\\
& \mathbf{CB}(P_{30}, P_{31}, P_{32}, P_{33}, t),\\
& s)
\end{aligned}
$$

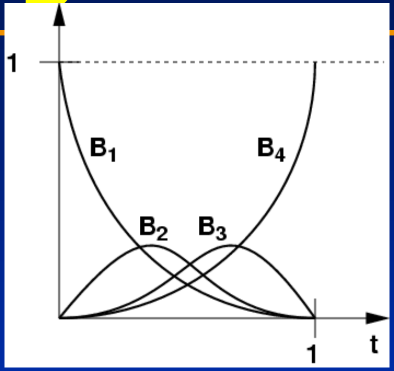


(a)

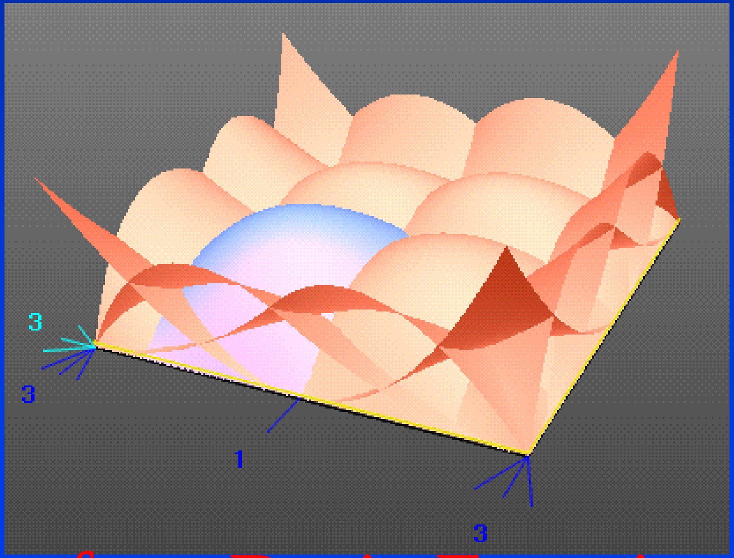


(b)

