CS130 : Computer Graphics

Lecture 16: Curves (cont.)

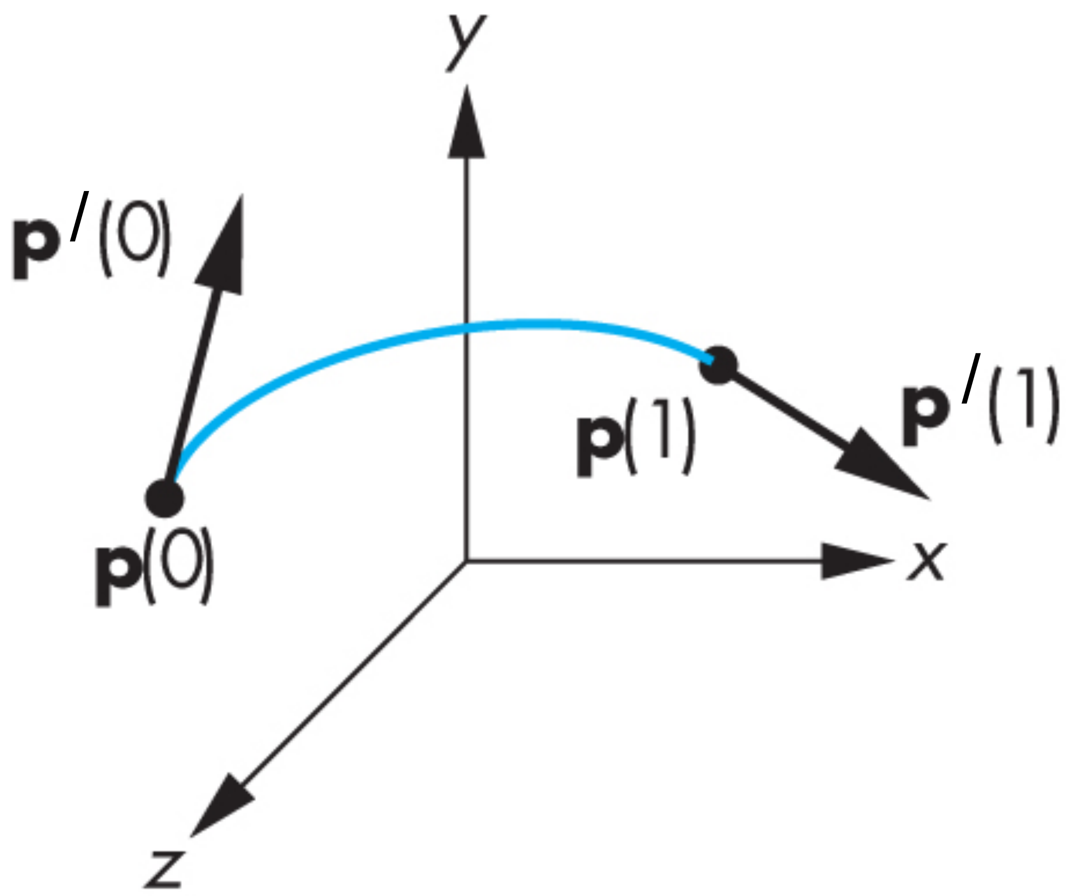
Tamar Shinar
Computer Science & Engineering
UC Riverside

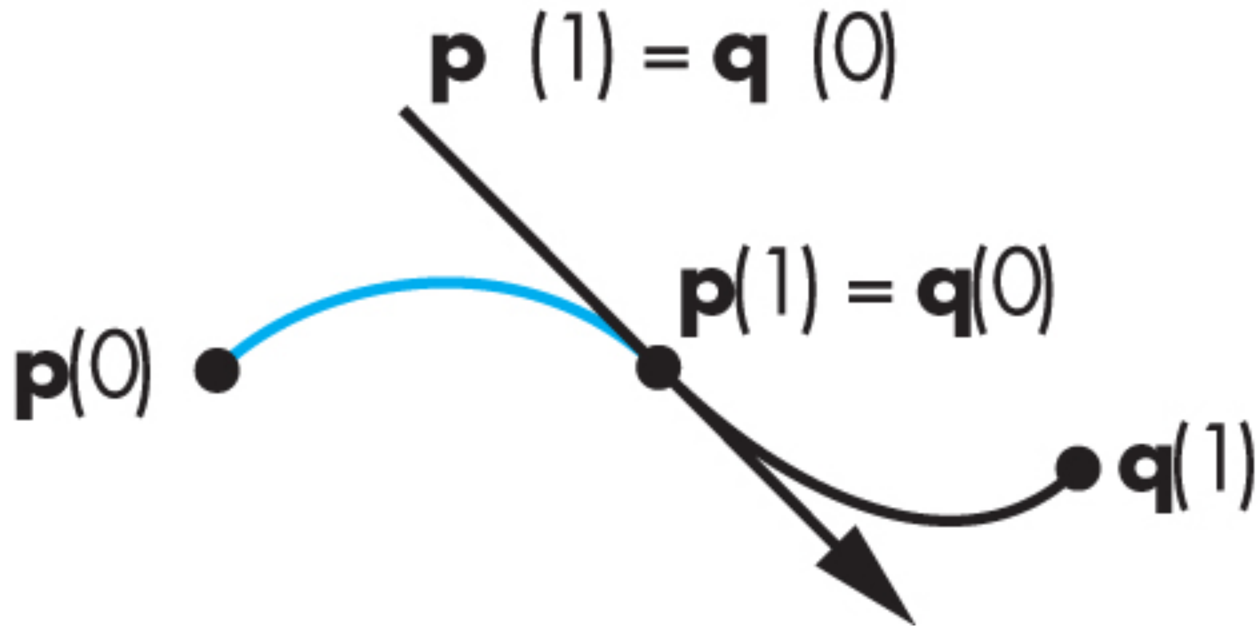
Cubic Hermite Curves

Cubic Hermite Curves

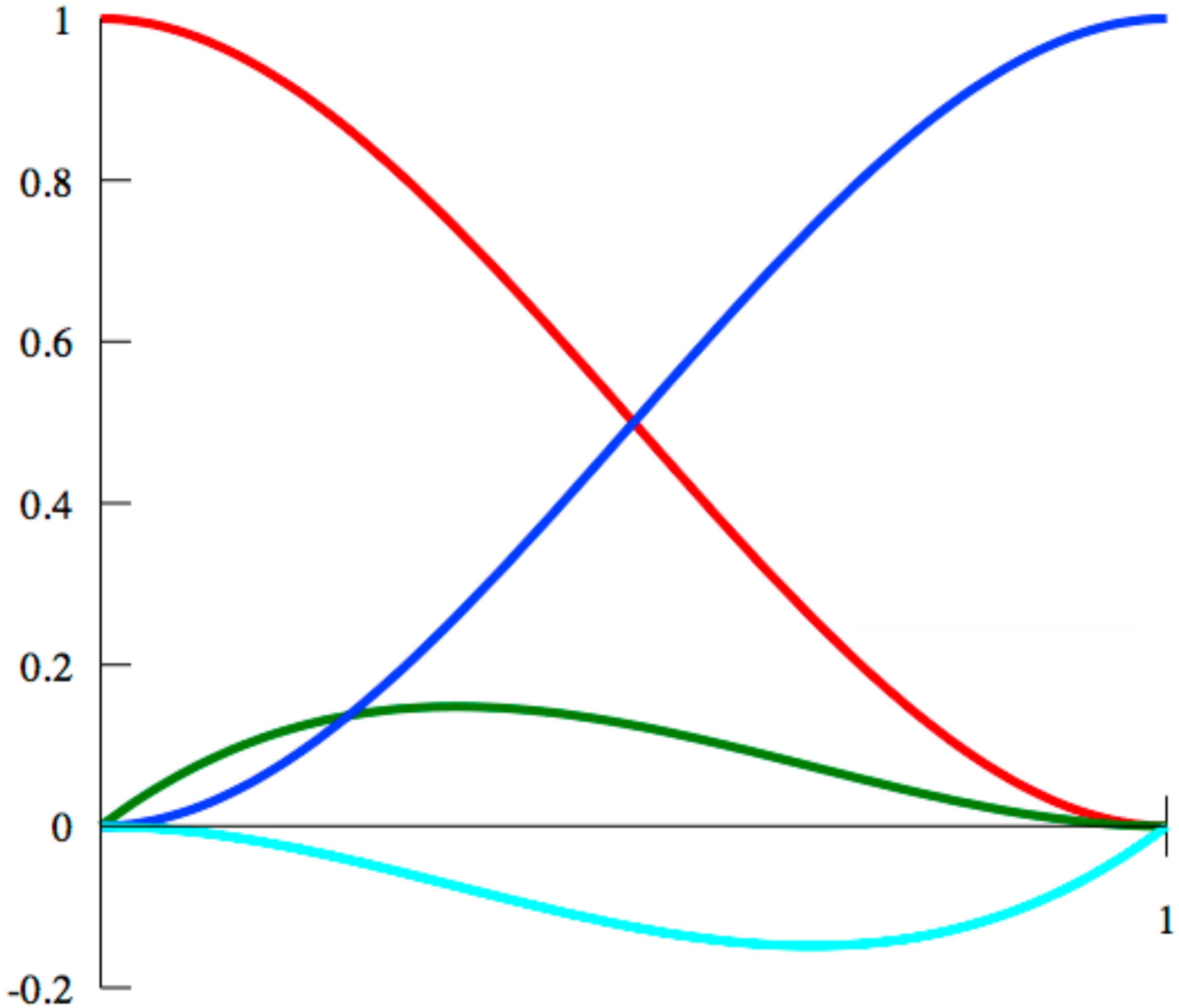
Specify endpoints and derivatives

construct curve with C^1 continuity





Hermite blending functions



[Wikimedia Commons]

