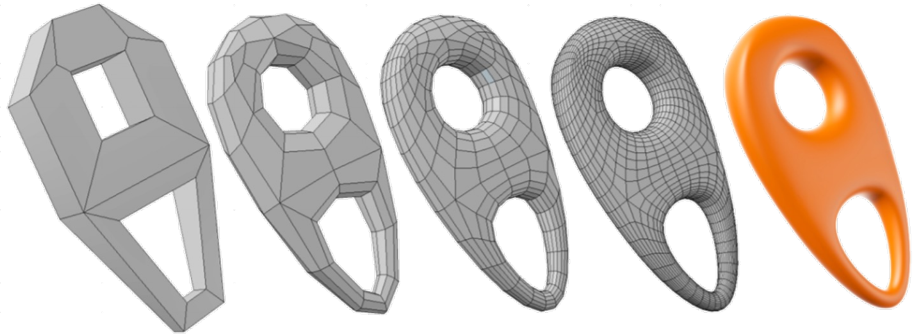
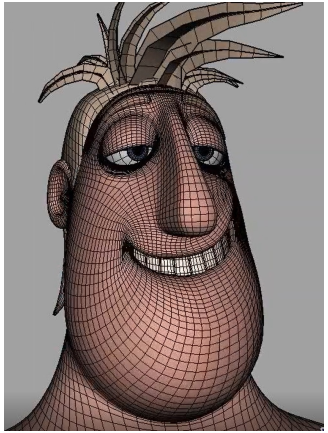
Lecture 9:

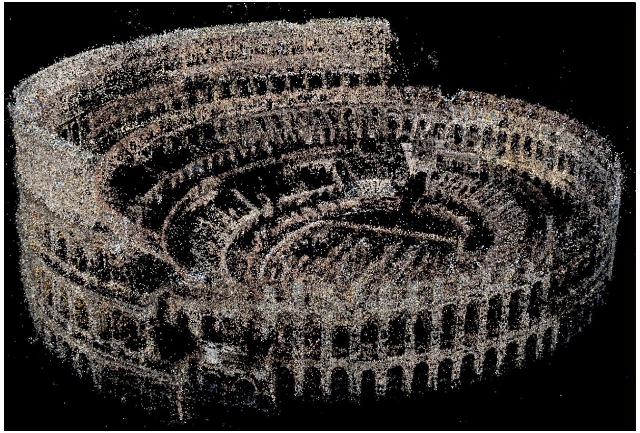
Geometry —— Curves and Surfaces

Review: Many Explicit Representations in Graphics

- Triangle/Quad meshes
- Bezier surfaces
- Subdivision surfaces
- NURBS
- Point cloud
- …



