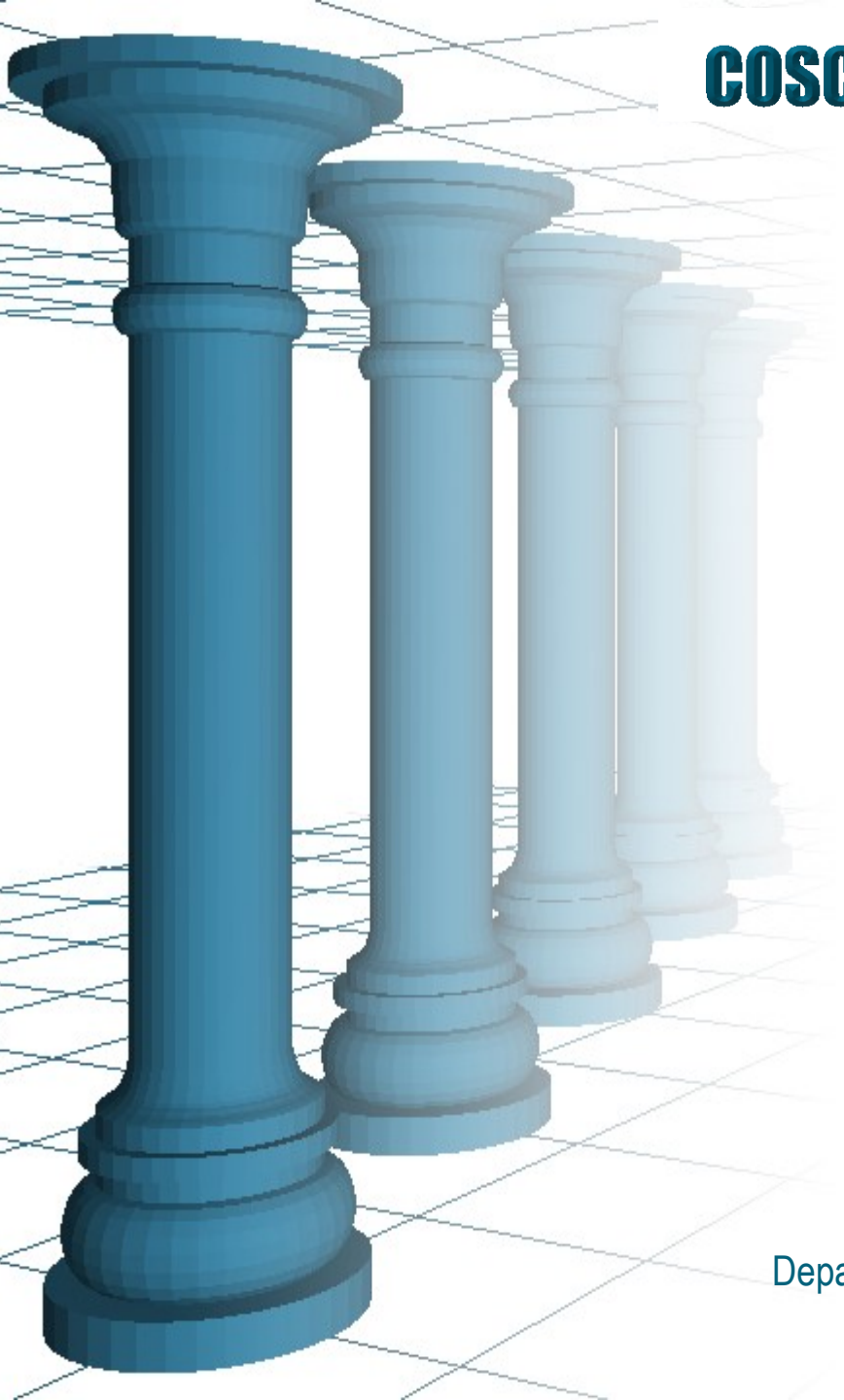


COSC363 Computer Graphics

10

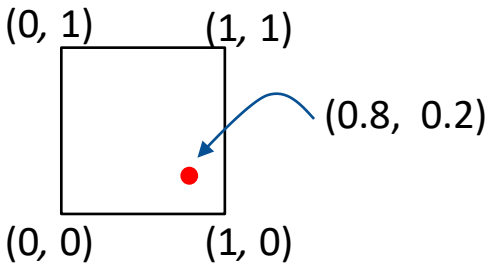
Parametric Domains and Bezier Surfaces

Crucibles and Moulds

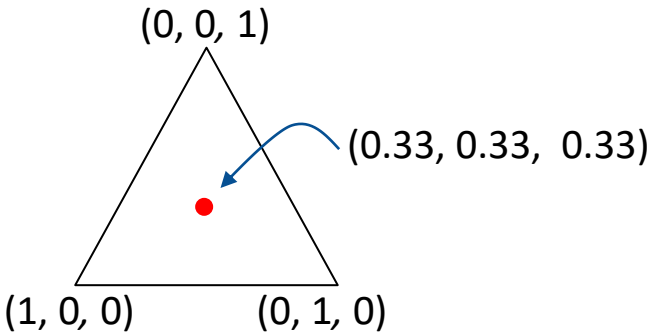
R. Mukundan (mukundan@canterbury.ac.nz)
Department of Computer Science and Software Engineering
University of Canterbury, New Zealand.

Parametric Domains

- A parametric domain (or parametric space) is a space defined by a regular shape, within which points are specified using parametric values in the range [0, 1].
- We consider two types of 2D parametric domains:
  - A **quad domain**: This is a unit square.



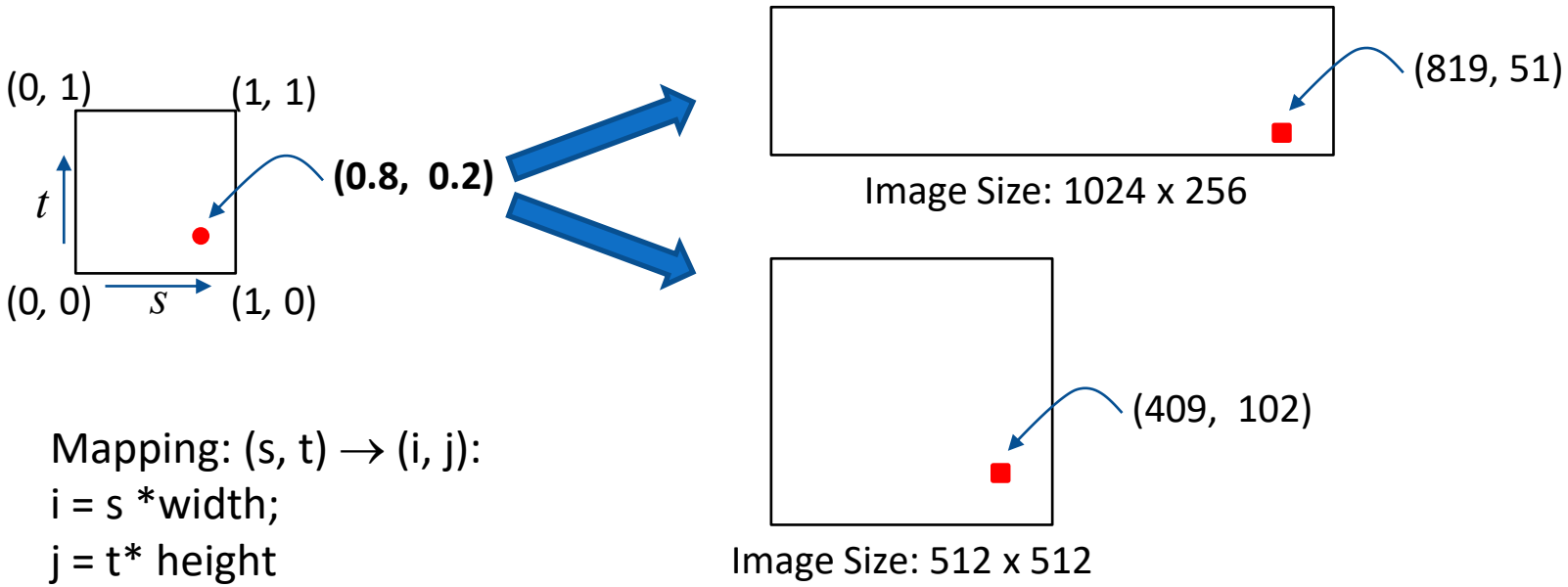
  - A **triangle domain**: This is an equilateral triangle



NOTE: The coordinates shown here are <u>not</u> 3D Cartesian coordinates. The triangle is a 2D domain. The coordinates are barycentric coordinates (explained later).

