Curves, Surfaces and Recursive Subdivision

- Conics: Curves and Quadrics: Surfaces
  - Implicit form
  - Parametric form
- Rational Bézier Forms
- Recursive Subdivision of Curves
- Recursive Subdivision of Surfaces

Conic Curves

Conic Sections (Implicit form)

- Ellipse

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad a, b > 0$$

- Hyperbola

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \quad a, b > 0$$

- Parabola

$$y^2 = 4ax \quad a > 0$$

Conic Sections (Parametric form)

- Ellipse

$$
\begin{aligned}
x(t) &= a\frac{1-t^2}{1+t^2} \\
y(t) &= b\frac{2t}{1+t^2} \quad (-\infty < t < +\infty)
\end{aligned}
$$

- Hyperbola

$$
\begin{aligned}
x(t) &= a\frac{1+t^2}{1-t^2} \\
y(t) &= b\frac{2t}{1-t^2} \quad (-\infty < t < +\infty)
\end{aligned}
$$

- Parabola

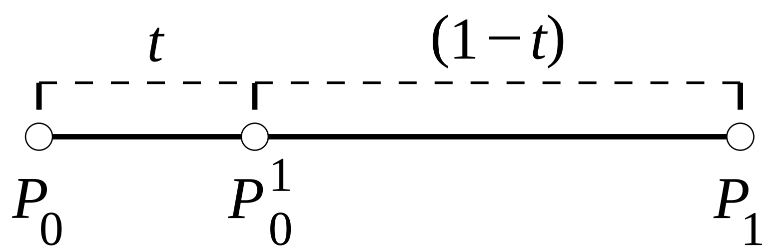
$$
\begin{aligned}
x(t) &= at^2 \\
y(t) &= 2at \quad (-\infty < t < +\infty)
\end{aligned}
$$

Constructing Curve Segments

Linear blend:

- Line segment from an affine combination of points

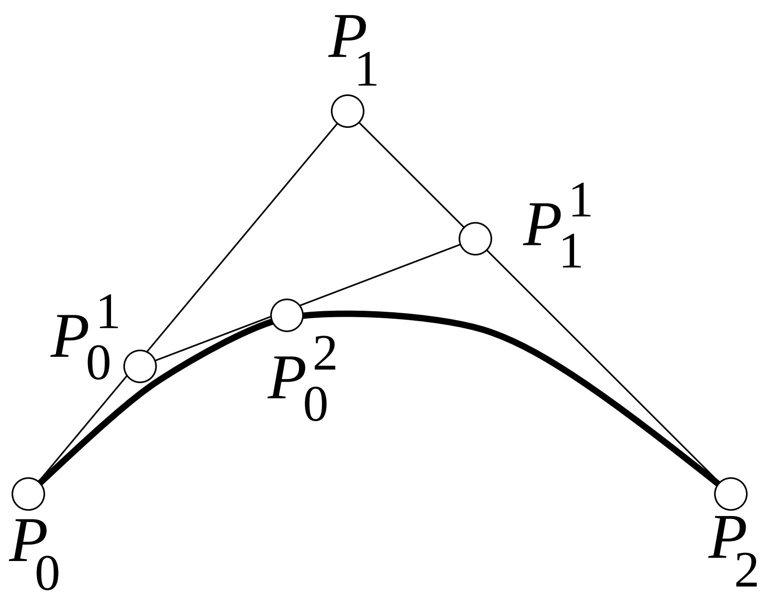
$$P_0^1(t) = (1-t)P_0 + tP_1$$



Quadratic blend:

- Quadratic segment from an affine combination of line segments

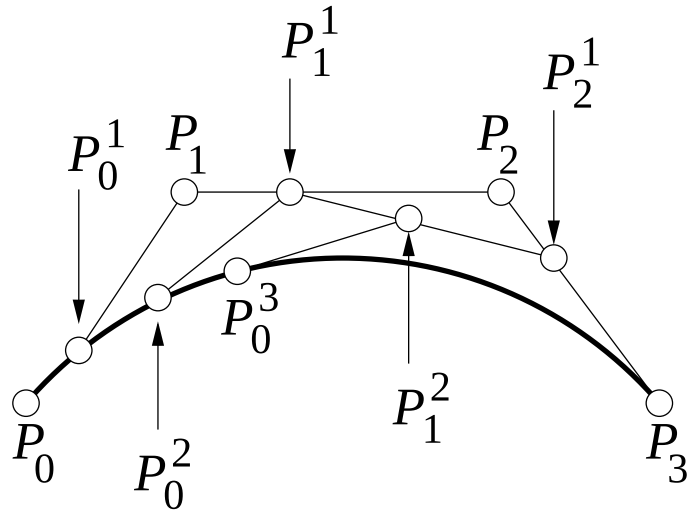
$$
\begin{aligned}
P_0^1(t) &= (1-t)P_0 + tP_1 \\
P_1^1(t) &= (1-t)P_1 + tP_2 \\
P_0^2(t) &= (1-t)P_0^1(t) + tP_1^1(t)
\end{aligned}
$$