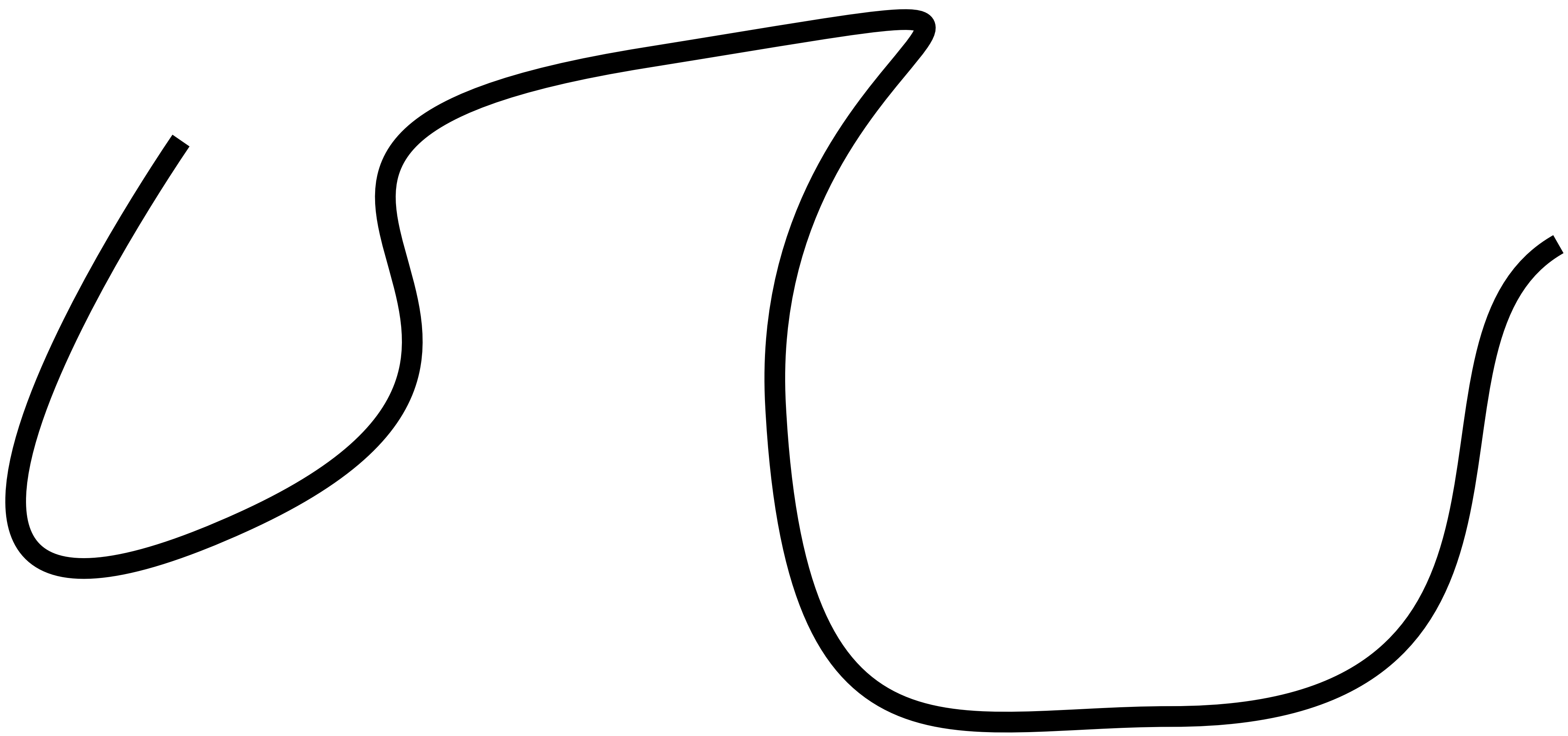
$$b_0(u) = 2u^3 - 3u^2 + 1$$

$$b_1(u) = -2u^3 + 3u^2$$

$$b_2(u) = u^3 - 2u^2 + u$$

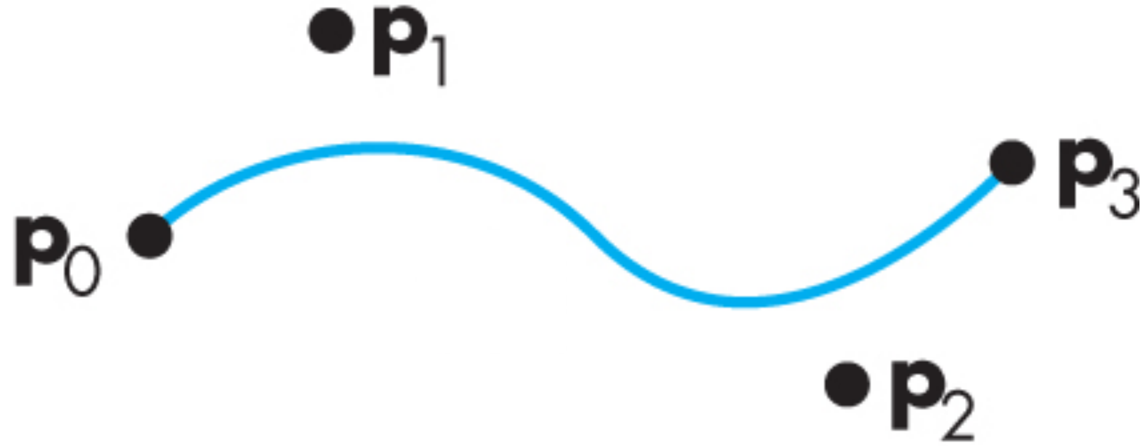
$$b_3(u) = u^3 - u^2$$

Example: keynote curve tool



Interpolating vs. Approximating Curves

Interpolating

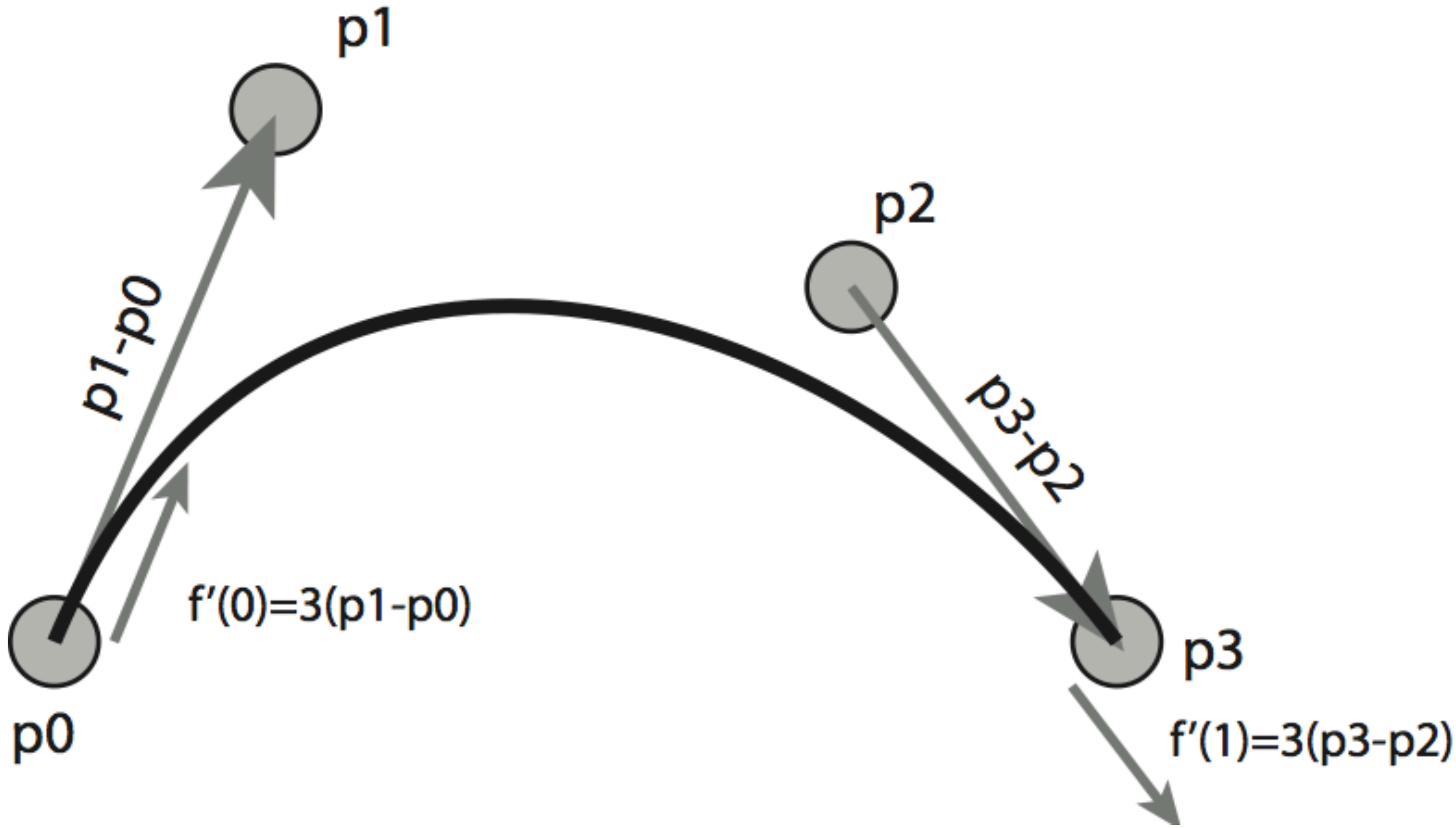


Approximating
(non-interpolating)

approximating

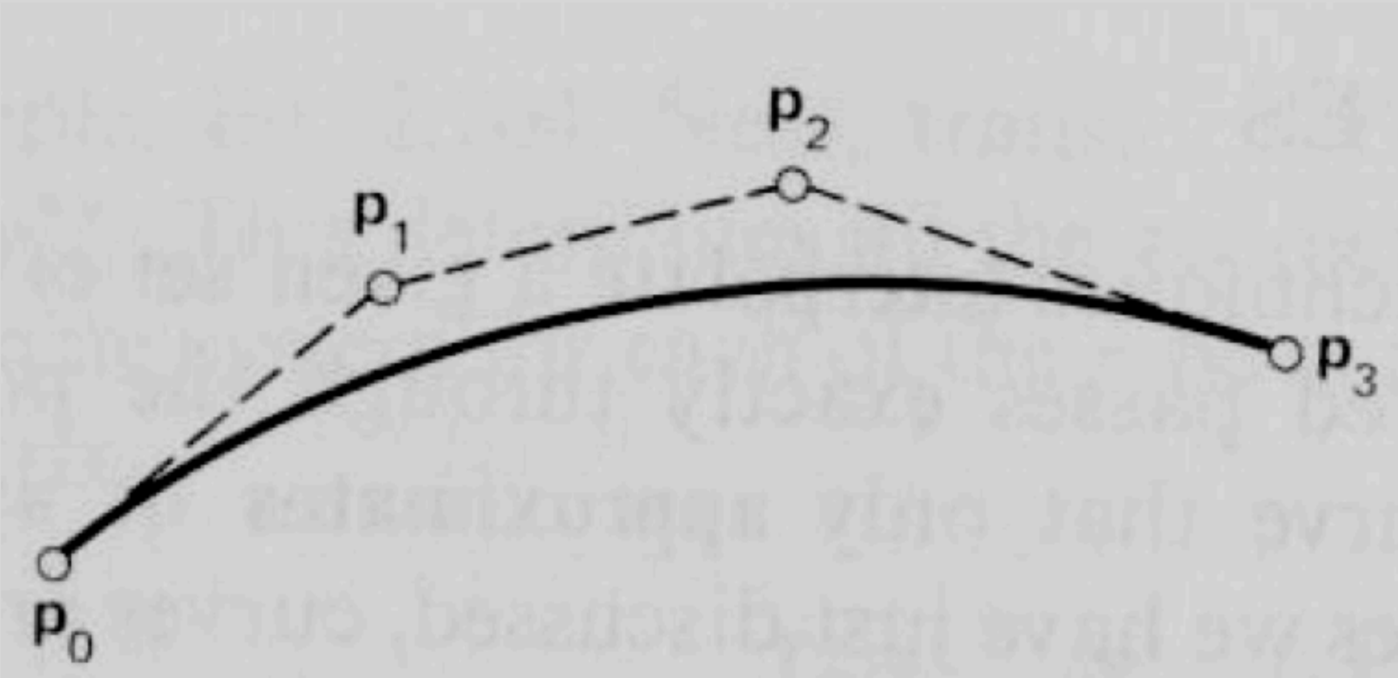
Cubic Bezier Curves

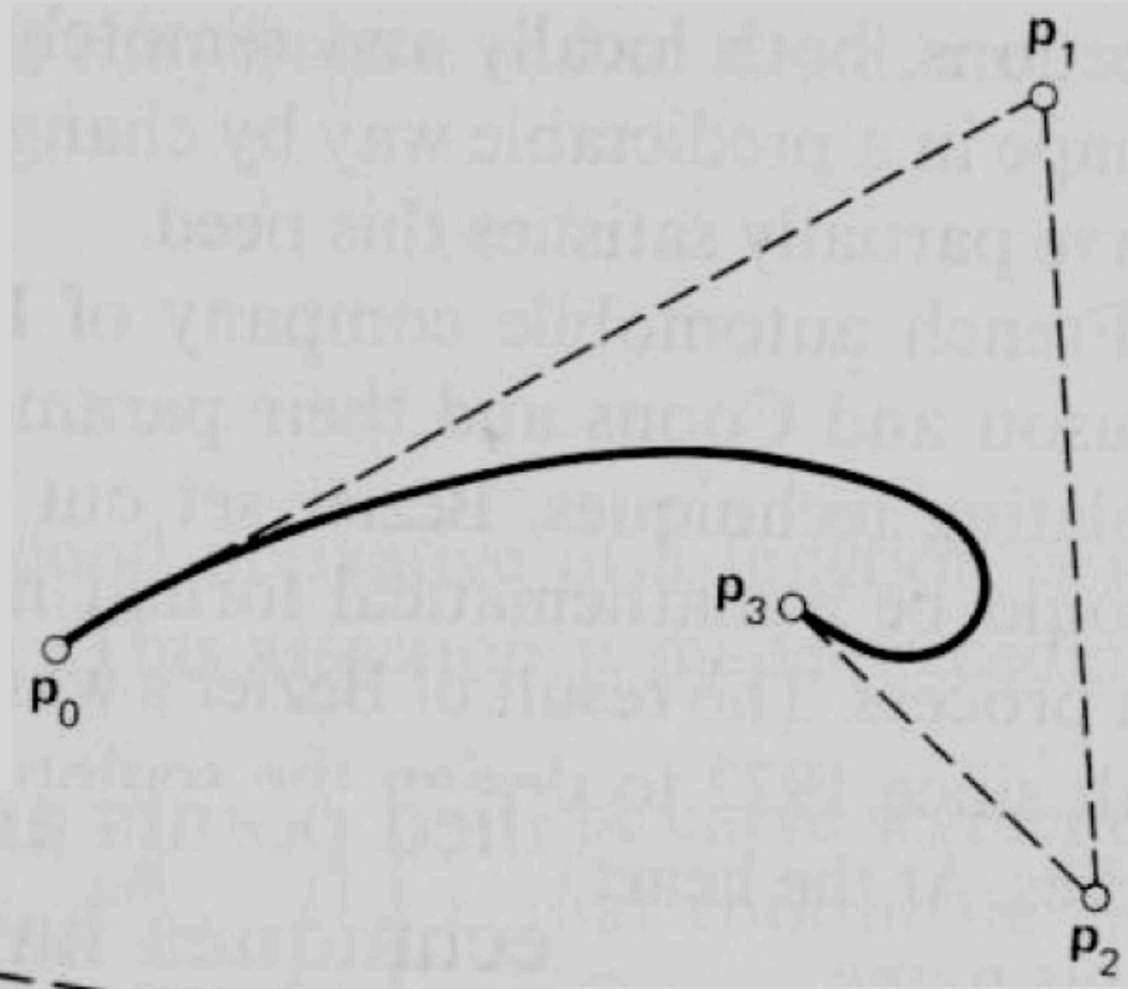
Cubic Bezier Curves

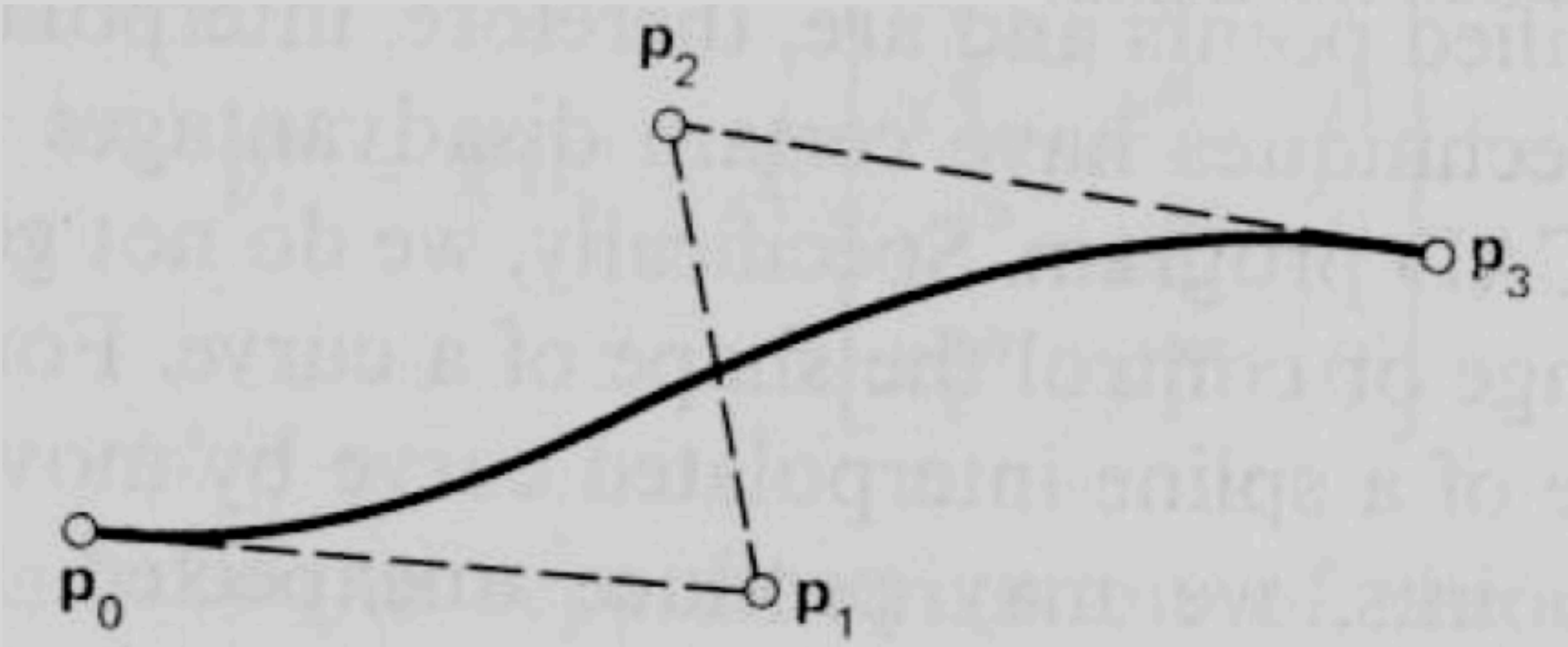


-The curve interpolates its first (u=0) and last (u = 1) control points
- first derivative at the beginning is the vector from first to second point, scaled by degree

