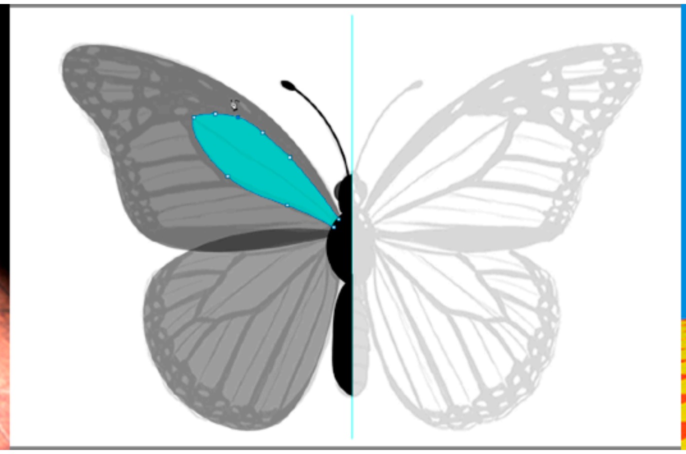
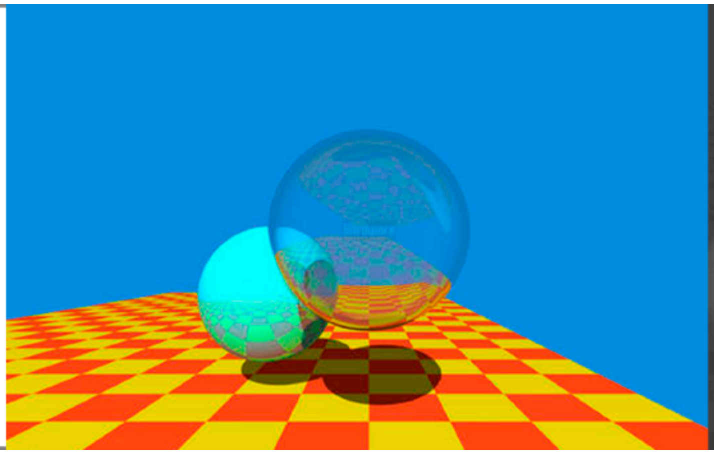
Introduction to Computer Graphics

GAMES101, Lingqi Yan, UC Santa Barbara

Lecture 11:

Geometry 2 (Curves and Surfaces)


http://www.cs.ucsb.edu/~lingqi/teaching/games101.html

Announcements

- Homework 3 deadline has been extended
  - To Thursday 23:59PM, Beijing time
- COVID-19 is getting worse in the US
  - Be careful, dude
  - Have to stream at home, network & lighting are worse

Last Lecture

- Introduction to geometry
  - Examples of geometry
  - Various representations of geometry
    - Implicit
    - Explicit

Today

- Explicit Representations
- Curves
  - Bezier curves
  - De Casteljau's algorithm
  - B-splines, etc.
- Surfaces
  - Bezier surfaces
  - Triangles & quads
    - Subdivision, simplification, regularization

Explicit Representations in Computer Graphics

Many Explicit Representations in Graphics

triangle meshes

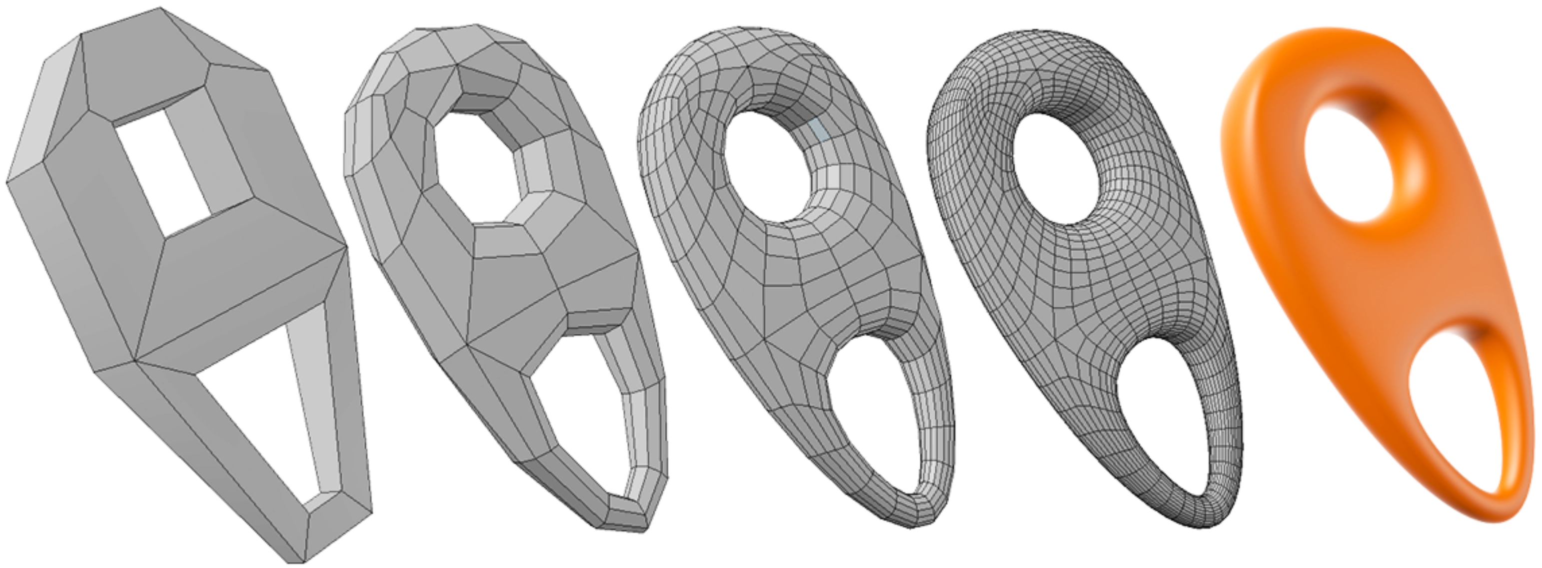
Bezier surfaces

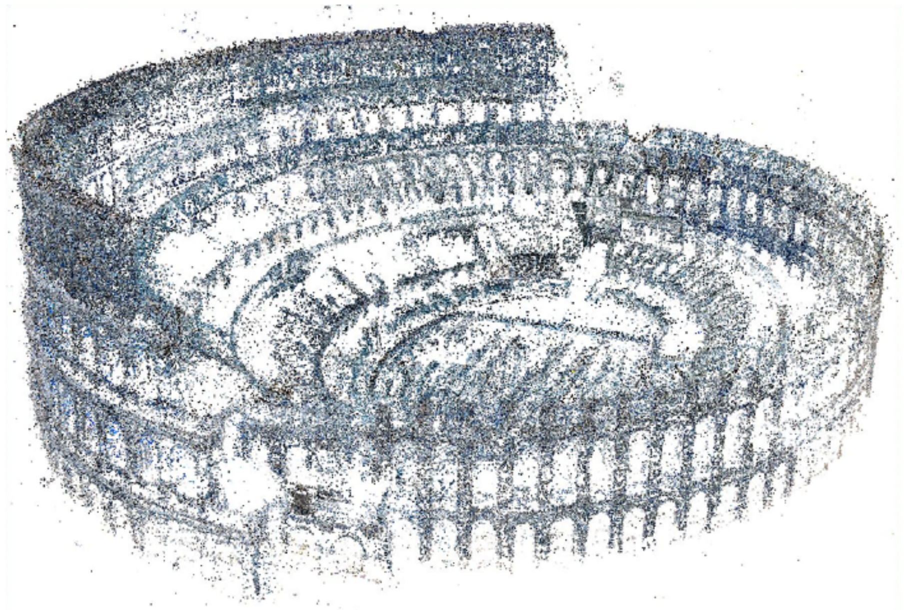
subdivision surfaces

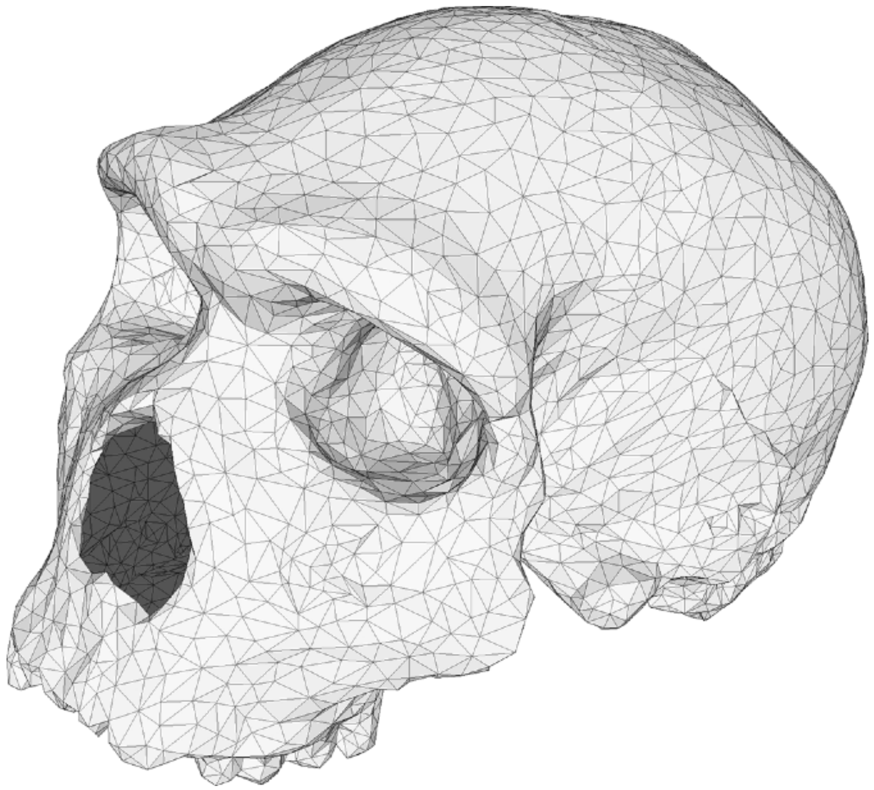
NURBS

point clouds

…







Point Cloud (Explicit)

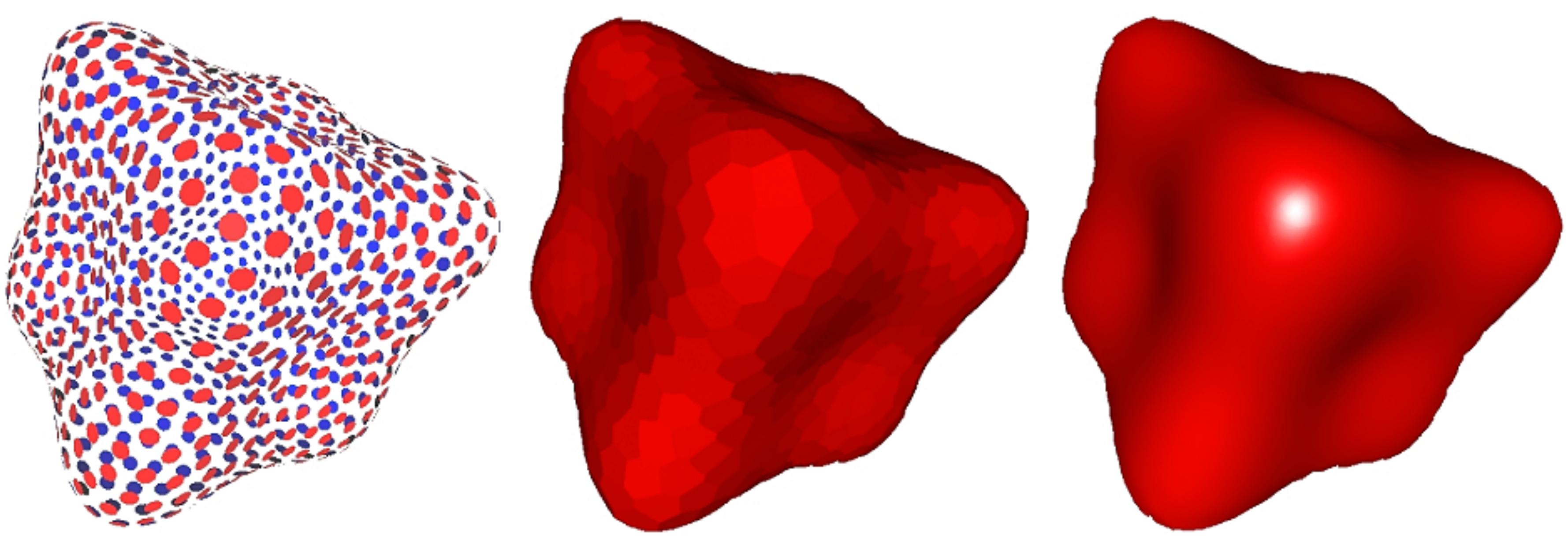
Easiest representation: list of points (x,y,z)

Easily represent any kind of geometry

Useful for LARGE datasets (>>1 point/pixel)