Quad Domain: A Trivial Example

- We used texture coordinates (s, t) defined in a parametric space to specify a point in an image of arbitrary size.

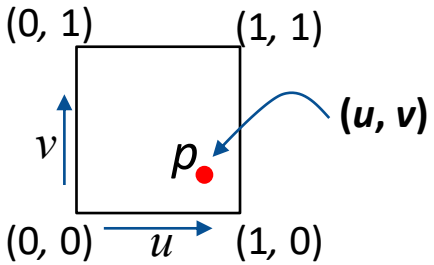


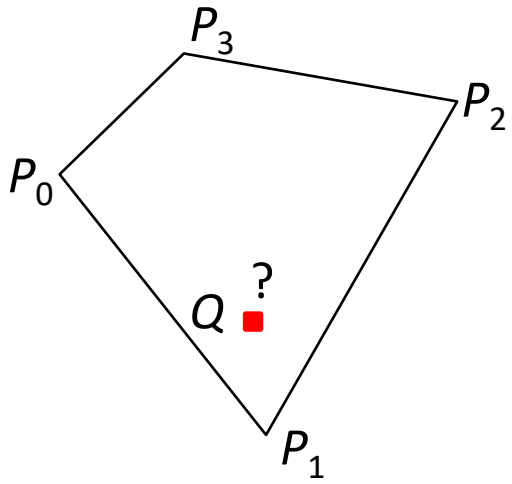
Mapping: (s, t) → (i, j):
i = s *width;
j = t* height

- This is a simple example because images have rectangular 2D shapes and one corner of the image is always (0, 0), closely matching the shape of the parametric domain.

Quad Domain: A General Example

- Suppose we want to map points from the parametric domain to a general quad in 3D space given by points $P_0..P_3$, where $P_i = (x_i, y_i, z_i)$





$0 \le \boldsymbol{u} \le 1$
$0 \le \boldsymbol{v} \le 1$

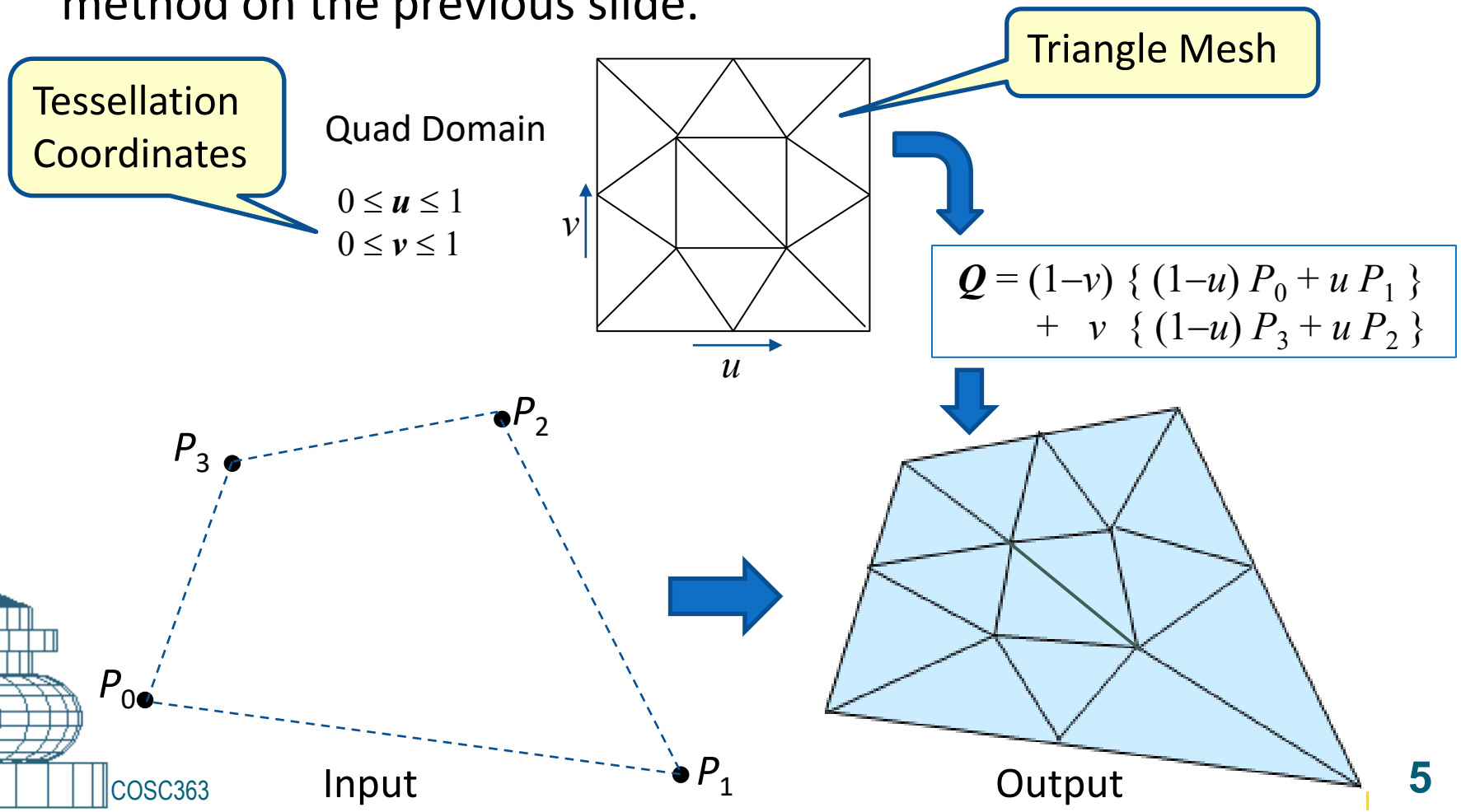
A 3D polygon of arbitrary size and shape

- We could use a bi-linear interpolation formula for the above mapping.

$$\boldsymbol{Q} = (1-v)\ \{\ (1-u)\ P_0 + u\ P_1\ \}$$
$$+\quad v\ \{\ (1-u)\ P_3 + u\ P_2\ \}$$

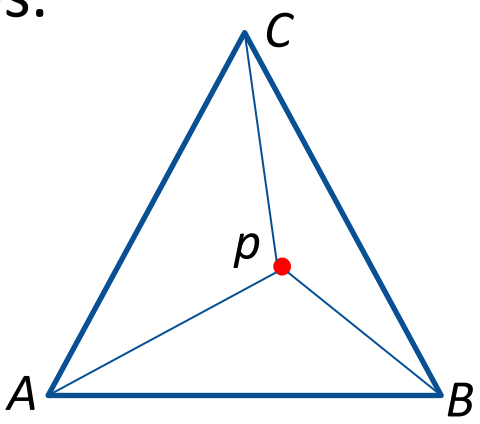
Quad Domain: Tessellation

- Tessellating a quad of arbitrary shape in 3D space is difficult. If we can tessellate a quad domain (which has a regular shape in 2D), then we can map the points to any quad in 3D using the method on the previous slide.



Triangle Domain: Barycentric Coordinates

- The barycentric coordinates (u, v, w) of a point inside a triangle ABC specify the position of the point relative to its vertices.