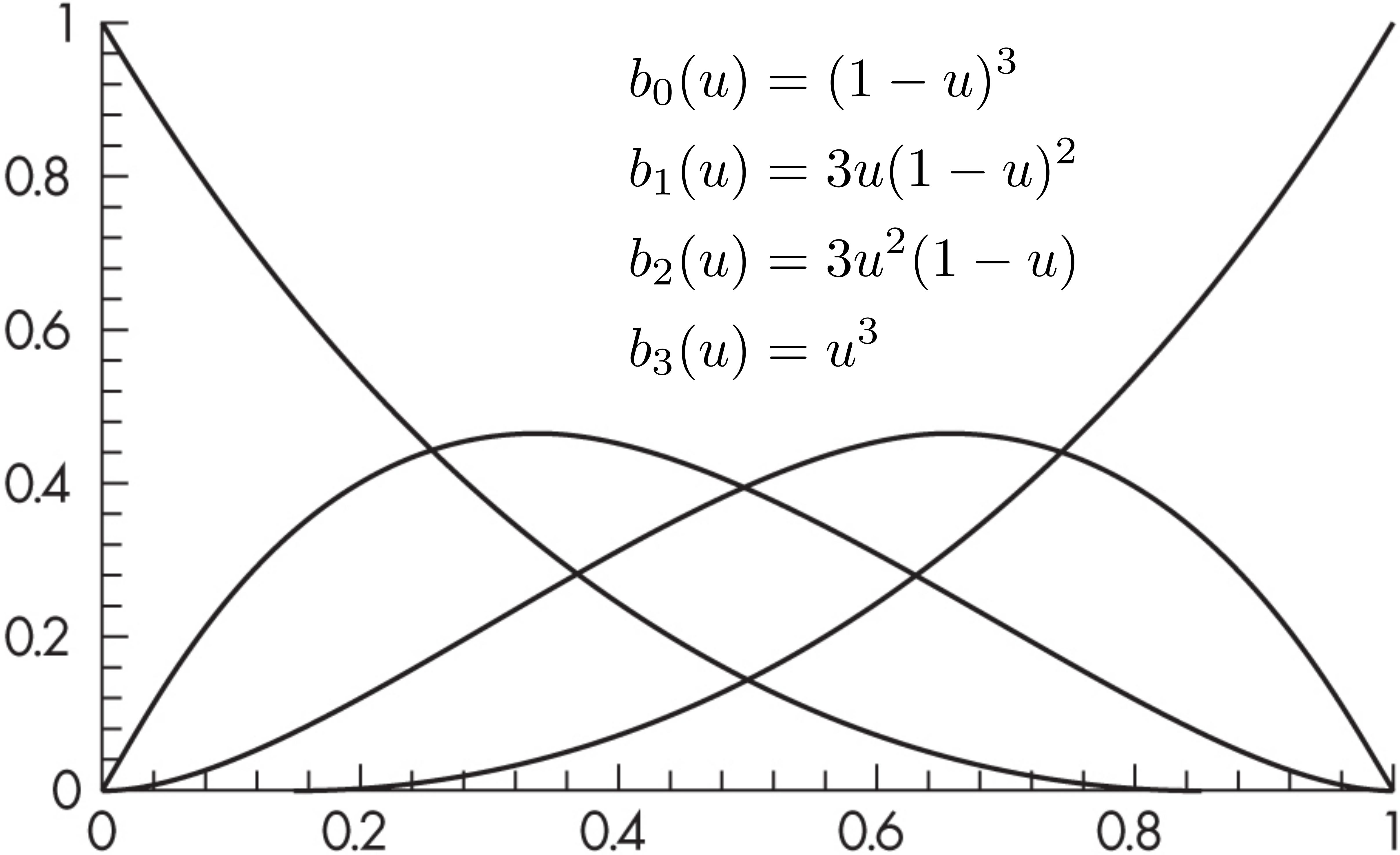
Cubic Bezier Curve Examples



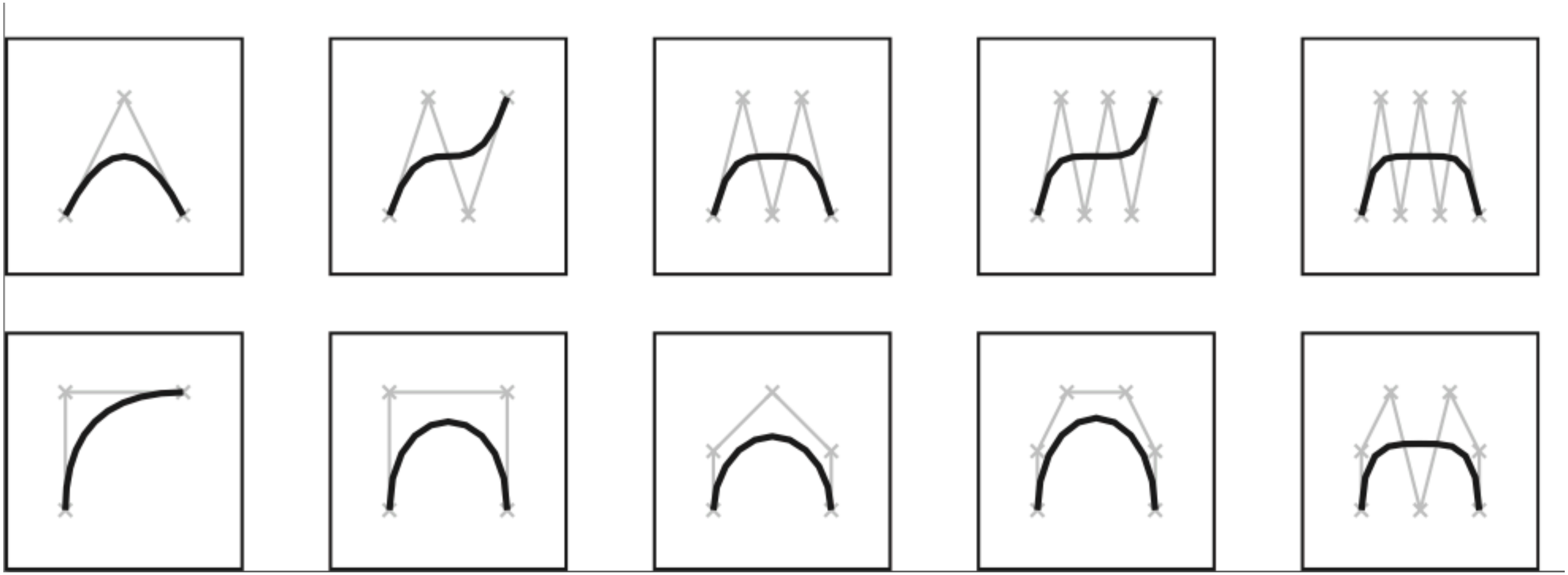




Cubic Bezier blending functions



Bezier Curves Degrees 2-6



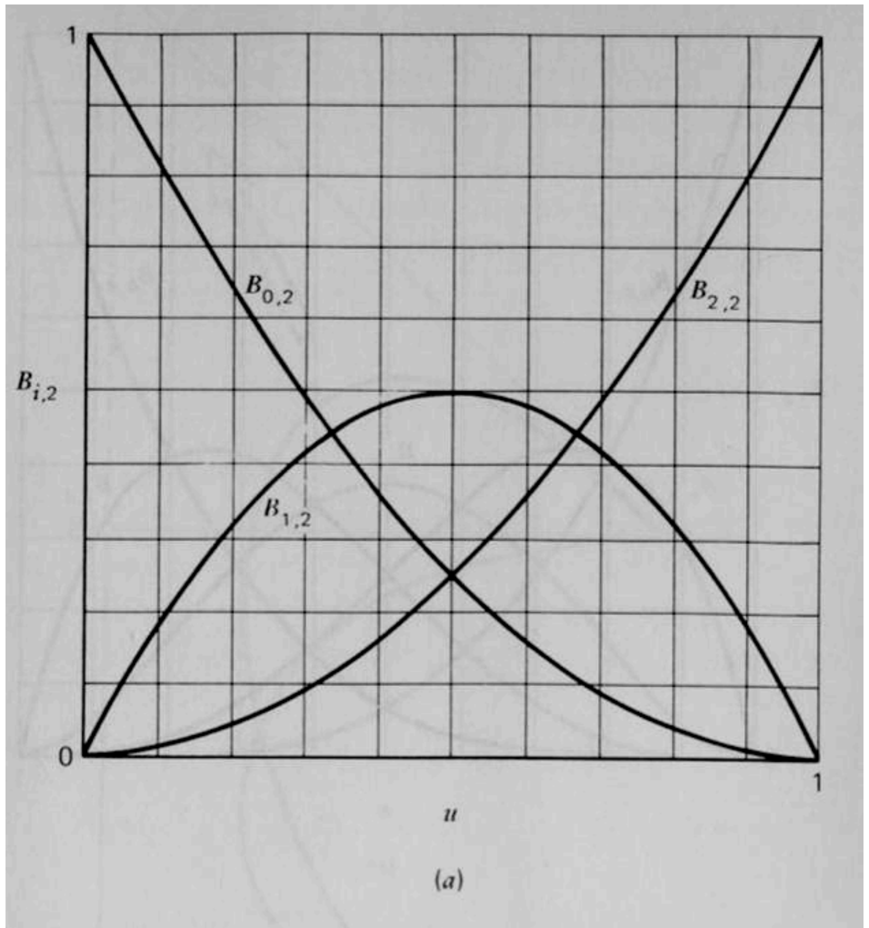
Bernstein Polynomials

- The blending functions are a special case of the Bernstein polynomials

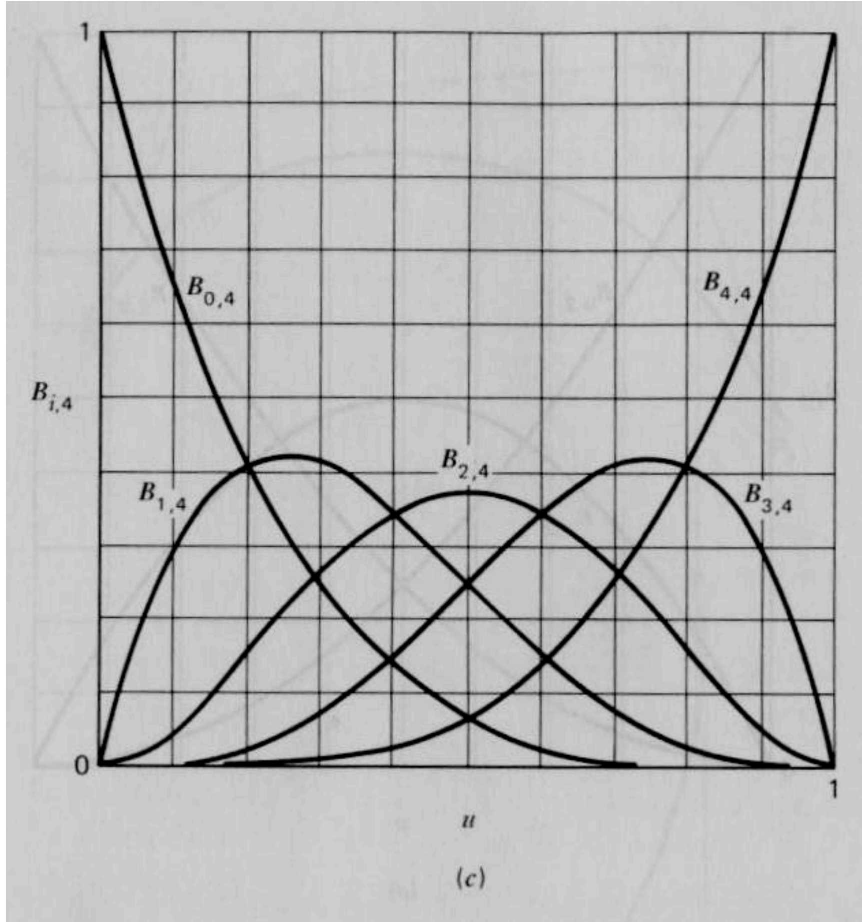
$$b_{\mathrm{kd}}(u) = \frac{d!}{k!(d-k)!} u^k (1-u)^{d-k}$$

- These polynomials give the blending polynomials for any degree Bezier form
  - All roots at 0 and 1
  - For any degree they all sum to 1
  - They are all between 0 and 1 inside (0,1)