Often converted into polygon mesh

Difficult to draw in undersampled regions



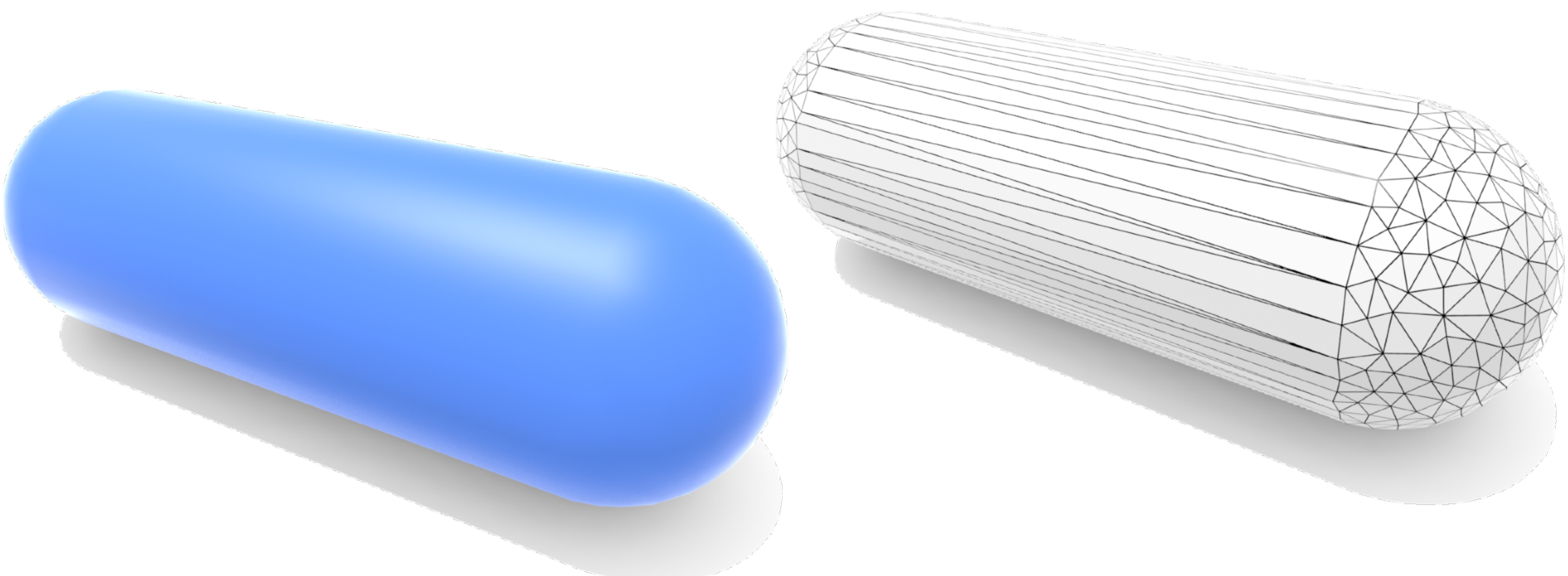
Polygon Mesh (Explicit)

Store vertices & polygons (often triangles or quads)

Easier to do processing / simulation, adaptive sampling

More complicated data structures

Perhaps most common representation in graphics



The Wavefront Object File (.obj) Format

Commonly used in Graphics research

Just a text file that specifies vertices, normals, texture coordinates **and their connectivities**

```
# This is a comment

v 1.000000 -1.000000 -1.000000
v 1.000000 -1.000000 1.000000
v -1.000000 -1.000000 1.000000
v -1.000000 -1.000000 -1.000000
v 1.000000 1.000000 -1.000000
v 0.999999 1.000000 1.000001
v -1.000000 1.000000 1.000000
v -1.000000 1.000000 -1.000000

vt 0.748573 0.750412
vt 0.749279 0.501284
vt 0.999110 0.501077
vt 0.999455 0.750380
vt 0.250471 0.500702
vt 0.249682 0.749677
vt 0.001085 0.750380
vt 0.001517 0.499994
vt 0.499422 0.500239
vt 0.500149 0.750166
vt 0.748355 0.998230
vt 0.500193 0.998728
vt 0.498993 0.250415
vt 0.748953 0.250920
```

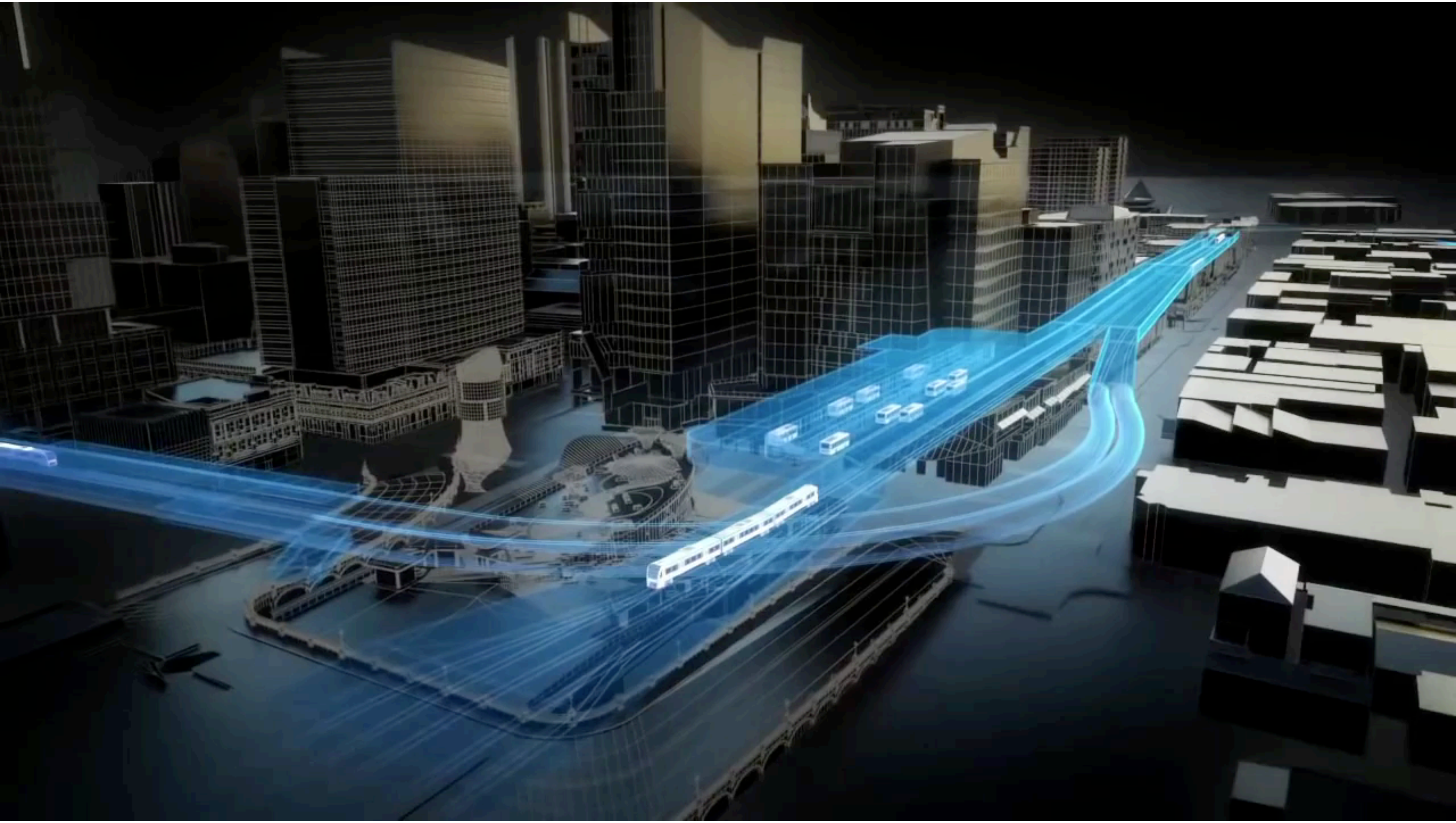
```

vn 0.000000 0.000000 -1.000000
vn -1.000000 -0.000000 -0.000000
vn -0.000000 -0.000000 1.000000
vn -0.000001 0.000000 1.000000
vn 1.000000 -0.000000 0.000000
vn 1.000000 0.000000 0.000001
vn 0.000000 1.000000 -0.000000
vn -0.000000 -1.000000 0.000000

f 5/1/1 1/2/1 4/3/1
f 5/1/1 4/3/1 8/4/1
f 3/5/2 7/6/2 8/7/2
f 3/5/2 8/7/2 4/8/2
f 2/9/3 6/10/3 3/5/3
f 6/10/4 7/6/4 3/5/4
f 1/2/5 5/1/5 2/9/5
f 5/1/6 6/10/6 2/9/6
f 5/1/7 8/11/7 6/10/7
f 8/11/7 7/12/7 6/10/7
f 1/2/8 2/9/8 3/13/8
f 1/2/8 3/13/8 4/14/8
```