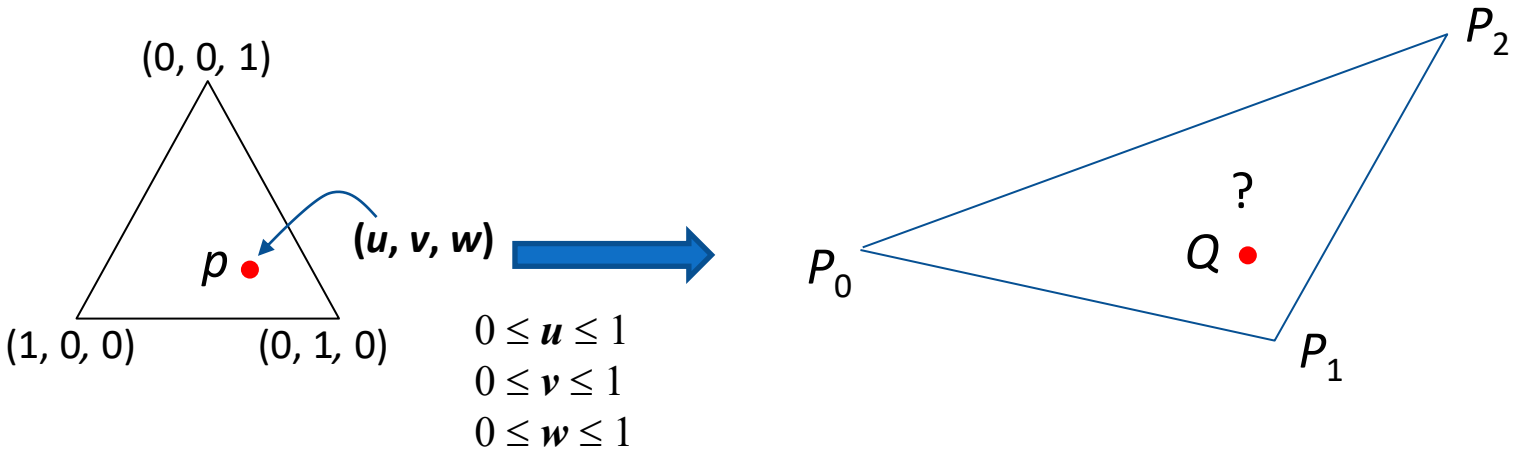


Barycentric coordinates of p: (u, v, w)
u = Area (PBC) / Area (ABC)
v = Area (PCA) / Area (ABC)
w = Area (PAB) / Area (ABC)

- Properties:
  - A = (1, 0, 0), B = (0, 1, 0), C = (0, 0, 1)
  - For any point inside the triangle, the barycentric coordinates will have a value between 0 and 1.
  - $u + v + w = 1$. E.g. p = (0.2, 0.4, 0.4)

Triangle Domain: Barycentric Mapping

- If we know the barycentric coordinates of a point inside a triangle, we can find its map on any other triangle!
- Consider a triangle in 3D space, with vertices P_0, P_1, P_2, where $P_i = (x_i, y_i, z_i)$.



- The barycentric mapping of the point *p* is given by:

$$\boldsymbol{Q} = u\,P_0 + v\,P_1 + w\,P_2$$

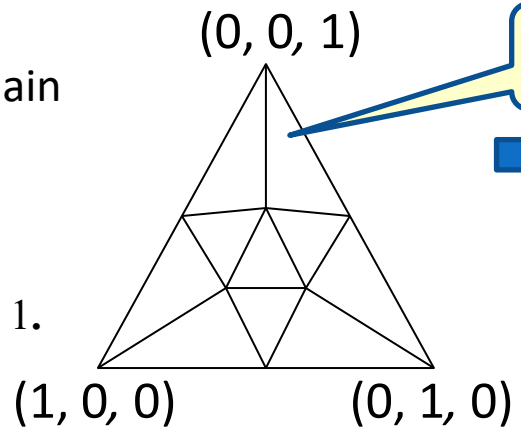
Triangle Domain: Tessellation

- Triangles in 3D space are tessellated with the help of a triangle domain.
- The triangle domain is tessellated and the points mapped to the 3D space using barycentric coordinates.

Tessellation Coordinates

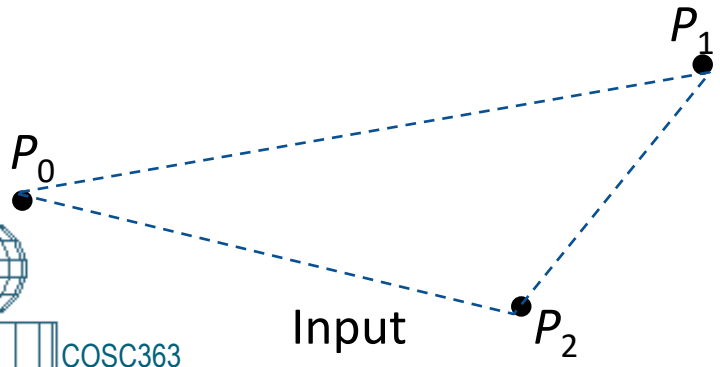
Triangle Domain

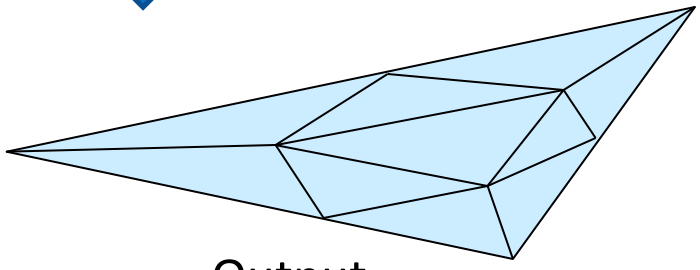
$$0 \le u \le 1$$
$$0 \le v \le 1$$
$$0 \le w \le 1,$$
$$u + v + w = 1.$$



Triangle Mesh

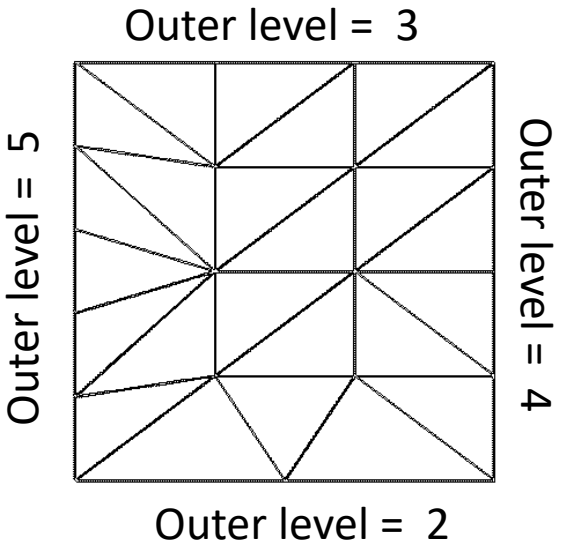
$$\boldsymbol{Q} = u\,P_0 + v\,P_1 + w\,P_2$$



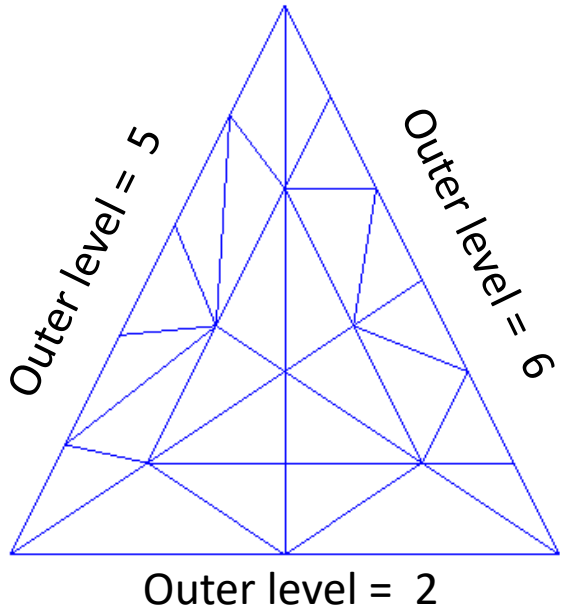


Outer Tessellation Levels

- Two types of tessellation levels are defined for parametric domains: Outer and Inner
- Outer tessellation levels specify the number of subdivisions along the edges of the domain