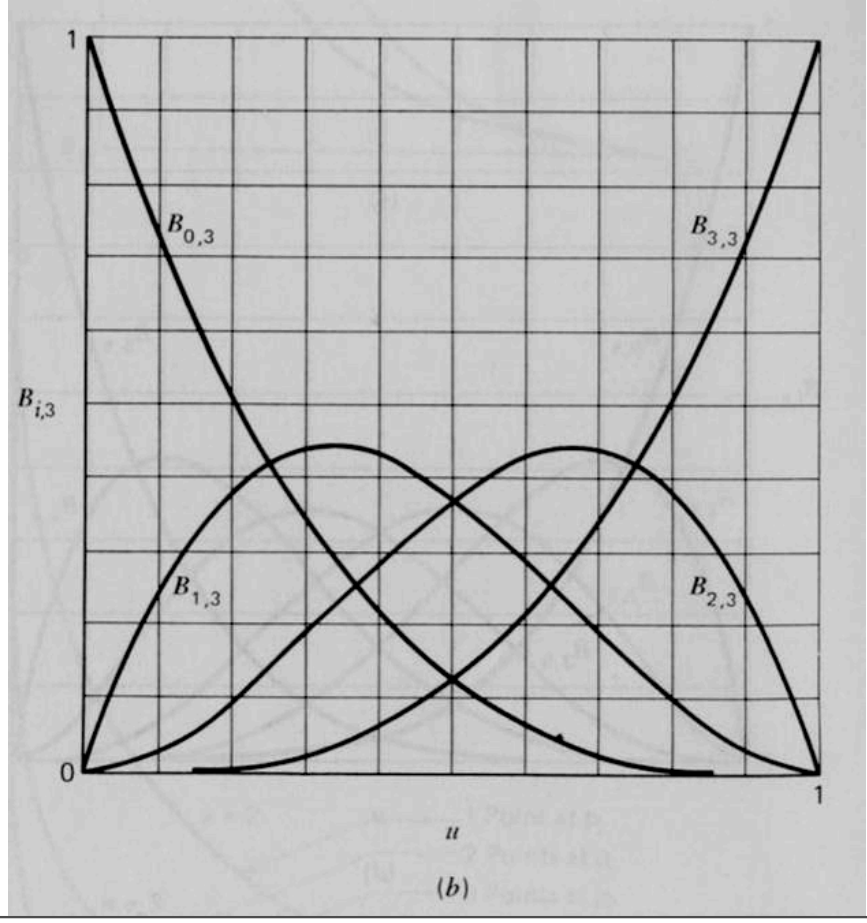
n = 3


1
0
1
u
$B_{i,2}$
$B_{0,2}$
$B_{1,2}$
$B_{2,2}$
(a)

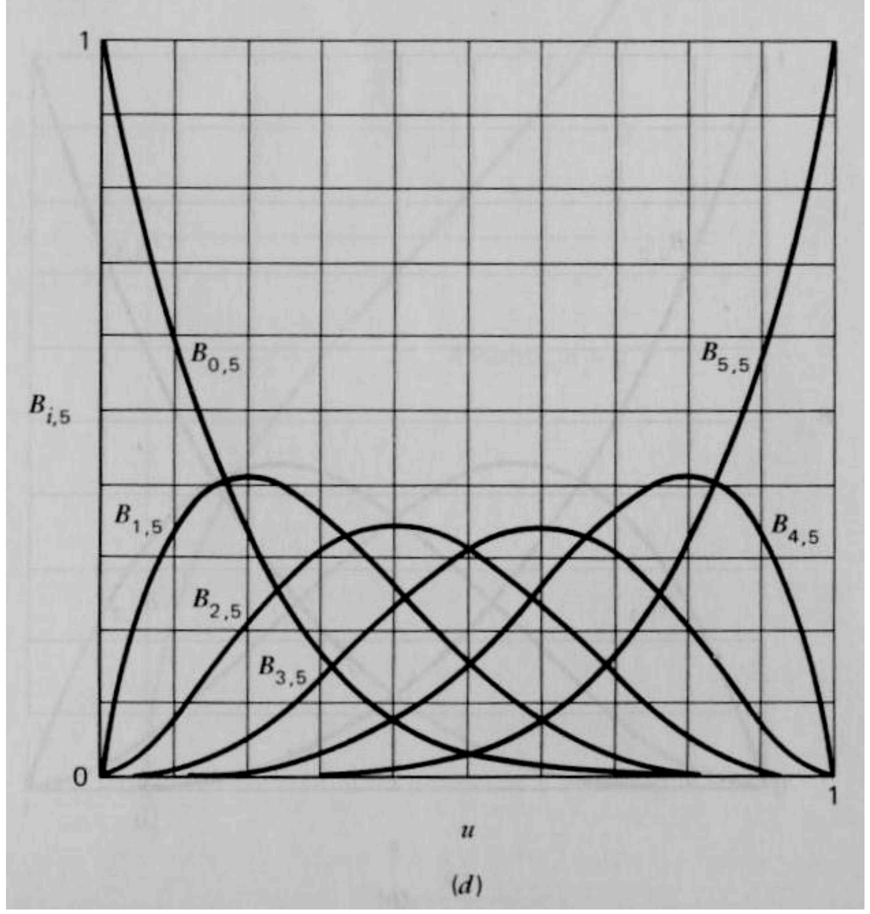
n = 5


1
0
1
u
$B_{i,4}$
$B_{0,4}$
$B_{1,4}$
$B_{2,4}$
$B_{3,4}$
$B_{4,4}$
(c)

n = 4


1
0
1
u
$B_{i,3}$
$B_{0,3}$
$B_{1,3}$
$B_{2,3}$
$B_{3,3}$
(b)

n = 6


1
0
1
u
$B_{i,5}$
$B_{0,5}$
$B_{1,5}$
$B_{2,5}$
$B_{3,5}$
$B_{4,5}$
$B_{5,5}$
(d)

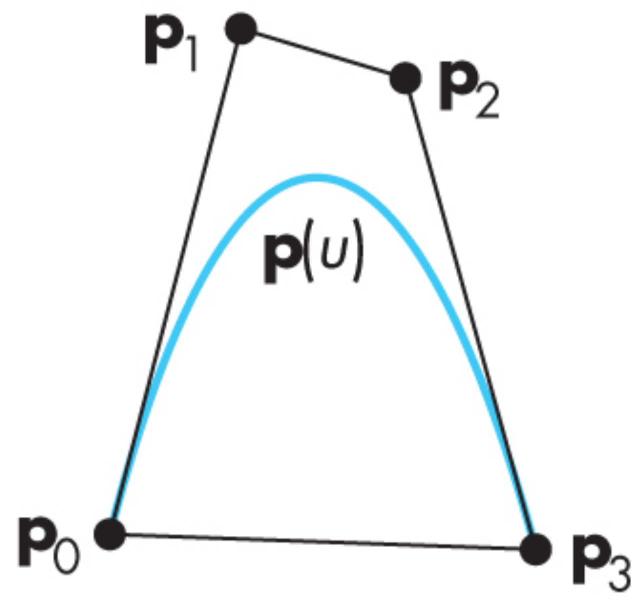
Bezier Curve Properties

- curve lies in the convex hull of the data
- variation diminishing
- symmetry
- affine invariant
- efficient evaluation and subdivision

Shirley Section 15.6.1 (p. 368)

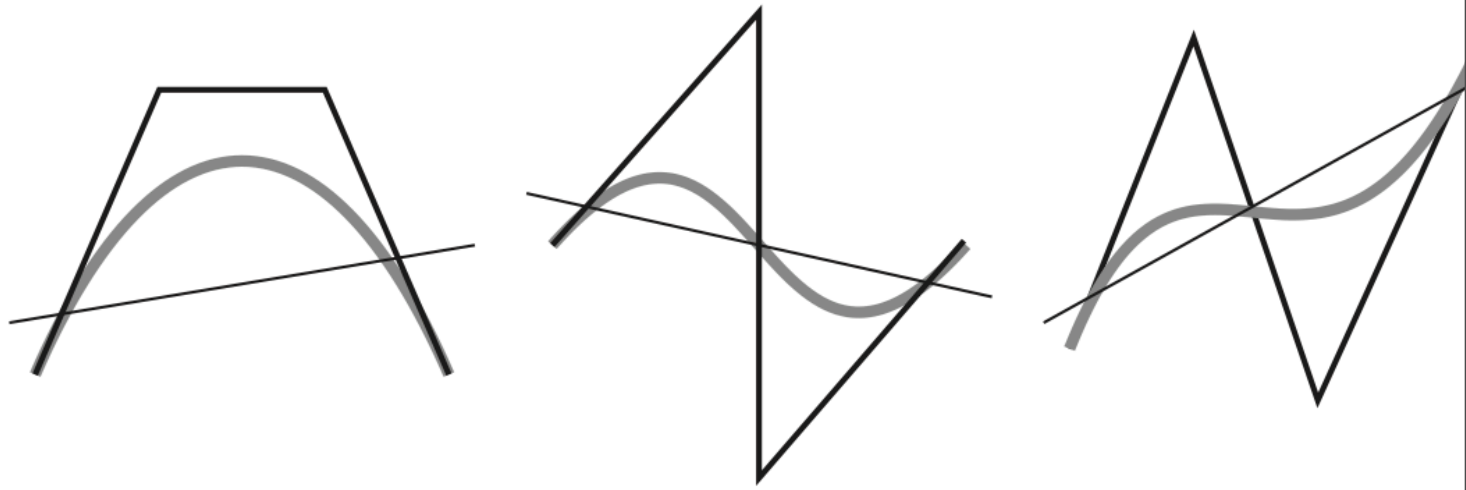
Bezier Curve Properties

- curve lies in the convex hull of the data
- variation diminishing
- symmetry
- affine invariant
- efficient evaluation and subdivision



Bezier Curve Properties

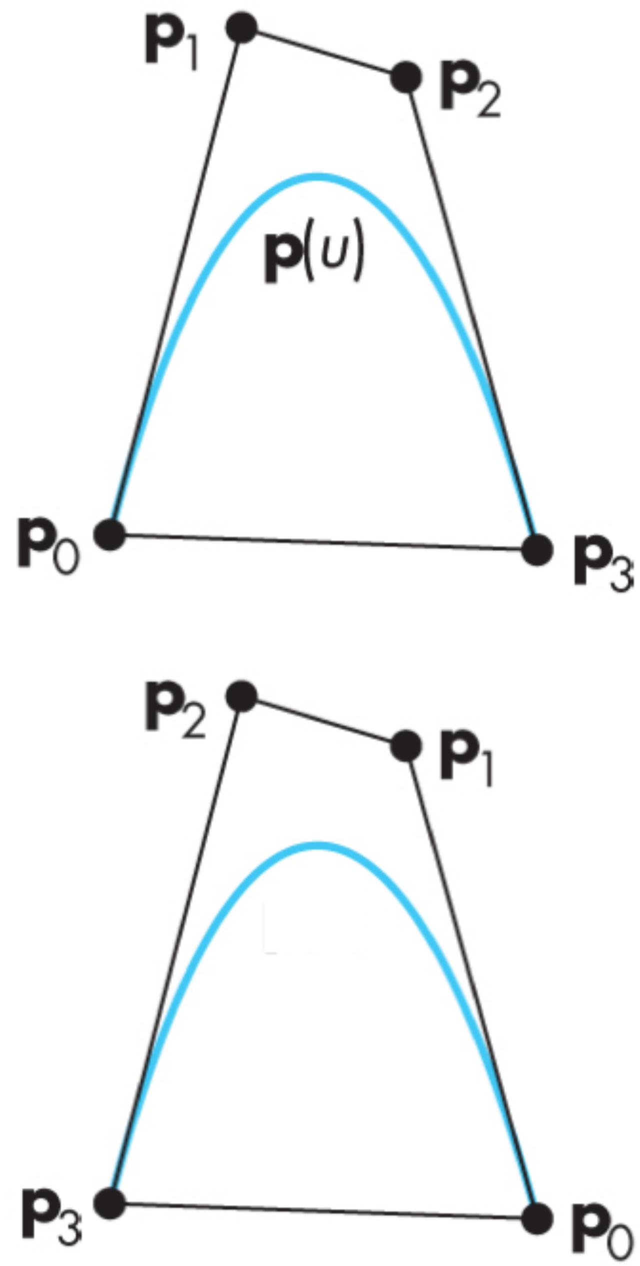
- curve lies in the convex hull of the data
- variation diminishing
- symmetry
- affine invariant
- efficient evaluation and subdivision



Shirley Section 15.6.1 (p. 368)