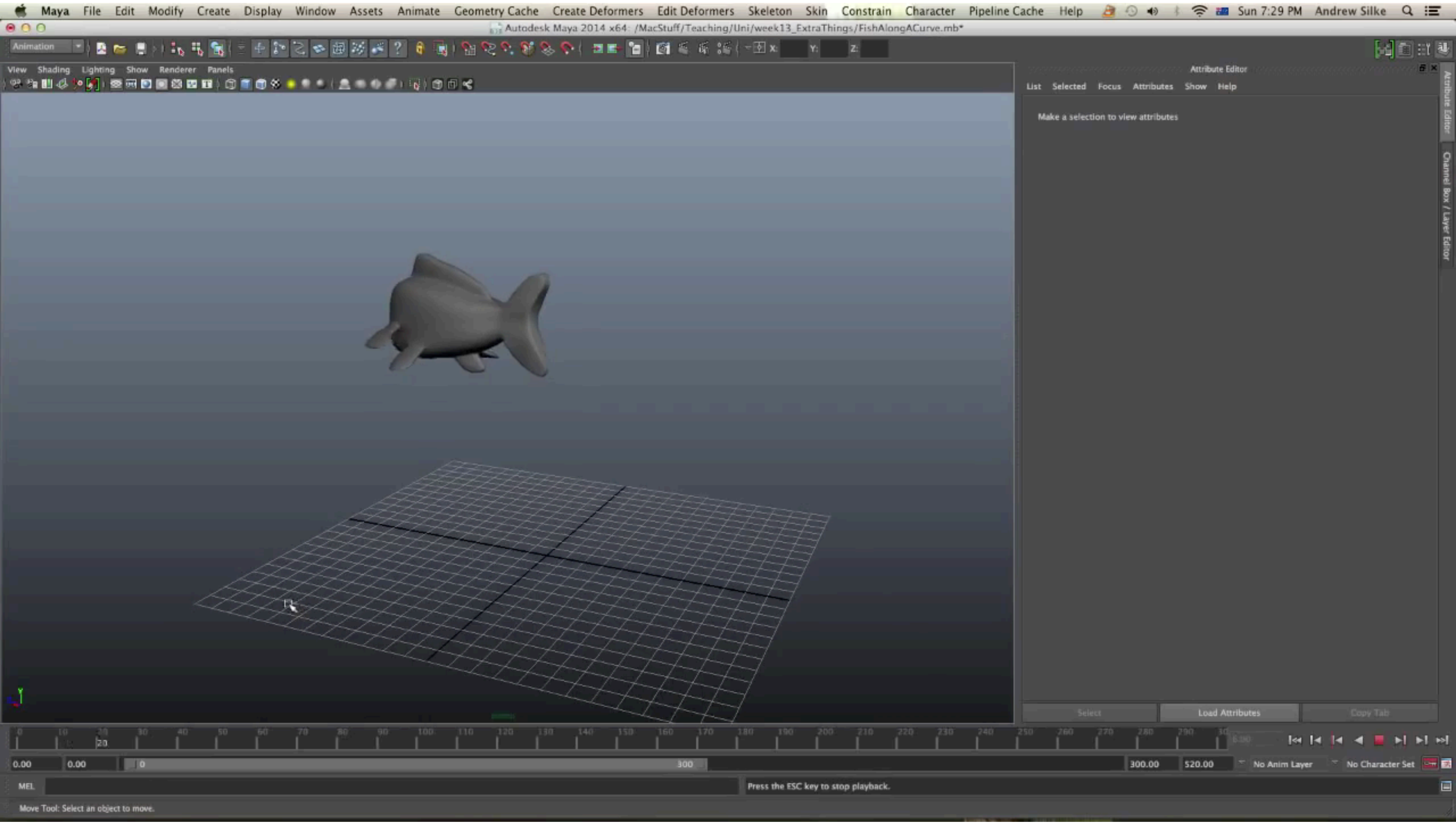
Curves

Camera Paths



Flythrough of proposed Perth Citylink subway, https://youtu.be/rIJMuQPwr3E

Animation Curves

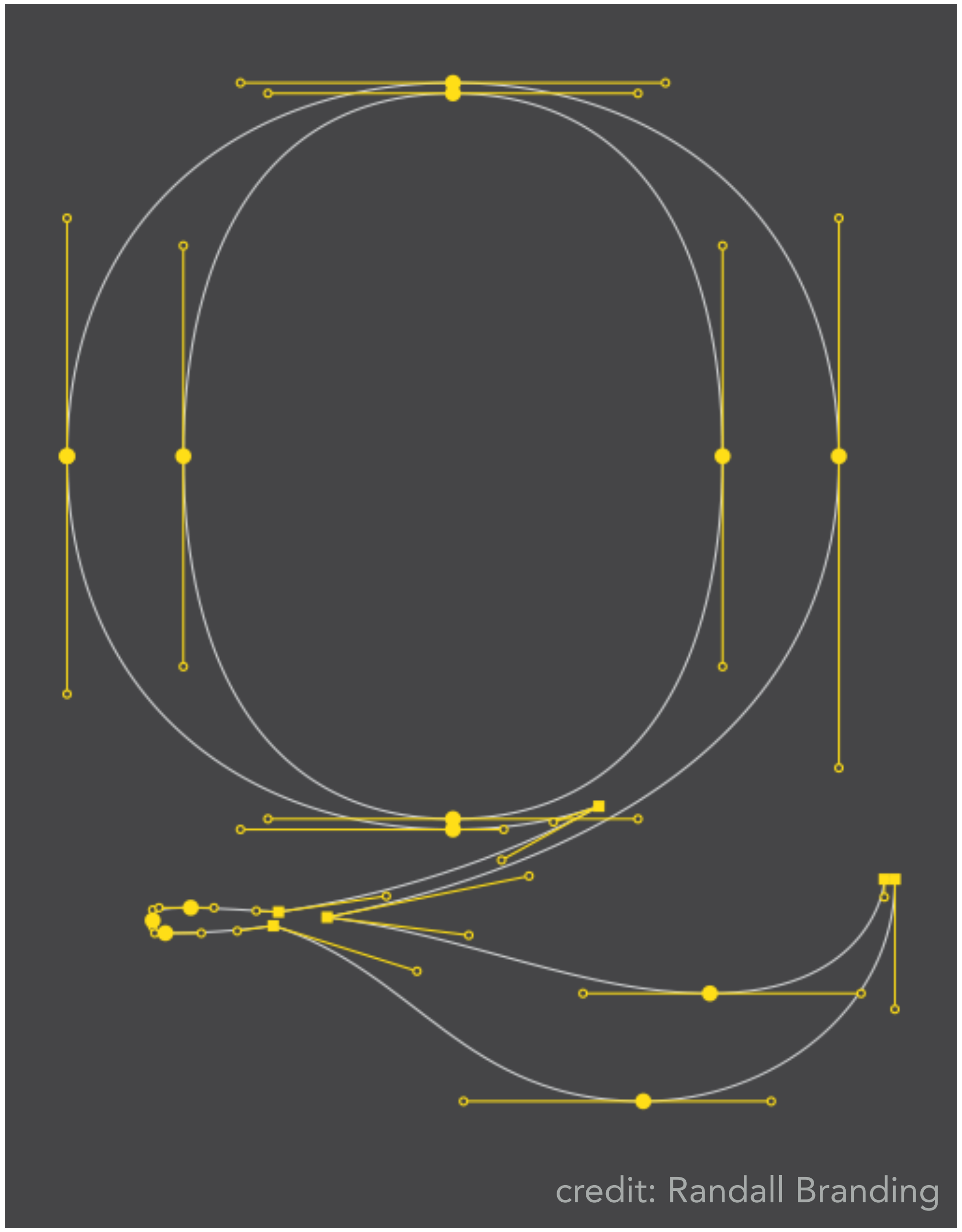


Maya Animation Tutorial: https://youtu.be/b-o5wtZlJPc

Vector Fonts

The Quick Brown Fox Jumps Over The Lazy Dog

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 0123456789

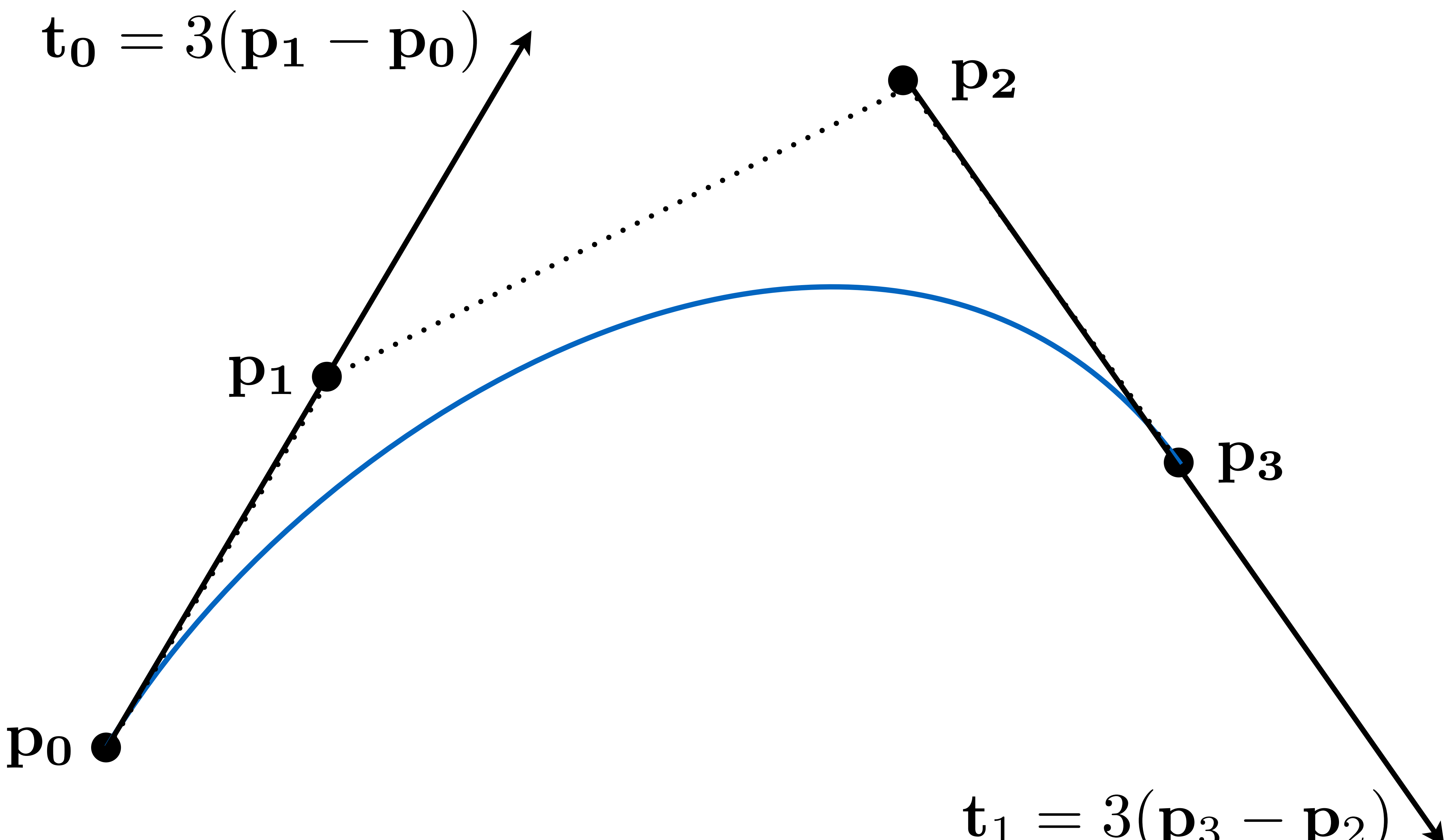


Baskerville font - represented as piecewise cubic Bézier curves

Bézier Curves

(贝塞尔曲线)

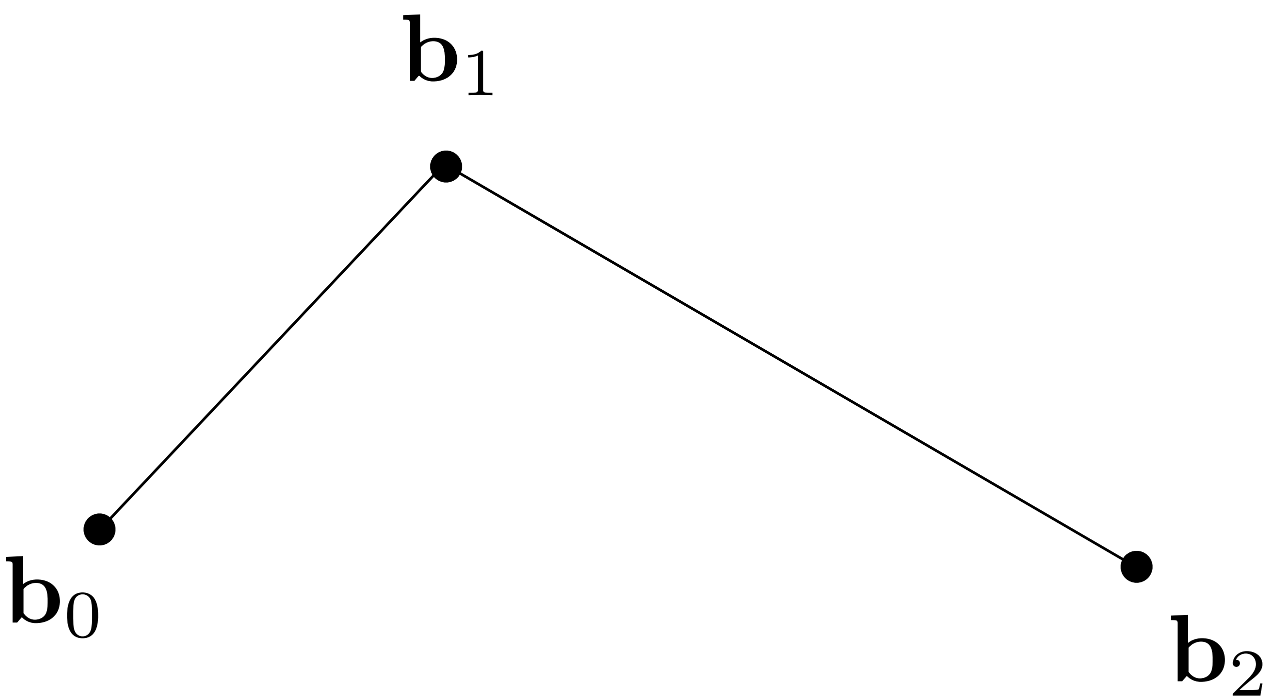
Defining Cubic Bézier Curve With Tangents



Evaluating Bézier Curves
(de Casteljau Algorithm)

Bézier Curves – de Casteljau Algorithm

Consider three points (quadratic Bezier)