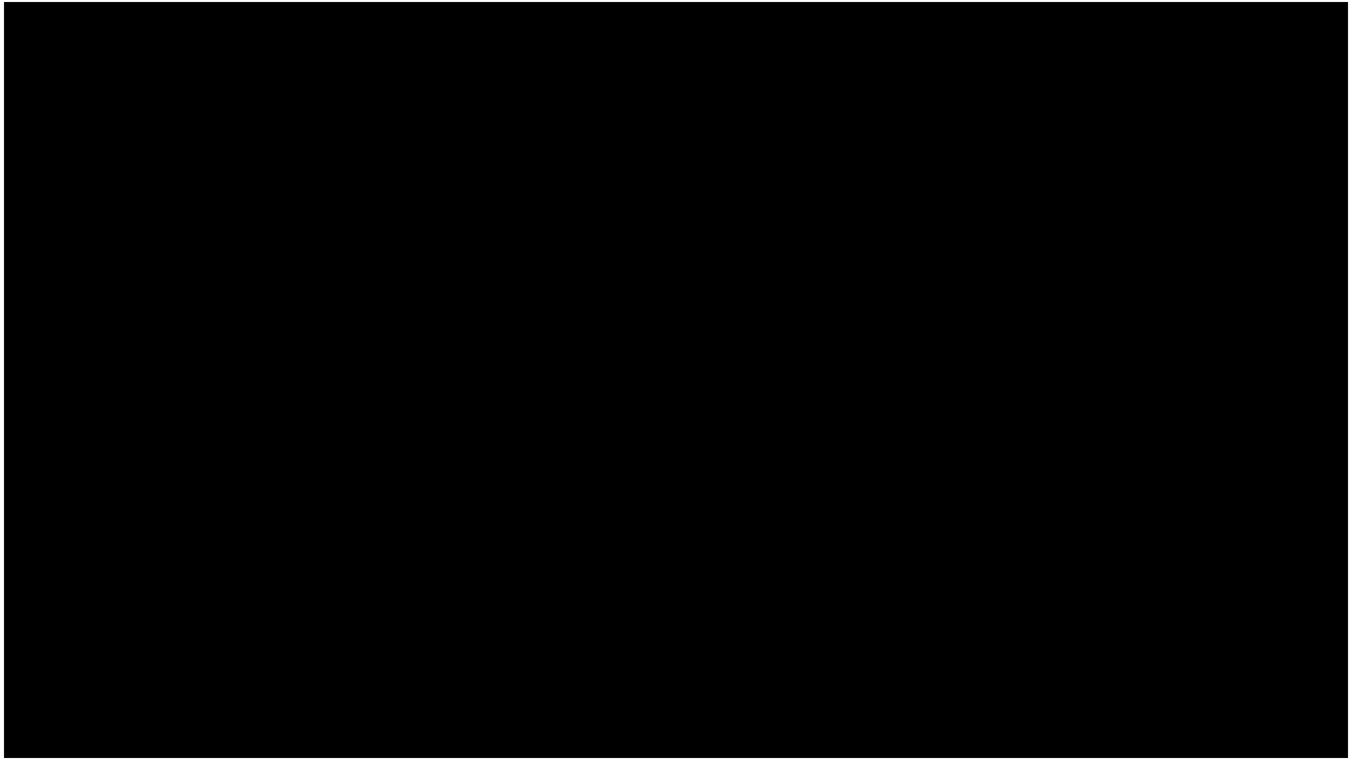




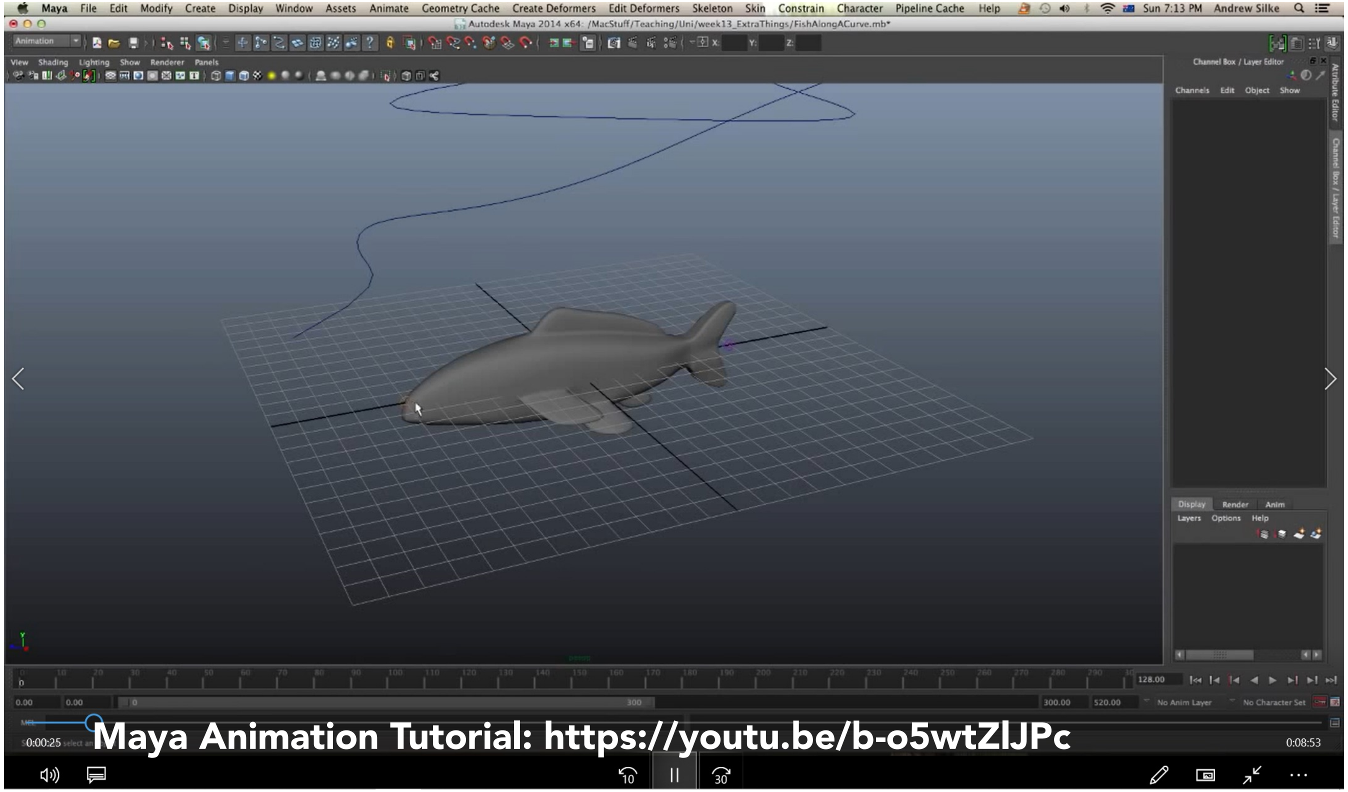
Topic

- Curves and Surfaces
  - Bezier curves
  - De Casteljau’s algorithm
  - Piecewise Bezier Curves
  - Bezier surfaces

Camera Paths

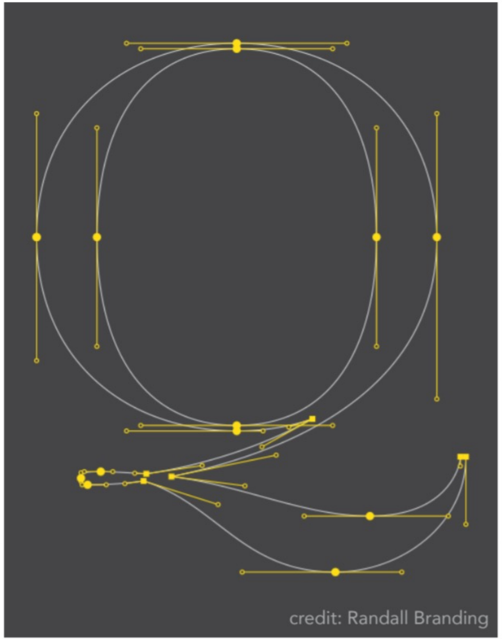


Animation Curves



Maya Animation Tutorial: https://youtu.be/b-o5wtZlJPc

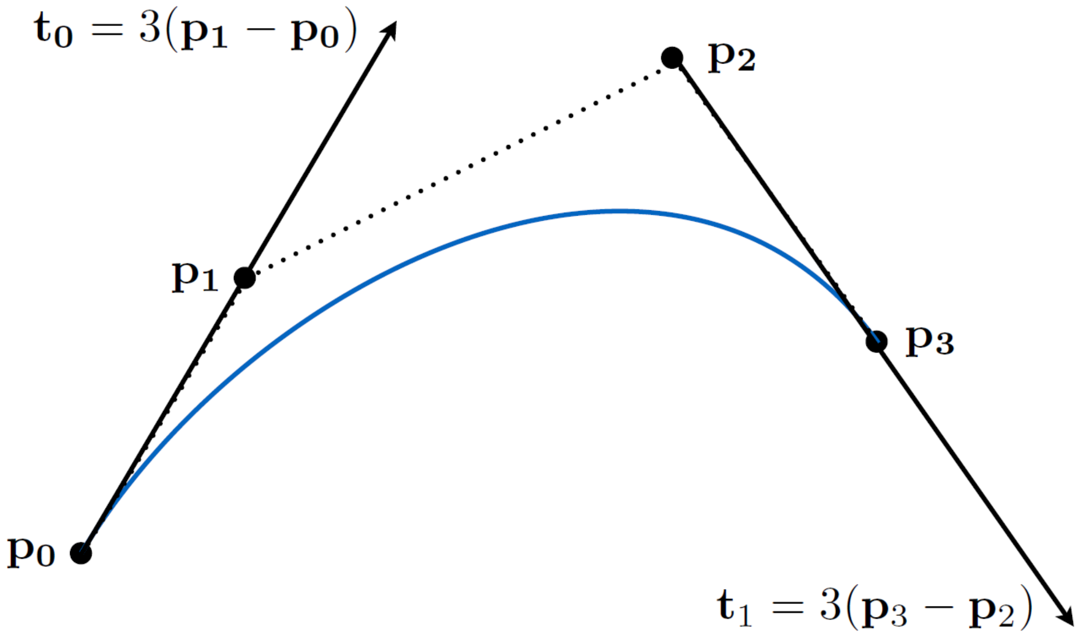
Vector Fonts



Baskerville font - represented as piecewise cubic Bézier curves

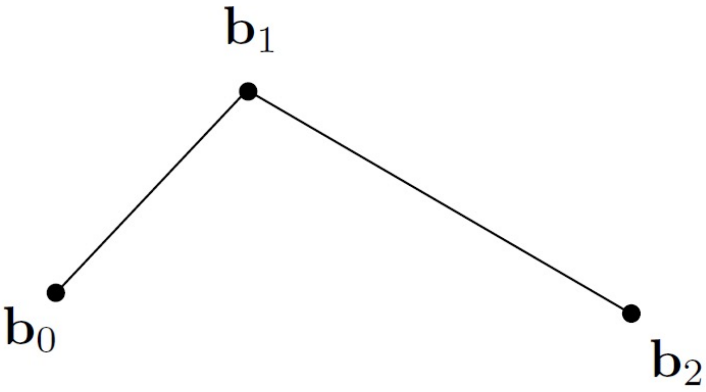
Bézier Curves
(贝塞尔曲线)

Defining Cubic Bézier Curve With Tangents



Bézier Curves – de Casteljau Algorithm

Consider three points (quadratic Bezier)