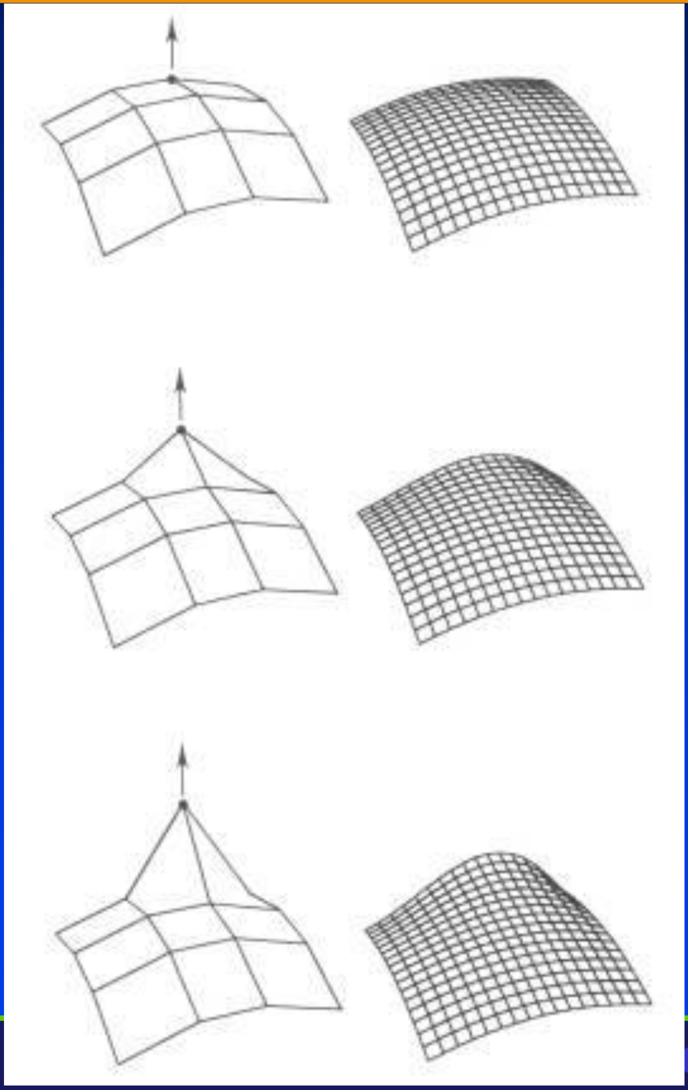
Editing Bicubic Bezier Patches



Curve Basis Functions

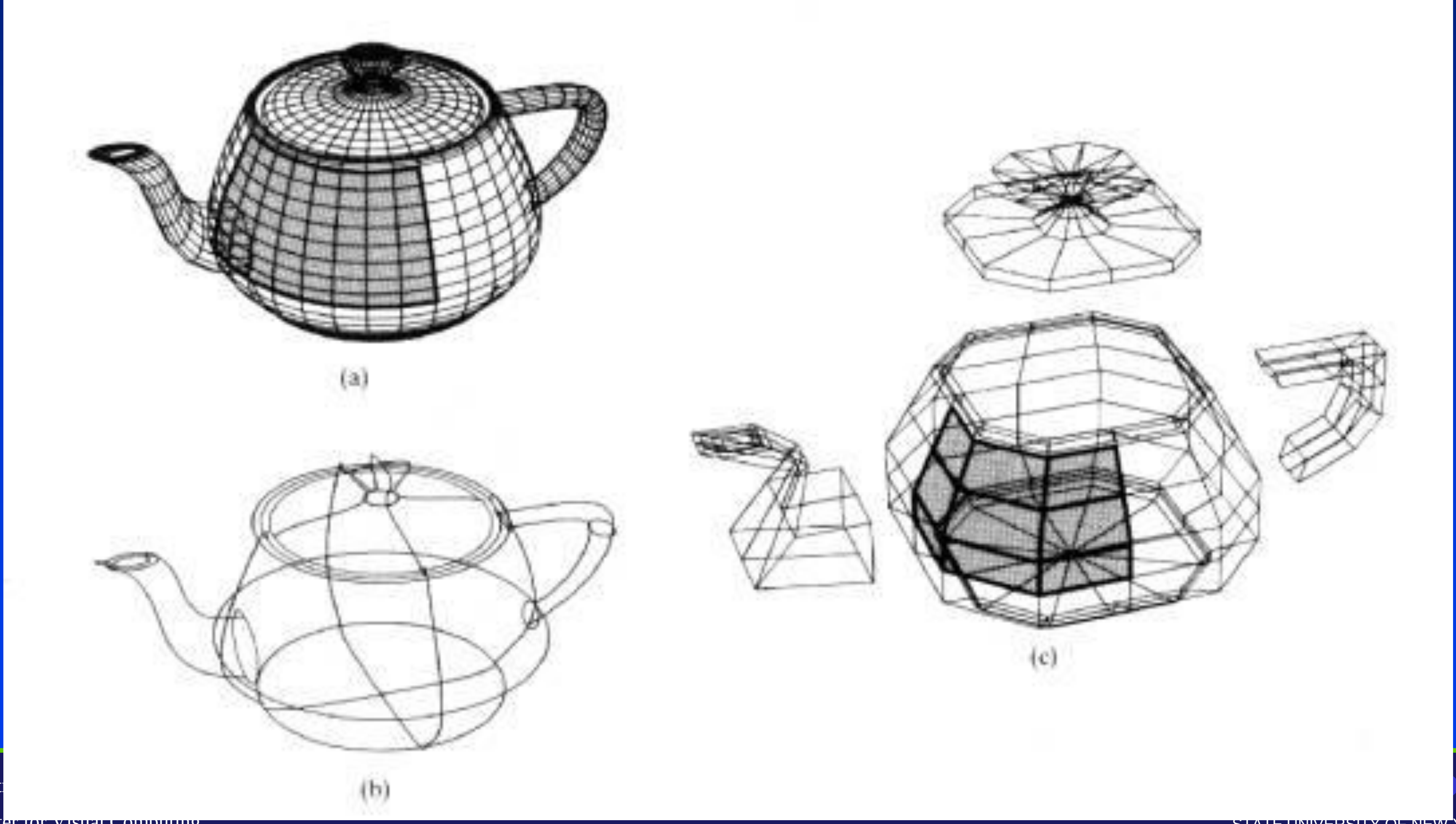


Surface Basis Functions



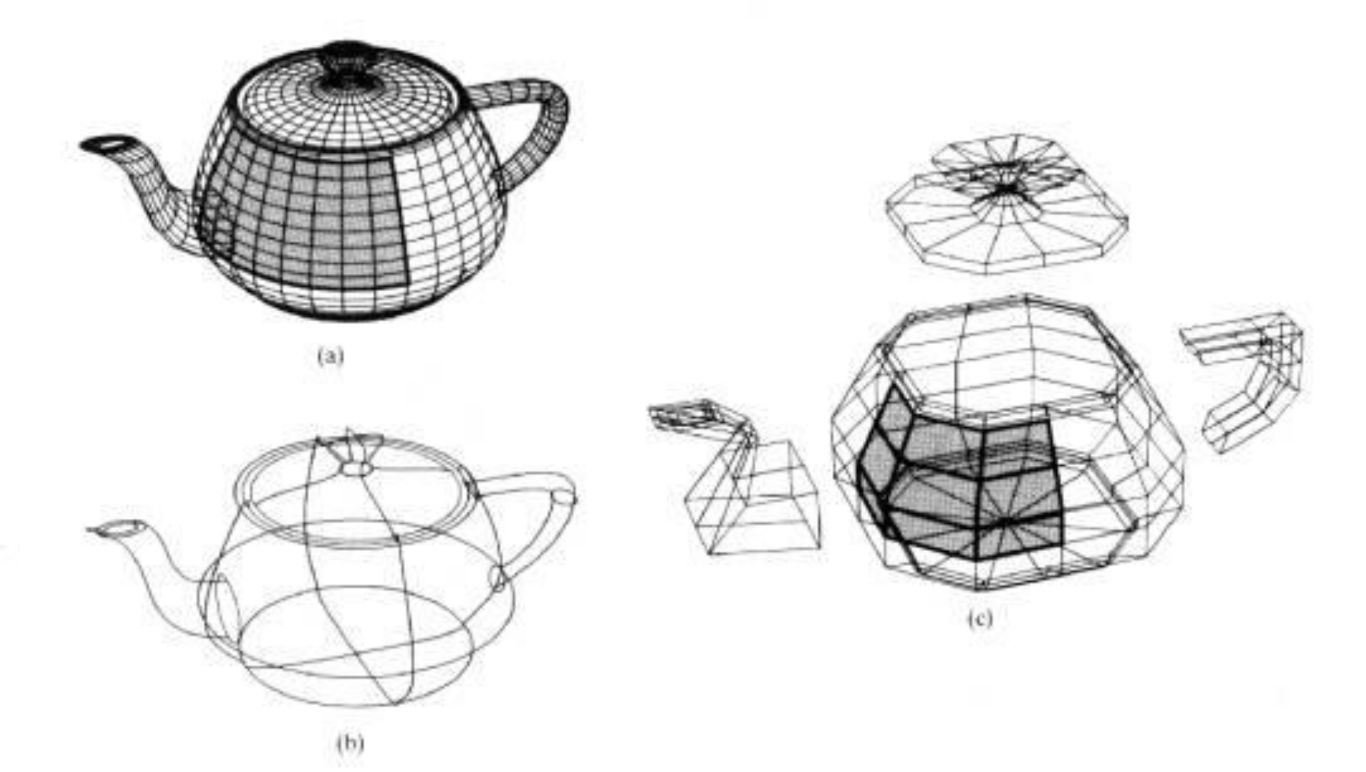
Modeling with Bicubic Bezier Patches

- Original teapot specified with Bezier patches

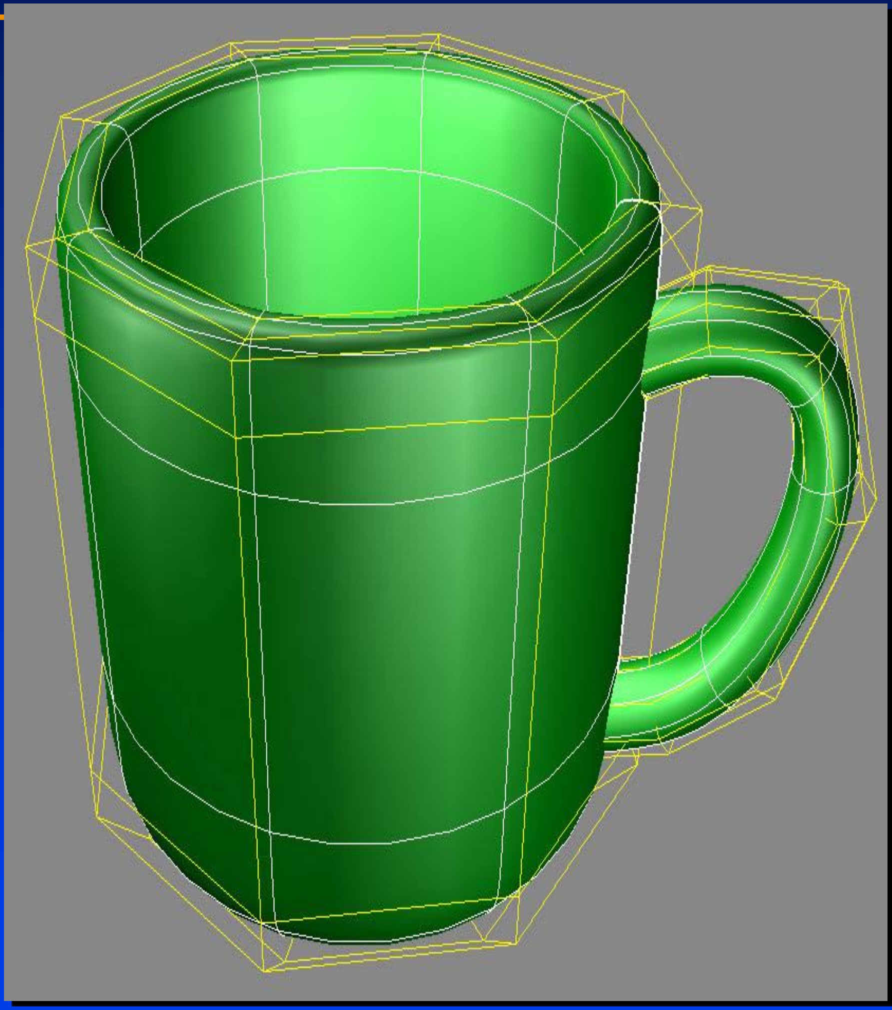


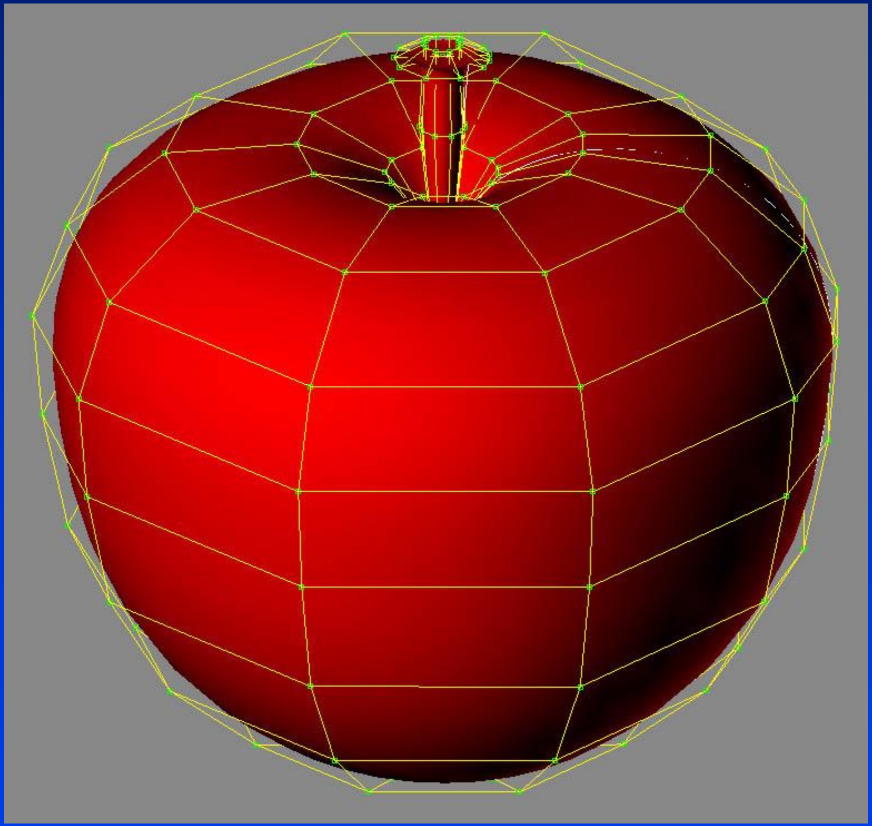
Modeling Difficulties

- Original teapot model:
- Intersecting surfaces at spout & handle, no bottom, a hole at the spout tip, a gap between lid & base