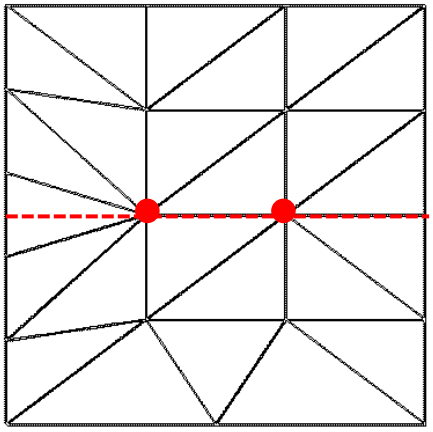
Quad Domain



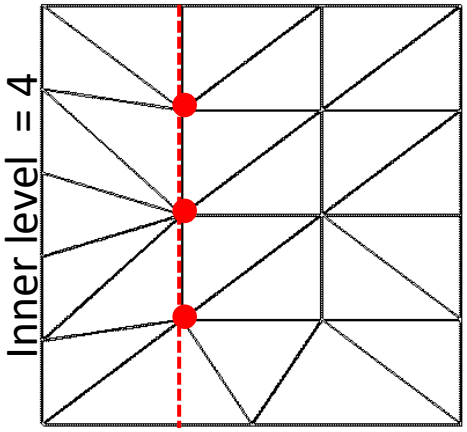
Triangle Domain

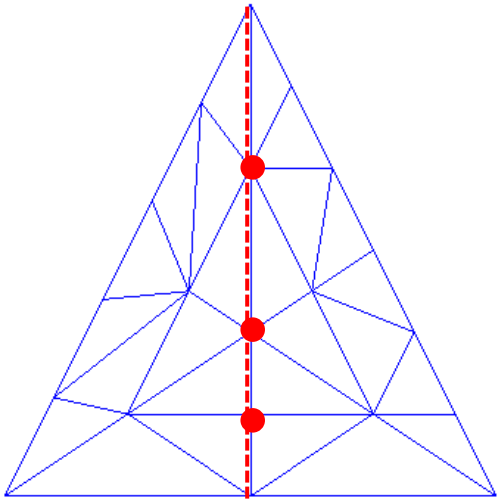
Inner Tessellation Levels

- Quad domain: The minimum number of subdivisions (or steps) along a path from one edge to the opposite edge.
  - Two inner levels corresponding to the horizontal direction and vertical direction of the paths.
- Triangle domain: The minimum number of subdivisions (or steps) along a path from one vertex to the opposite edge.
  - Only one inner level corresponding to all three directions.



Inner level = 3

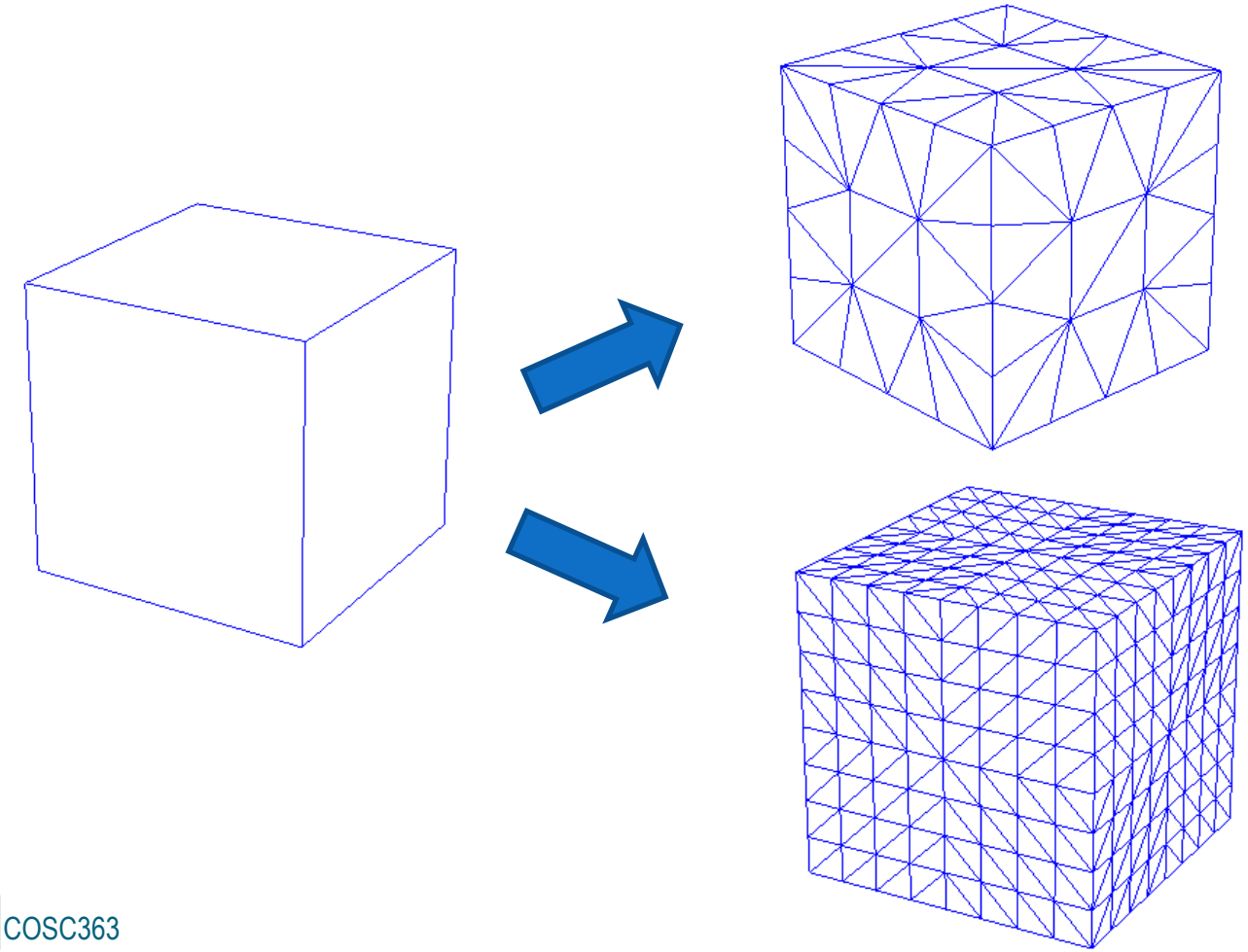




Inner level = 4

Tessellation

- The methods given on slides 5, 8 provide only tessellation of planar surfaces (quads or triangles) without modifying the surface geometry. Such tessellations have limited use.



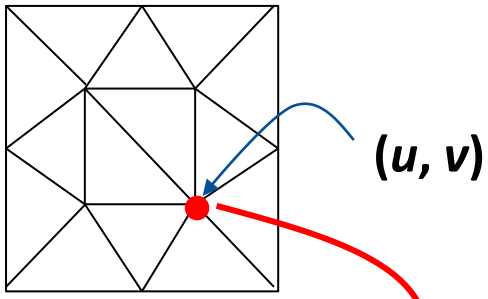
Linear Blending Functions

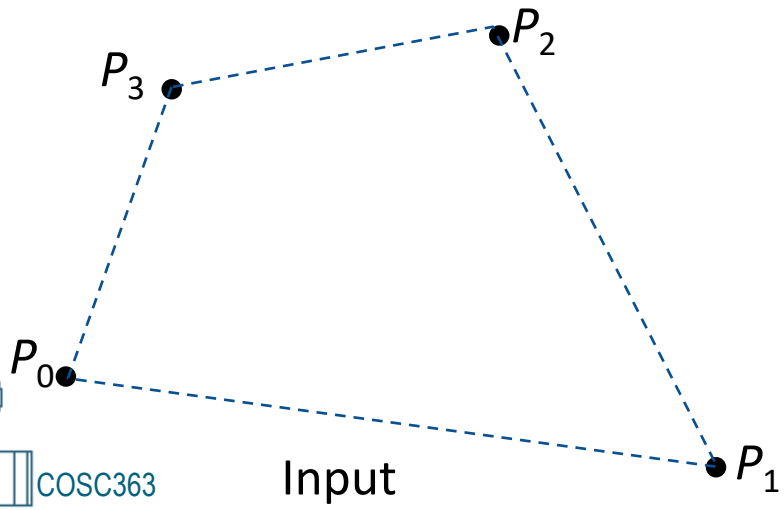
- The quad domain mapping method combines a set of vertices P_0, P_1, P_2, P_3 using linear blending functions in u, v.