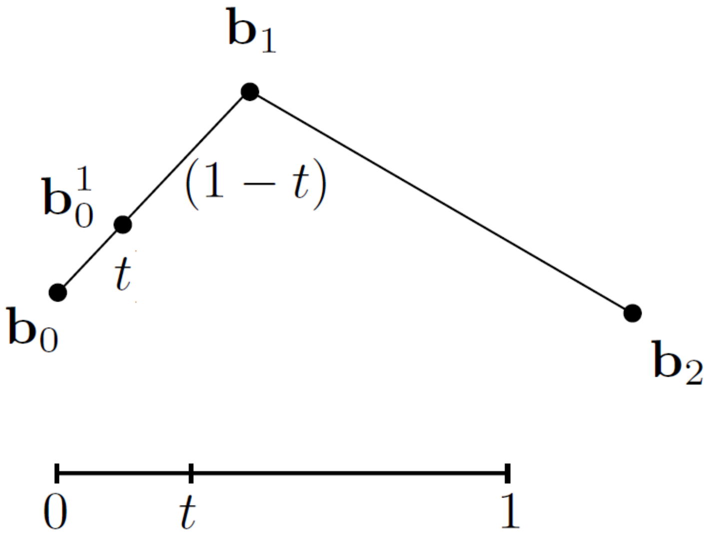


Pierre Bézier
1910 – 1999

Paul de Casteljau
b. 1930

Bézier Curves – de Casteljau Algorithm

Insert a point using linear interpolation

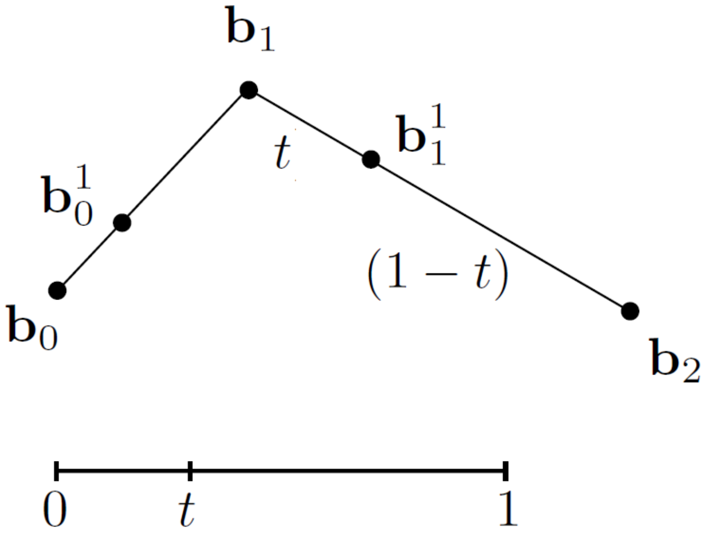


Pierre Bézier
1910 – 1999

Paul de Casteljau
b. 1930

Bézier Curves – de Casteljau Algorithm

Insert on both edges

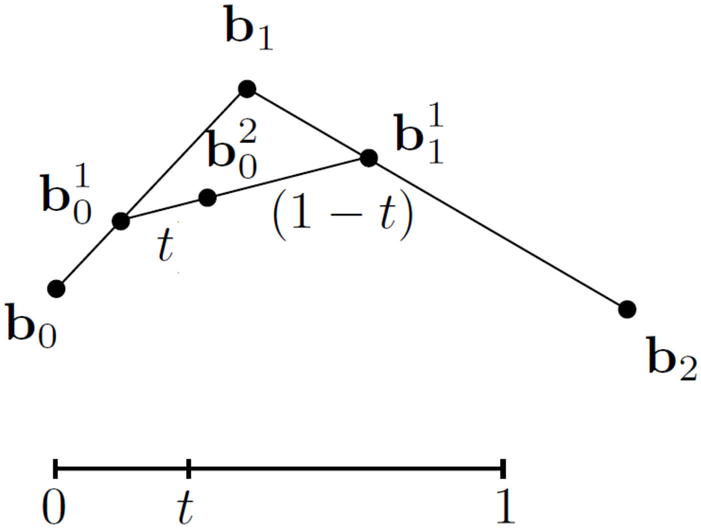


Pierre Bézier
1910 – 1999

Paul de Casteljau
b. 1930

Bézier Curves – de Casteljau Algorithm

Repeat recursively

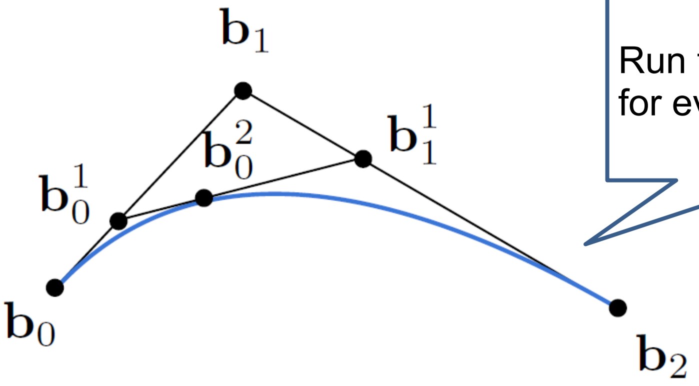


Pierre Bézier
1910 – 1999

Paul de Casteljau
b. 1930

Bézier Curves – de Casteljau Algorithm

Algorithm defines the curve



Run the same algorithm for every t in [0,1]