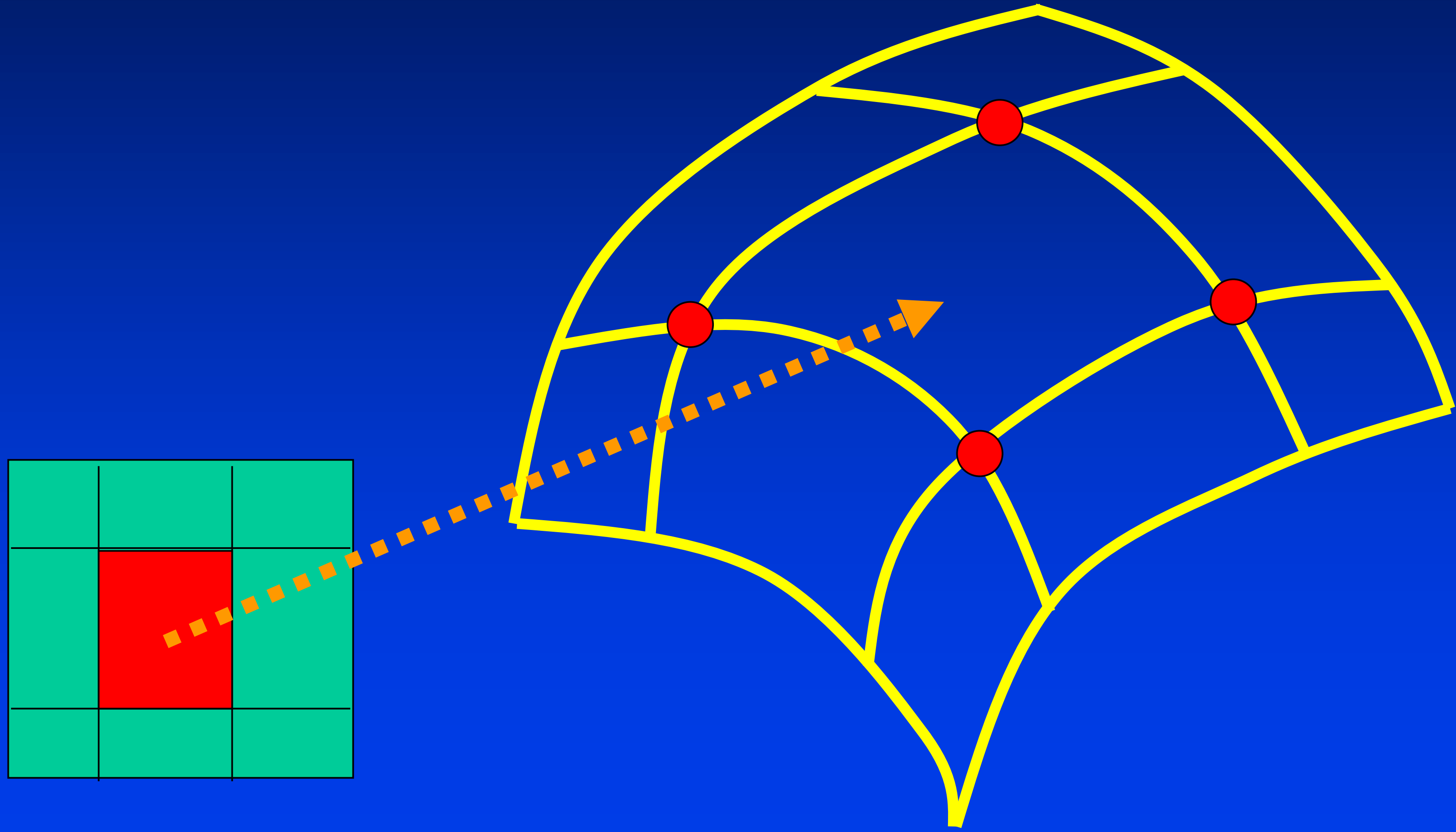


NURBS Surface Examples



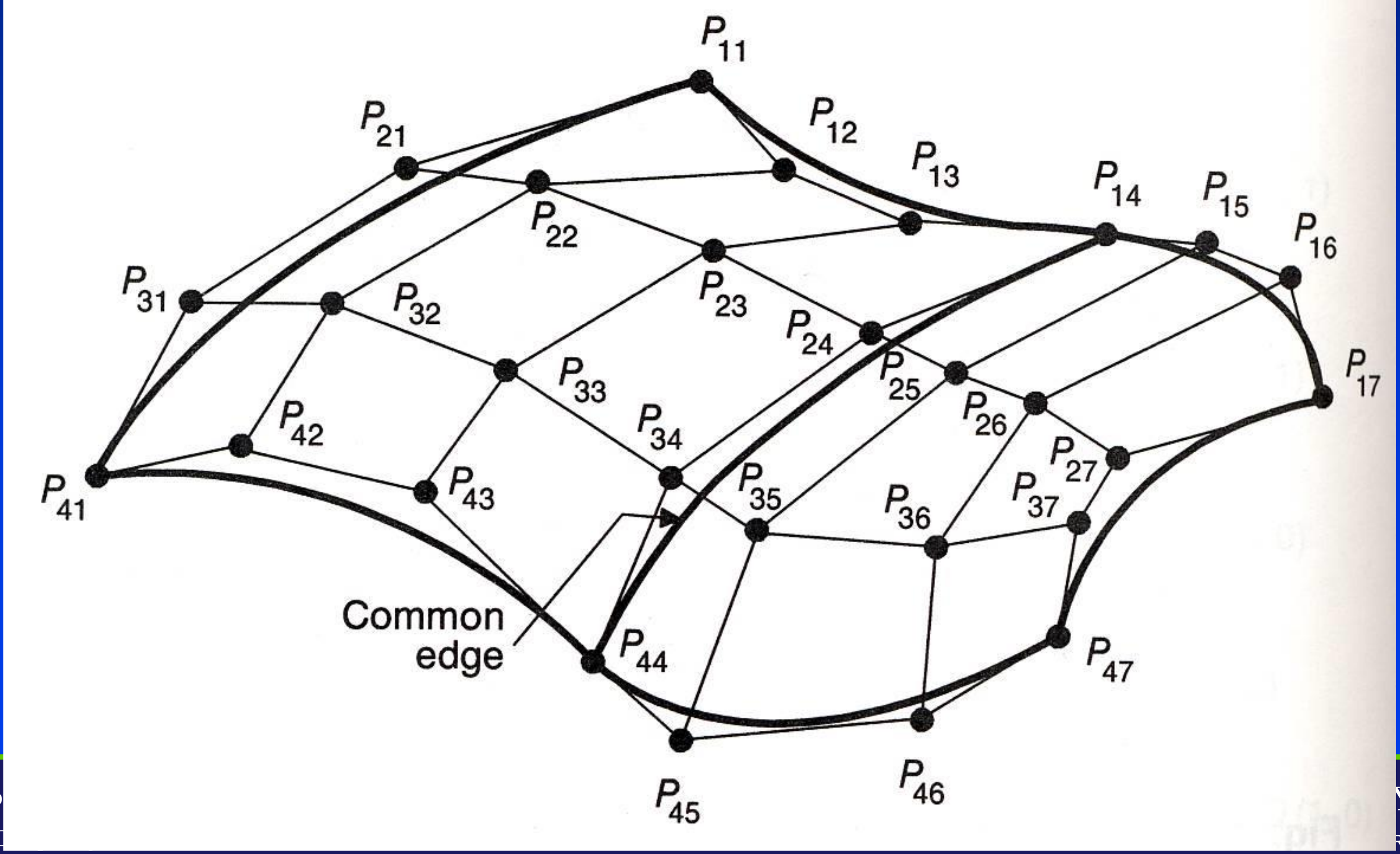


Rectangular Surface



Adjacent Bézier Patches

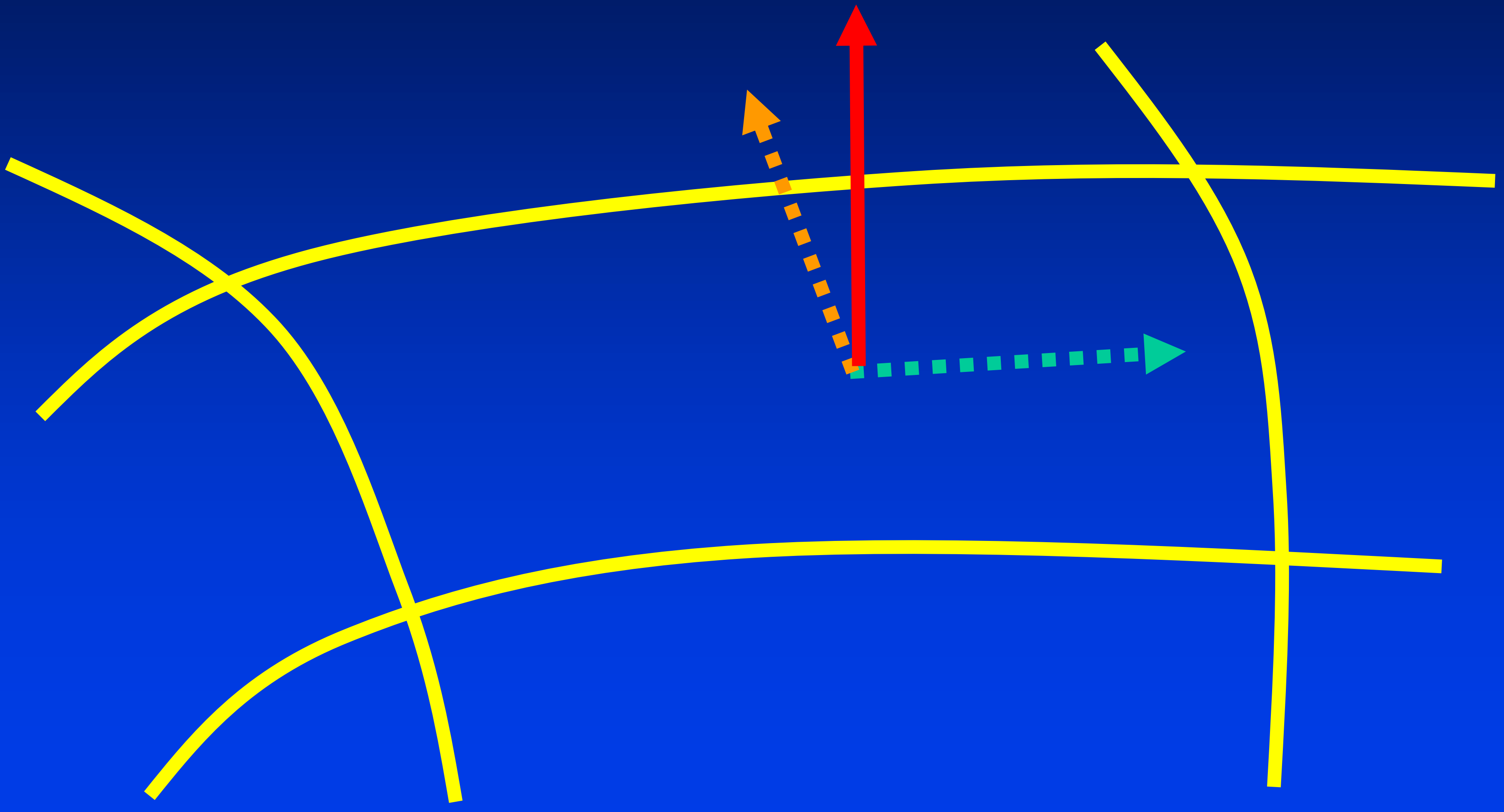
- Continuity conditions across the common, shared boundary



Rendering Curves and Surfaces

- One way of rendering a curve/surface is to compute intersections with rays from the eye through each pixel.
  - costly for real-time rendering
- Another approach is to evaluate the curve or surface at enough points to approximate it with standard flat objects (i.e. lines or polygons)
- Recursive subdivision techniques can also be used and are very efficient - good for adaptive rendering.

Surface Normal



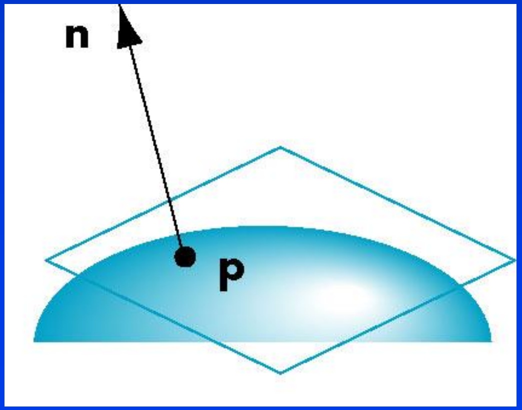
Normals

We can differentiate with respect to u and v to obtain the normal at any point **p**

$$\frac{\partial \mathbf{p}(u,v)}{\partial u} = \begin{bmatrix} \partial \mathrm{x}(u,v)/\partial u \\ \partial \mathrm{y}(u,v)/\partial u \\ \partial \mathrm{z}(u,v)/\partial u \end{bmatrix}$$

$$\frac{\partial \mathbf{p}(u,v)}{\partial v} = \begin{bmatrix} \partial \mathrm{x}(u,v)/\partial v \\ \partial \mathrm{y}(u,v)/\partial v \\ \partial \mathrm{z}(u,v)/\partial v \end{bmatrix}$$

$$\mathbf{n} = \frac{\partial \mathbf{p}(u,v)}{\partial u} \times \frac{\partial \mathbf{p}(u,v)}{\partial v}$$



Normals to Surfaces

$$\frac{\partial}{\partial s}Q(s,t) = T^{\mathrm{T}} \bullet M^{\mathrm{T}} \bullet \boldsymbol{G} \bullet M \bullet \frac{\partial}{\partial s}S$$

$$= T^{\mathrm{T}} \bullet M^{\mathrm{T}} \bullet \boldsymbol{G} \bullet M \bullet \begin{bmatrix} 3s^2 & 2s & 1 & 0 \end{bmatrix}^{\mathrm{T}}$$

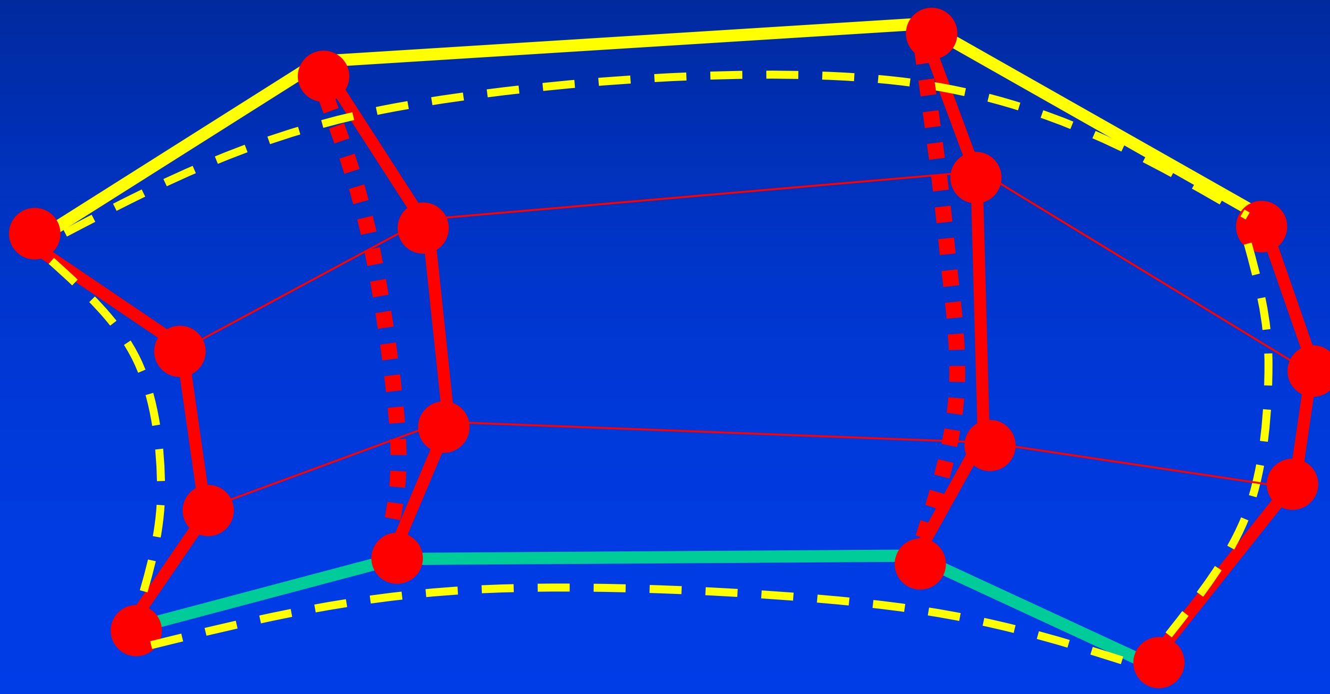
$$\frac{\partial}{\partial t}Q(s,t) = \frac{\partial}{\partial t}\left(T^{\mathrm{T}}\right) \bullet M^{\mathrm{T}} \bullet \boldsymbol{G} \bullet M \bullet S$$

$$= \begin{bmatrix} 3t^2 & 2t & 1 & 0 \end{bmatrix}^{\mathrm{T}} \bullet M^{\mathrm{T}} \bullet \boldsymbol{G} \bullet M \bullet S$$

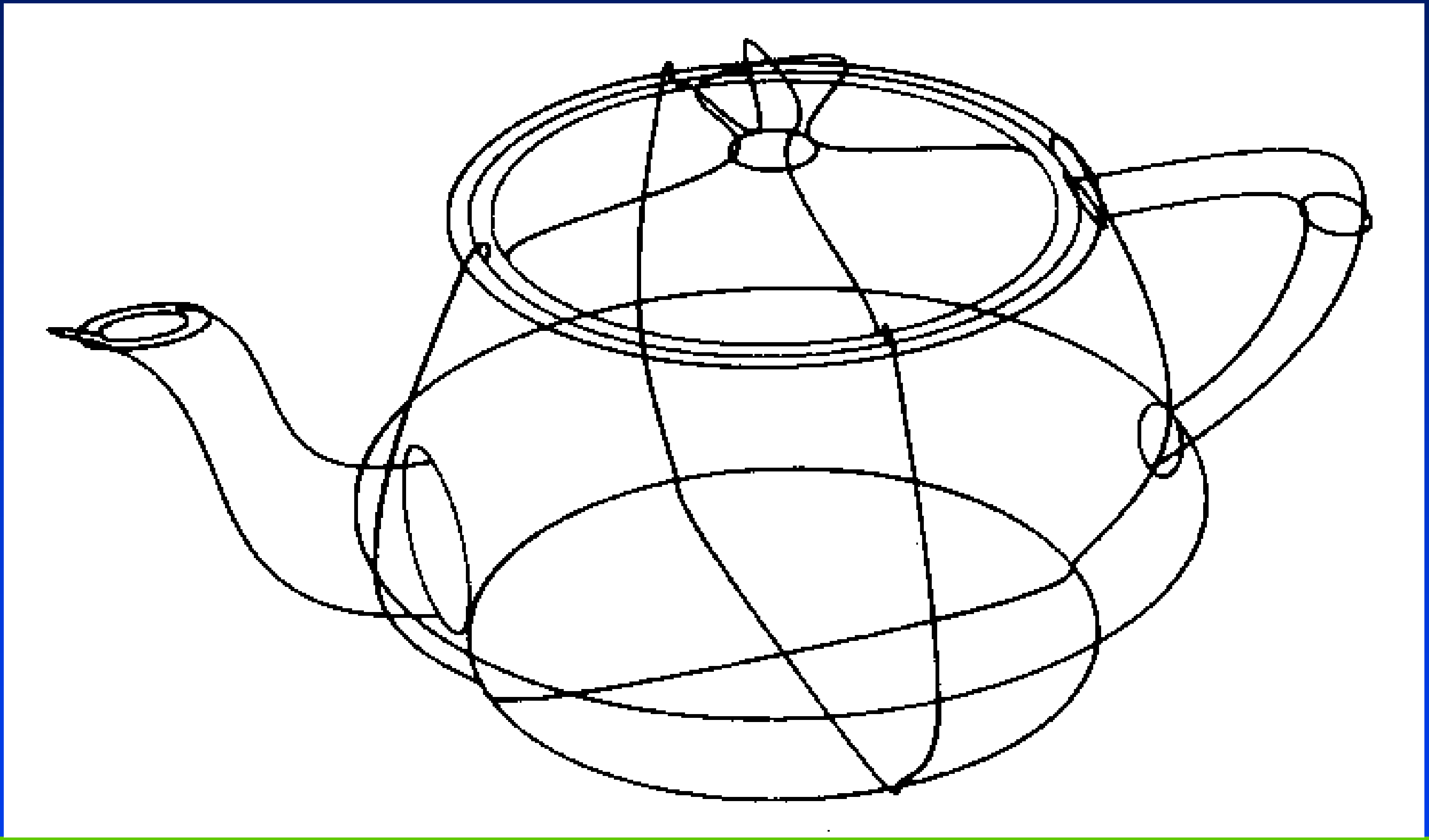
$$\frac{\partial}{\partial s}Q(s,t) \times \frac{\partial}{\partial t}Q(s,t) \longleftarrow$$ **normal vector**