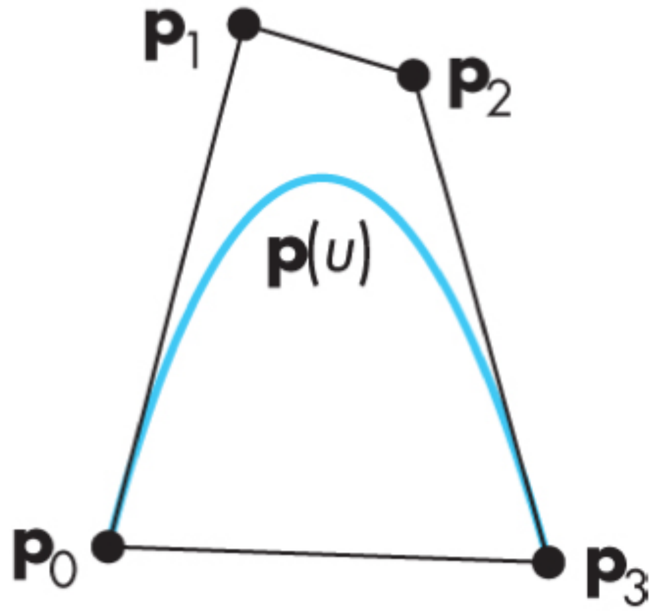
Bezier Curve Properties

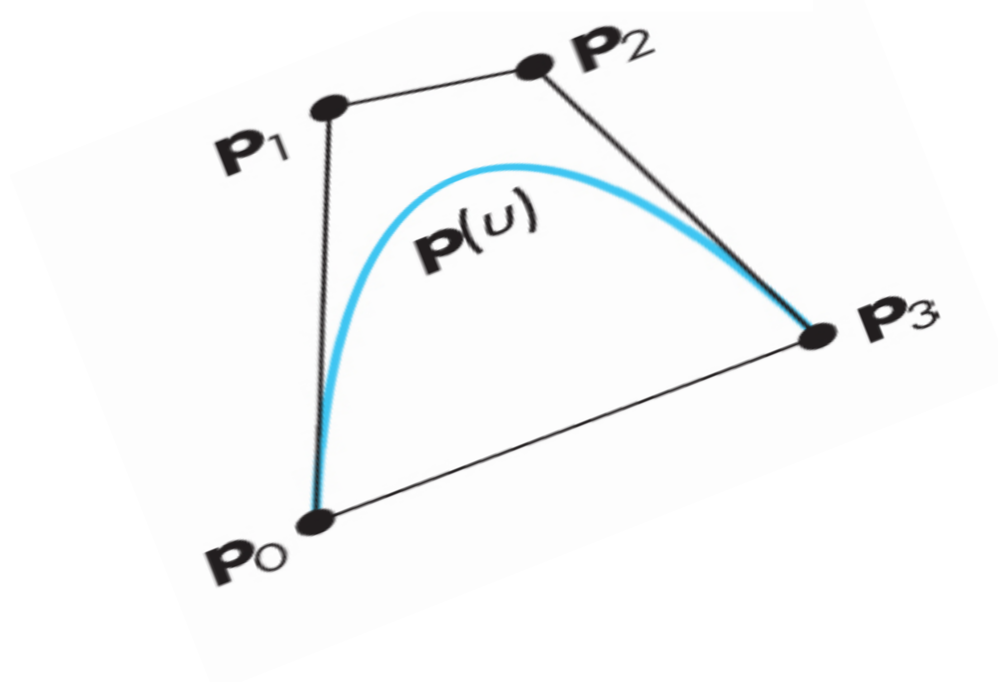
- curve lies in the convex hull of the data
- variation diminishing
- symmetry
- affine invariant
- efficient evaluation and subdivision



Bezier Curve Properties

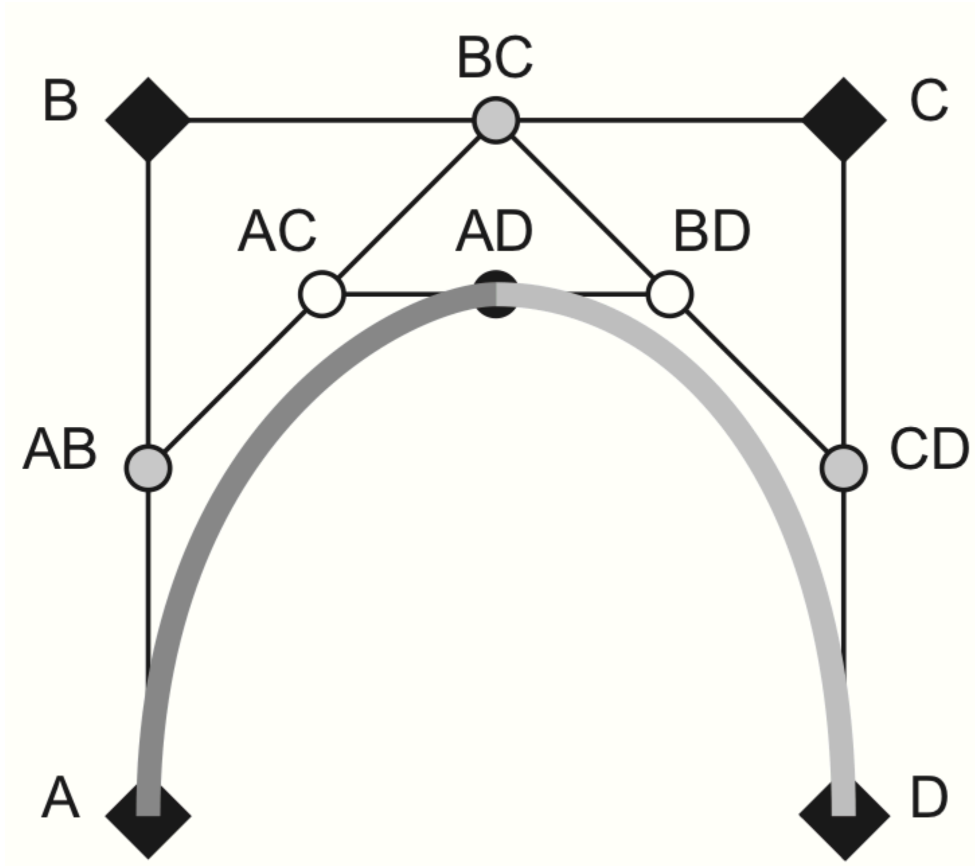
- curve lies in the convex hull of the data
- variation diminishing
- symmetry
- affine invariant
- efficient evaluation and subdivision





Bezier Curve Properties

- curve lies in the convex hull of the data
- variation diminishing
- symmetry
- affine invariant
- efficient evaluation and subdivision



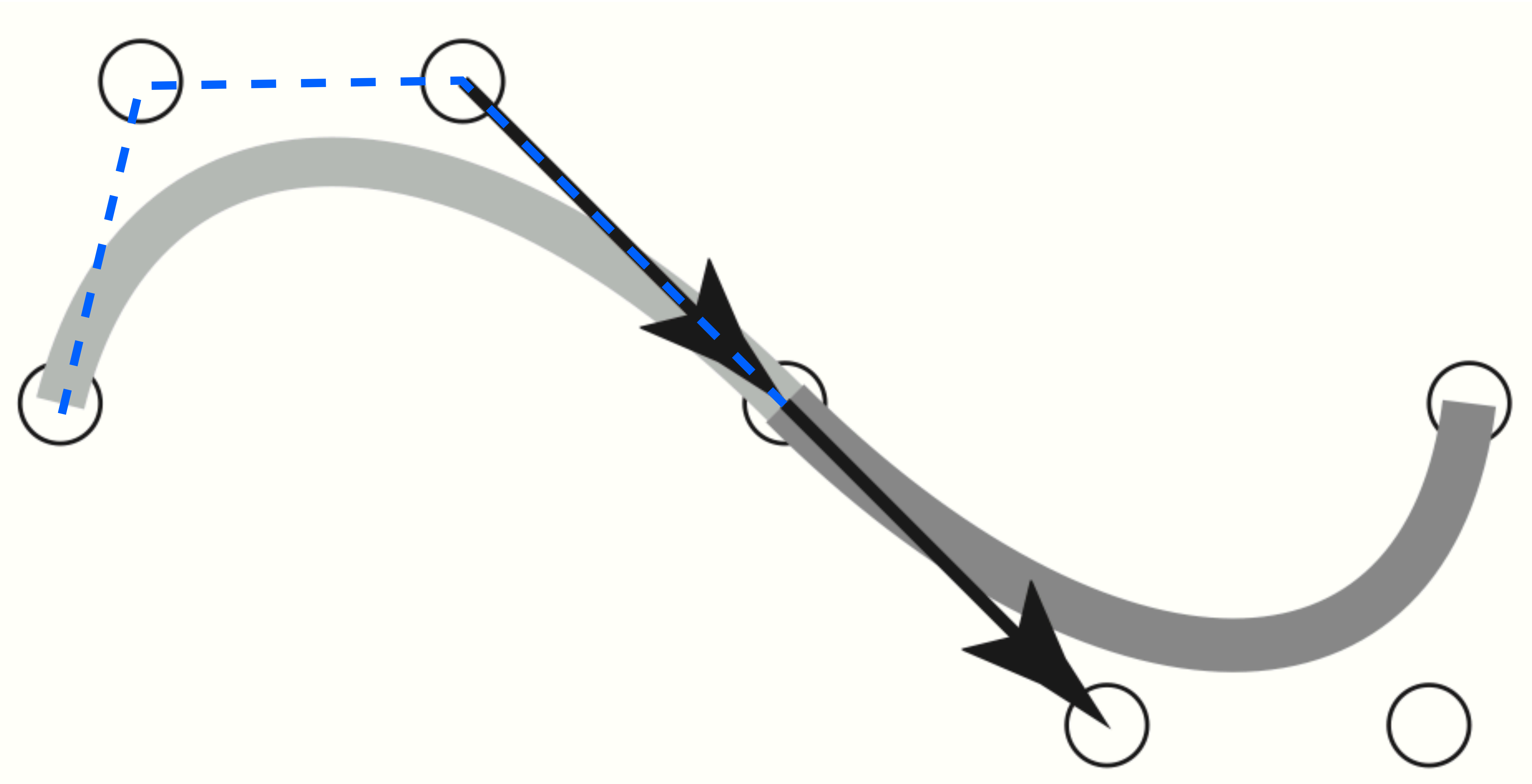
Shirley Section 15.6.1 (p. 368)

Bezier Curve Properties

- curve lies in the convex hull of the data
- variation diminishing
- symmetry
- affine invariant
- efficient evaluation and subdivision

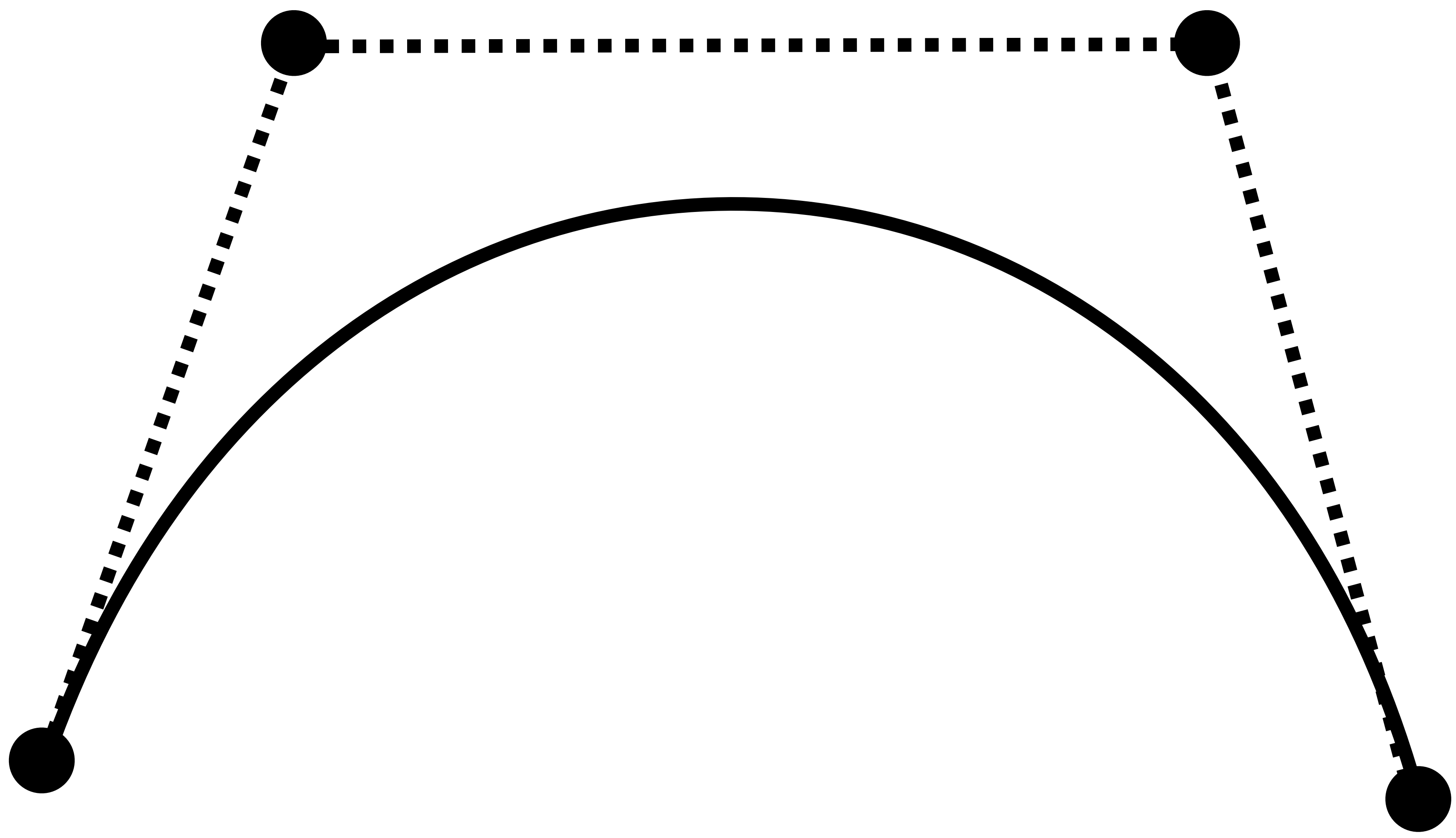
Shirley Section 15.6.1 (p. 368)