Cubic blend:

- Cubic segment from an affine combination of quadratic segments

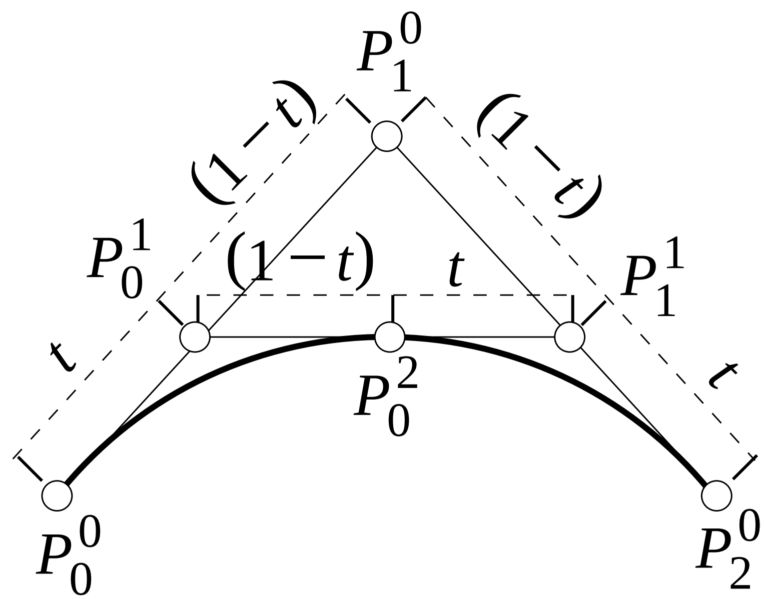
$$
\begin{aligned}
P_0^1(t) &= (1-t)P_0 + tP_1 \\
P_1^1(t) &= (1-t)P_1 + tP_2 \\
P_0^2(t) &= (1-t)P_0^1(t) + tP_1^1(t) \\
P_2^1(t) &= (1-t)P_2 + tP_3 \\
P_1^2(t) &= (1-t)P_1^1(t) + tP_2^1(t) \\
P_0^3(t) &= (1-t)P_0^2(t) + tP_1^2(t)
\end{aligned}
$$



- The pattern should be evident for higher degrees

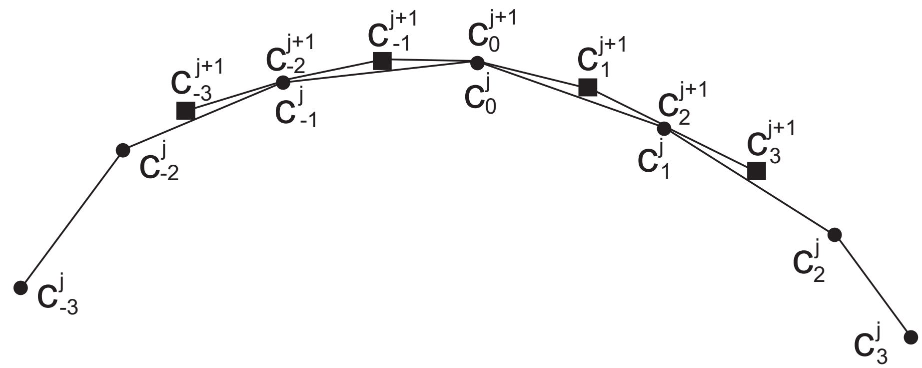
Geometric view (de Casteljau Algorithm):

- Join the points P_i by line segments
- Join the $t : (1 - t)$ points of those line segments by line segments
- Repeat as necessary
- The $t : (1 - t)$ point on the final line segment is a point on the curve
- The final line segment is tangent to the curve at t



Subdivision of Polygons

Four Point Scheme



Four point scheme: the filled circles are the level j control points, the filled squares are the level $j+1$ control points.