Regular Surface

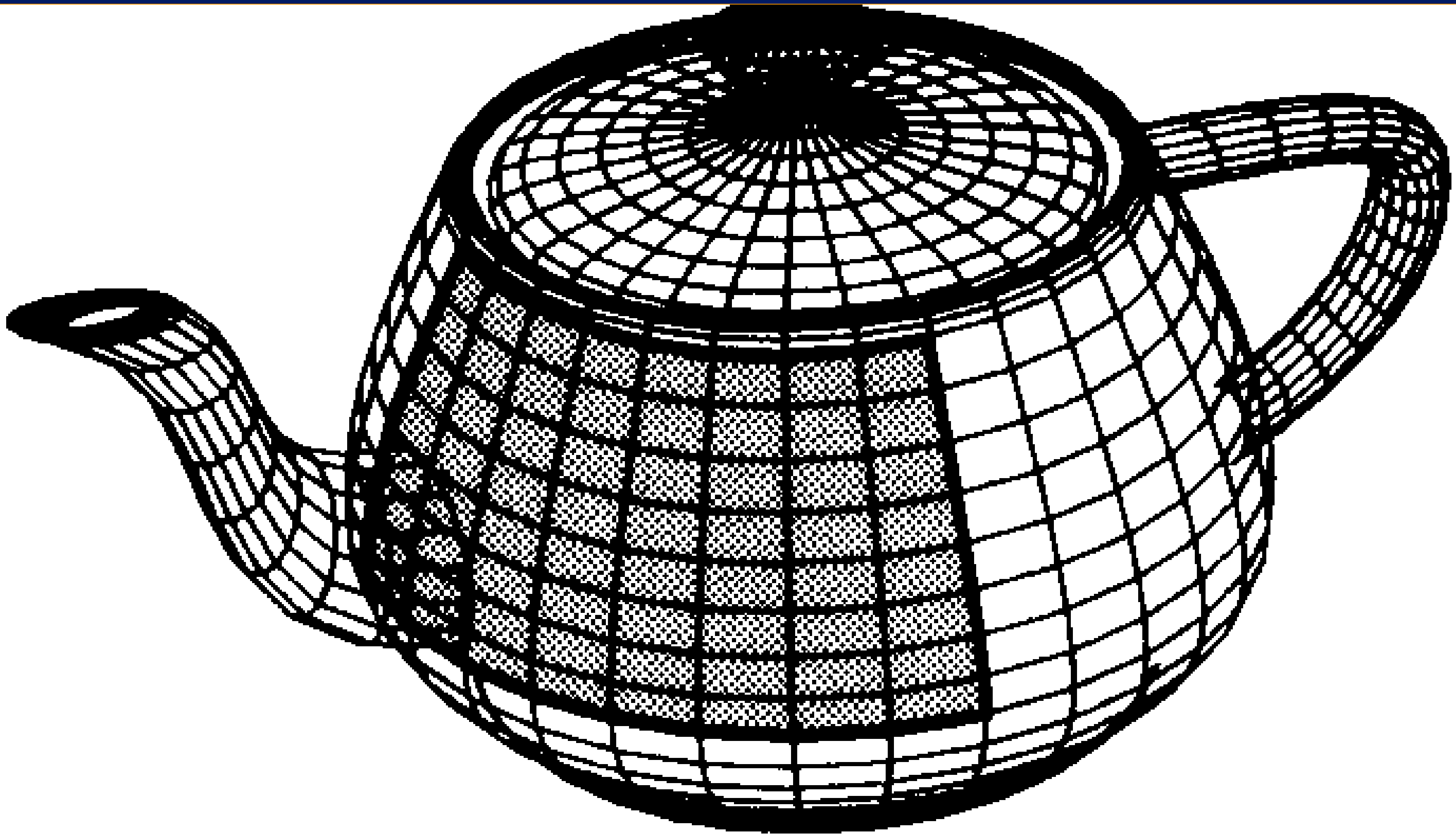
- Generated from a set of control points.



The Utah Teapot: 32 Bezier Patches



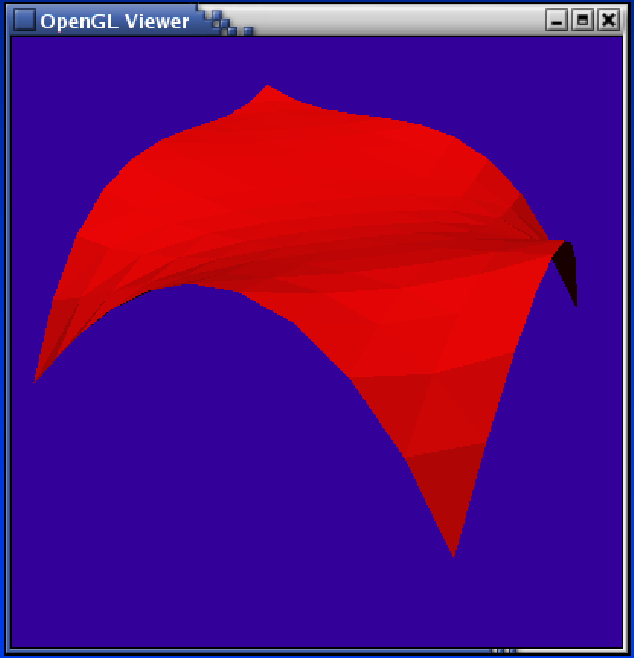
Utah Teapot: Polygon Representation



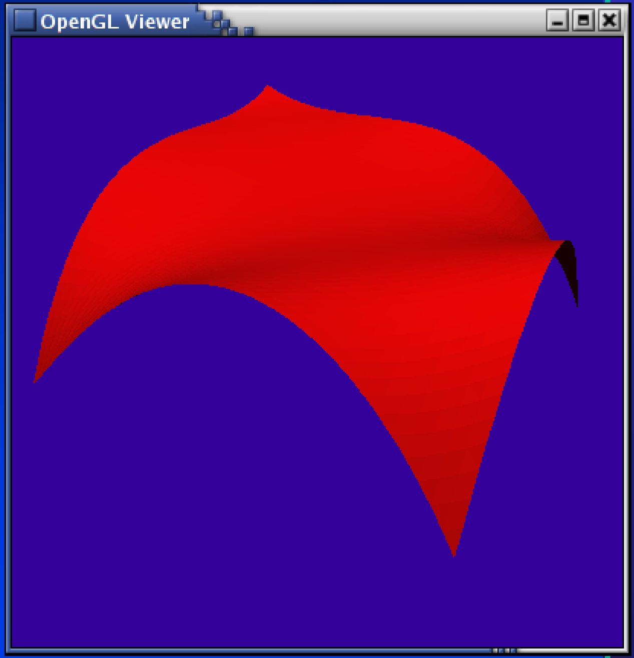
Displaying Bezier Patch

- Given 16 control points (Bicubic Bezier Patch) and a tessellation resolution, create a triangle mesh