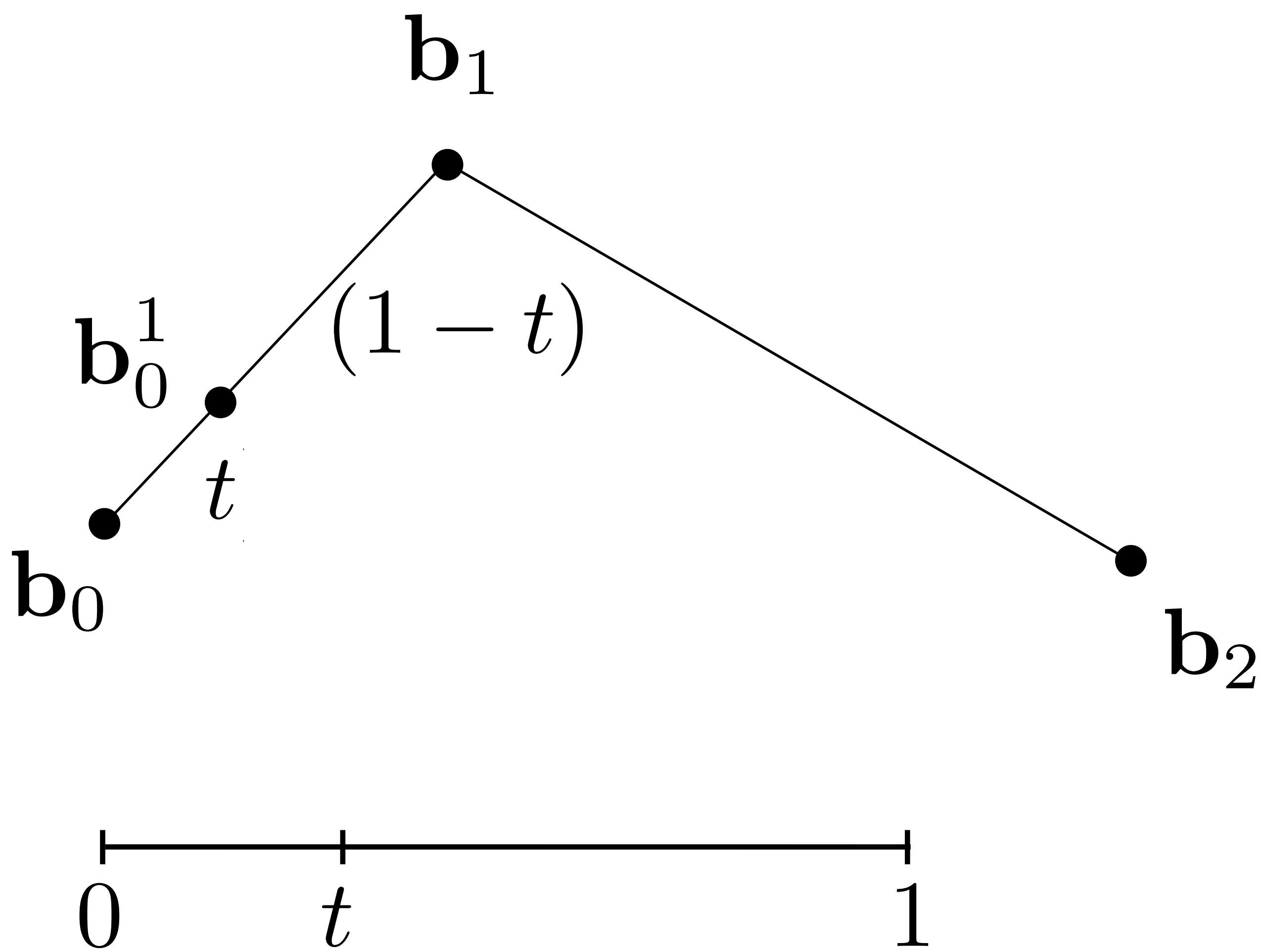


Pierre Bézier
1910 – 1999

Paul de Casteljau
b. 1930

Bézier Curves – de Casteljau Algorithm

Insert a point using linear interpolation

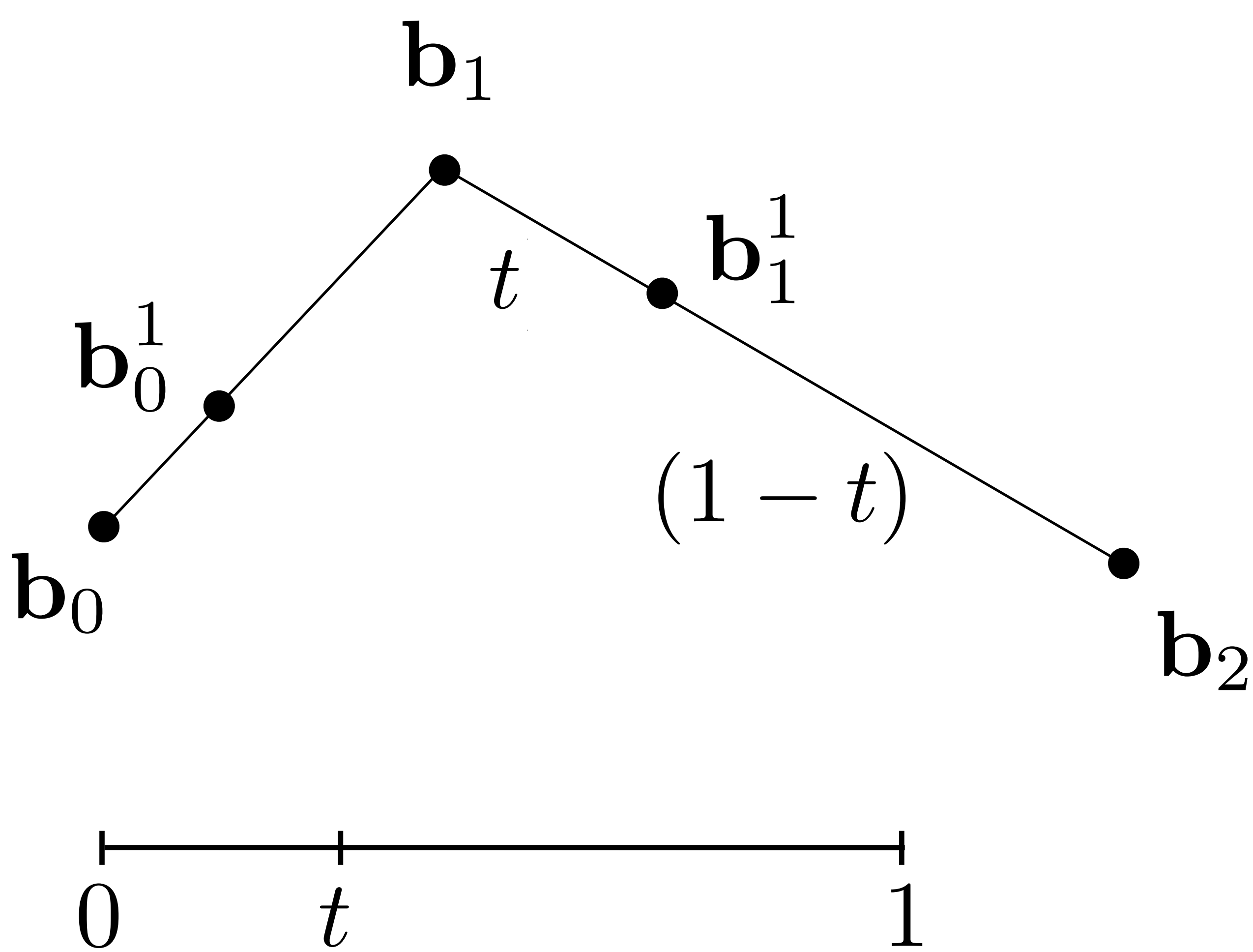


Pierre Bézier
1910 – 1999

Paul de Casteljau
b. 1930

Bézier Curves – de Casteljau Algorithm

Insert on both edges

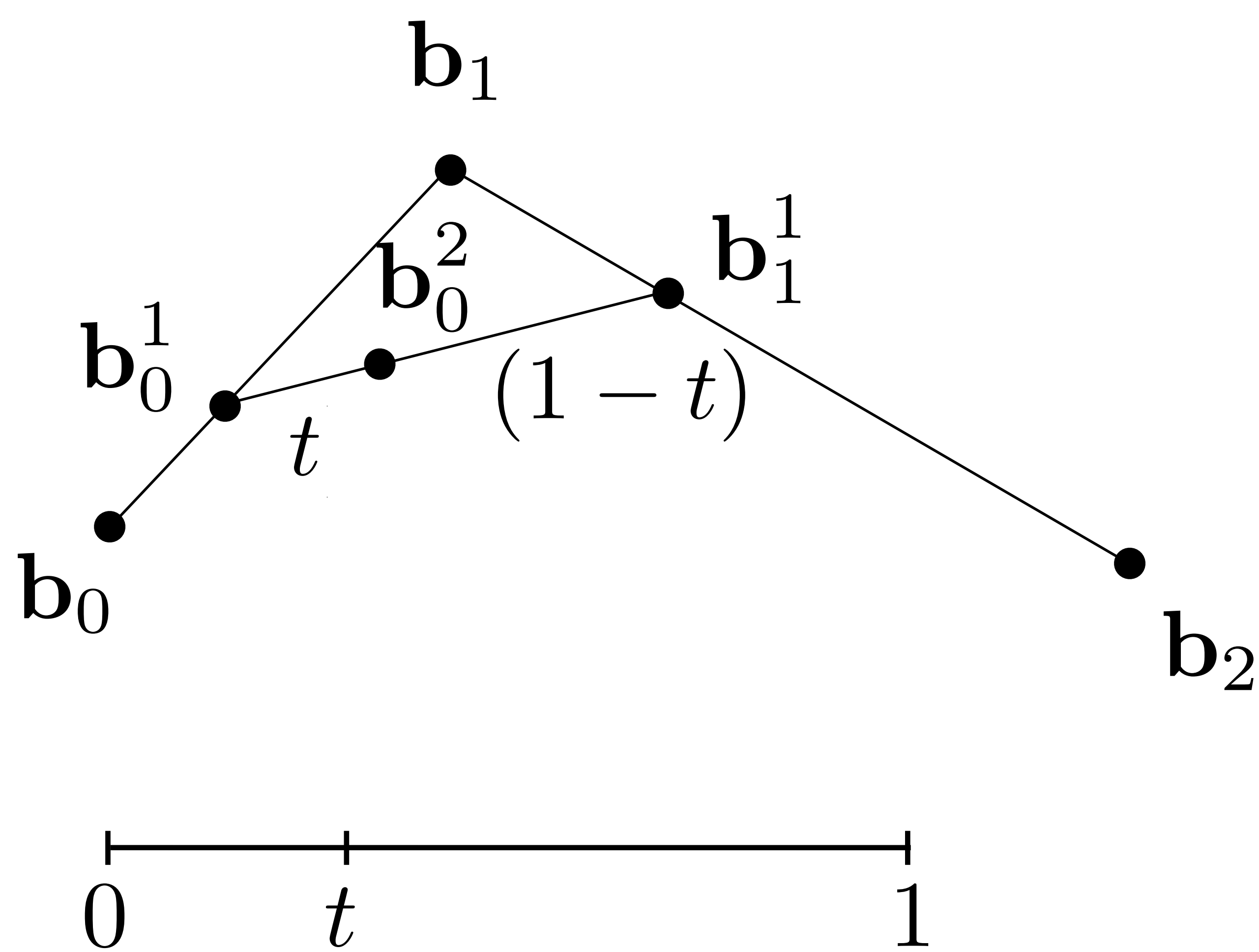


Pierre Bézier
1910 – 1999

Paul de Casteljau
b. 1930

Bézier Curves – de Casteljau Algorithm

Repeat recursively

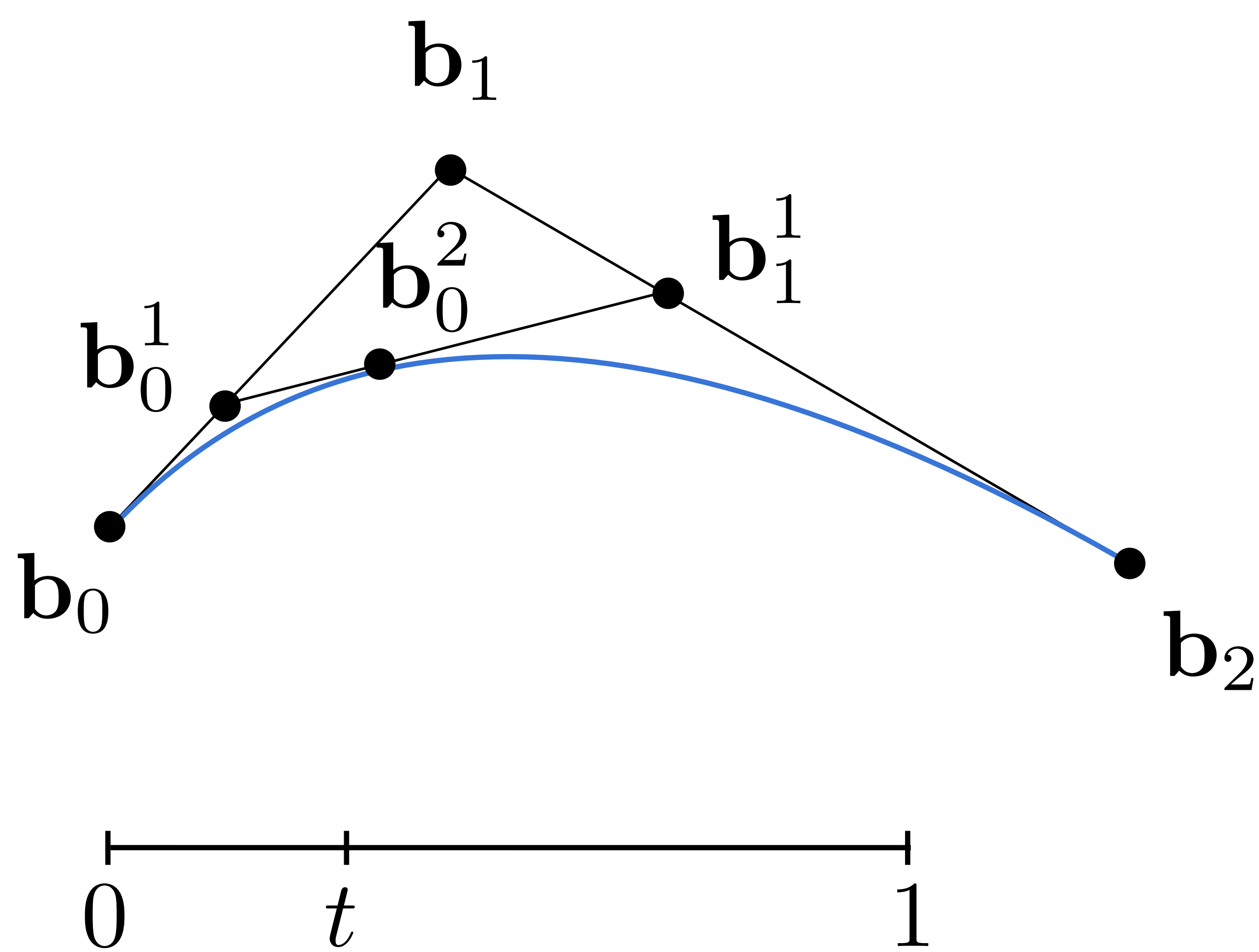


Pierre Bézier