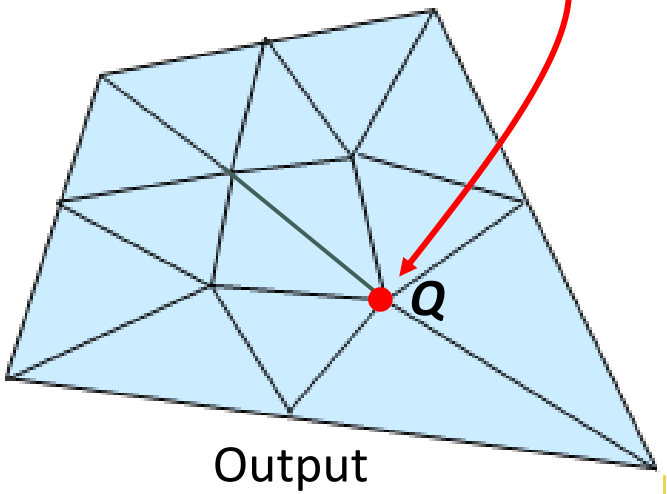
$$\boldsymbol{Q} = (1-v)\,(1-u)\,\boldsymbol{P_0} + (1-v)\,u\,\boldsymbol{P_1} + v\,(1-u)\,\boldsymbol{P_3} + v\,u\,\boldsymbol{P_2}$$

- **Blending functions**: $(1-u)$, u, $(1-v)$, v.
- **Patch vertices**: P_0, P_1, P_2, P_3

  The patch vertices define the shape of the tessellated surface

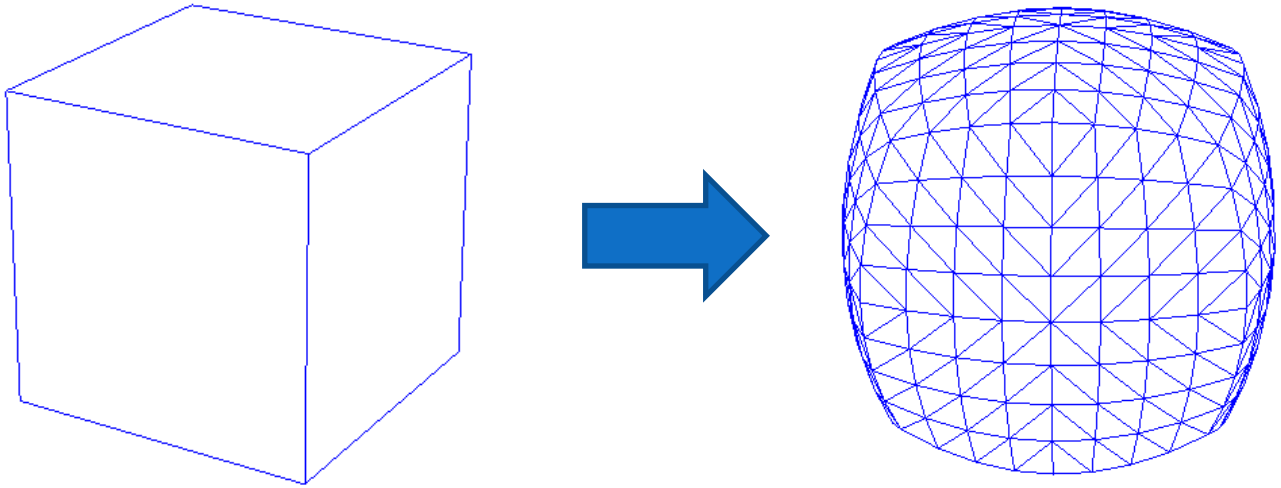






Tessellation

- Tessellations (subdivisions) of surfaces are commonly used for smoothing or approximation.
- Such surfaces can be generated using higher order blending functions in the tessellation coordinates *u*, *v*.
- Higher order blending functions will require more (> 4) patch vertices

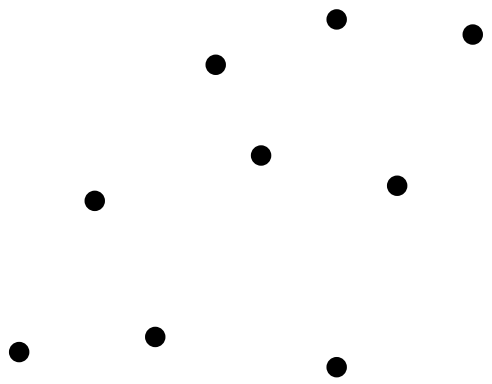


Higher Order Blending Functions

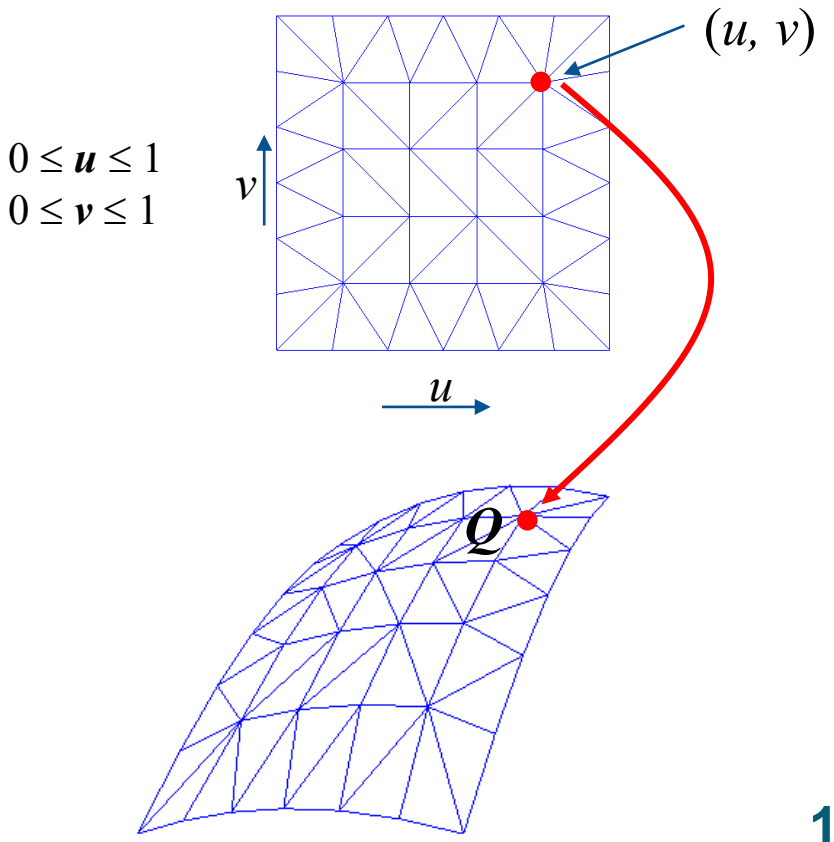
- The following example uses quadratic functions in u, v:
- **Blending functions**: $(1-u)^2$, $2u(1-u)$, u^2, $(1-v)^2$, $2v(1-v)$, v^2.
- **Patch vertices**: $P_0,\ldots,P_8$

The patch vertices are control points defining the shape of the tessellated surface.

The generated surface may not pass through all patch vertices.



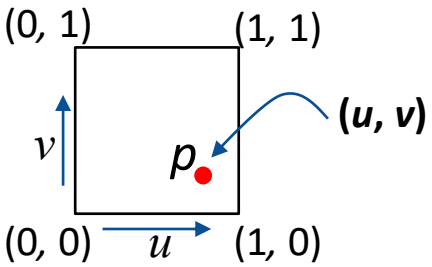
A 3x3 Patch



Higher Order Blending Functions

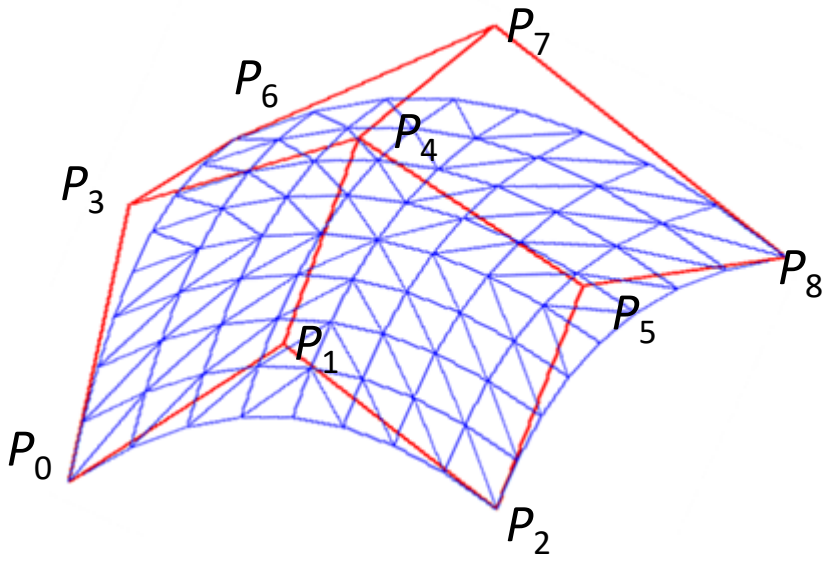
- Example:

$$\begin{aligned} \boldsymbol{Q} = \quad & (1-v)^2 \{ (1-u)^2 P_0 + 2u(1-u) P_1 + u^2 P_2 \} \\ & + 2v(1-v) \{ (1-u)^2 P_3 + 2u(1-u) P_4 + u^2 P_5 \} \\ & + v^2 \{ (1-u)^2 P_6 + 2u(1-u) P_7 + u^2 P_8 \} \end{aligned}$$