Joining Cubic Bezier Curves

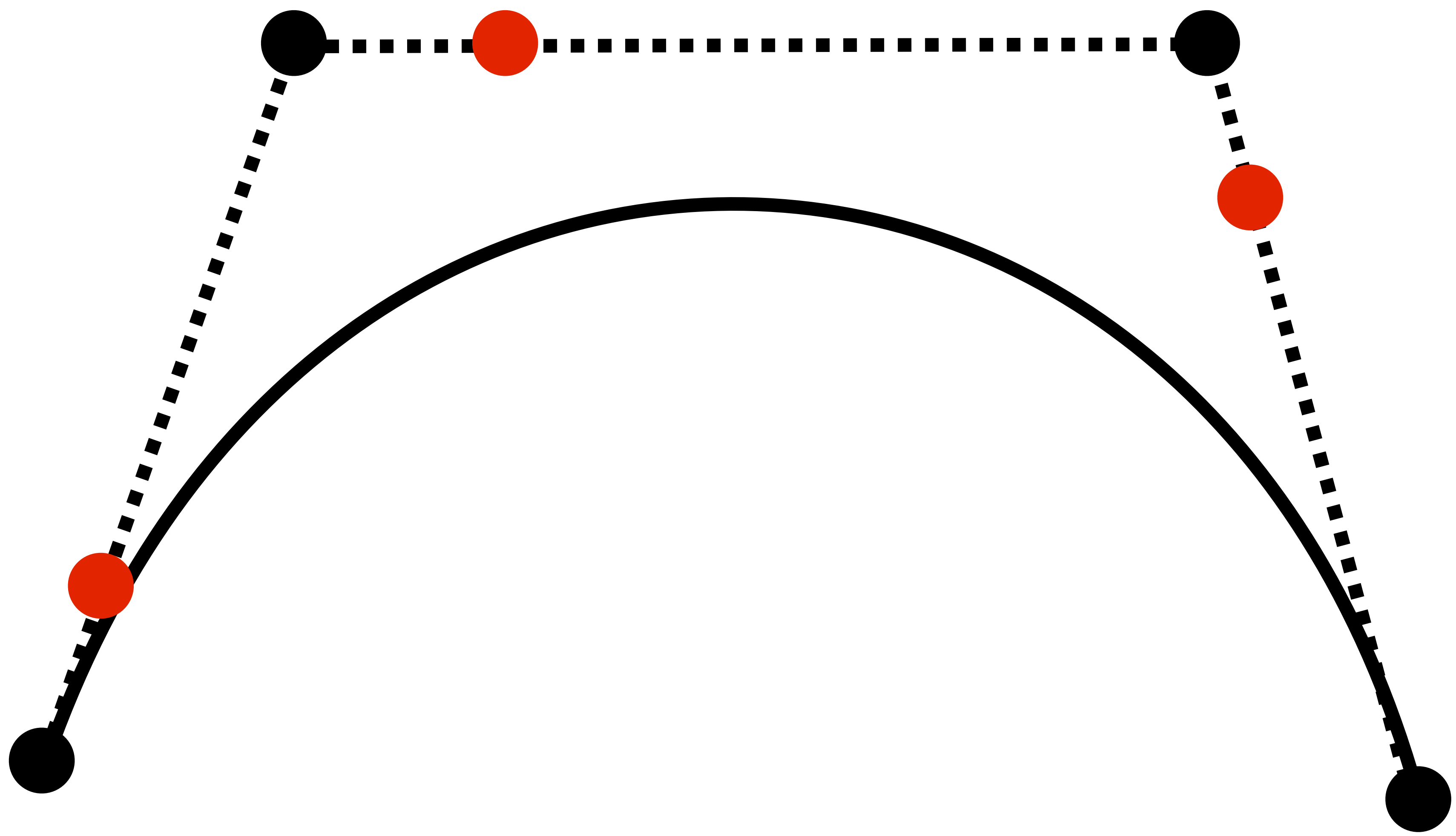


for C1 continuity, the vectors must line up and be the same length
for G1 continuity, the vectors need only line up