1910 – 1999

Paul de Casteljau
b. 1930

Bézier Curves – de Casteljau Algorithm

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



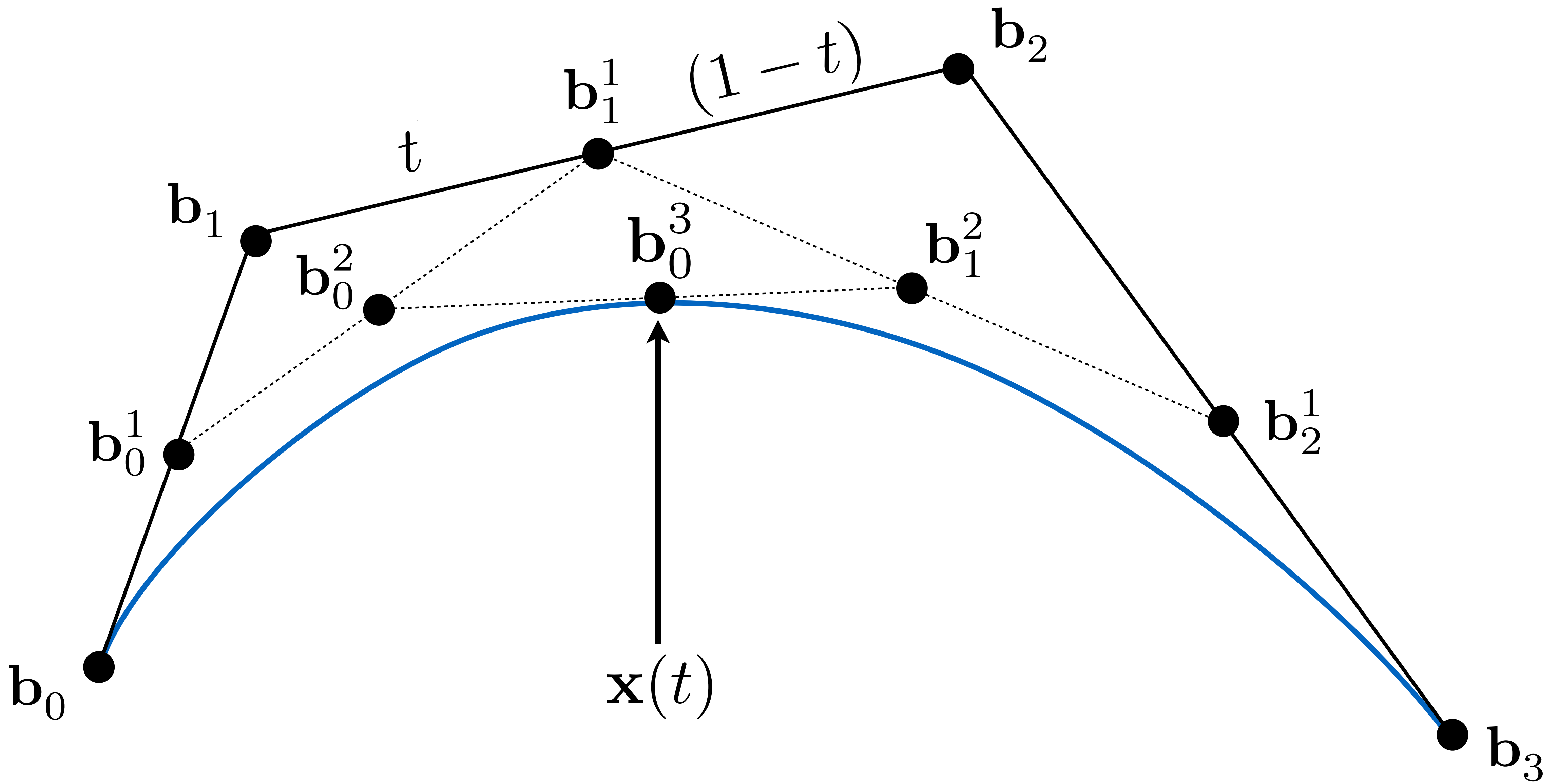
Pierre Bézier
1910 – 1999

Paul de Casteljau
b. 1930

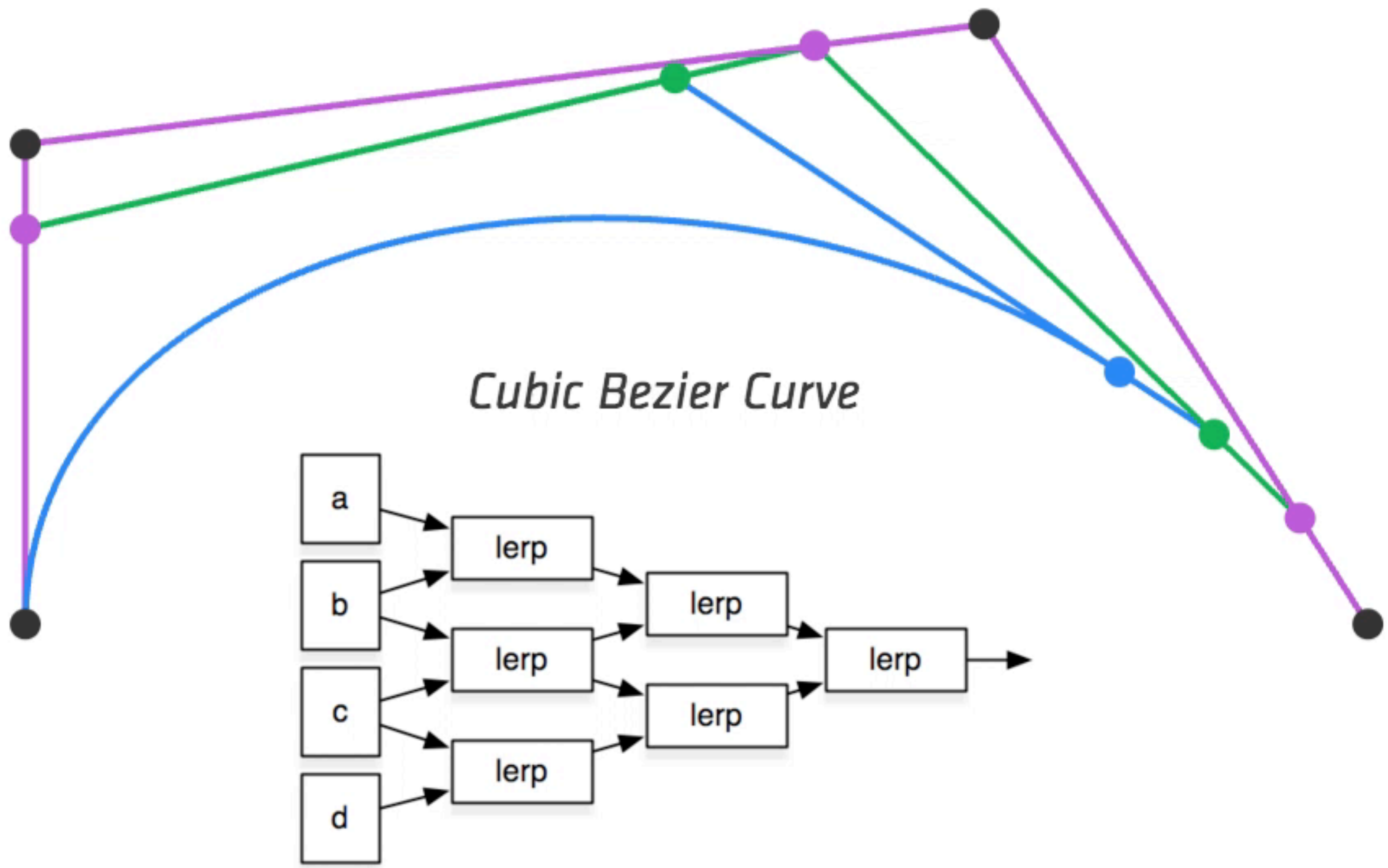
Cubic Bézier Curve – de Casteljau

Four input points in total

Same recursive linear interpolations



Visualizing de Casteljau Algorithm

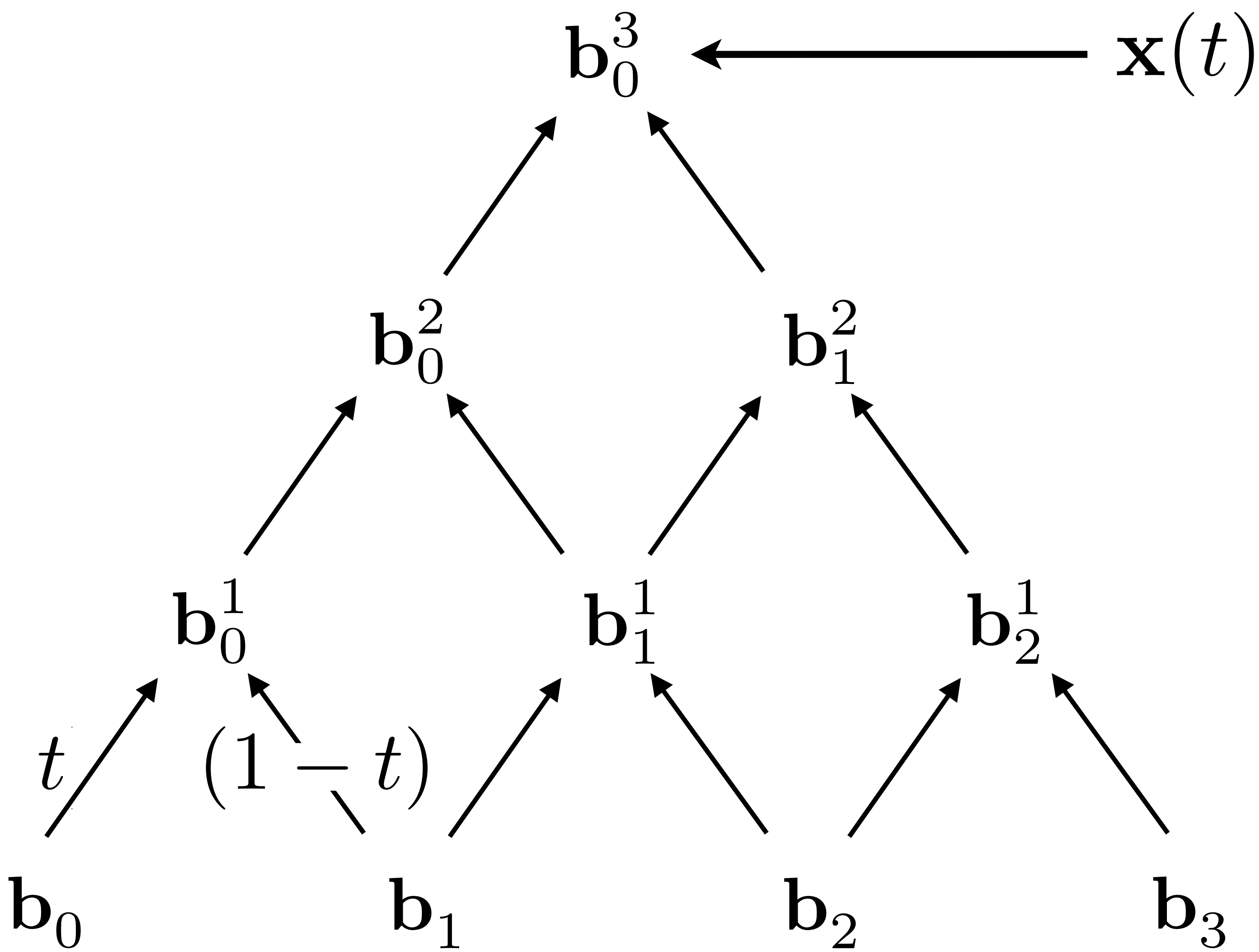


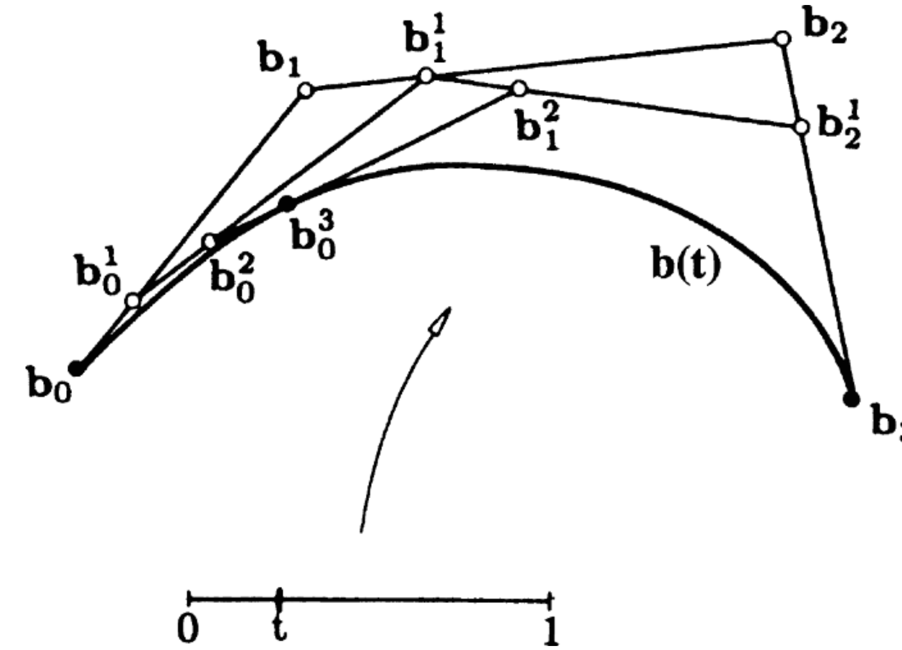
Animation: Steven Wittens, Making Things with Maths, http://acko.net