resolution:
5x5 vertices



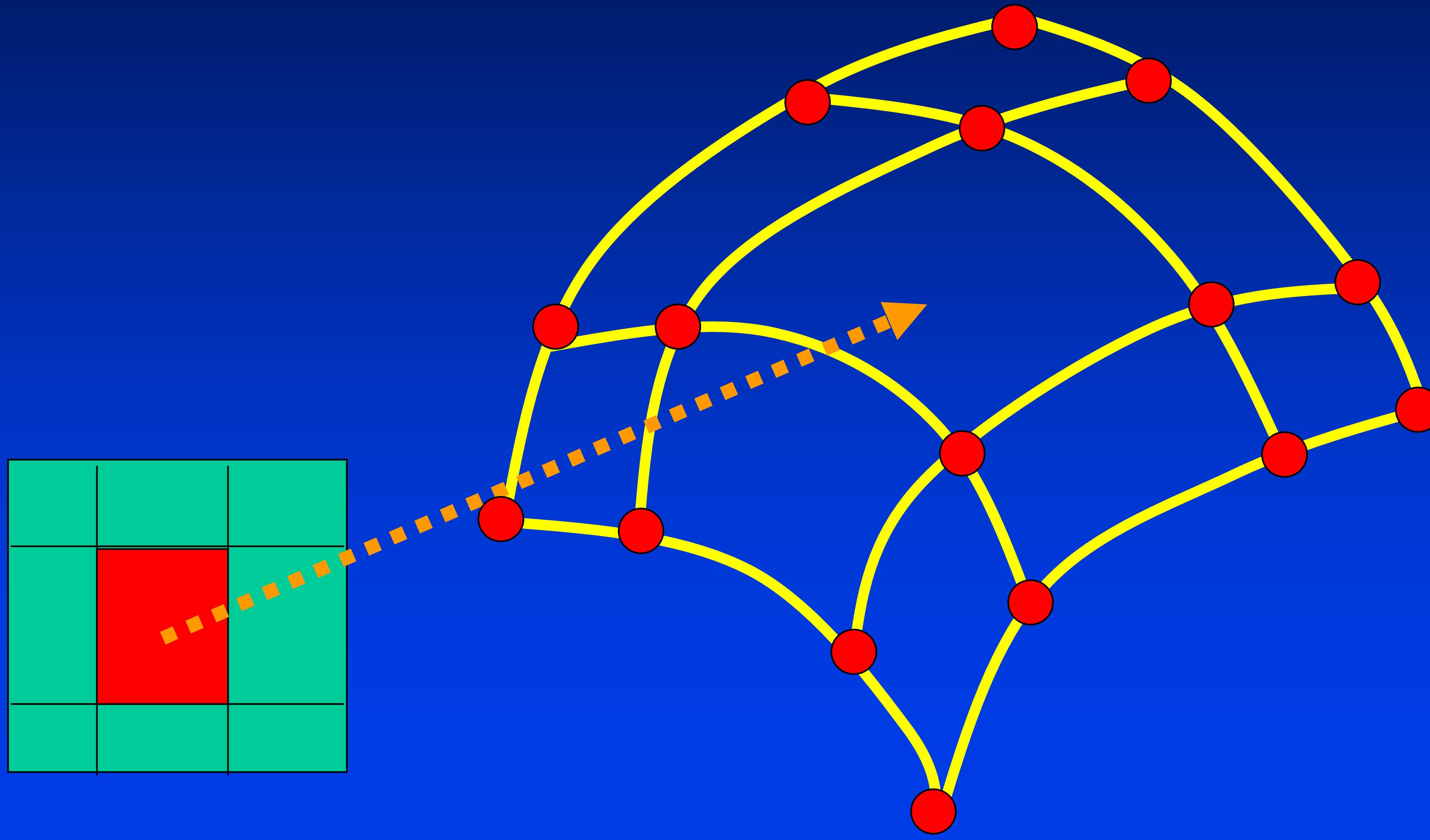
resolution:
11x11 vertices

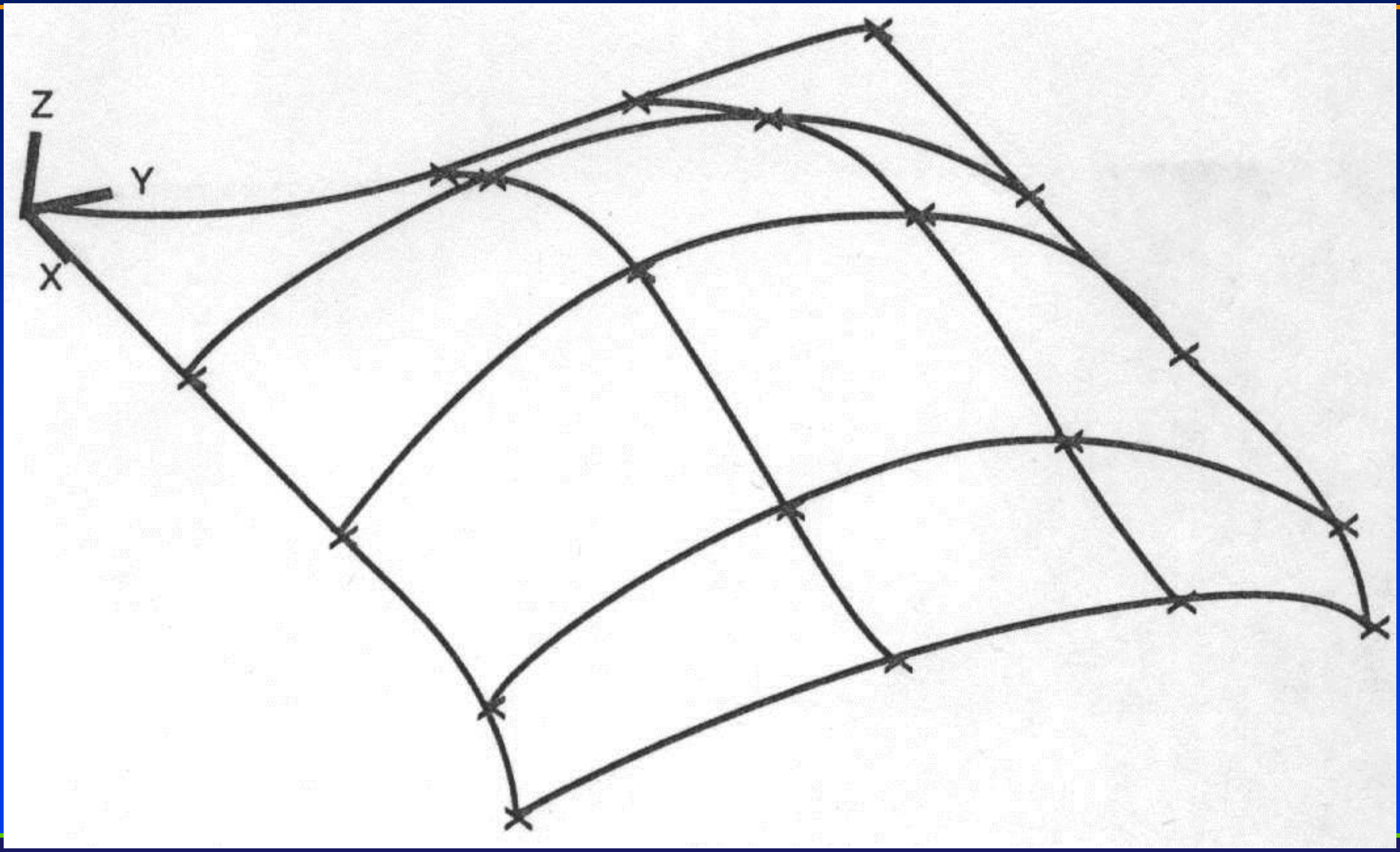


resolution:
41x41 vertices

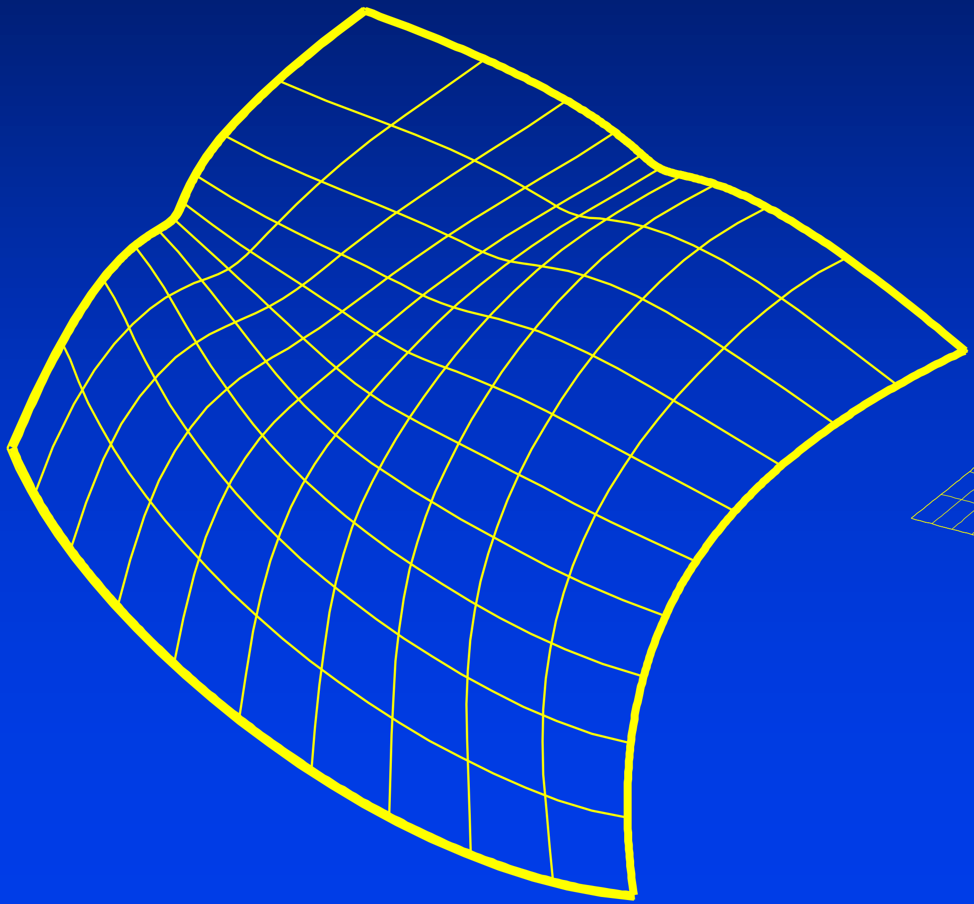
Rendering the Teapot

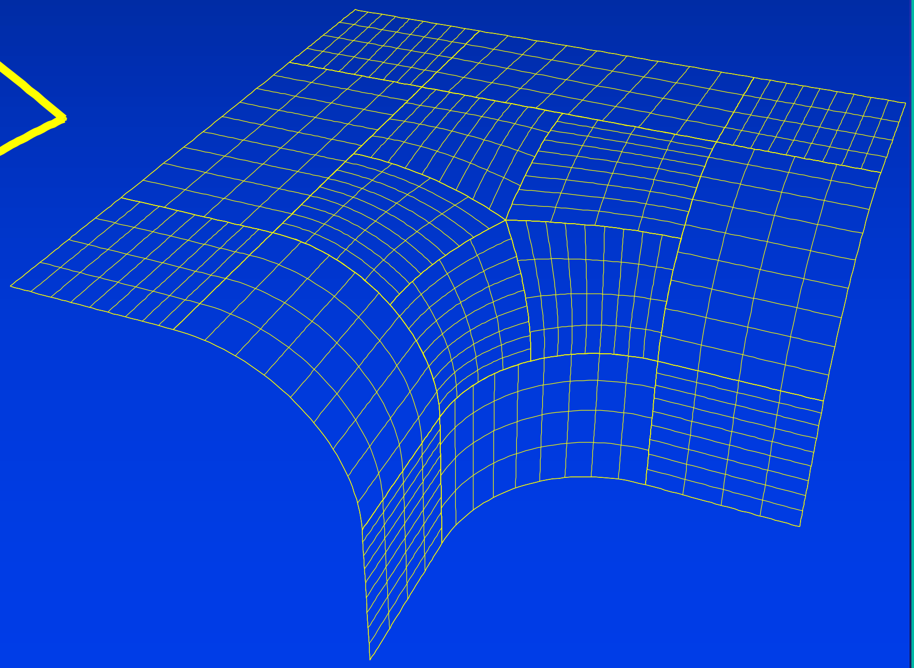
Curve Network



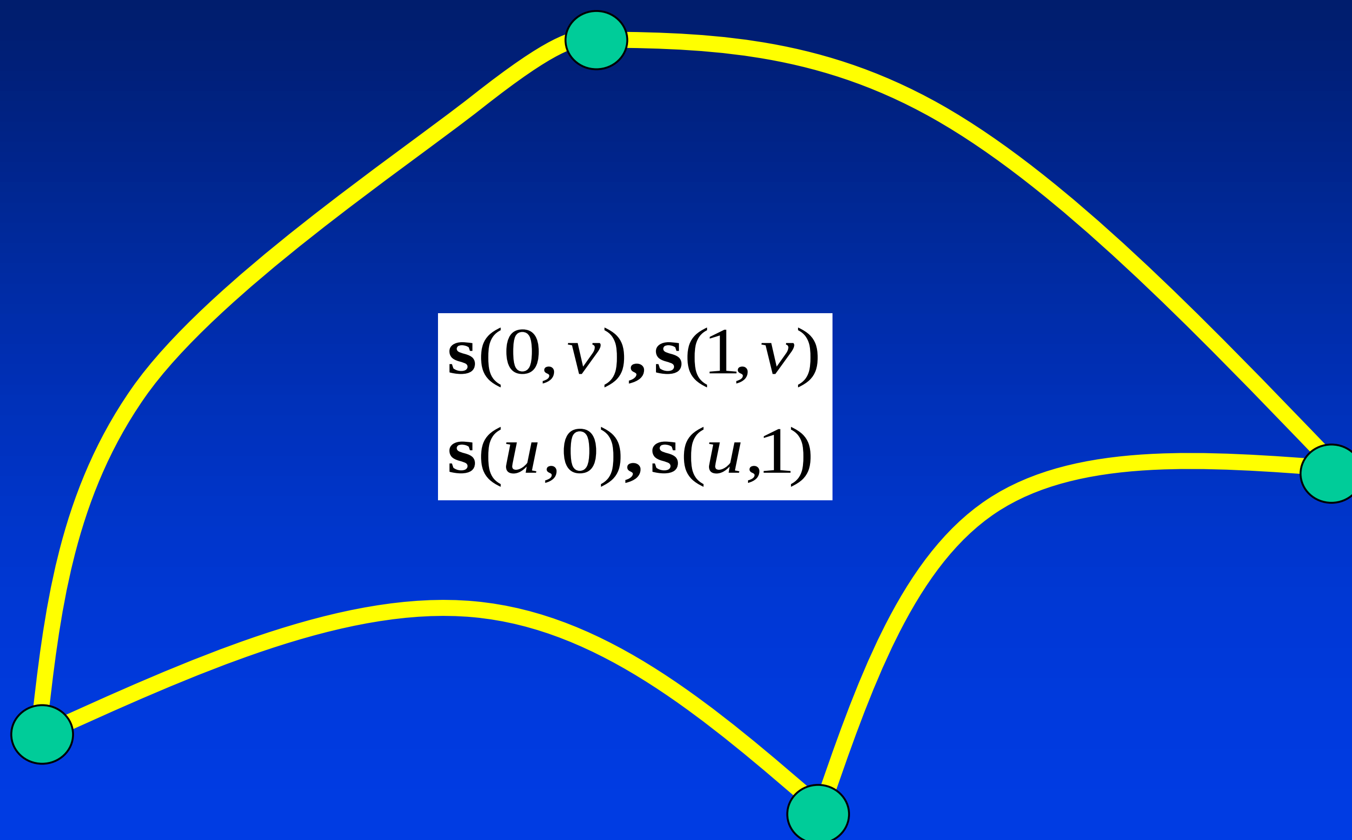

Z
Y
X

Transfinite Method and N-side Hole Filling

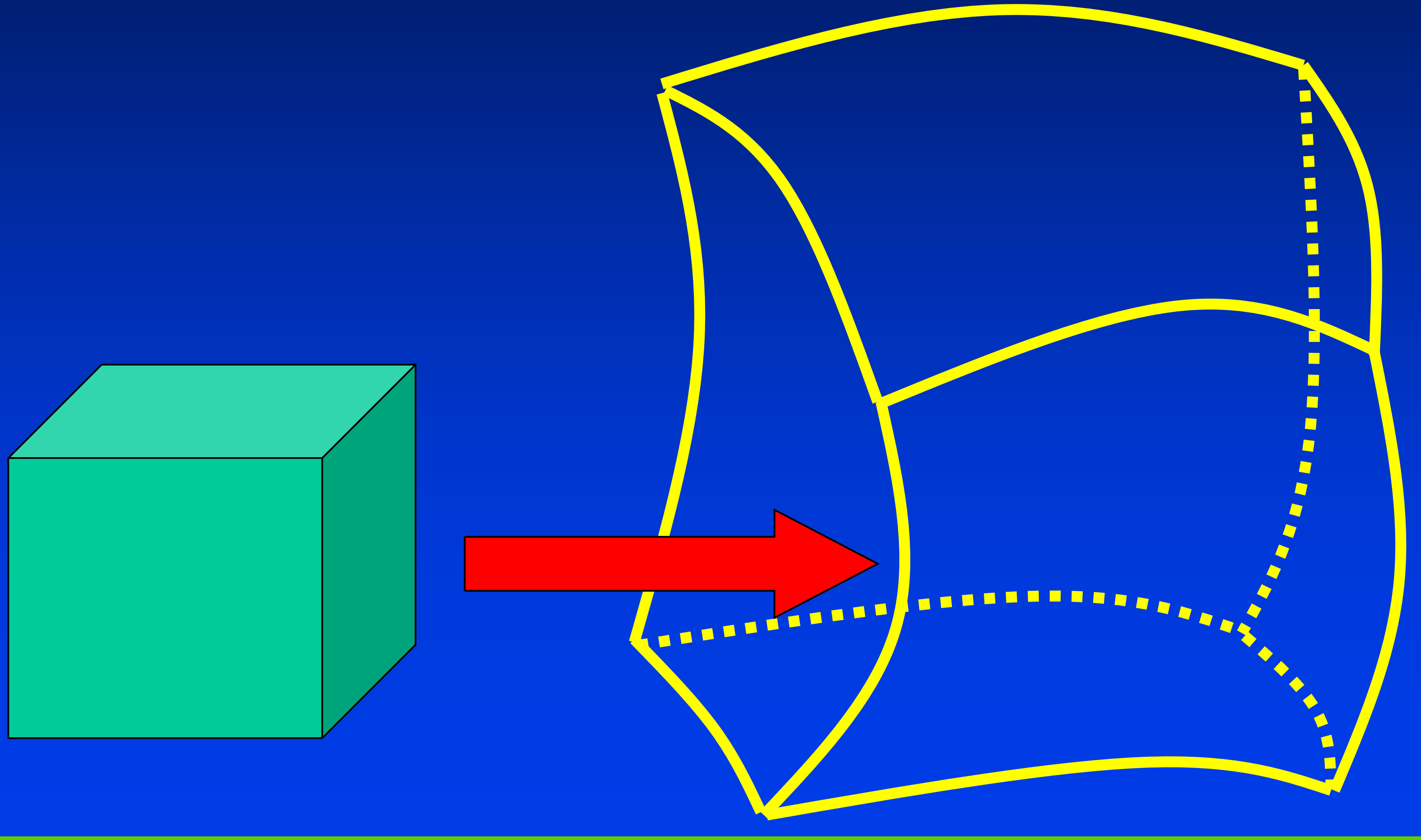




Coons Patch



Solid



Parametric Solids

- Tricubic solid

$$\mathbf{p}(u,v,w) = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3}\mathbf{a}_{ijk}u^i v^j w^k$$
$$u,v,w \in [0,1]$$

- Bezier solid

$$\mathbf{p}(u,v,w) = \sum_{i}\sum_{j}\sum_{k}\mathbf{p}_{ijk}B_i(u)B_j(v)B_k(w)$$

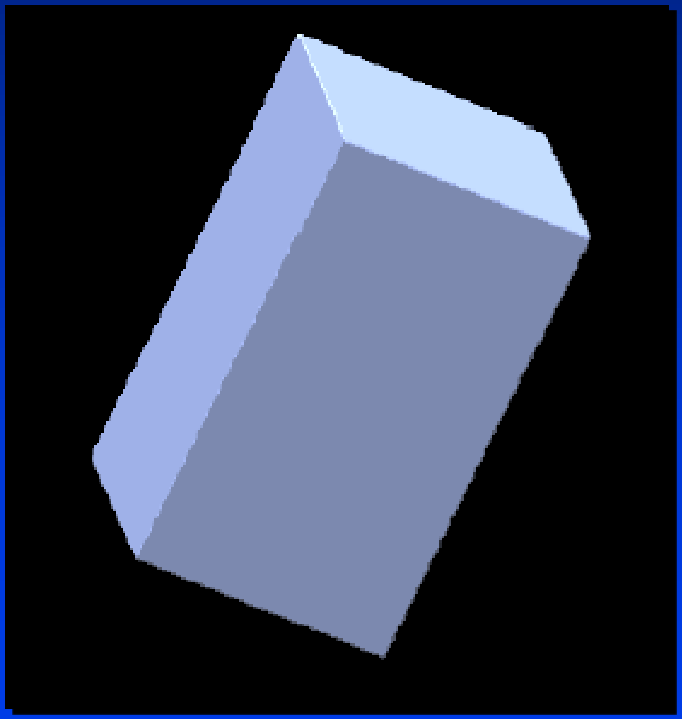
- B-spline solid

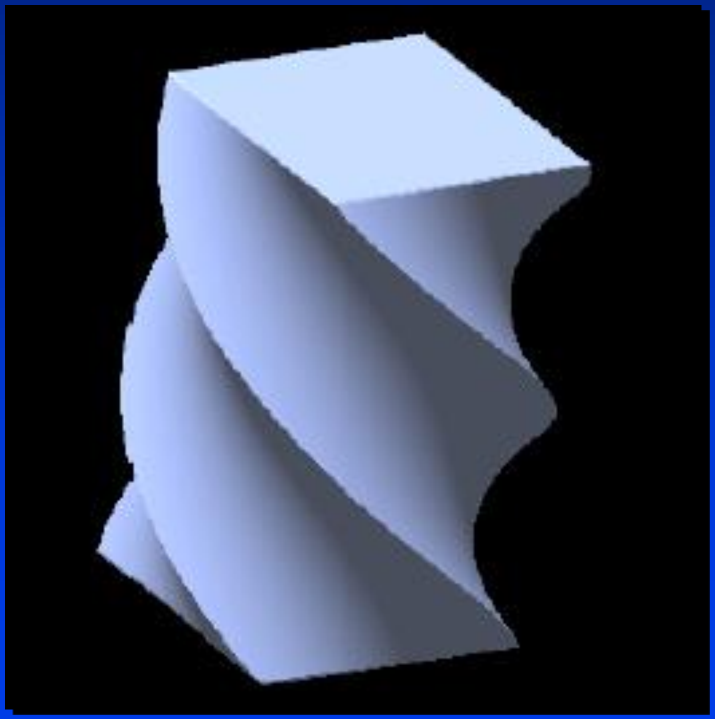
$$\mathbf{p}(u,v,w) = \sum_{i}\sum_{j}\sum_{k}\mathbf{p}_{ijk}B_{i,I}(u)B_{j,J}(v)B_{k,K}(w)$$

- NURBS solid

$$\mathbf{p}(u,v,w) = \frac{\sum_{i}\sum_{j}\sum_{k}\mathbf{p}_{ijk}q_{ijk}B_{i,I}(u)B_{j,J}(v)B_{k,K}(w)}{\sum_{i}\sum_{j}\sum_{k}q_{ijk}B_{i,I}(u)B_{j,J}(v)B_{k,K}(w)}$$

Free-Form Deformation





Free-form Deformation

- Any geometric objects can be embedded into a space
- The surrounding space is represented by using commonly-used, popular splines