"Corner cutting" recursive subdivision

Successive linear interpolation

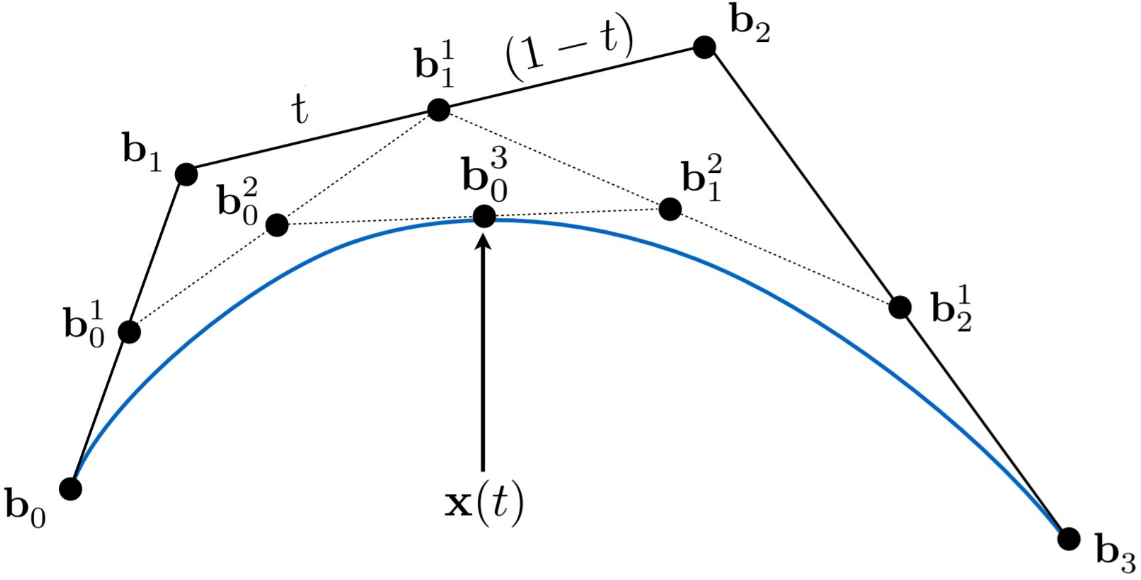
Pierre Bézier
1910 – 1999

Paul de Casteljau
b. 1930

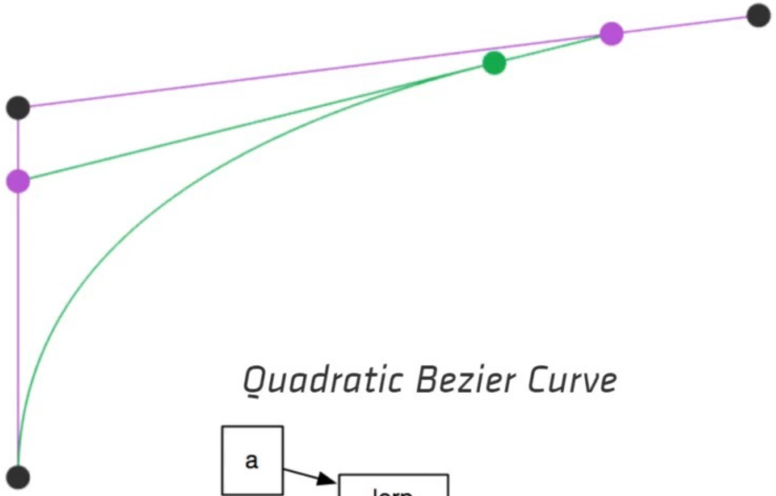
Cubic Bézier Curve – de Casteljau

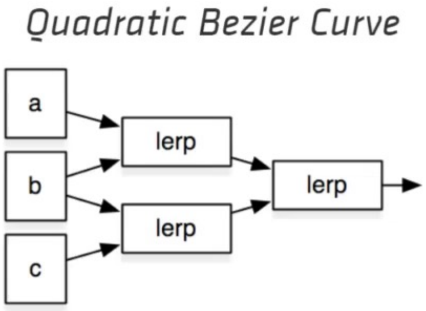
Four input points in total

Same recursive linear interpolations

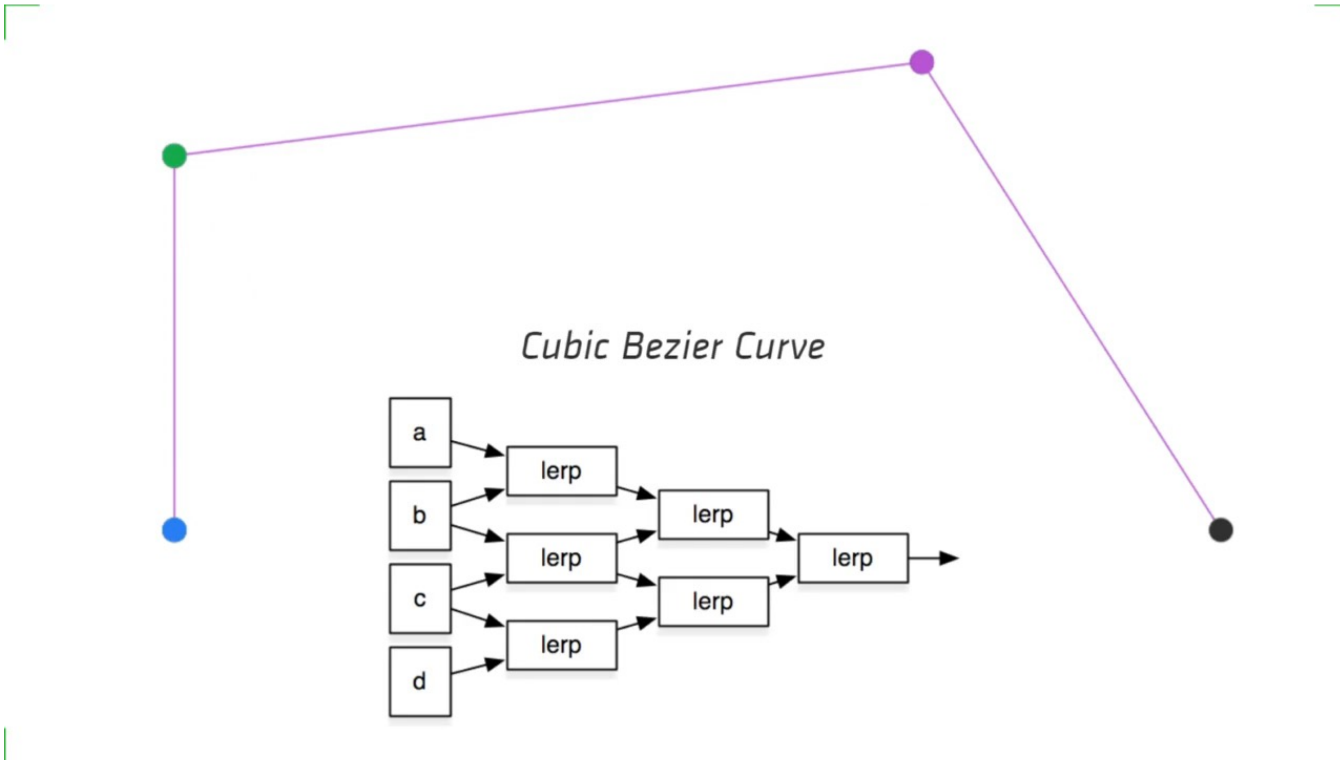


Visualizing de Casteljau Algorithm (Demo)





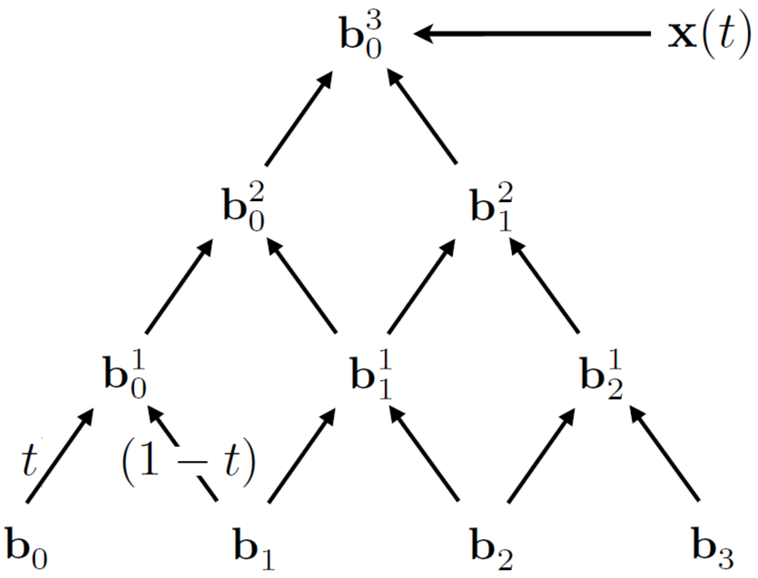
Visualizing de Casteljau Algorithm (Demo)

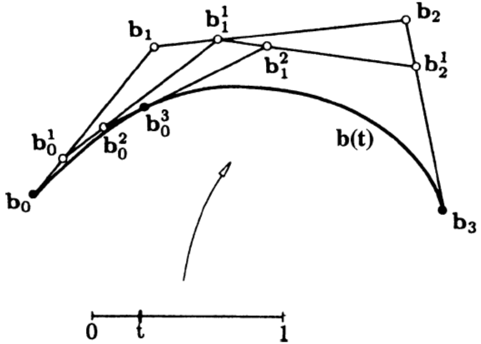


Evaluating Bézier Curves
Algebraic Formula

Bézier Curve – Algebraic Formula

- de Casteljau algorithm gives a pyramid of coefficients

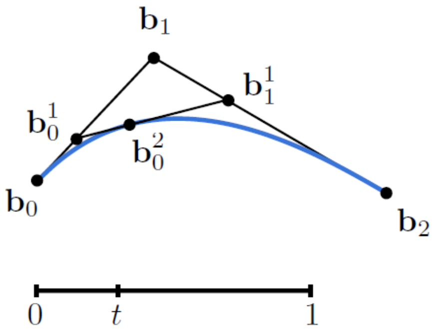




Every rightward arrow is multiplication by t,
Every leftward arrow by (1–t)