$0 \le \boldsymbol{u} \le 1$

$0 \le \boldsymbol{v} \le 1$



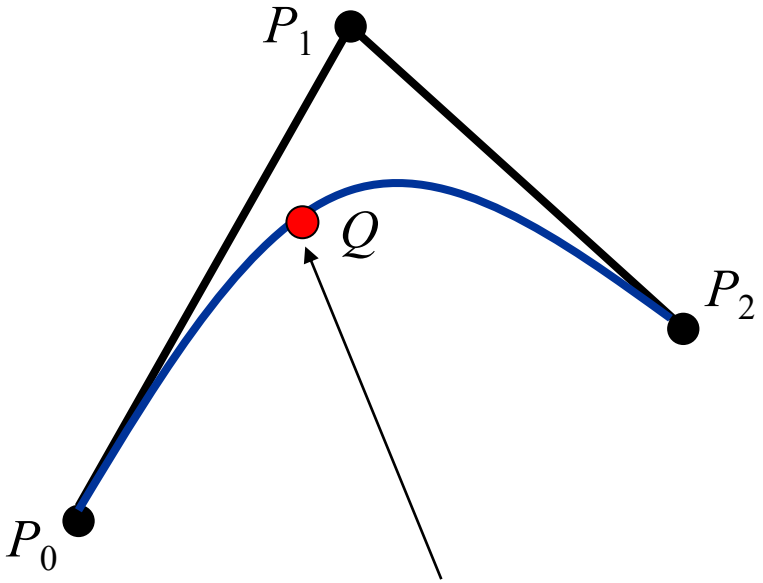
Quadratic Bezier Curve

- A quadratic **Bezier curve** is defined using 3 control points.
- The blending functions are **Bernstein polynomials** of degree 2:

$f_0(u) = (1-u)^2$

$f_1(u) = 2u(1-u)$

$f_2(u) = u^2$



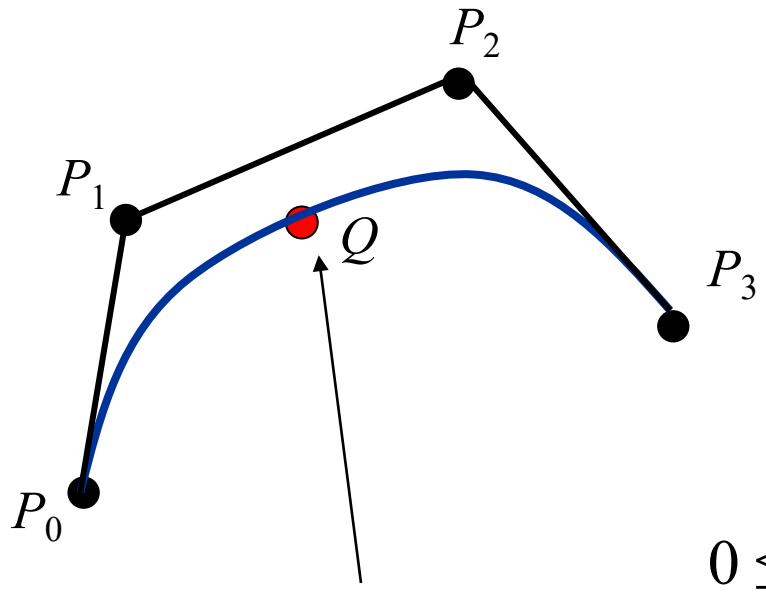
$0 \le u \le 1.$

$$\boldsymbol{Q} = (1-u)^2 P_0 + 2(1-u)u P_1 + u^2 P_2$$

Cubic Bezier Curve

- A cubic Bezier curve is defined using 4 control points.
- The blending functions are Bernstein polynomials of degree 3:

$$f_0(u) = (1-u)^3$$
$$f_1(u) = 3(1-u)^2 t$$
$$f_2(u) = 3(1-u)u^2$$
$$f_3(u) = u^3$$



$$0 \le u \le 1.$$

$$\boldsymbol{Q} = (1-u)^3 P_0 + 3(1-u)^2 u P_1 + 3(1-u)u^2 P_2 + u^3 P_3$$

n^{th} degree Bezier Curve

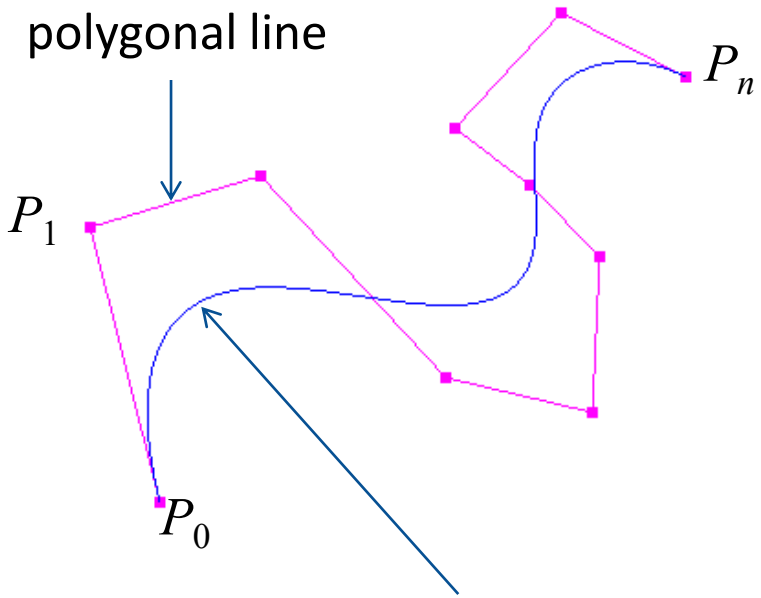
- An n^{th} degree Bezier curve is defined using n+1 control points: $P_0 \ldots P_n$.
- The blending functions are Bernstein polynomials of degree n.