Evaluating Bézier Curves
Algebraic Formula

Bézier Curve – Algebraic Formula

de Casteljau algorithm gives a pyramid of coefficients

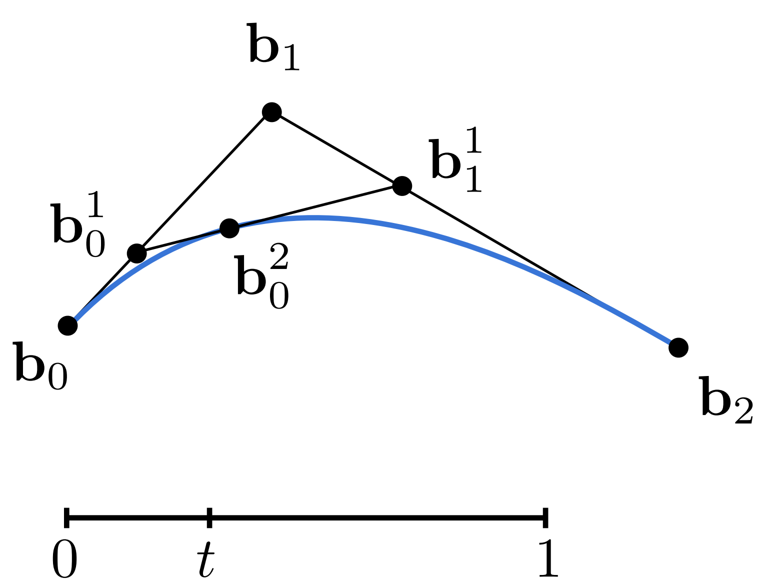




Every rightward arrow is multiplication by t,
Every leftward arrow by (1–t)

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 – General 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:

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

Bézier Curve – Algebraic Formula: Example

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:

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

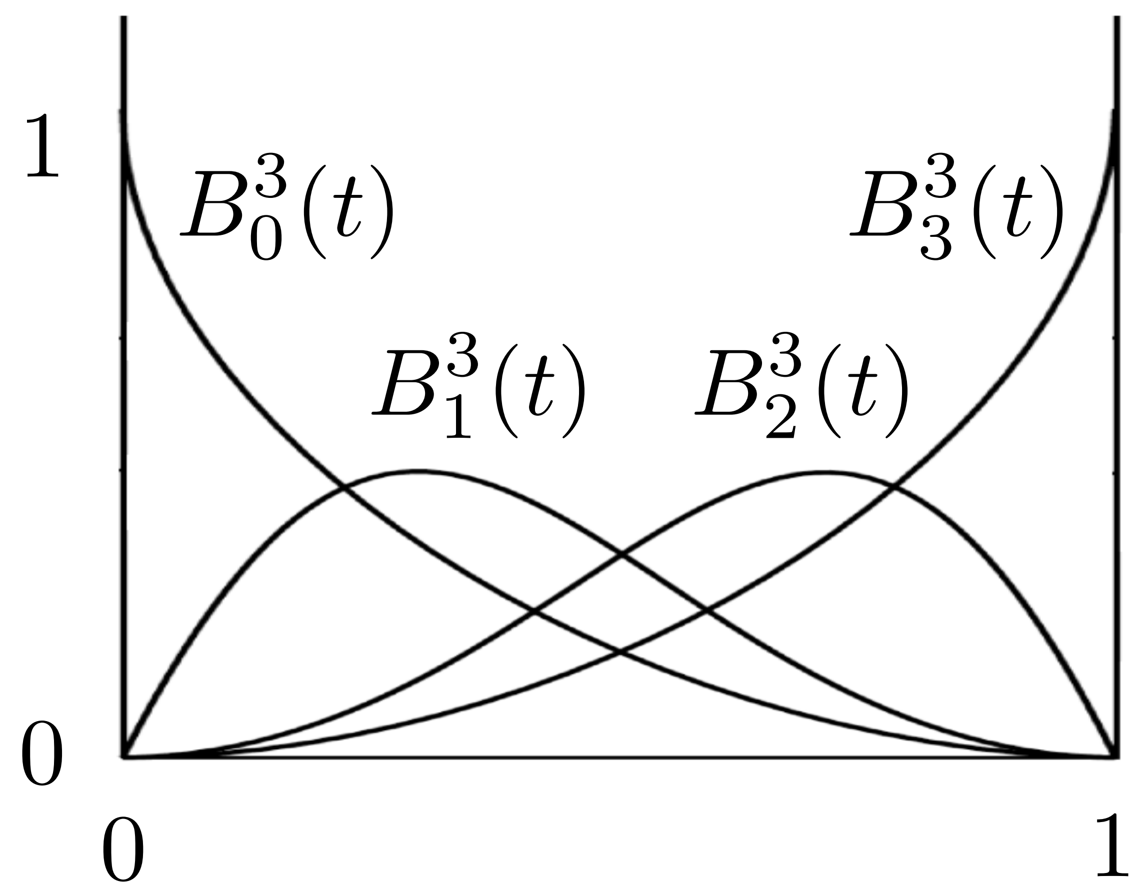
These points define a Bezier curve in 3D that is a cubic polynomial in t:

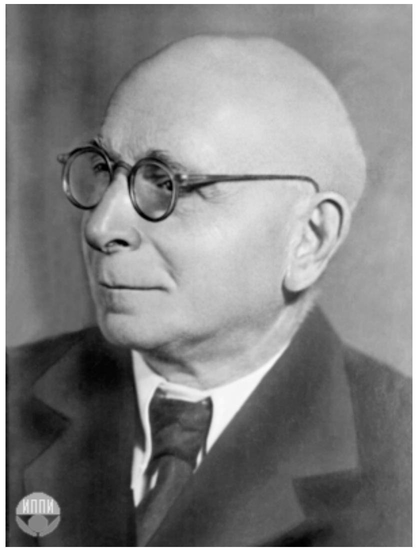
$$\mathbf{b}^n(t) = \mathbf{b}_0\,(1-t)^3 + \mathbf{b}_1\,3t(1-t)^2 + \mathbf{b}_2\,3t^2(1-t) + \mathbf{b}_3\,t^3$$

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

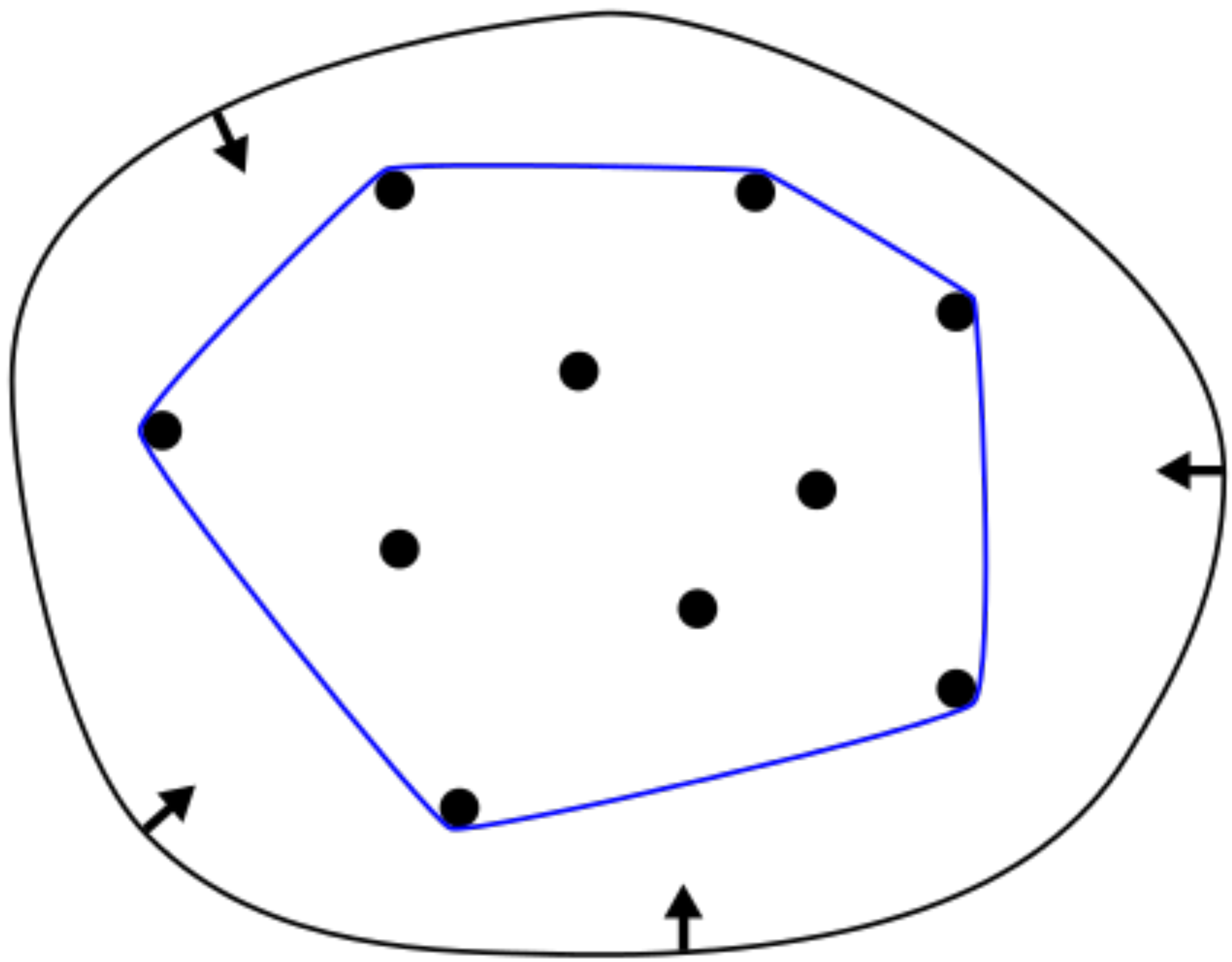
Affine transformation property

- Transform curve by transforming control points

Convex hull property

- Curve is within convex hull of control points

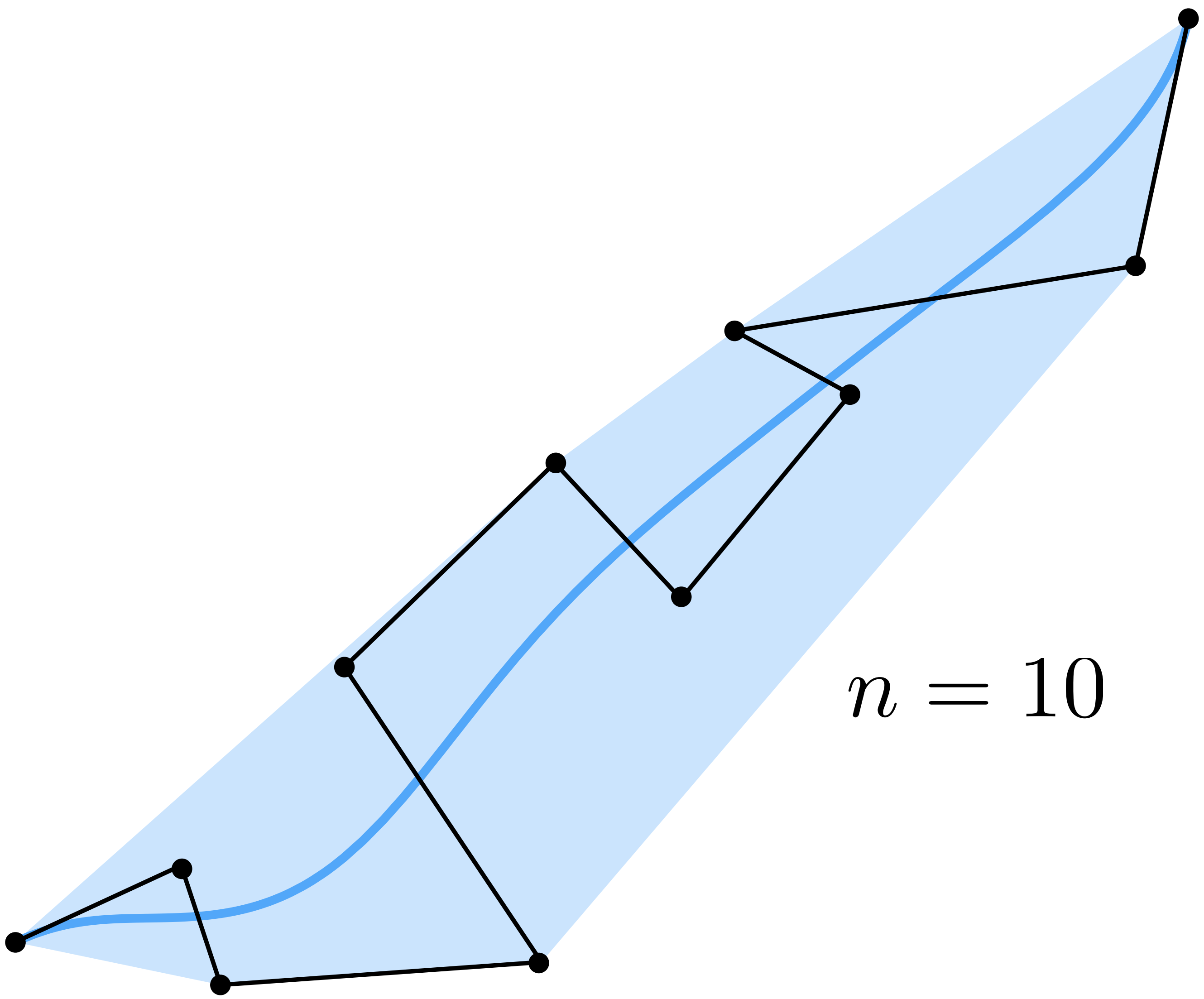
BTW: What's a Convex Hull



[from Wikipedia]

Piecewise Bézier Curves

Higher-Order Bézier Curves?