For four-point scheme we need to consider only 7 control points; these 7 points completely define the piece of the curve around a control point. We can consider a set of 7 control points *on any subdivision level*, as we do not care how small our piece of the curve is. Note that we can compute the positions of the seven control points on level $j+1$ from the positions of similar seven control points on level j, using a 7×7 submatrix $\mathbf{S}$ of the infinite subdivision matrix.

The local subdivision matrix for the four-point scheme is:

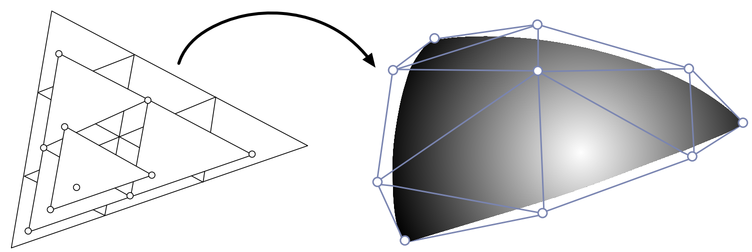
$$\begin{pmatrix} c_{-3}^{j+1} \\ c_{-2}^{j+1} \\ c_{-1}^{j+1} \\ c_{0}^{j+1} \\ c_{1}^{j+1} \\ c_{2}^{j+1} \\ c_{3}^{j+1} \end{pmatrix} = \begin{pmatrix} -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} \end{pmatrix} \begin{pmatrix} c_{-3}^{j} \\ c_{-2}^{j} \\ c_{-1}^{j} \\ c_{0}^{j} \\ c_{1}^{j} \\ c_{2}^{j} \\ c_{3}^{j} \end{pmatrix}$$

Quadric Surfaces

Implicit form

Parametric form

Constructing Surface Patches

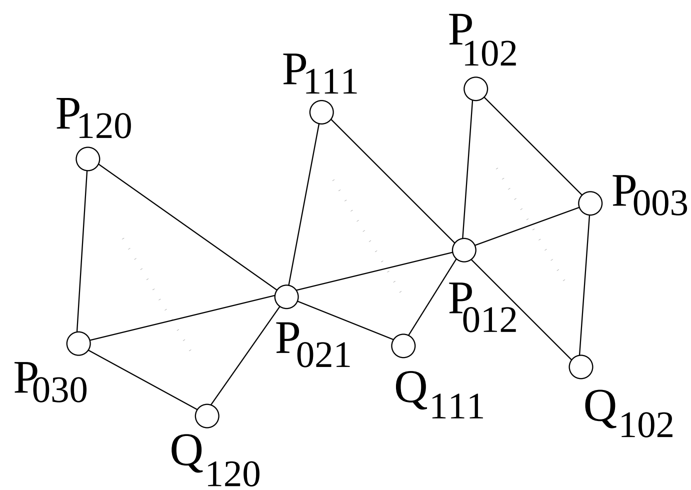


Triangular deCasteljau:

- Join adjacently indexed P_{ijk} by triangles
- Find $r : s : t$ barycentric point in each triangle
- Join adjacent points by triangles
- Repeat
  - Final point is the surface point $P(r, s, t)$
  - final triangle is tangent to the surface at $P(r, s, t)$
- Triangle up/down schemes become tetrahedral up/down schemes

Properties:

- Each boundary curve is a Bézier curve
- Patches will be joined smoothly if pairs of boundary triangles are planar as shown



Tensor Product Patches

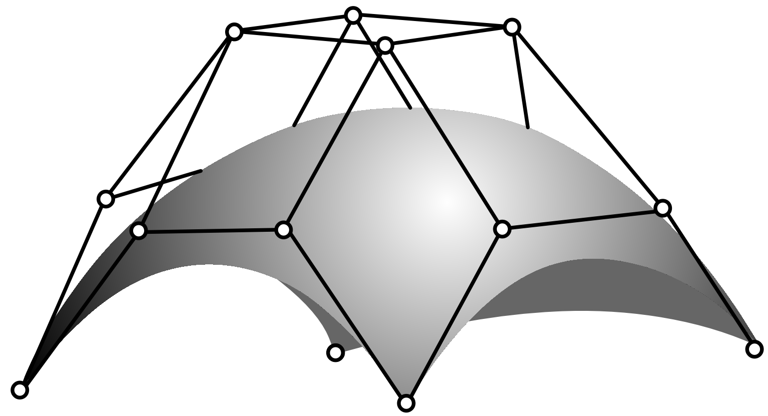
Tensor Product Patches:

- The *control polygon* is the polygonal mesh with vertices $P_{i,j}$
- The *patch basis functions* are products of curve basis functions