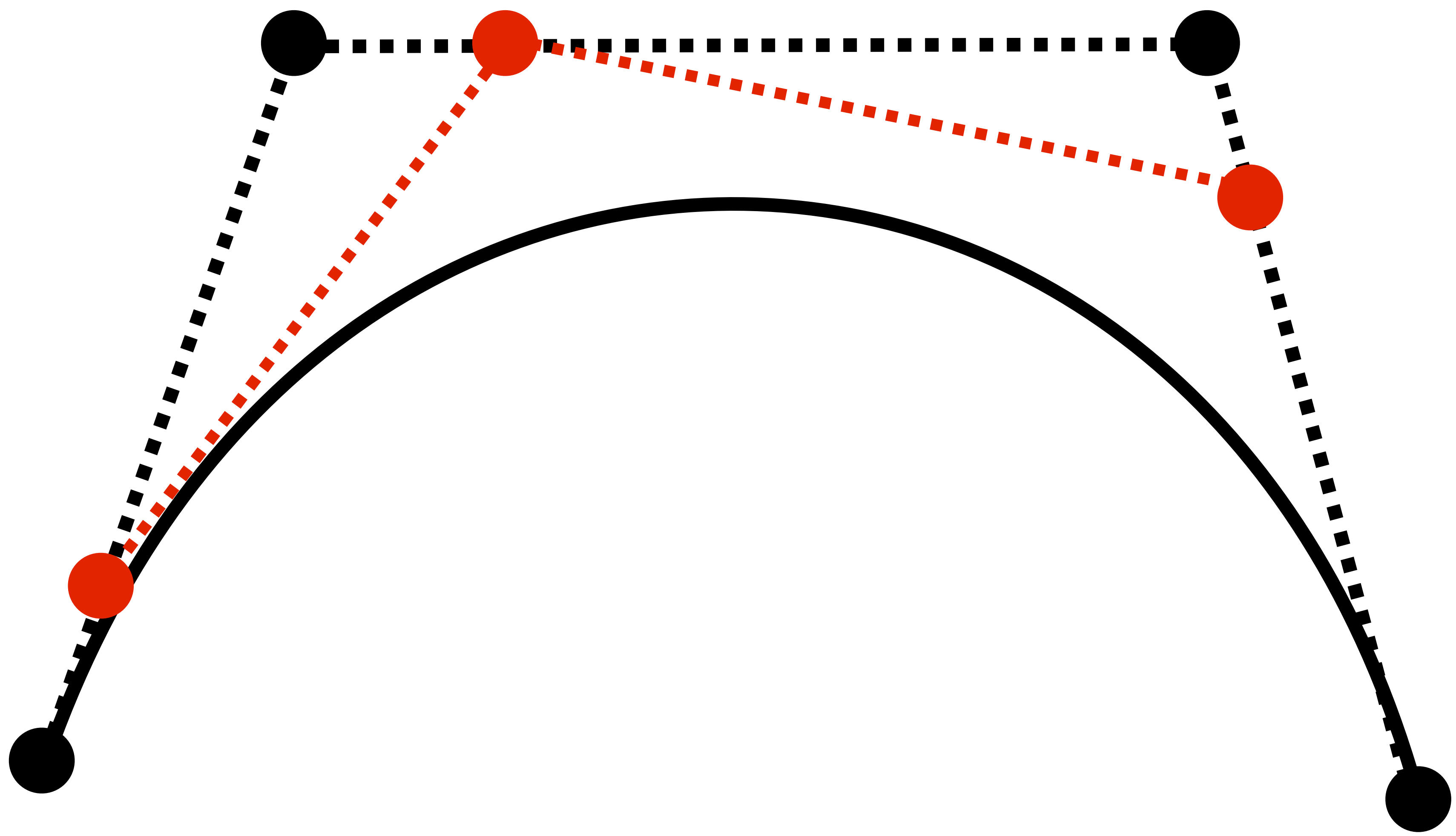
Geometric Construction



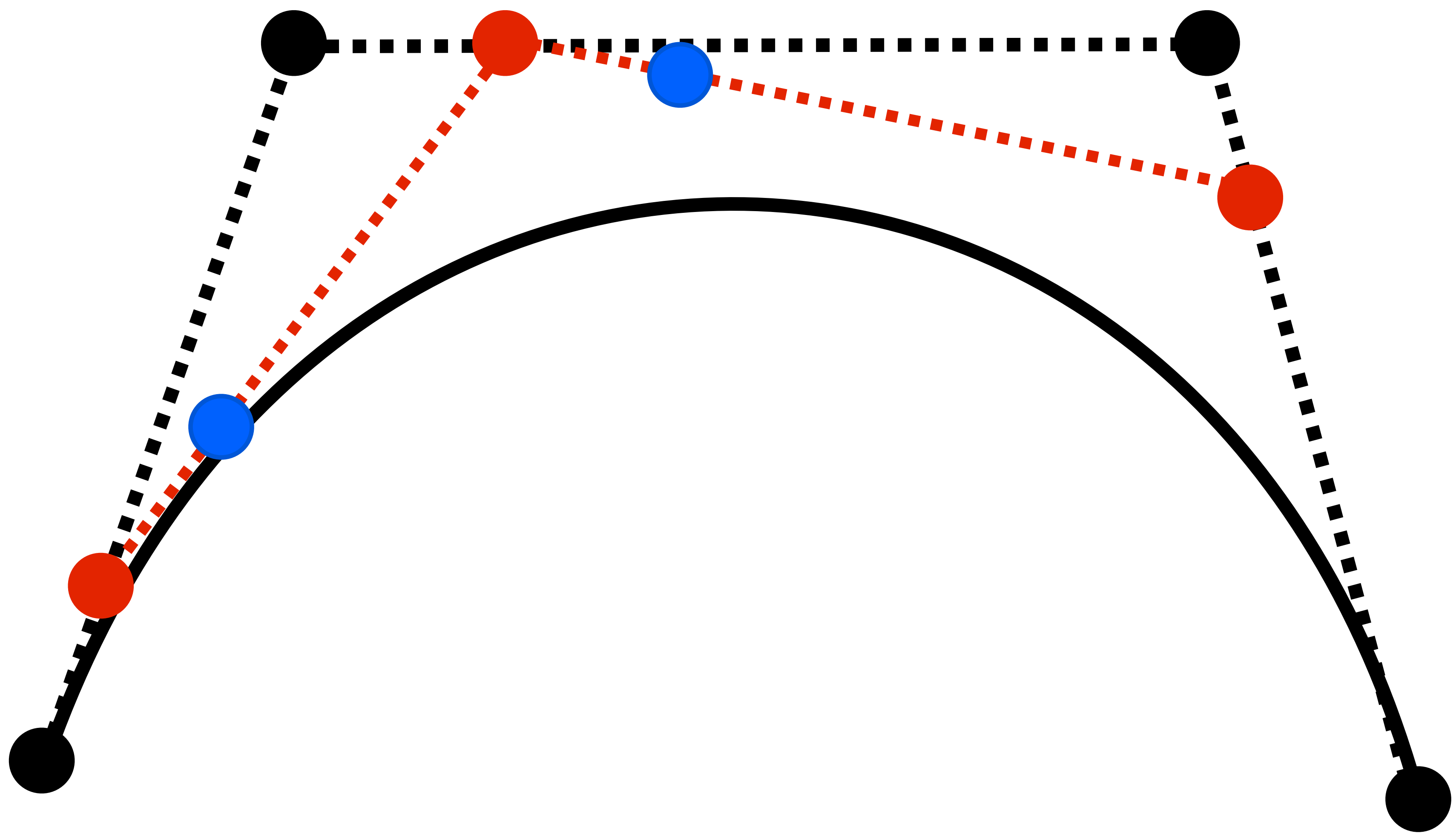
Geometric Construction



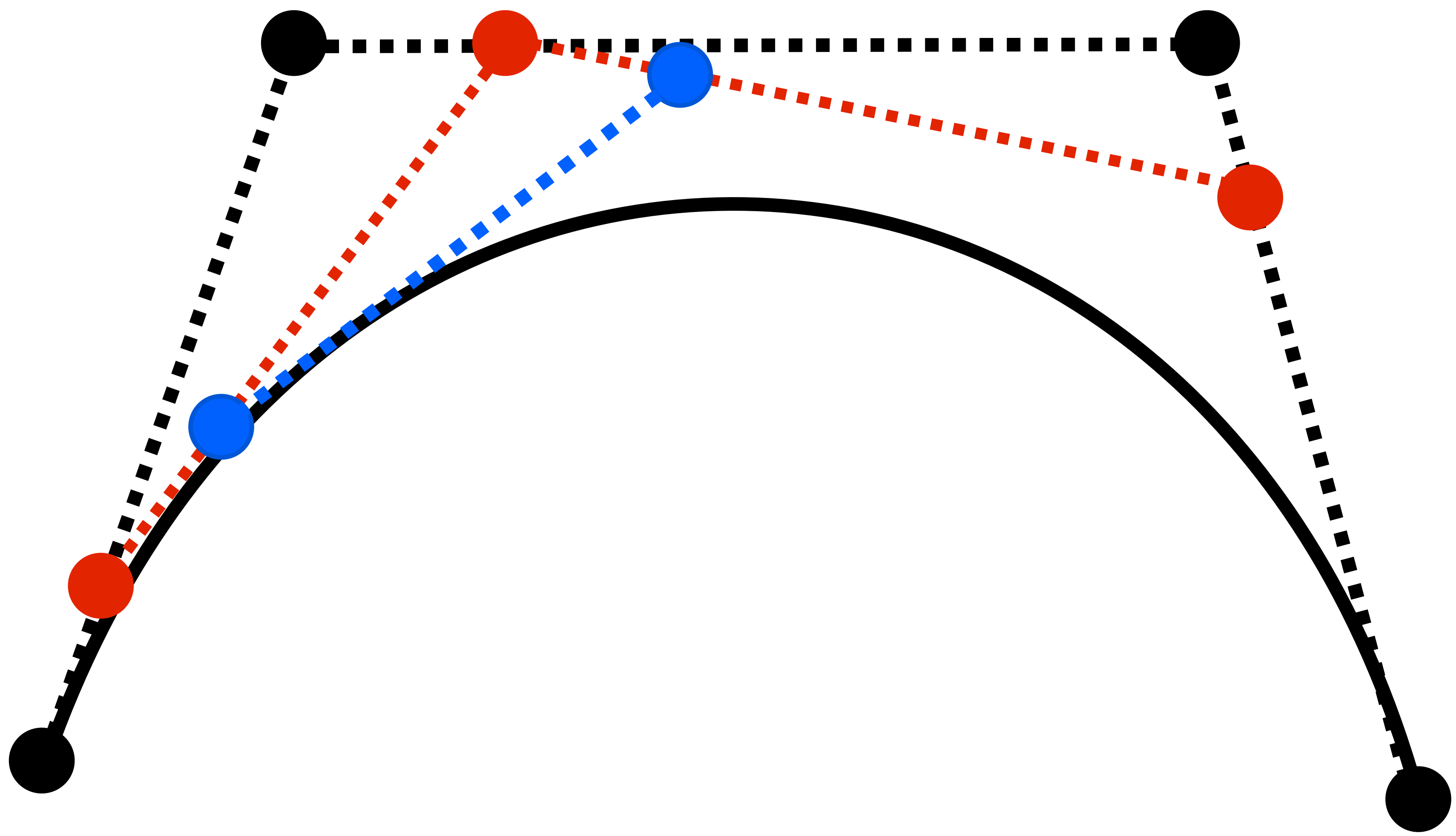
Geometric Construction



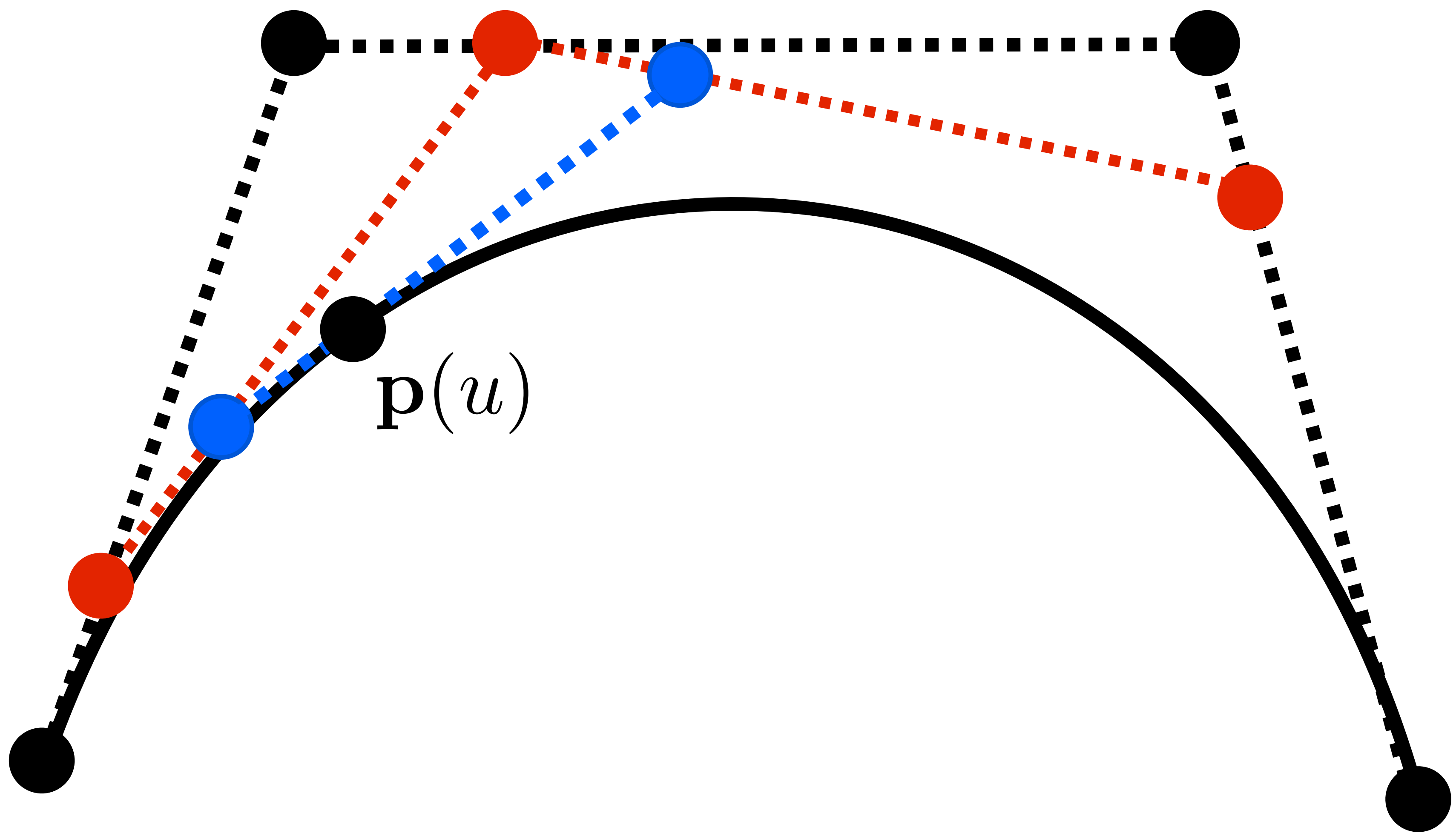
Geometric Construction



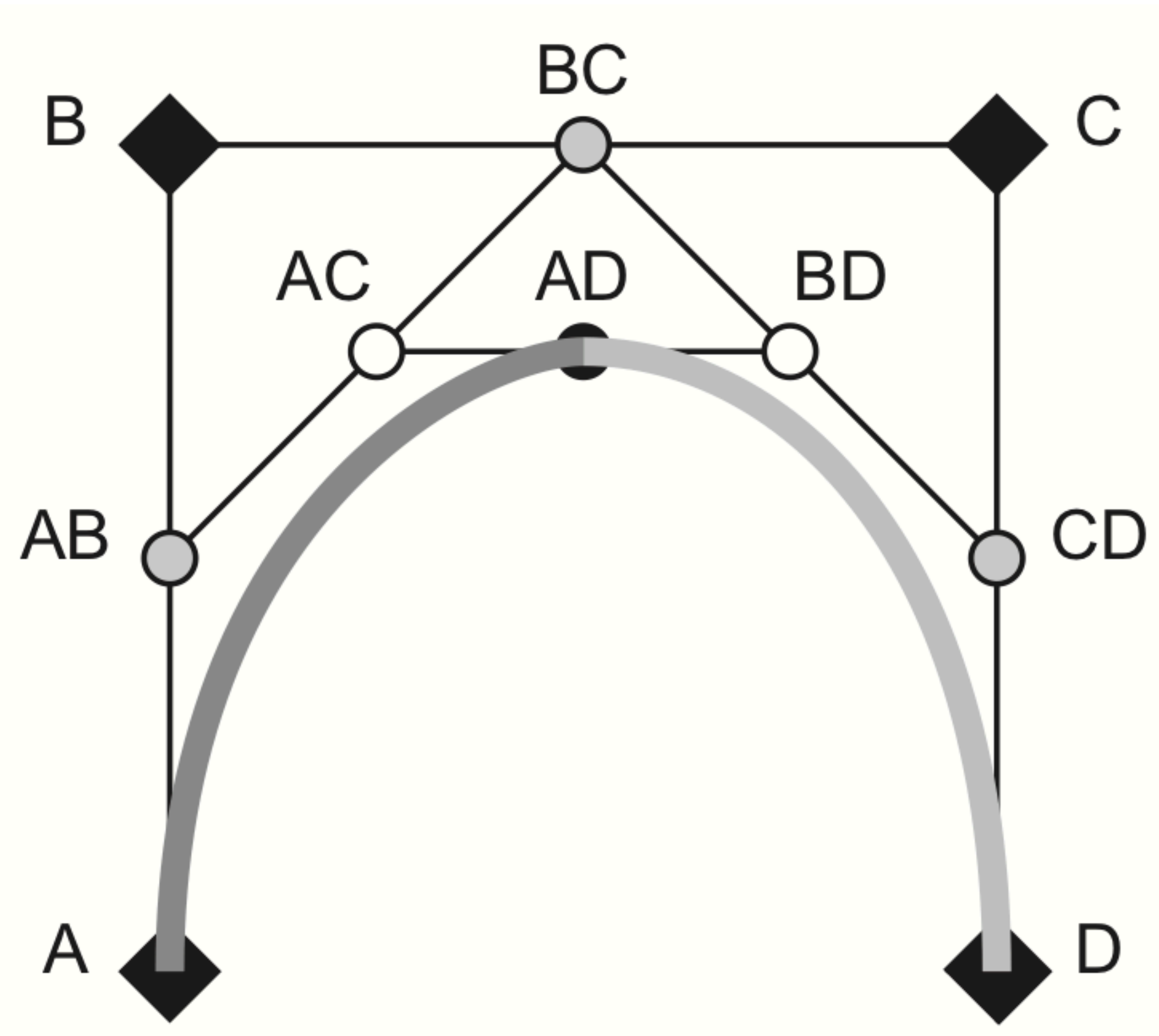
Geometric Construction

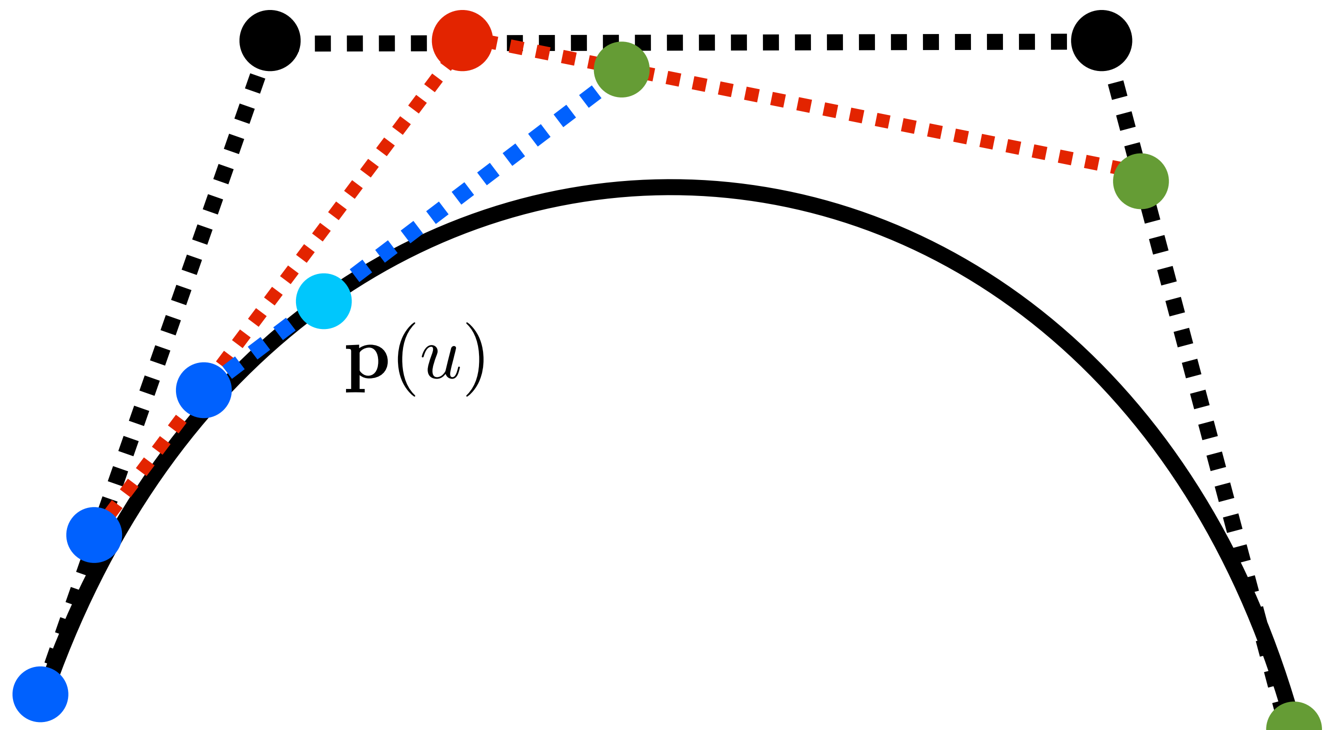


Geometric Construction



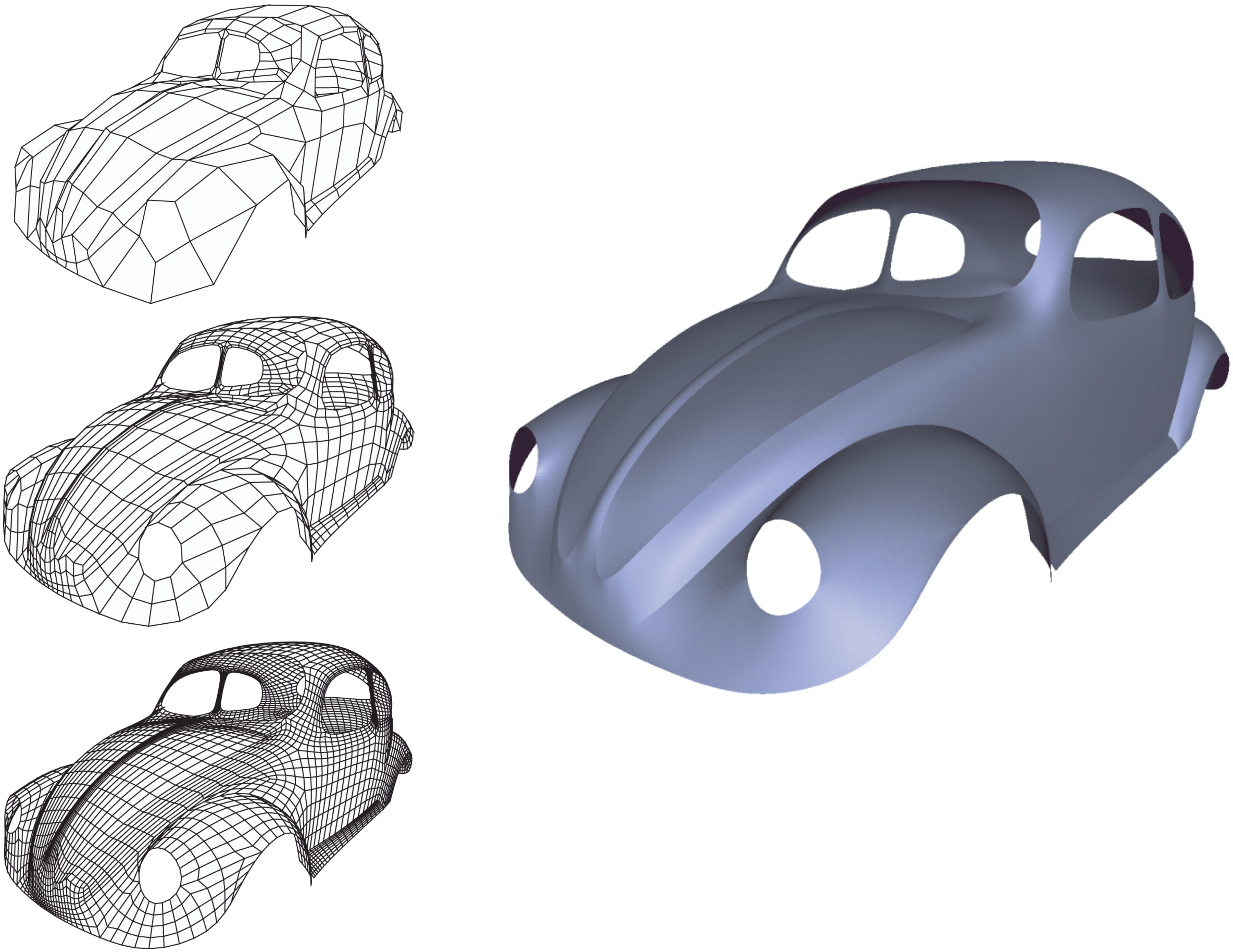
Bezier subdivision





de Casteljau algorithm
Left: Subdivide the curve at the point u=.5

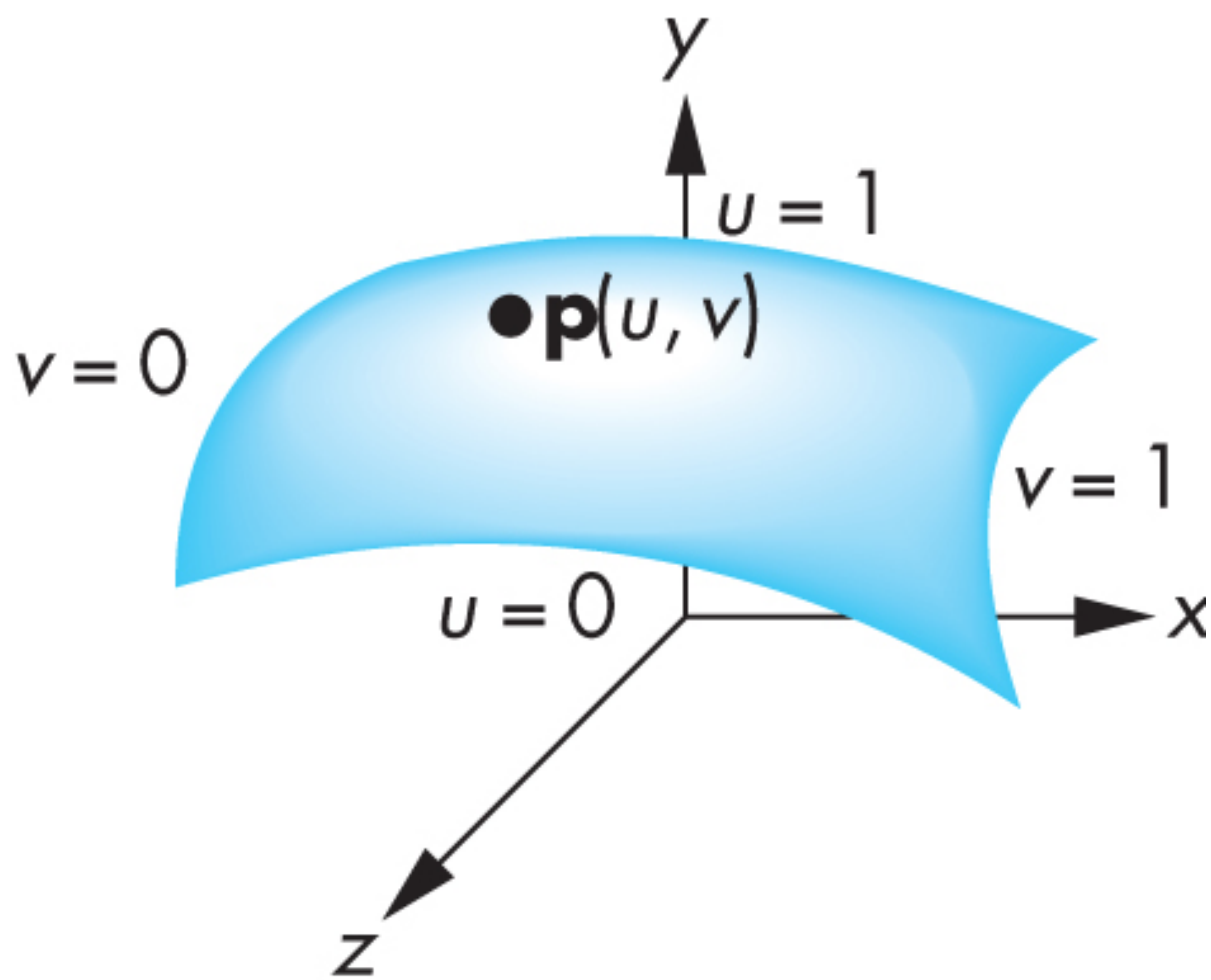
Recursive Subdivision for Rendering



Surfaces

Parametric Surface