- Free-form deformation of the surrounding space
- All the embedded (geometric) objects are deformed accordingly, the quantitative measurement of deformation is obtained from the displacement vectors of the trivariate splines that define the surrounding space
- Essentially, the deformation is governed by the trivariate, volumetric splines
- Very popular in graphics and related fields

Surrounding Space represented by Parametric Solids

- Tricubic solid

$$\mathbf{p}(u,v,w)=\sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3}\mathbf{a}_{ijk}u^i v^j w^k$$

$$u,v,w\in[0,1]$$

- Bezier solid

$$\mathbf{p}(u,v,w)=\sum_{i}\sum_{j}\sum_{k}\mathbf{p}_{ijk}B_i(u)B_j(v)B_k(w)$$

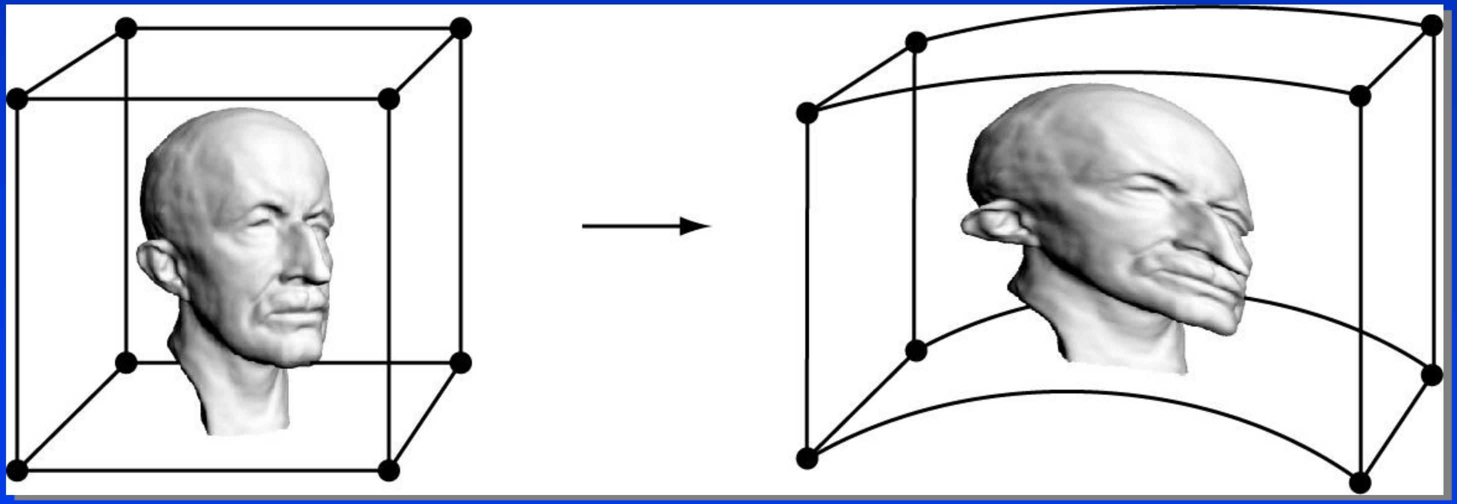
- B-spline solid

$$\mathbf{p}(u,v,w)=\sum_{i}\sum_{j}\sum_{k}\mathbf{p}_{ijk}B_{i,I}(u)B_{j,J}(v)B_{k,K}(w)$$

- NURBS solid

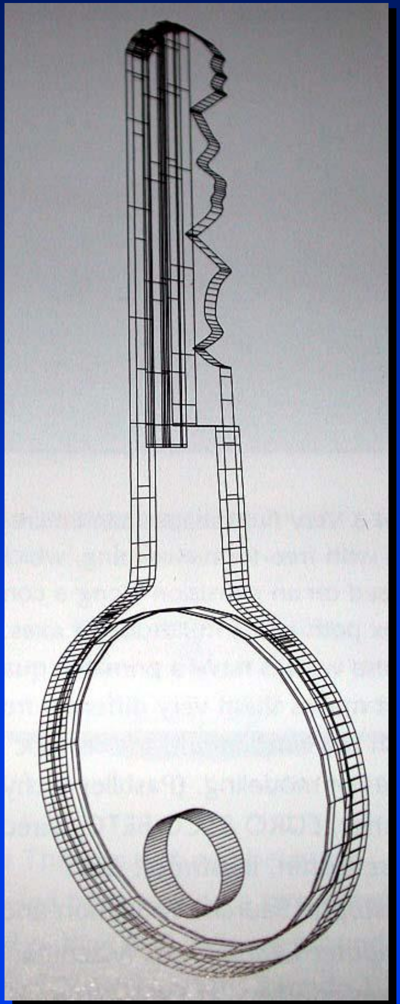
$$\mathbf{p}(u,v,w)=\frac{\sum_{i}\sum_{j}\sum_{k}\mathbf{p}_{ijk}q_{ijk}B_{i,I}(u)B_{j,J}(v)B_{k,K}(w)}{\sum_{i}\sum_{j}\sum_{k}q_{ijk}B_{i,I}(u)B_{j,J}(v)B_{k,K}(w)}$$

Free-form Deformations



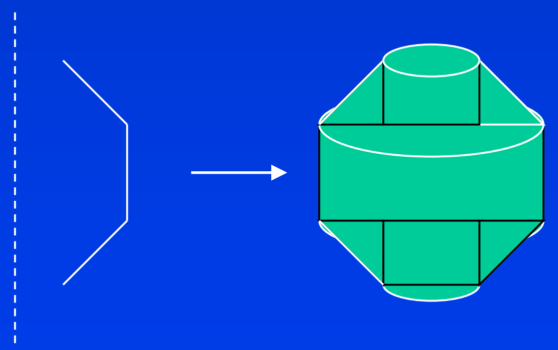
Procedural Modeling

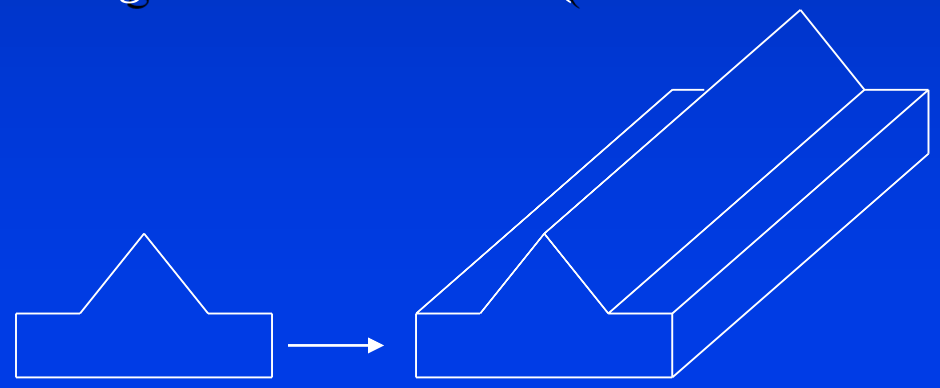
- Being applied to shape geometry
- For example, simple extrusion
- Extrude: grow a 2D shape in the third dimension