$$f_i^{(n)}(t) = \binom{n}{i}(1-t)^{n-i}t^i$$

$$i = 0, 1, 2 \ldots n.$$

$$P(t) = \sum_{i=0}^{n} f_i^{(n)}(t) P_i$$

$$0 \le t \le 1.$$



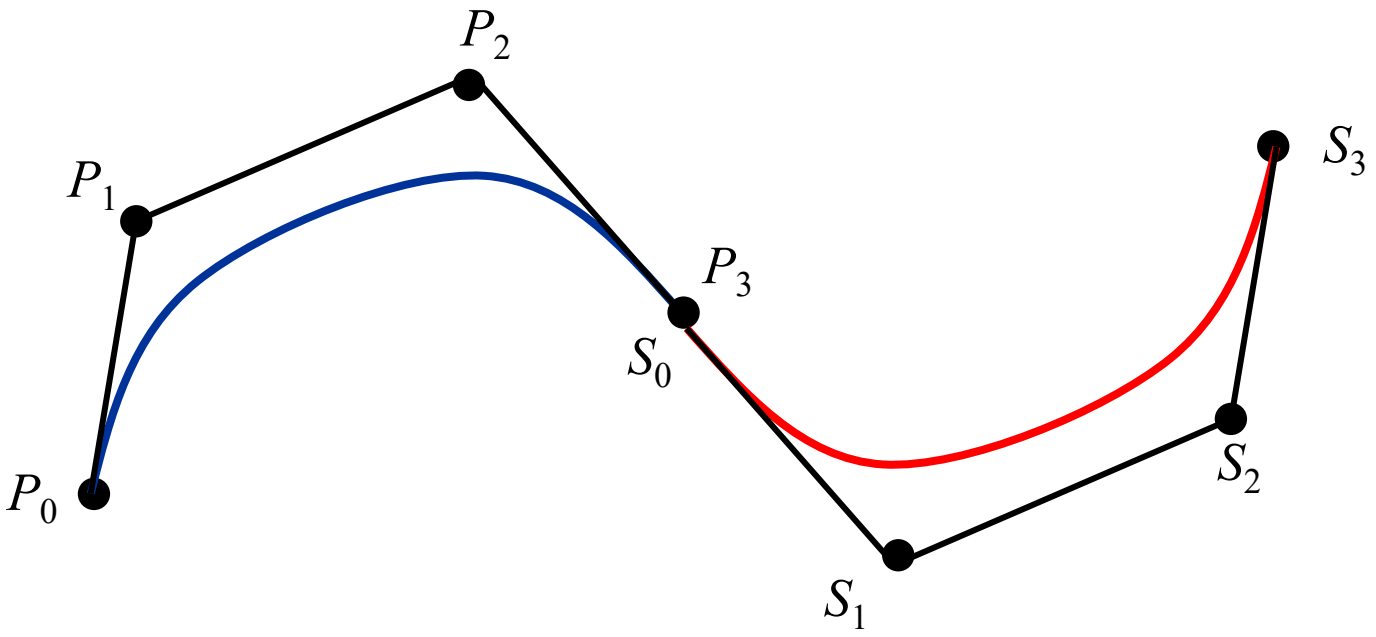
A Bezier curve of degree 9

Properties of Bezier Curves

- The Bernstein polynomials form a partition of unity (sum of the polynomial values = 1), and therefore a Bezier curve is always generated using a convex combination of control points.
- The Bezier curve always lies within the convex hull of the control points.
- The Bezier curve passes through the first and the last control points, and has the corresponding edges of the control polygonal line as its tangents.
- The curve is affine invariant: An affine transformation of a Bezier curve is the same as the Bezier curve generated using the transformed control points.

Joining Bezier Curves (Splines)

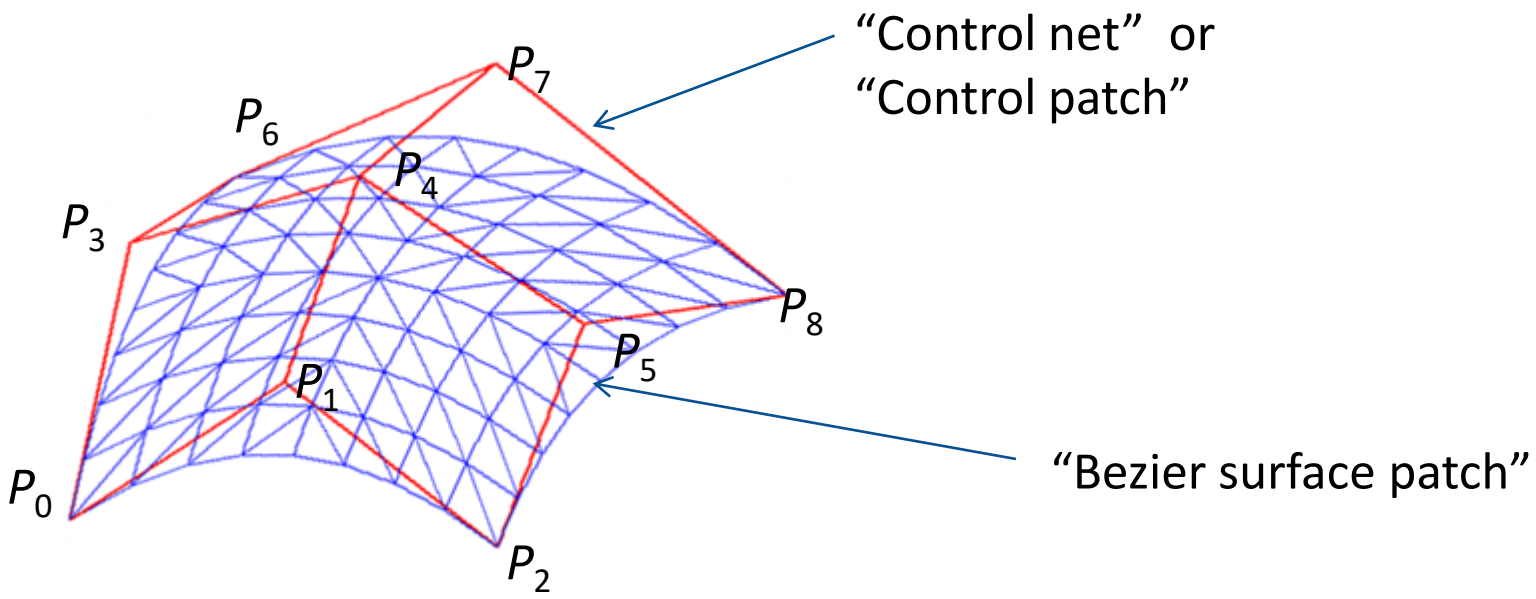
- A set of cubic Bezier curves may be joined together to form on large curve.
- Such curves are called splines. Their shape is controlled by adjusting the positions of the control points.
- We have to ensure tangential continuity at joints (P_3)



Control points P_2, P_3, S_0, S_1 must be collinear

From Curves to Patches

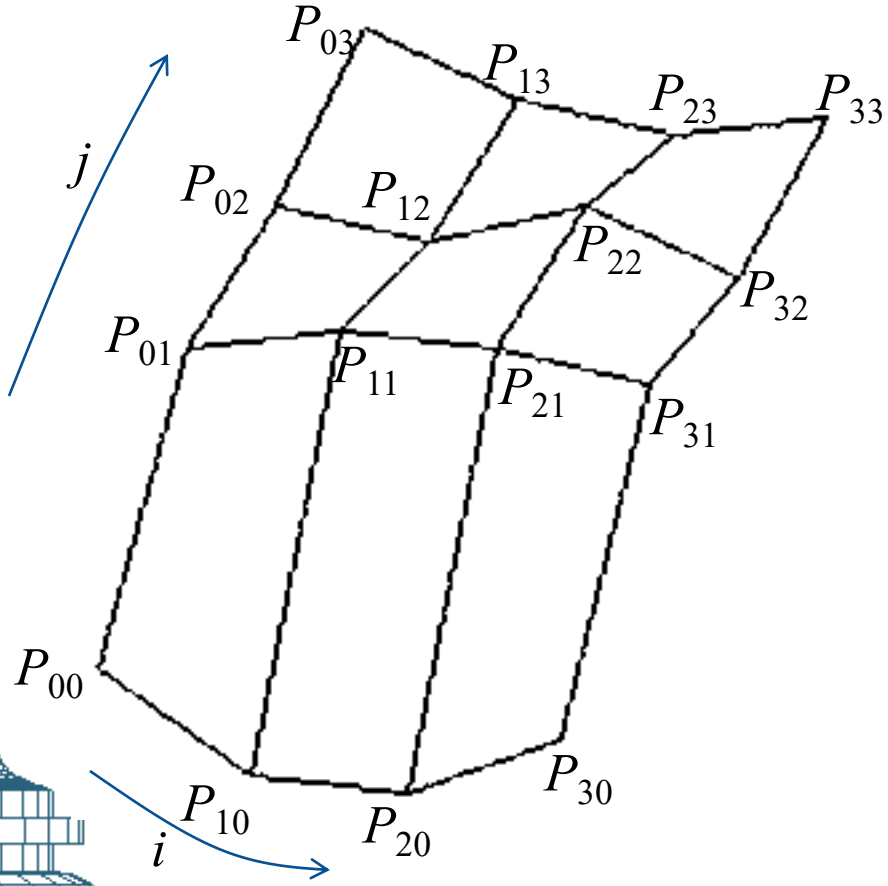
- The method for generating one-parameter Bezier curves $P(u)$ can be extended to get two-parameter Bezier surfaces $P(u, v)$ by using a grid of control points instead of a polygonal line.
- The generated surface is called a ***Bezier surface patch***.