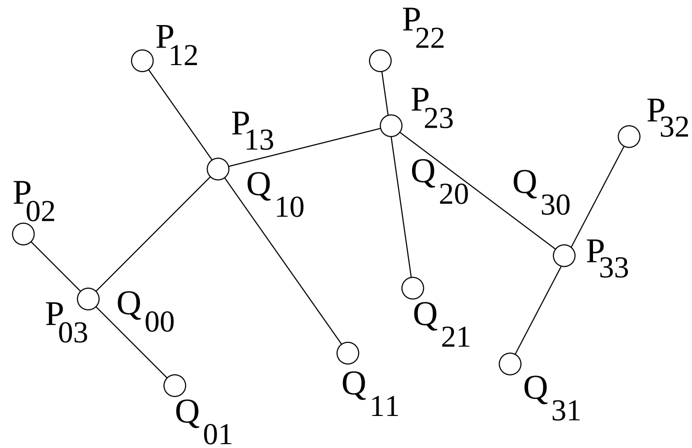
$$P(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{n} P_{i,j} B_{i,j}^{n}(s,t)$$

where

$$B_{i,j}^{n}(s,t) = B_{i}^{n}(s) B_{j}^{n}(t)$$



Smoothly Joined Patches:



- Can be achieved by ensuring that

$$(P_{i,n} - P_{i,n-1}) = \beta(Q_{i,1} - Qi, 0) \text{ for } \beta > 0$$

(and correspondingly for other boundaries)

Rendering via Subdivision:

- Divide up into polygons:
  1. By stepping

$$
\begin{aligned}
s &= 0, \delta, 2\delta, \ldots, 1 \\
t &= 1, \gamma, 2\gamma, \ldots, 1
\end{aligned}
$$

     and joining up sides and diagonals to produce a triangular mesh
  2. By subdividing and rendering the control polygon

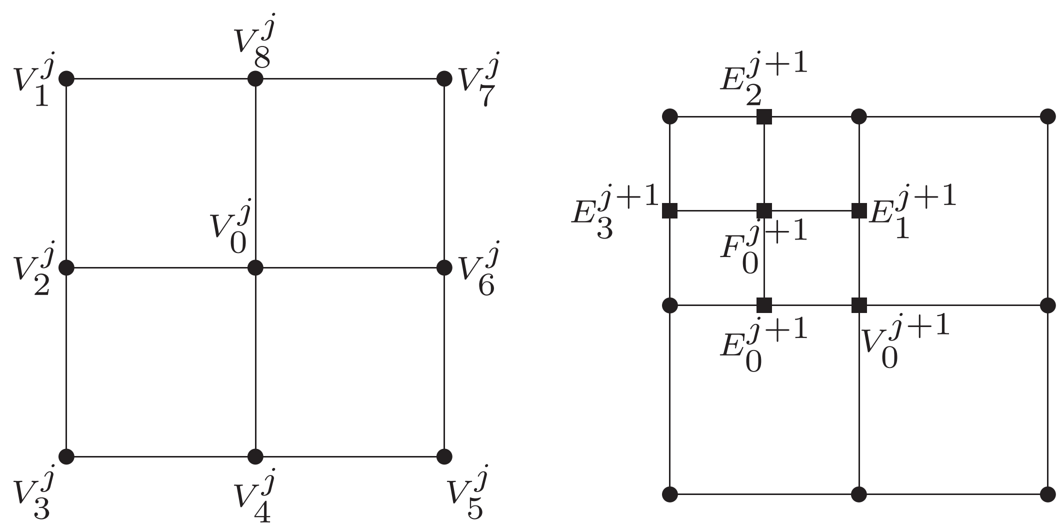
Subdivision for Polyhedra

Regular Polyhedra (Platonic Solids)