Sweeping Objects

- Define a polygon by its edges
- Sweep it along a path
- The path taken by the edges form a surface - the sweep surface
- Special cases
  - Translational sweeping – sweep along a straight line (extraction)
  - Rotational sweeping – rotate profiling curves about an axis (surface of revolution)

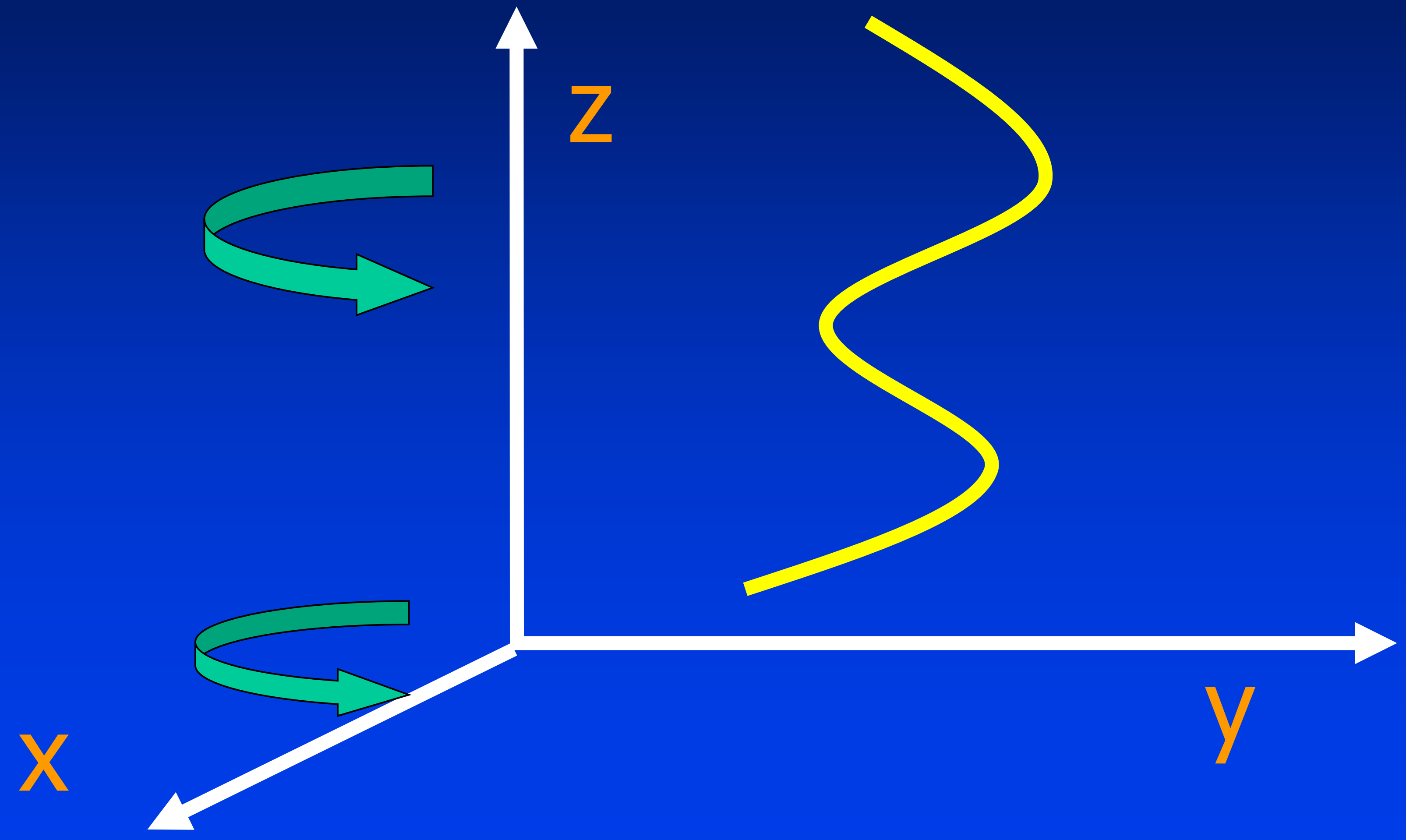




Department of Computer Science
Center for Visual Computing

STONY BROOK
STATE UNIVERSITY OF NEW YORK

Surface of Revolution



Surfaces of Revolution

- Geometric construction
  - Specify a planar curve profile on y-z plane
  - Rotate this profile with respect to z-axis
- Procedure-based model
- What kinds of shape can we model?
- Review: three dimensional rotation w.r.t. z-axis

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Surfaces of Revolution

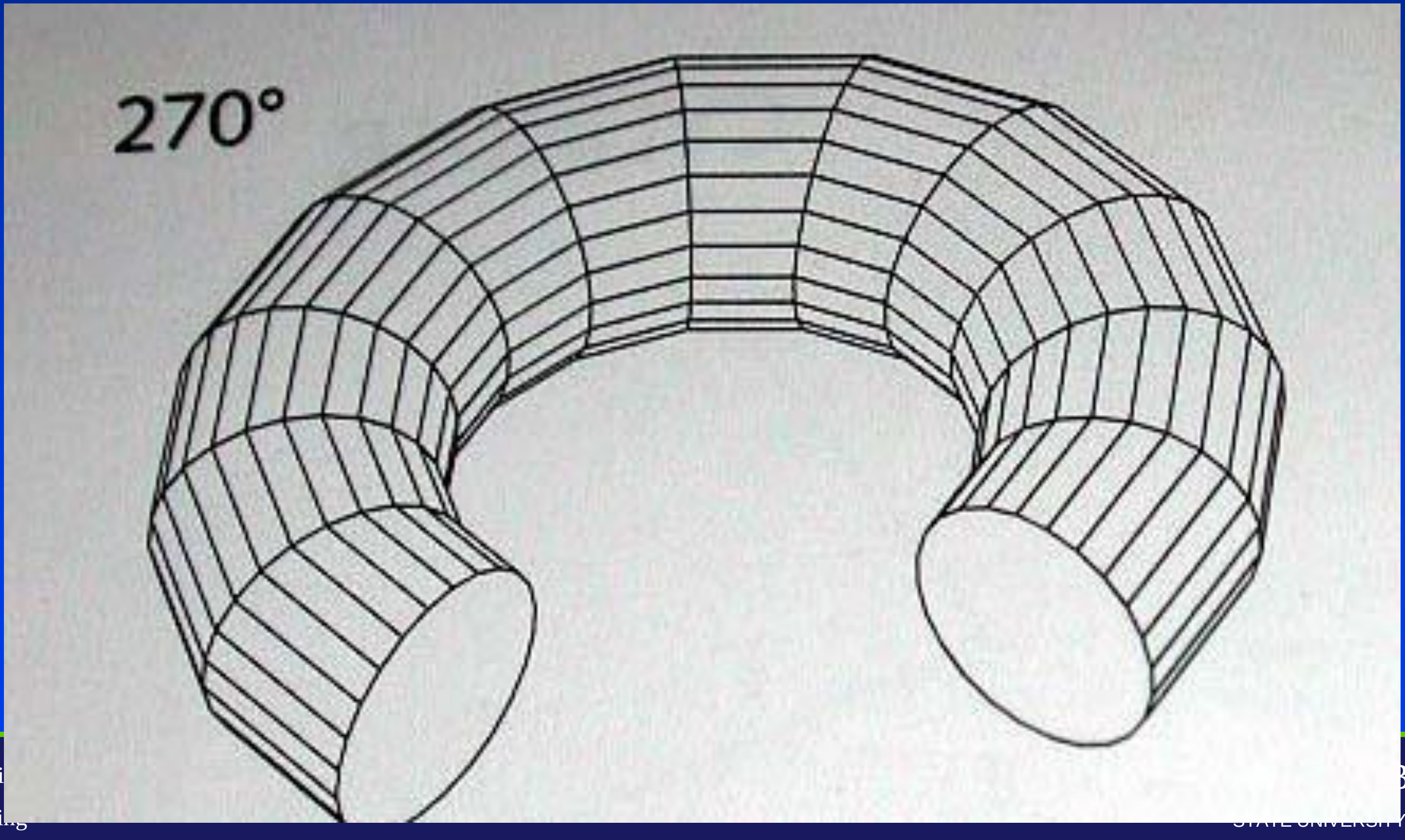
- Mathematics: surfaces of revolution

$$\mathbf{c}(u) = \begin{bmatrix} 0 \\ y(u) \\ z(u) \end{bmatrix}$$

$$\mathbf{s}(u,v) = \begin{bmatrix} -y(u)\sin(v) \\ y(u)\cos(v) \\ z(u) \end{bmatrix}$$

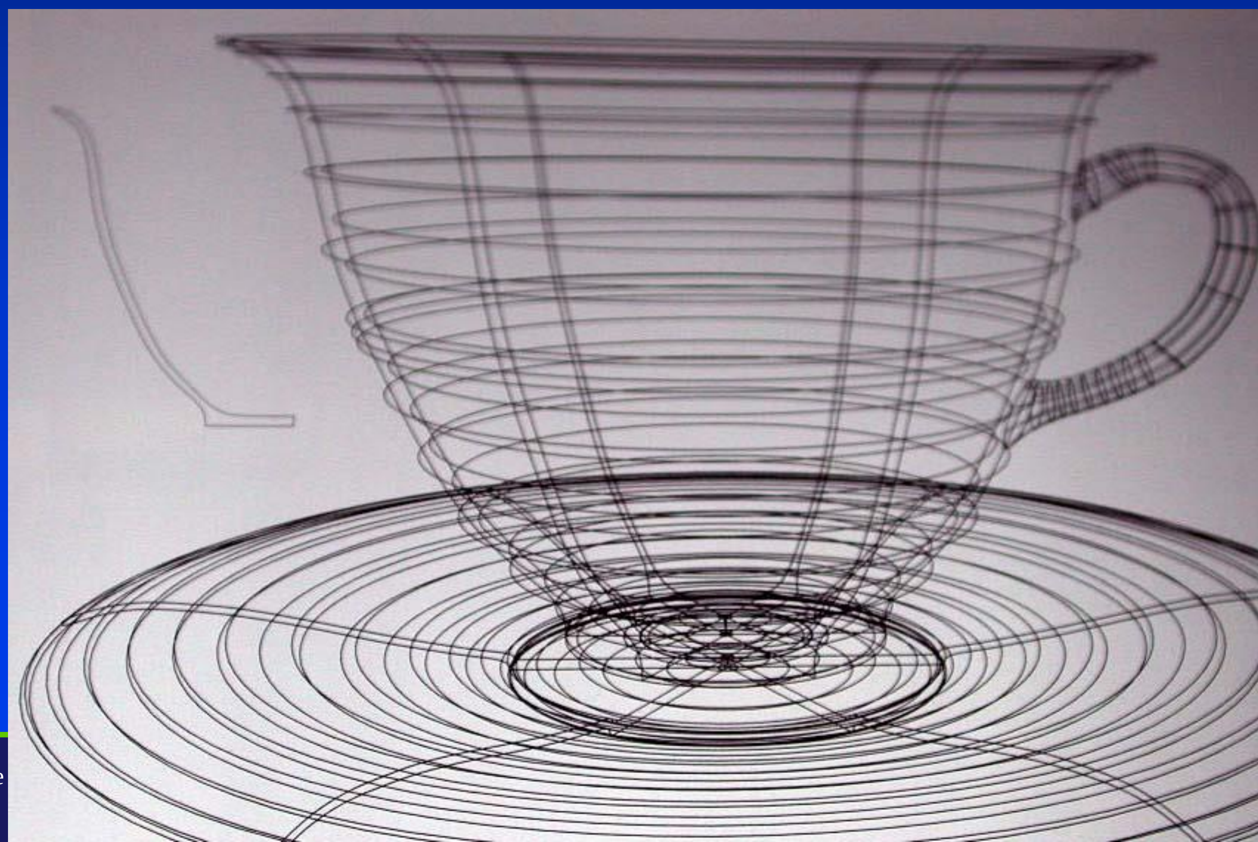
Sweeping

- Sweep a shape over a path to form a generalized cylinder



Revolution

- Revolve a shape around an axis to create an object with rotational symmetry



General Sweeping Objects

- The path maybe any curve
- The polygon that is swept may be transformed as it is moved along the path
  - Scale, rotate with respect to path orientation, ….
- One common way to specify is:
  - Give a poly-line (sequence of line segments) as the path
  - Give a poly-line as the shape to sweep
  - Give a transformation to apply at the vertex of each path segment
- Difficult to avoid self-intersection