Bézier Curve – Algebraic Formula

- Example: quadratic Bézier curve from three points



$$\mathbf{b}_0^1(t) = (1-t)\mathbf{b}_0 + t\mathbf{b}_1$$

$$\mathbf{b}_1^1(t) = (1-t)\mathbf{b}_1 + t\mathbf{b}_2$$

$$\mathbf{b}_0^2(t) = (1-t)\mathbf{b}_0^1 + t\mathbf{b}_1^1$$

$$\mathbf{b}_0^2(t) = (1-t)^2\mathbf{b}_0 + 2t(1-t)\mathbf{b}_1 + t^2\mathbf{b}_2$$

Bézier Curve – Algebraic Formula

Bernstein form of a Bézier curve of order n:

$$\mathbf{b}^n(t) = \mathbf{b}_0^n(t) = \sum_{j=0}^{n} \mathbf{b}_j B_j^n(t)$$

Bézier curve order n
(vector polynomial of degree n)

Bernstein polynomial
(scalar polynomial of degree n)

Bézier control points
(vector in R^N)

Bernstein polynomials:

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

The Binomial Theorem Formula

Reduce the 'a' term and increase the 'b' term each time

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k$$

The sigma sign tells us to add up the terms

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

$$(a+b)^n = \binom{n}{0} a^n b^0 + \binom{n}{1} a^{n-1} b^1 + \binom{n}{2} a^{n-2} b^2 + \cdots + \binom{n}{n} a^0 b^n$$

Binomial Expansions

$$(a+b)^1 = a + b$$

$$(a+b)^2 = a^2 + 2ab + b^2$$

$$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

$$(a+b)^4 = a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$$

$$(a+b)^5 = a^5 + 5a^4b + 10a^3b^2 + 10a^2b^3 + 5ab^4 + b^5$$

Bézier Curve – Algebraic Formula

Bernstein form of a Bézier curve of order n:

$$\mathbf{b}^n(t) = \sum_{j=0}^{n} \mathbf{b}_j B_j^n(t)$$

Example: assume n = 3 and we are in R^3

i.e. we could have control points in 3D such as:

$$\mathbf{b}_0 = (0,2,3),\ \mathbf{b}_1 = (2,3,5),\ \mathbf{b}_2 = (6,7,9),\ \mathbf{b}_3 = (3,4,5)$$

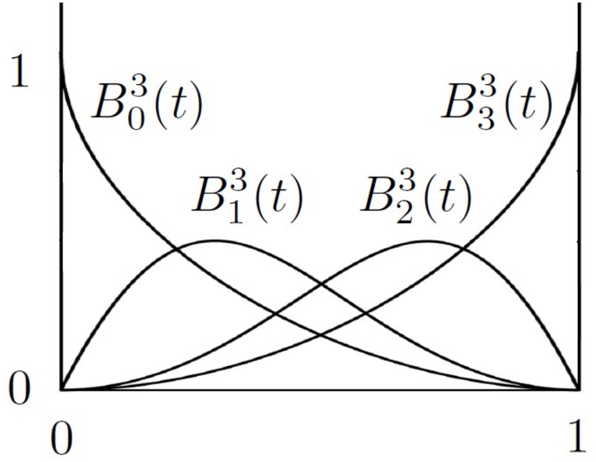
These points define a Bezier curve in 3D that is a cubic polynomial in t:

$$\mathbf{b}^n(t) = \mathbf{b}_0 (1-t)^3 + \mathbf{b}_1 3t(1-t)^2 + \mathbf{b}_2 3t^2(1-t) + \mathbf{b}_3 t^3$$

Cubic Bézier Basis Functions

Bernstein polynomials:

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



Sergei N. Bernstein
1880 – 1968

Properties of Bézier Curves

Interpolates endpoints

- For cubic Bézier: $\mathbf{b}(0) = \mathbf{b}_0; \quad \mathbf{b}(1) = \mathbf{b}_3$

Tangent to end segments

- Cubic case: $\mathbf{b}'(0) = 3(\mathbf{b}_1 - \mathbf{b}_0); \quad \mathbf{b}'(1) = 3(\mathbf{b}_3 - \mathbf{b}_2)$