$n = 10$

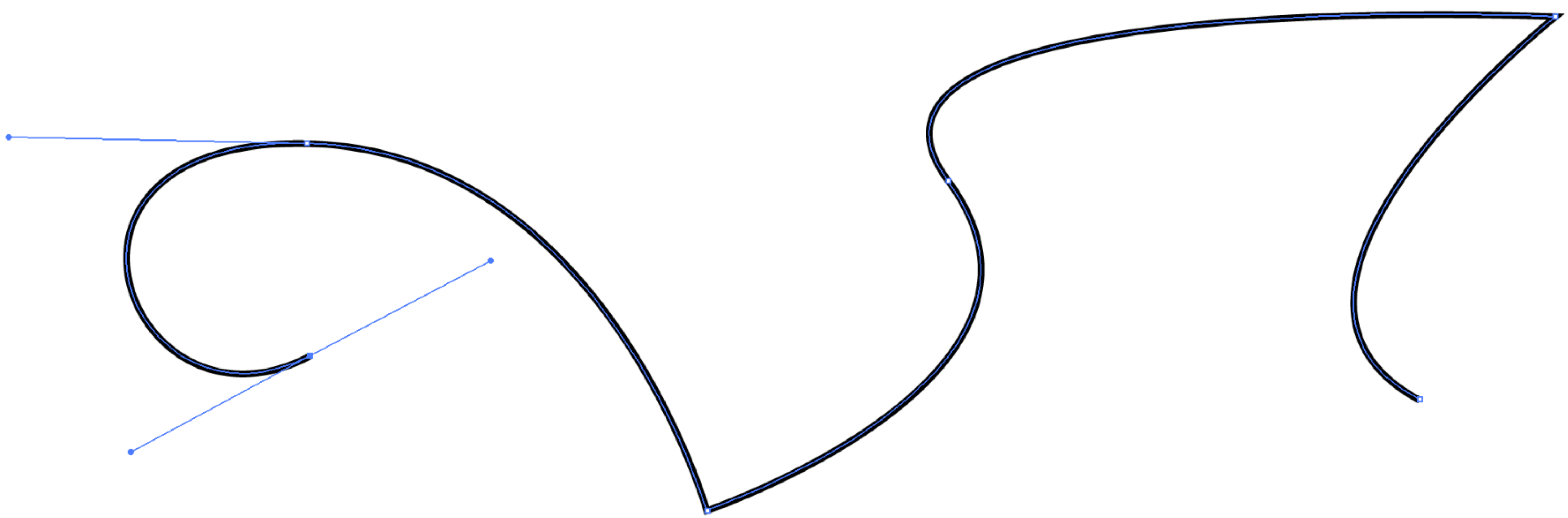
Very hard to control!

Uncommon

Piecewise Bézier Curves

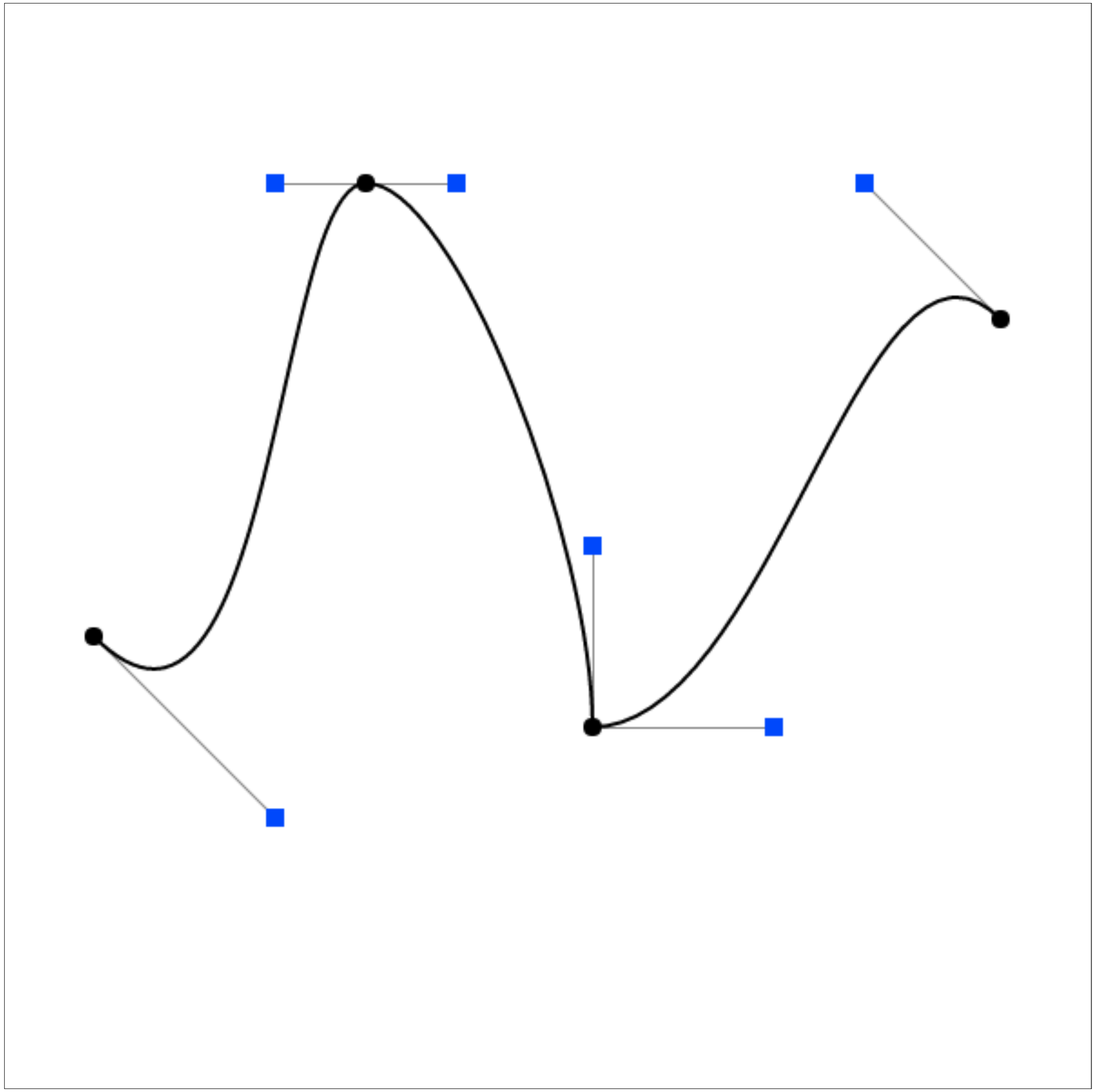
Instead, chain many low-order Bézier curve

Piecewise cubic Bézier the most common technique



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

Demo – Piecewise Cubic Bézier Curve



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

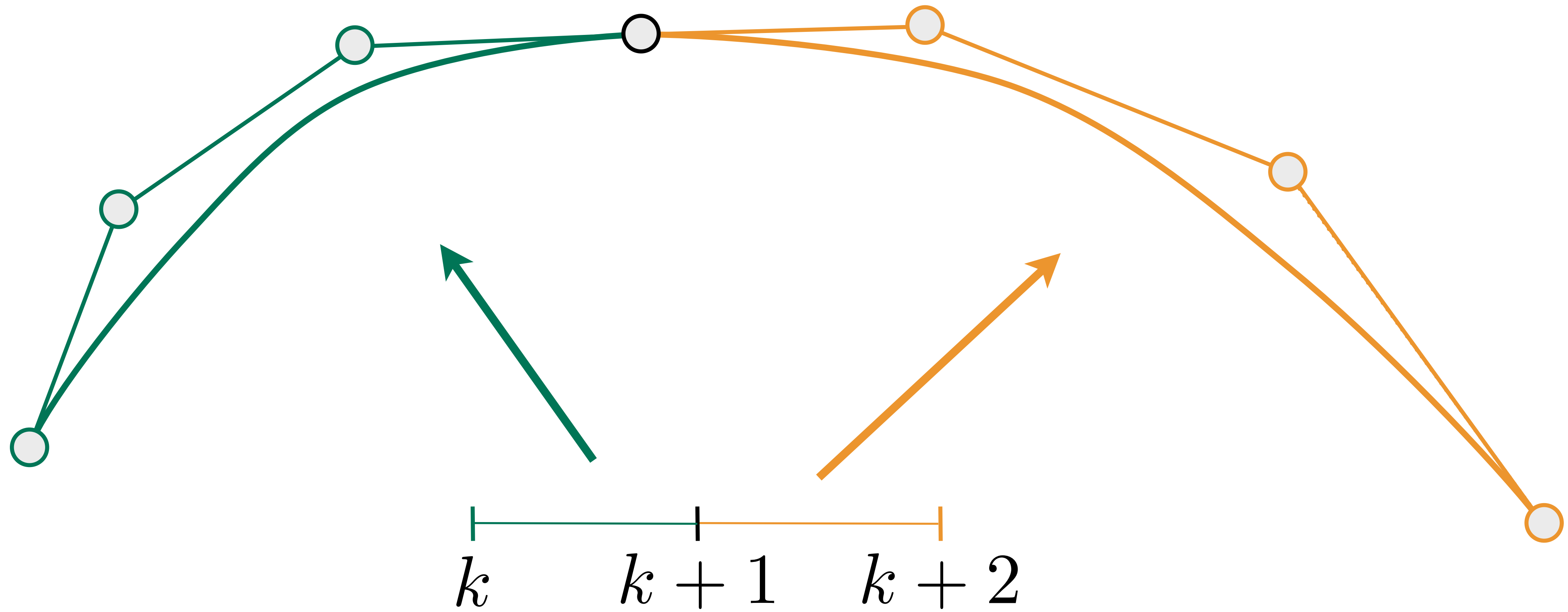
Piecewise Bézier Curve – Continuity

Two Bézier curves

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

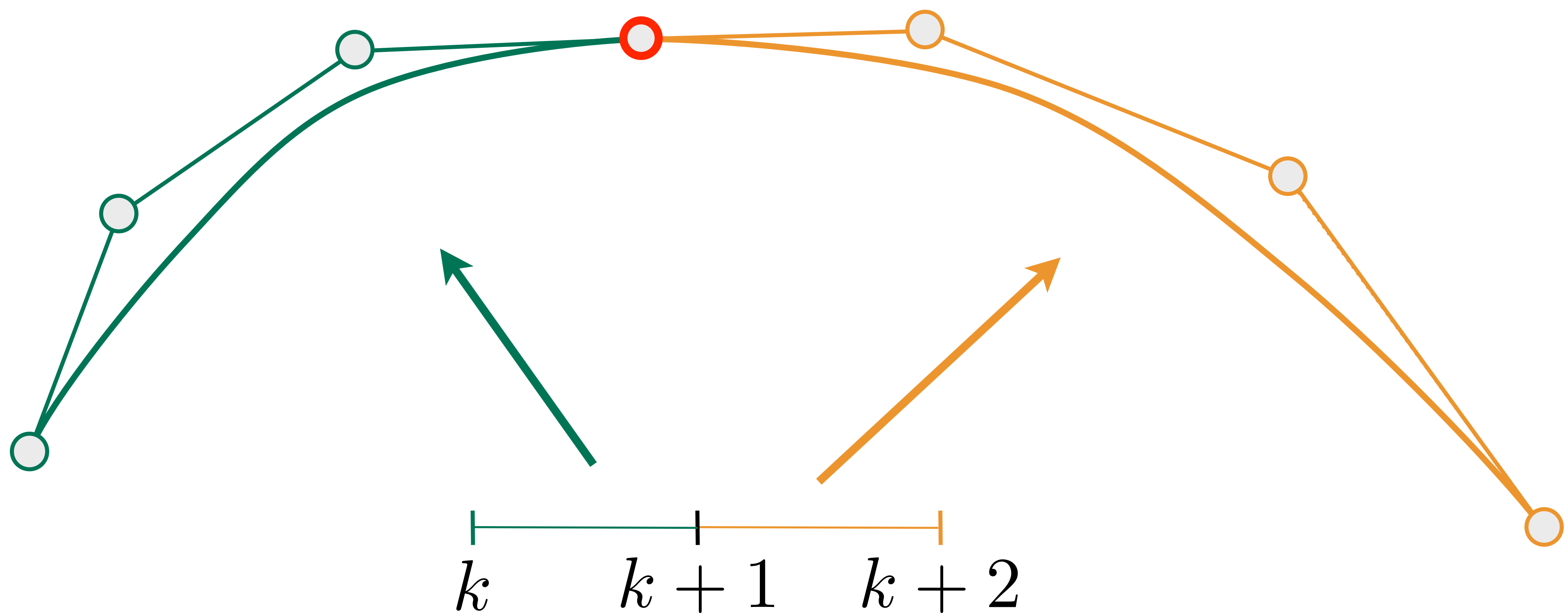
$$\mathbf{b} : [k+1, k+2] \to \mathbb{R}^N$$