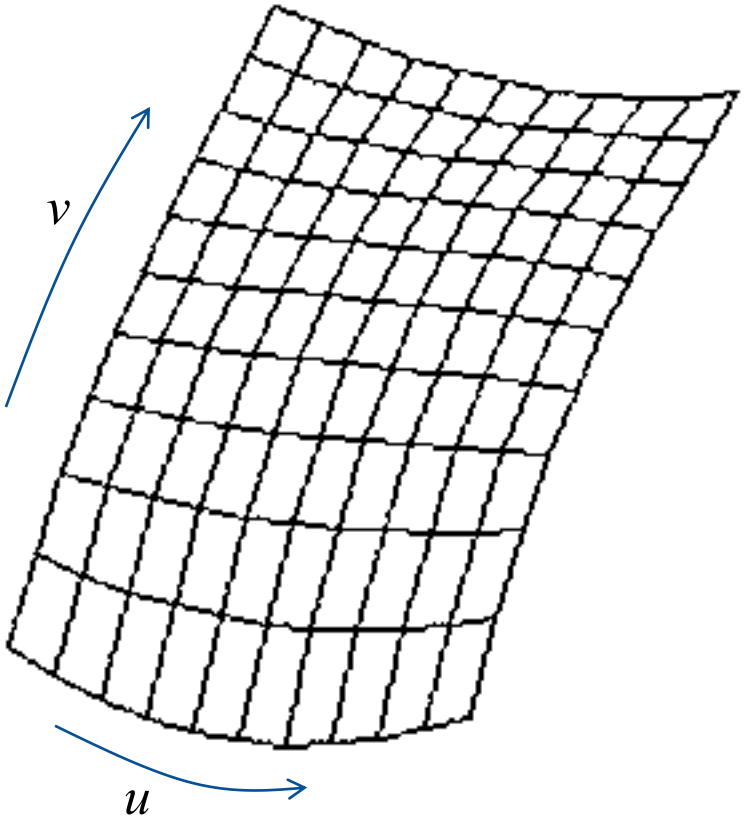
A 3x3 Control Patch

Bezier Surface Patch

- A 4x4 grid of control points is used to generate a bi-cubic Bezier patch.



4x4 Control patch



Bi-cubic Bezier surface patch

Bi-cubic Bezier Surface Patch

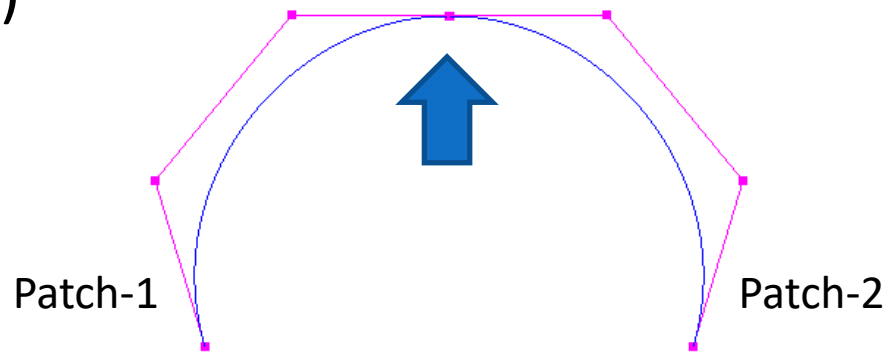
A *Bi-cubic Bezier surface patch* is generated using two sets of Bernstein polynomials of degree 3: $f_i(u)$ and $f_j(v)$. (See Slide 17)

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

$$
\begin{aligned}
0 \le u \le 1 \\
0 \le v \le 1
\end{aligned}
$$

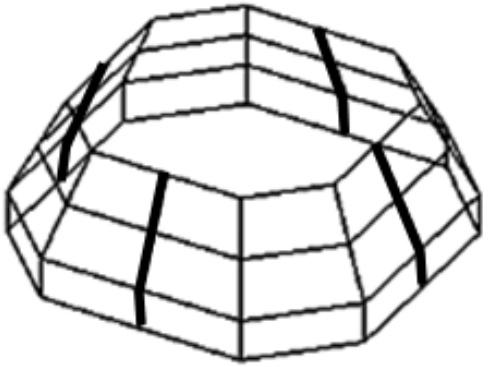
Modelling Complex Shapes

- A coarse description of a shape could be specified using several 4x4 control patches. Each control patch is then used to create a Bezier surface patch.
- We can get a smooth surface *if* the Bezier patches do not have any discontinuity along the edges where two patches meet. This can be achieved by making sure that the corresponding polygonal edges are collinear (for curves) or the corresponding polygons of the control patches are coplanar (for surfaces)

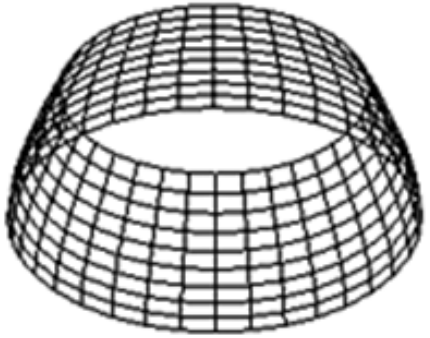


Modelling Complex Shapes

- The Utah Teapot is defined using 32 control patches each of size 4x4.
- The grids are defined such that polygons on either side of an edge where two grids meet are coplanar.
- Bi-cubic Bezier surface patches are generated for each control patch to get the shape of the teapot.

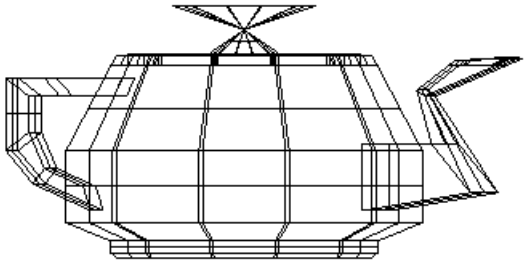


4 Control patches are joined together along edges shown as thick lines

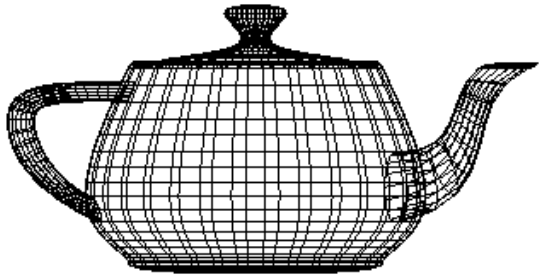


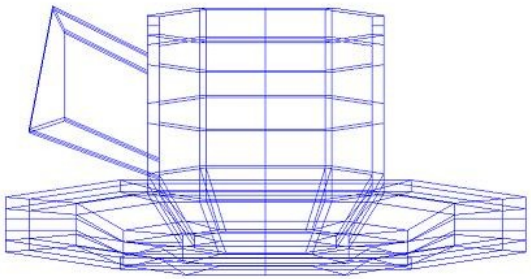
Teapot's upper body

The Teapot Family

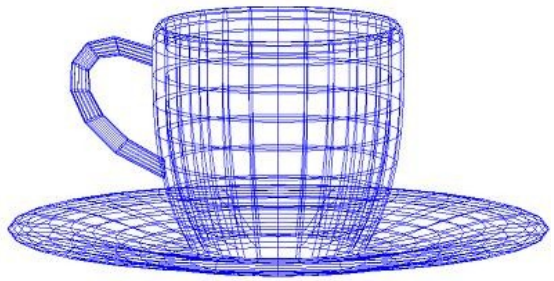


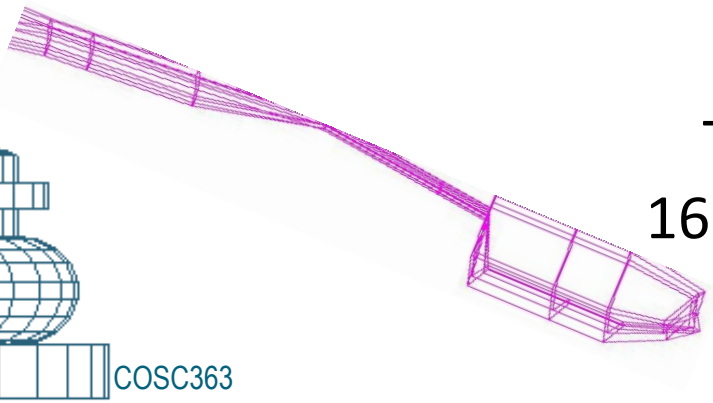
The Teapot
32 *Control patches*





The Teacup
26 *Control patches*





The Teaspoon
16 *Control patches*