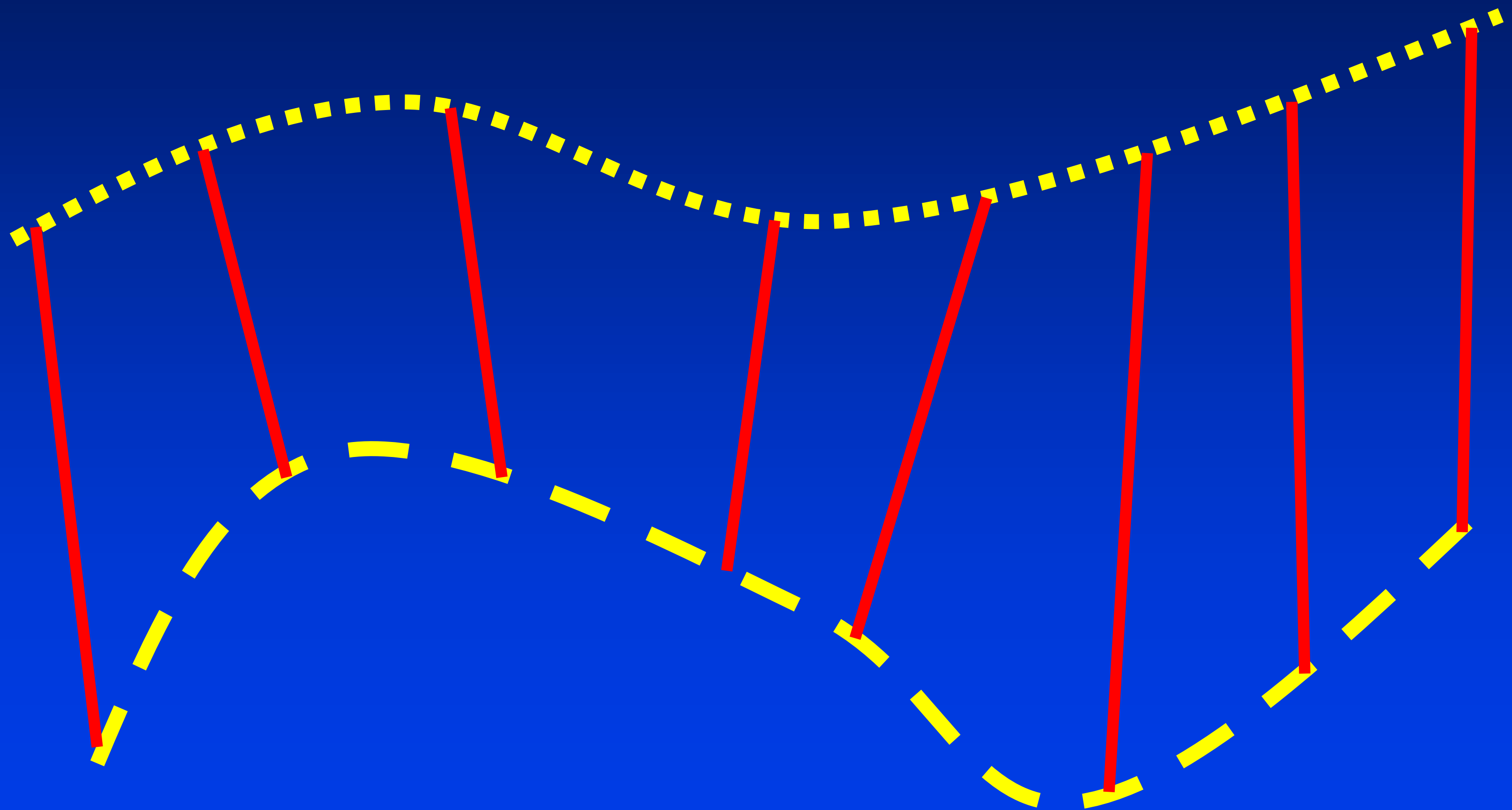
Sweeping Objects: Rendering

- Convert to polygons
  – Break path into short segments
  – Create a copy of the sweep polygon at each segment
  – Join the corresponding vertices between the polygons
  – May need things like end-caps on surfaces of revolution and extrusions
- Normals come from sweep polygon and path orientation
- Sweep polygon defines one texture parameter, sweep path defines the other

Tori Example

```
Vector3       points[2][8];
int           start_i = 0;
int           end_i = 1;
for ( int i = 0 ; i < 8 ; i++ )
   points[start_i][i] = TorusPoint(7,i);
for ( int j = 0 ; j < 8 ; j++ ) {
   glBegin(GL_TRIANGLE_STRIP);
   for ( int i = 0 ; i < 8 ; i++ ) {
       glVertex3fv(points[start_i][i];
       points[end_i][i] = TorusPoint[j][i];
       glVertex3fv(points[end_i][i];
   }
   glVertex3fv(points[start_i][0]);
   glVertex3fv(points[end_i][0]);
   glEnd();
   int temp = start_i; start_i = end_i; end_i = temp;
}
```

Ruled surfaces

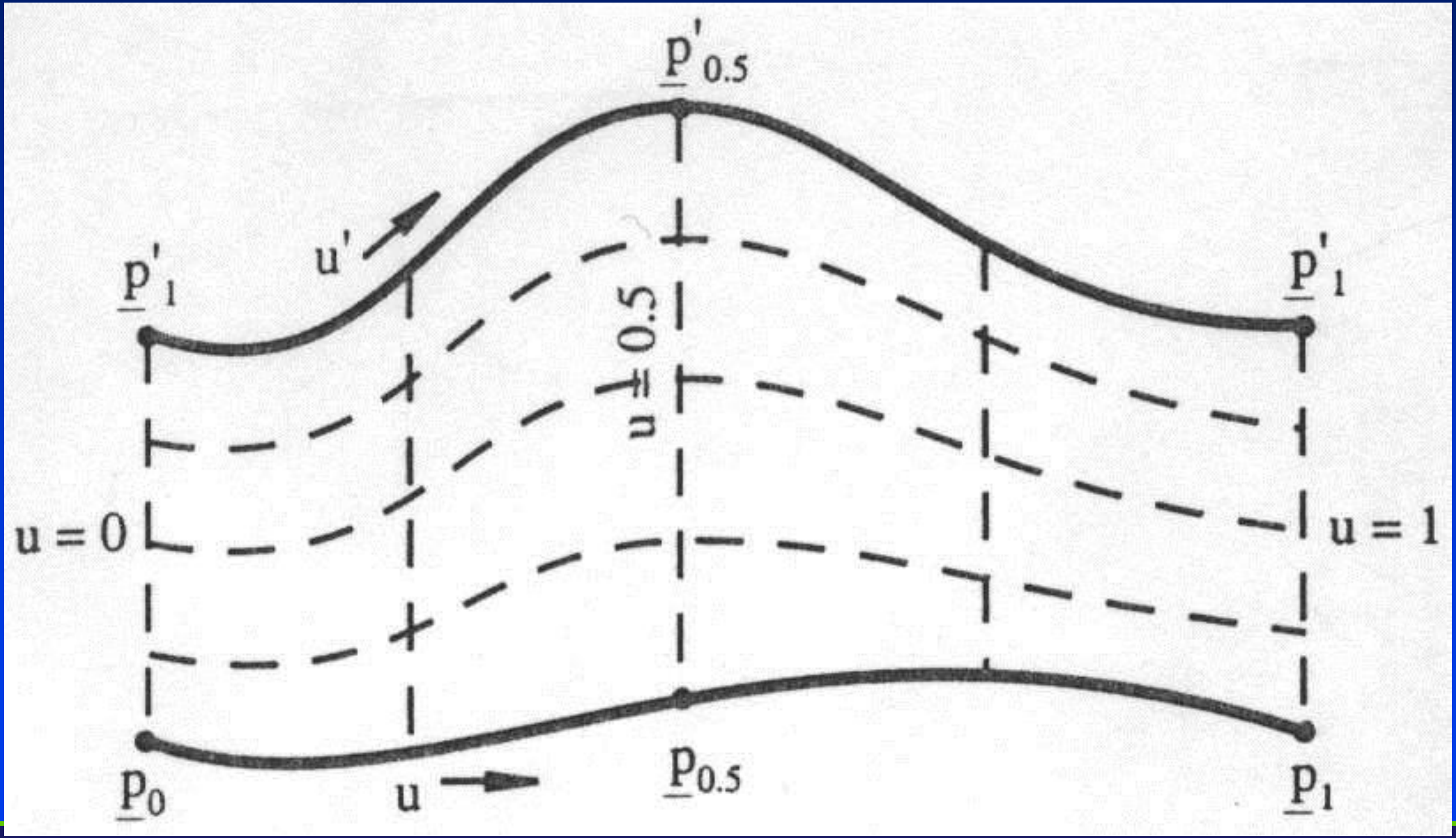


Ruled Surfaces

- Move one straight line along a curve, or join two parametric curves by straight lines
- Example: plane, cone, cylinder
- Cylindrical surface
- Surface equation

$$\mathbf{s}(u,v) = (1-v)\mathbf{a}(u) + v\mathbf{b}(u)$$
$$\mathbf{s}(u,v) = (1-v)\mathbf{s}(u,0) + v\mathbf{s}(u,1)$$
$$\mathbf{s}(u,v) = \mathbf{p}(u) + v\mathbf{q}(u)$$

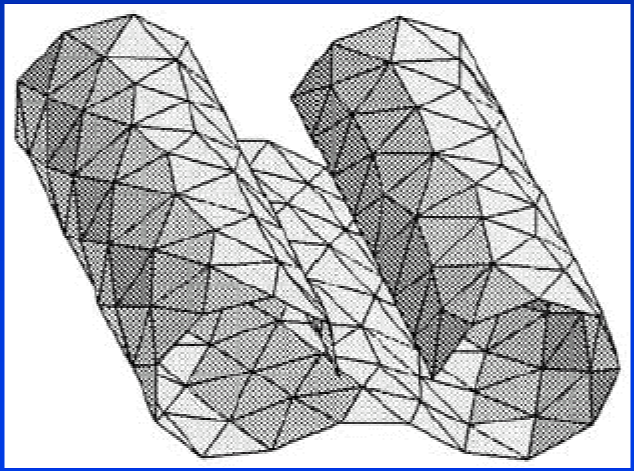
- Isoparametric lines
- Generalized cylinder
- Bending by roller

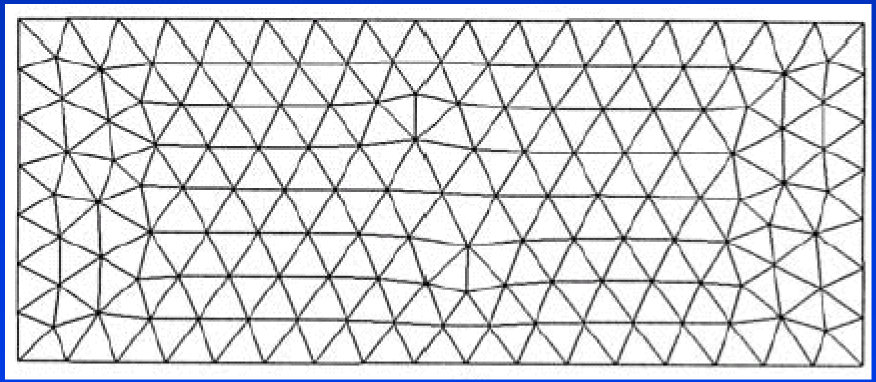

$\underline{p}'_{0.5}$
u'
$\underline{p}'_1$
$u = 0.5$
$u = 0$
$u = 1$
$\underline{p}_0$
u
$\underline{p}_{0.5}$
$\underline{p}_1$

Developable Surfaces

- Deform a surface to planar shape without length/area changes
- Unroll a surface to a plane without stretching/distorting
- Example: cone, cylinder
- Developable surfaces vs. Ruled surfaces
- More examples???

Developable Surface





Developable Surfaces

Planar quad strip (Meshes)