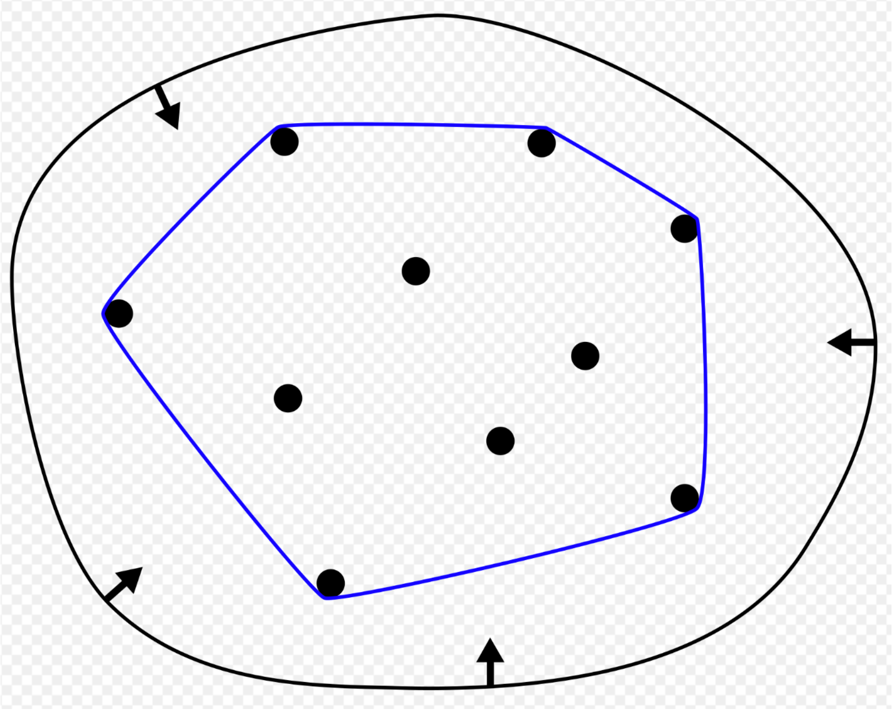
Affine transformation property

- Transform curve by transforming control points

Convex hull property

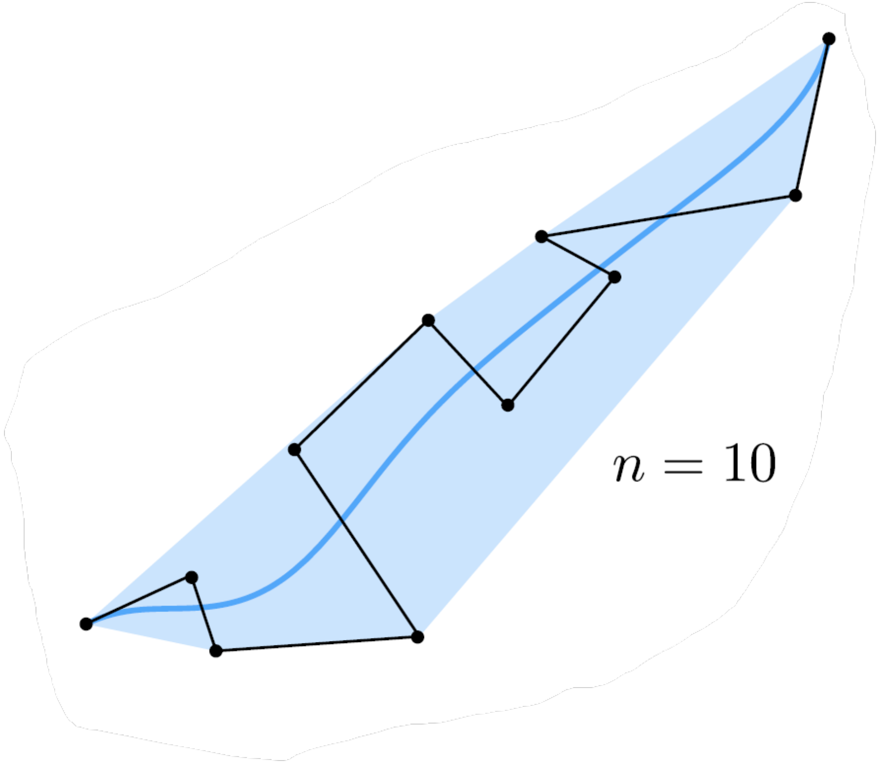
- Curve is within convex hull of control points

Convex hull



Piecewise Bézier Curves

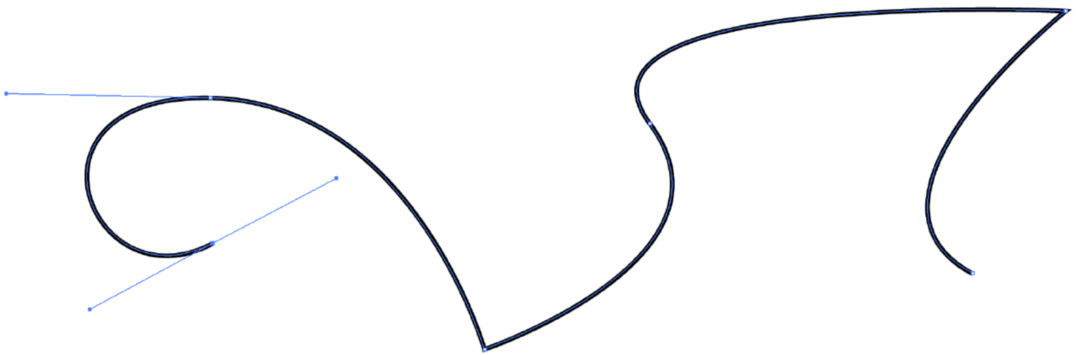
Higher-Order Bézier Curves?



Very hard to control!
Uncommon

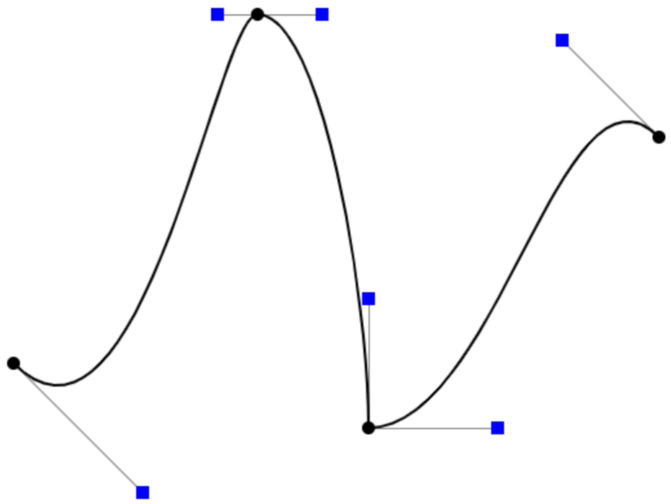
Piecewise Bézier Curves

- Instead, chain many low-order Bézier curve
- **Piecewise cubic Bézier** the most common technique



Widely used (fonts, paths, Illustrator, Keynote, ...)

Piecewise Cubic Bézier Curves (Demo)



I' will show this!

http://math.hws.edu/eck/cs424/notes2013/canvas/bezier.html

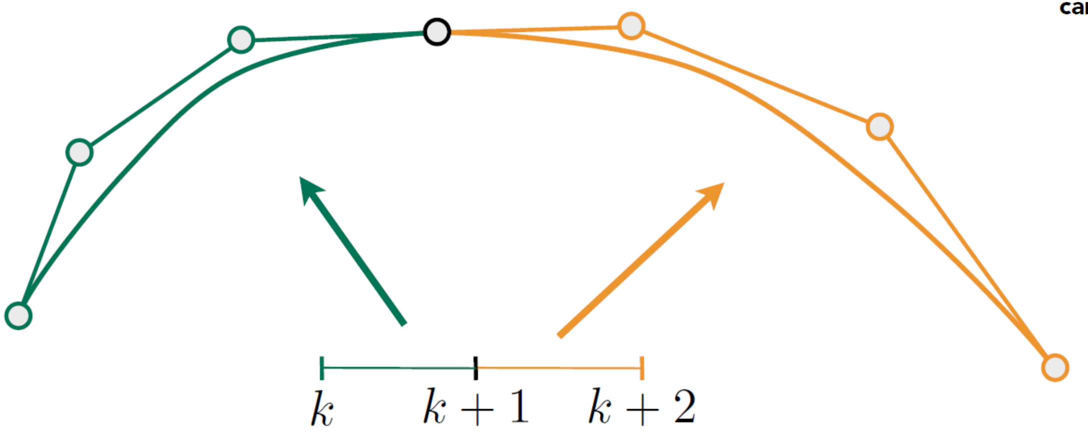
Piecewise Bézier Curves - Continuity

- Two Bézier curves

$$\mathbf{a} : [k, k+1] \to \mathbb{R}^N$$

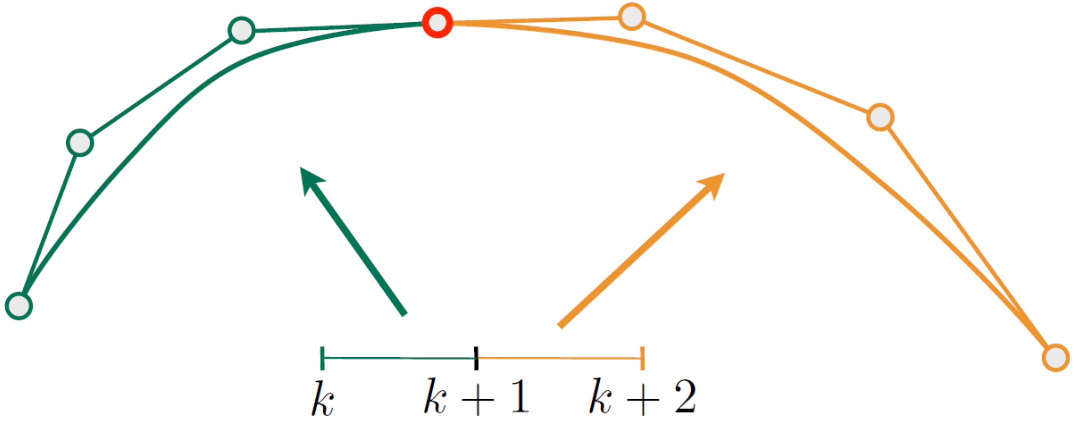
$$\mathbf{b} : [k+1, k+2] \to \mathbb{R}^N$$