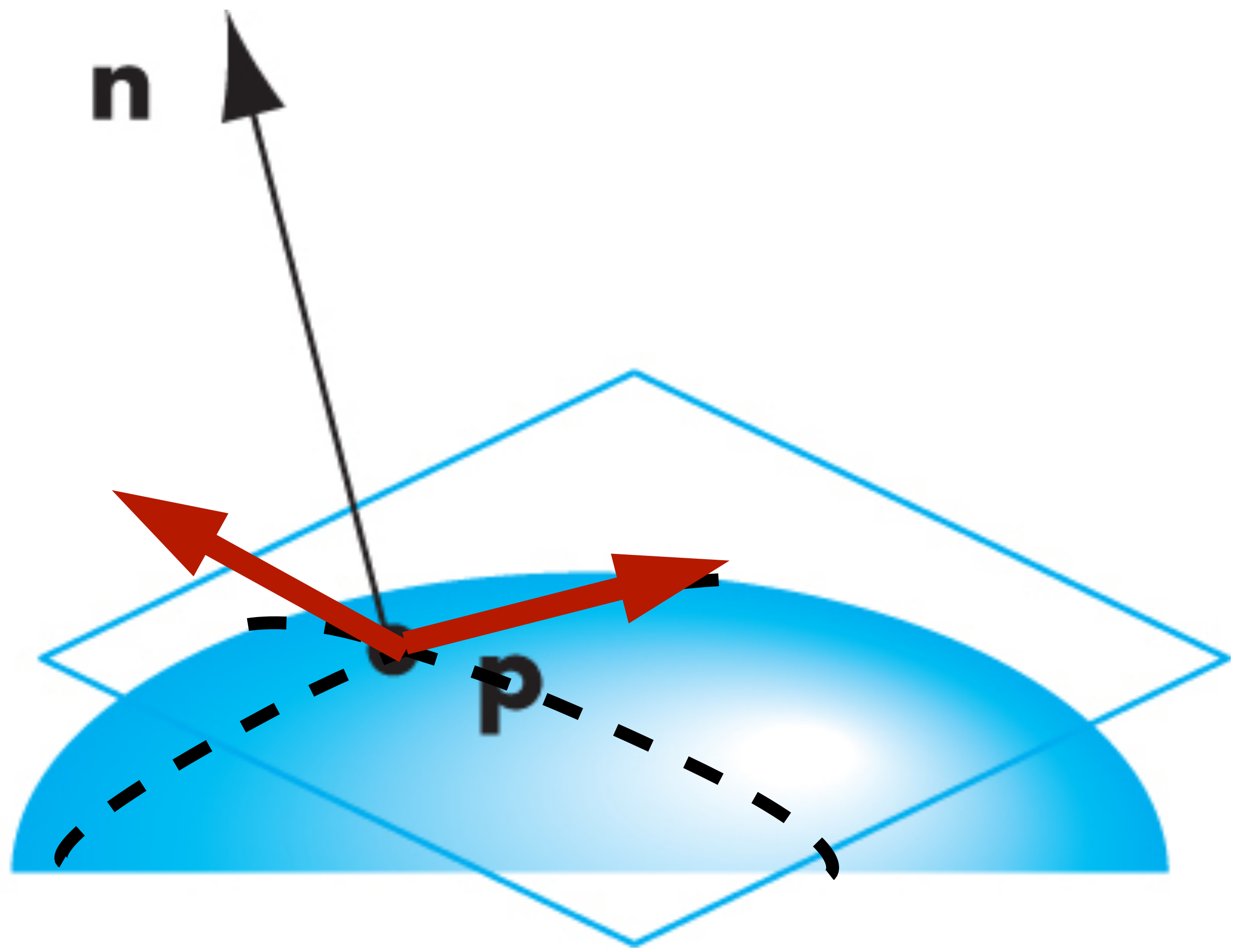
$$x = x(u, v)$$
$$y = y(u, v)$$
$$z = z(u, v)$$



Parametric Surface - tangent plane

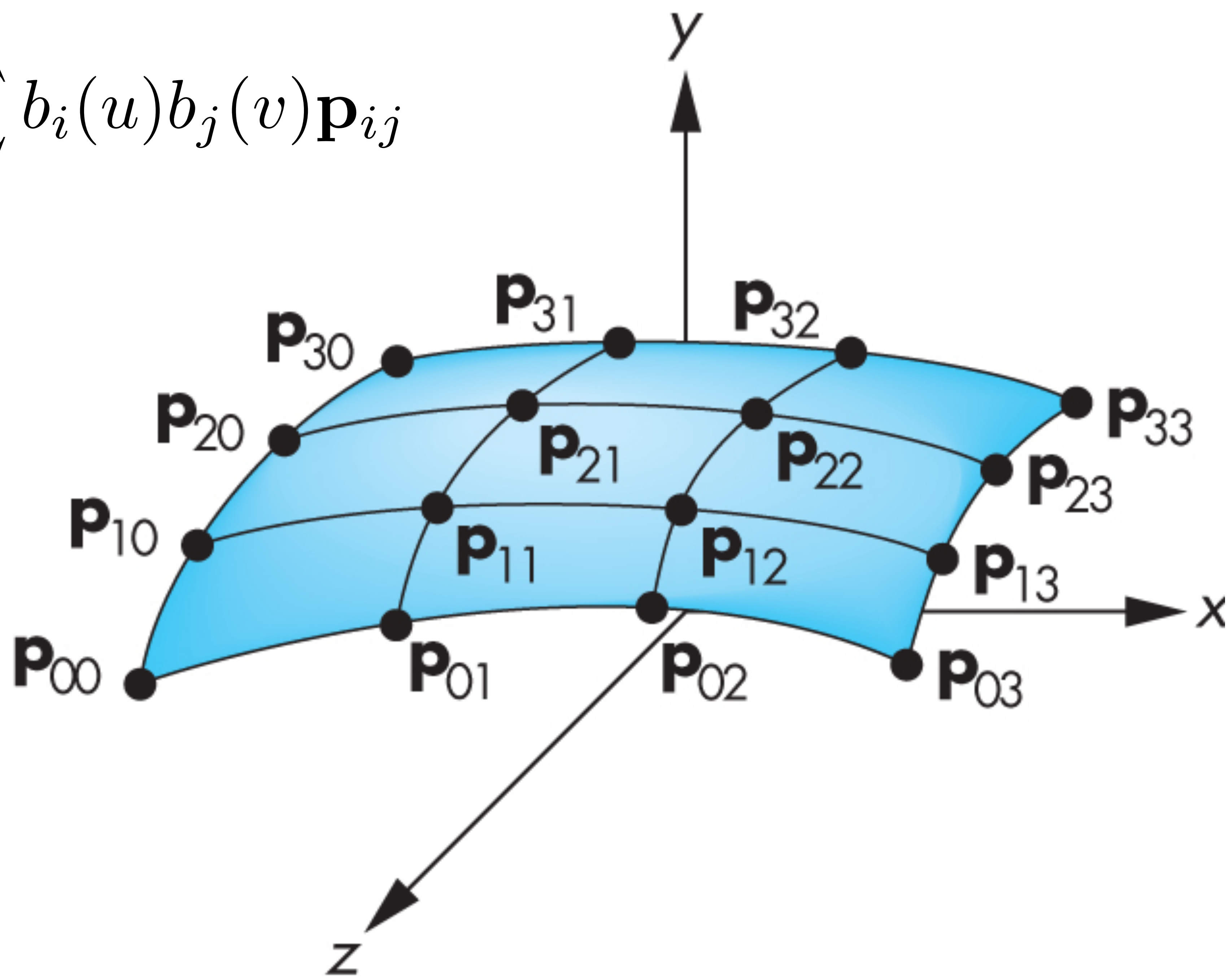
$$\mathbf{t}_u = \begin{pmatrix} \frac{\partial x}{\partial u} \\ \frac{\partial y}{\partial u} \\ \frac{\partial z}{\partial u} \end{pmatrix}$$

$$\mathbf{t}_v = \begin{pmatrix} \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial v} \end{pmatrix}$$



Bicubic Surface Patch

$$\mathbf{f}(u,v) = \sum_i \sum_j b_i(u) b_j(v) \mathbf{p}_{ij}$$



Bezier Surface Patch

$$\mathbf{f}(u, v) = \sum_i \sum_j b_i(u) b_j(v) \mathbf{p}_{ij}$$

Patch lies in convex hull