- Tetrahedron
- Octahedron
- Icosahedron
- Hexahedron (Cube)
- Dodecahedron

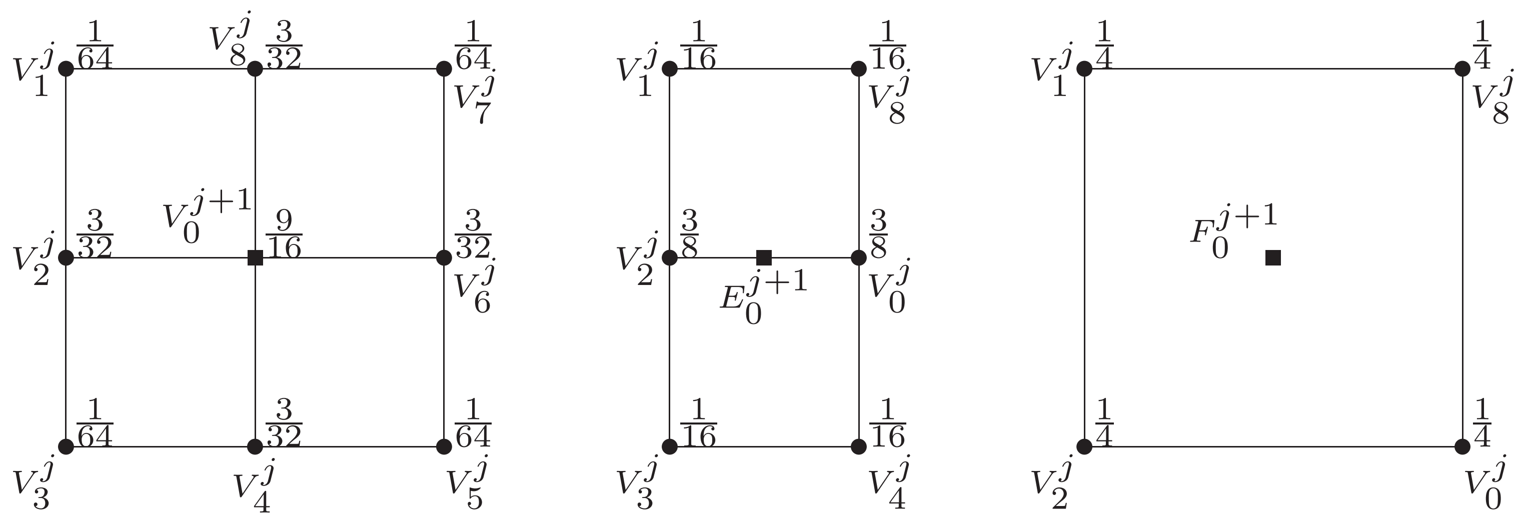
Catmull Clark

Refinement rule used by Catmull-Clark subdivision scheme is as follows. New vertices are added on each edge and in the center. When connected, 4 new level $j+1$ quadrilaterals are produced from the single level j quadrilateral.



Catmull-Clark subdivision scheme. Circles are the j level and Squares are the $j+1$ level.

The vertex rule, edge rule and face rule are shown in the following figure. Each black circle represents a vertex at level j; we compute the position of the vertex at level $j+1$ marked by the black square. Note that for the vertex rule, the control vertex with weight $\frac{9}{16}$ and the new vertex aren't necessarily aligned as they are in the figure.



- Vertex rule:

$$V_0^{j+1} = \frac{9}{16}V_0^j + \frac{3}{32}(V_2^j + V_4^j + V_6^j + V_8^j) + \frac{1}{64}(V_1^j + V_3^j + V_5^j + V_7^j)$$

- Edge rule:

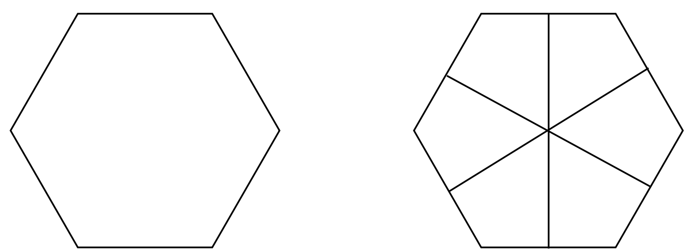
$$E_1^{j+1} = \frac{3}{8}(V_0^j + V_2^j) + \frac{1}{16}(V_1^j + V_3^j + V_4^j + V_8^j)$$

- Face rule:

$$F_0^{j+1} = \frac{1}{4}(V_1^j + V_2^j + V_0^j + V_8^j)$$

Arbitrary Meshes

We have defined Catmull-Clark scheme on quadrilaterals; it can be extended to handle arbitrary polygonal meshes. Observe that if we do one step of refinement, splitting each edge into two and inserting a new vertex for each face (see below Figure), we get a mesh which has only quadrilateral faces. On all other steps of subdivision standard rule described above can be applied.



Splitting a hexagon into quadrilaterals.

Reading Assignment and News

Chapter 11 pages 569 - 583, of Recommended Text.

Please also track the News section of the Course Web Pages for the most recent Announcements related to this course.

(http://www.cs.utexas.edu/users/bajaj/graphics25/cs354/)