Assuming integer partitions here, can generalize



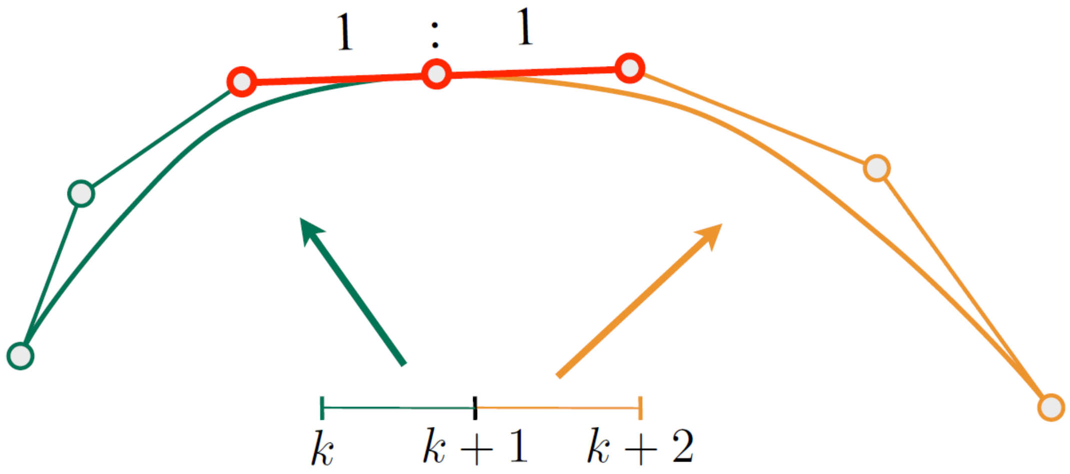
Piecewise Bézier Curves - Continuity

C^0 continuity: $\mathbf{a}_n = \mathbf{b}_0$



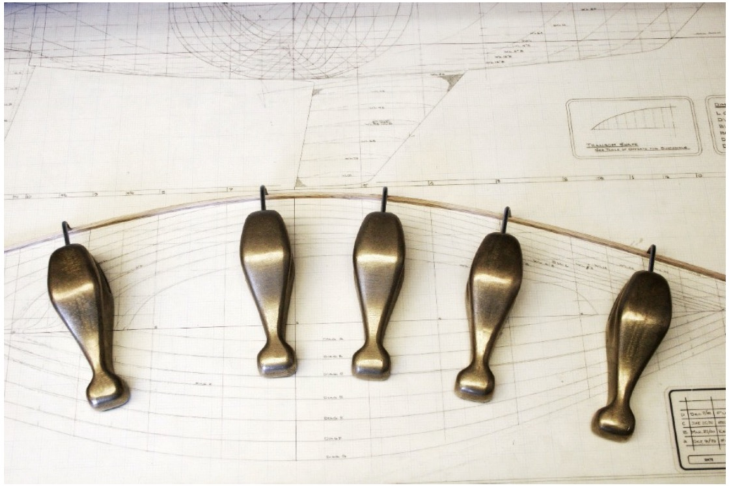
Piecewise Bézier Curves - Continuity

C¹ continuity: $$\mathbf{a}_n = \mathbf{b}_0 = \frac{1}{2}\left(\mathbf{a}_{n-1} + \mathbf{b}_1\right)$$



Other types of splines

- Spline
  - a continuous curve constructed so as to pass through a given set of points and have a certain number of continuous derivatives.
  - In short, a curve under control



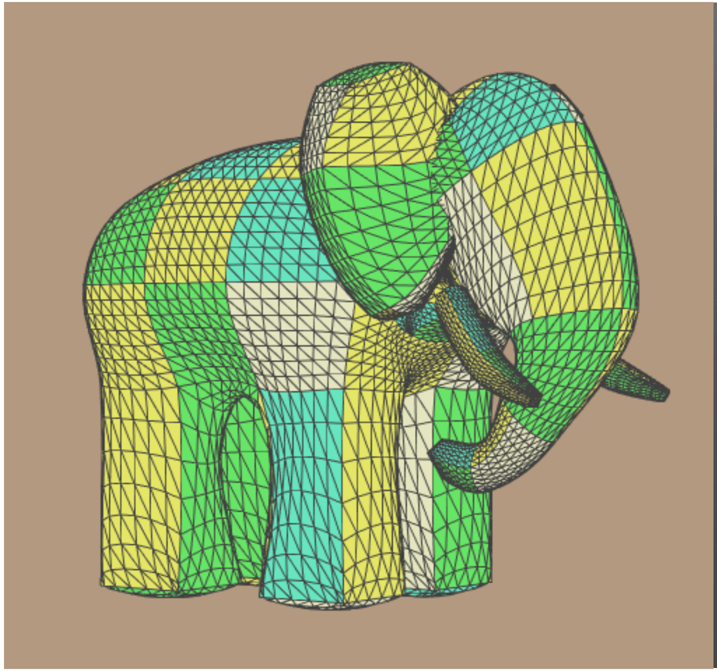
A Real Draftsman's Spline
http://www.alatown.com/spline-history-architecture/

Other types of splines

- B-splines
  - Short for basis splines
  - Require more information than Bezier curves
  - Satisfy all important properties that Bézier curves have (i.e. superset)

Bézier Surfaces

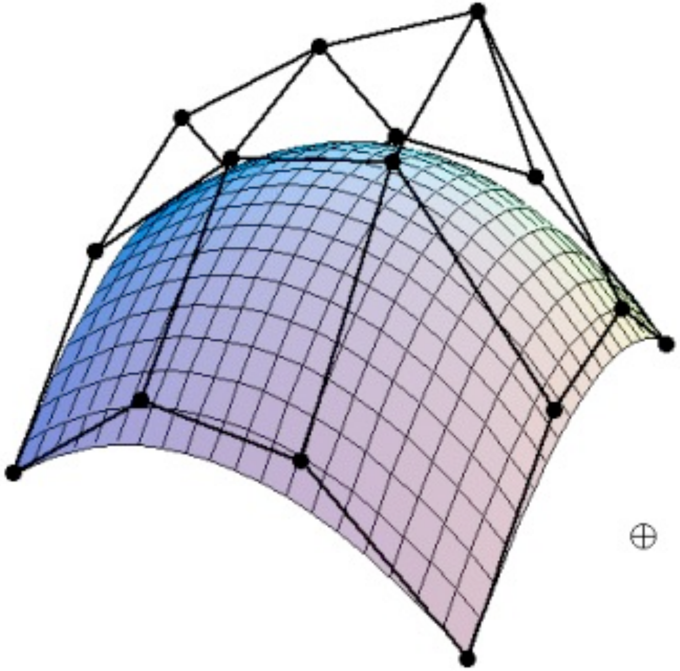
Bézier Surfaces



Ed Catmull's "Gumbo" model

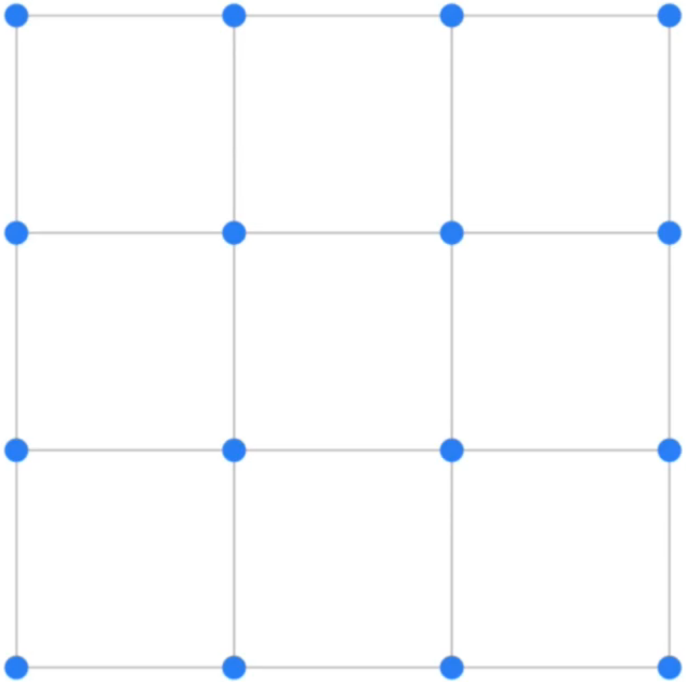
Utah Teapot

Bicubic Bézier Surface Patch



Bezier surface and 4 x 4 array of control points

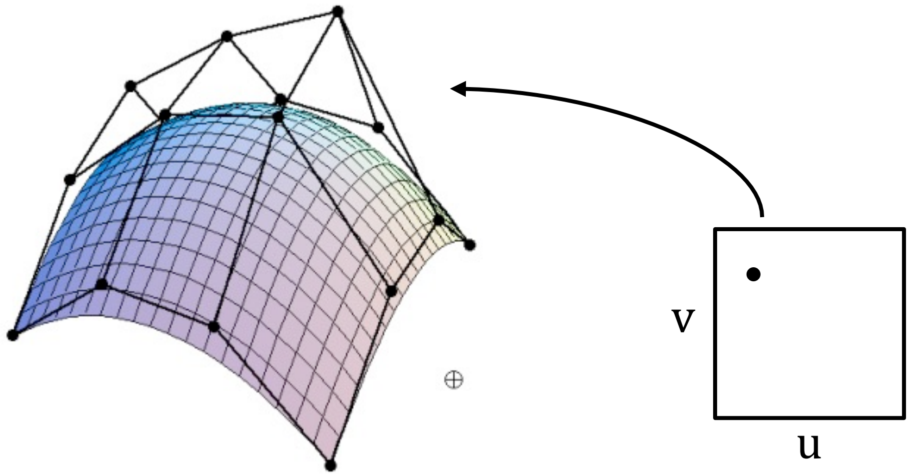
Visualizing Bicubic Bézier Surface Patch (Demo)



https://acko.net/files/fullfrontal/fullfrontal/wdcode/online.html

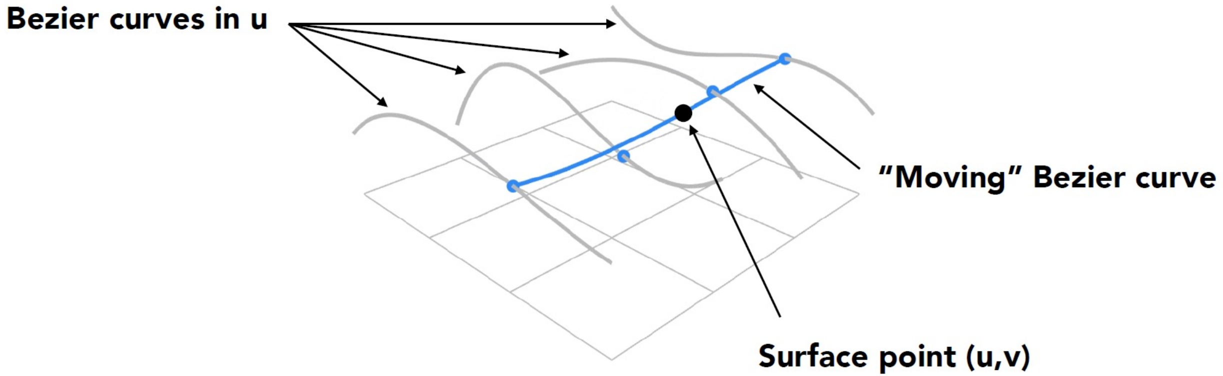
Evaluating Surface Position For Parameters (u,v)

- For bi-cubic Bezier surface patch,
- Input: 4x4 control points
- Output is 2D surface parameterized by (u,v) in [0,1]*[0,1]

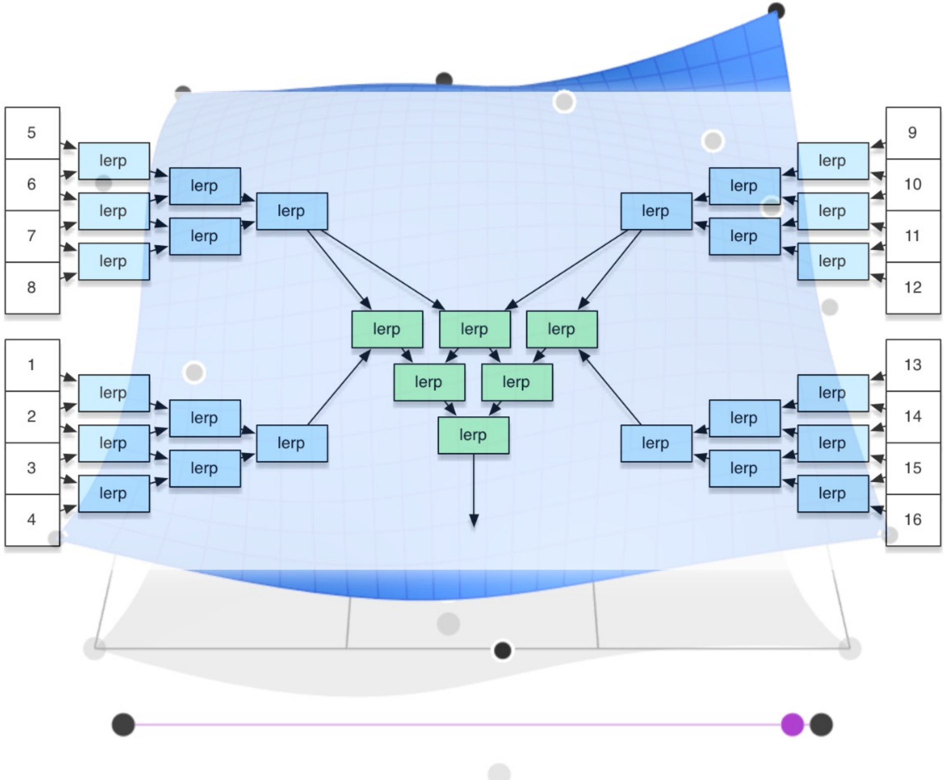


Method: Separable 1D de Casteljau Algorithm

- Goal: Evaluate surface position corresponding to (u,v)
- (u,v)-separable application of de Casteljau algorithm
  - Use de Casteljau to evaluate point u on each of the 4 Bezier curves in u. This gives 4 control points for the "moving" Bezier curve
  - Use 1D de Casteljau to evaluate point v on the "moving" curve



Method: Separable 1D de Casteljau Algorithm



Thank you!