Assuming integer partitions here,
can generalize



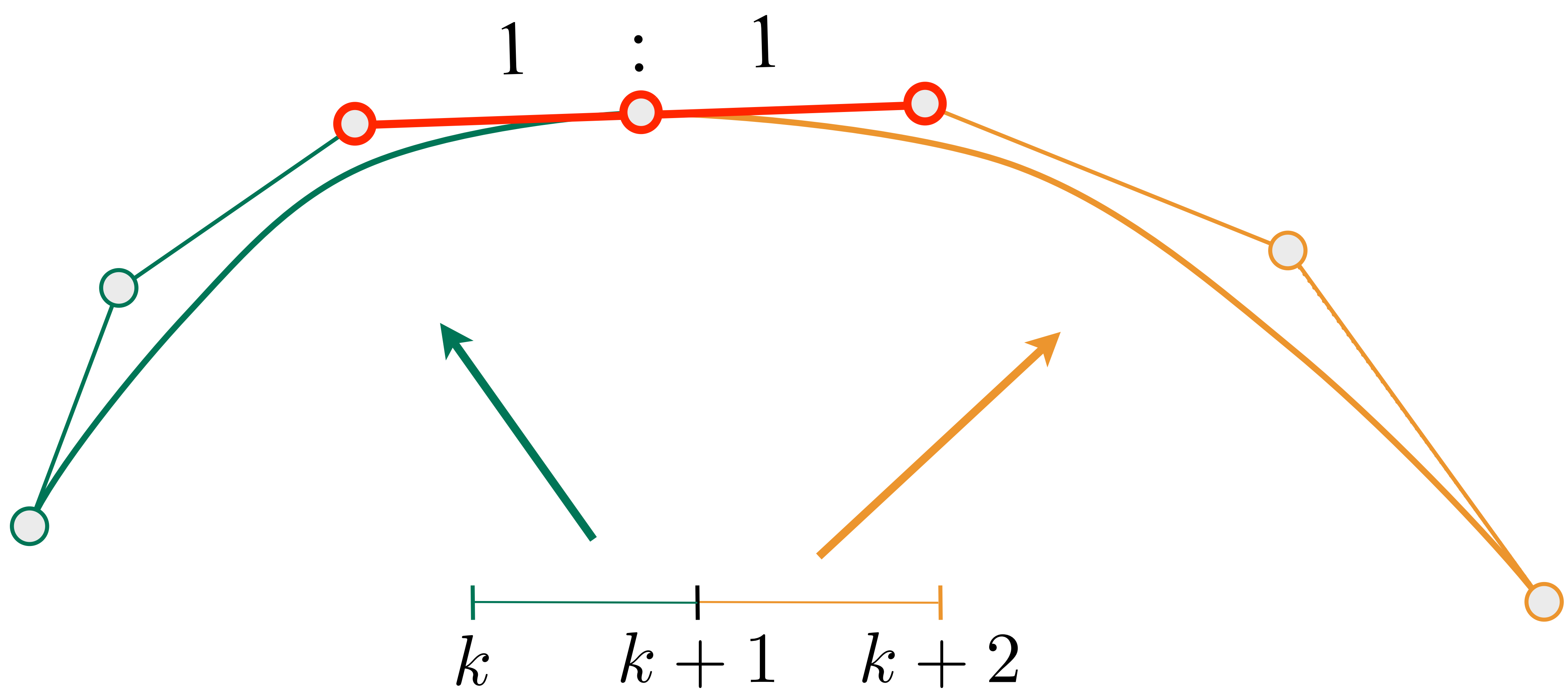
Piecewise Bézier Curve – Continuity

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



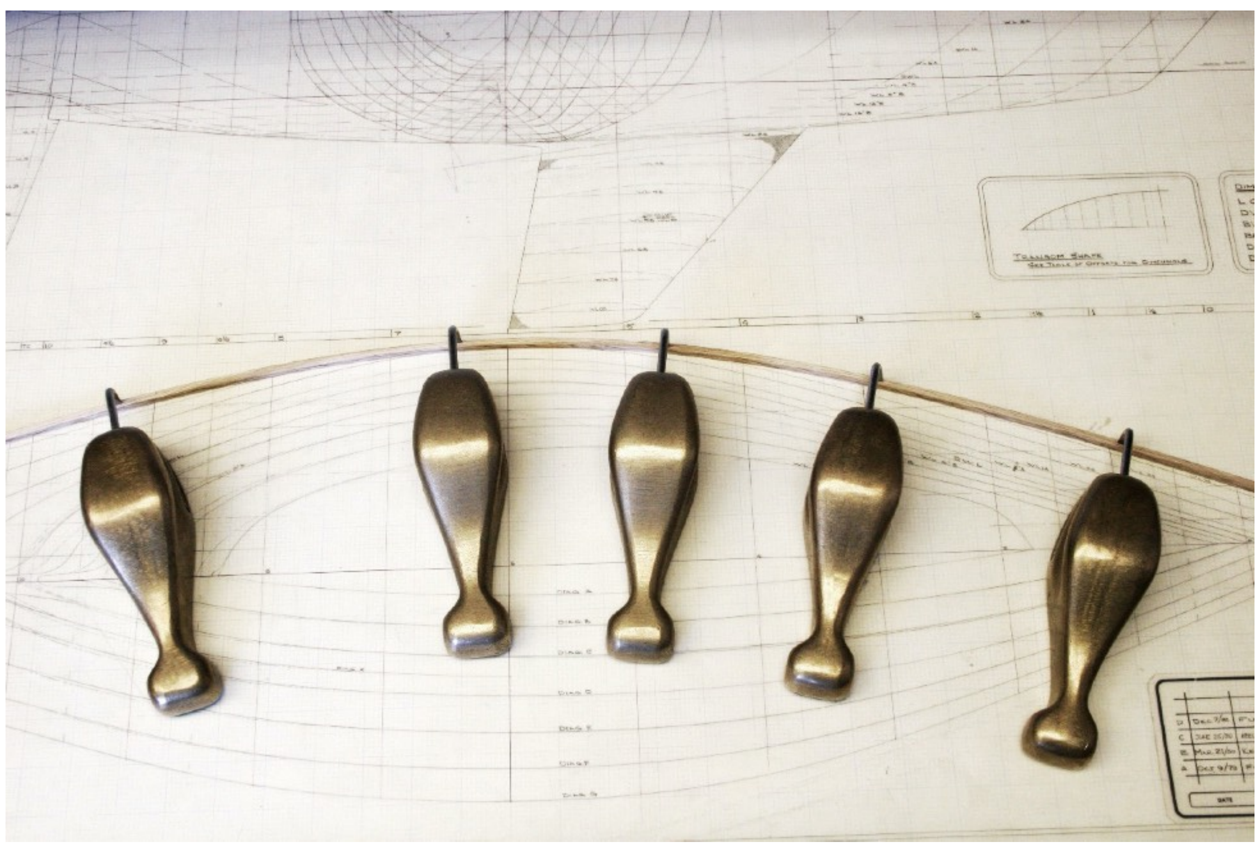
Piecewise Bézier Curve – Continuity

C^1 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)

https://en.wikipedia.org/wiki/B-spline

Important Note

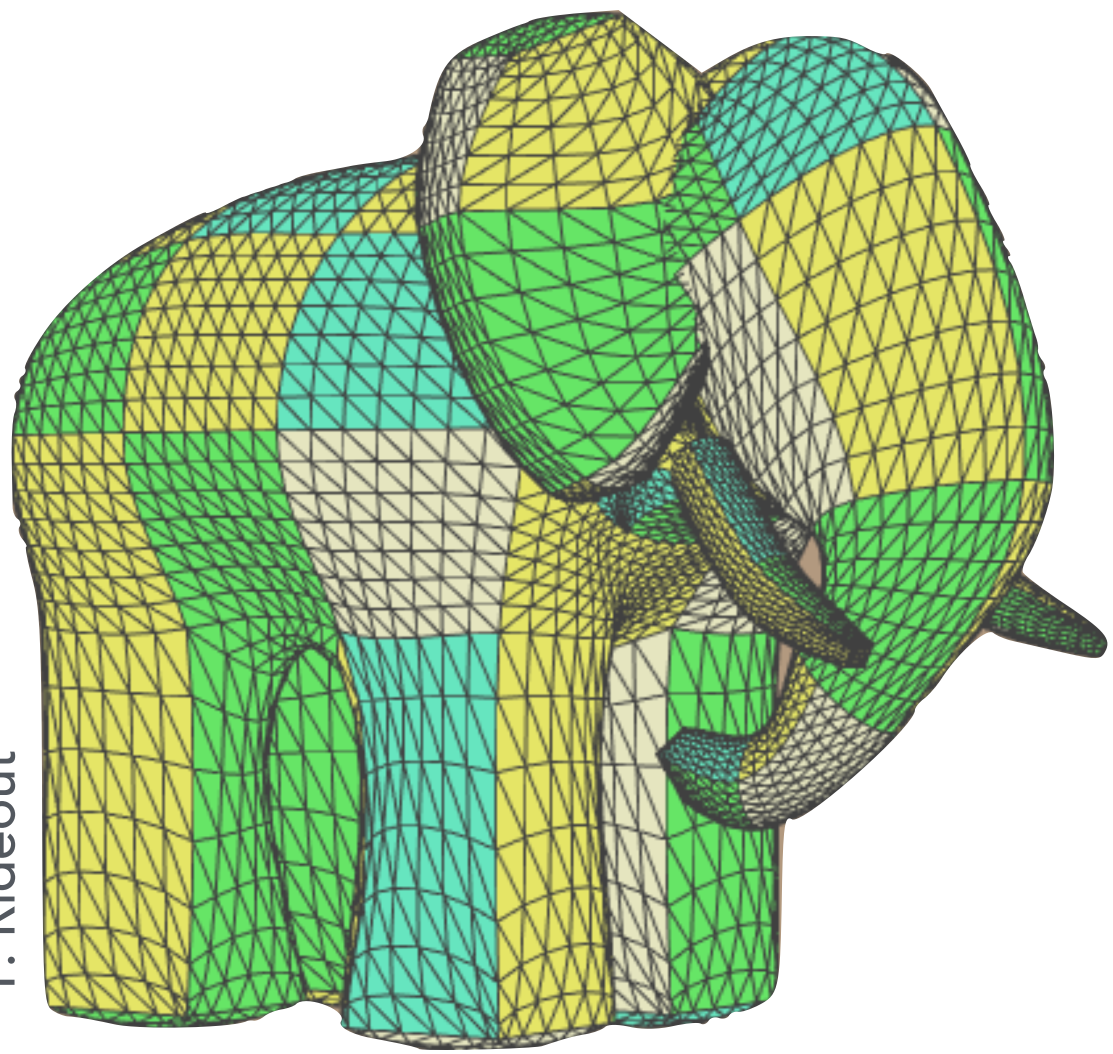
- In this course
  - We do not cover B-splines and NURBS
  - We also do not cover operations on curves (e.g. increasing/decreasing orders, etc.)
  - To learn more / deeper, you are welcome to refer to Prof. Shi-Min Hu's course: https://www.bilibili.com/video/av66548502?from=search&seid=65256805876131485

Today

- Curves
  - Bezier curves
  - De Casteljau’s algorithm
  - B-splines, etc.
- Surfaces
  - Bezier surfaces
  - Subdivision surfaces (triangles & quads)

Bézier Surfaces

Extend Bézier curves to surfaces



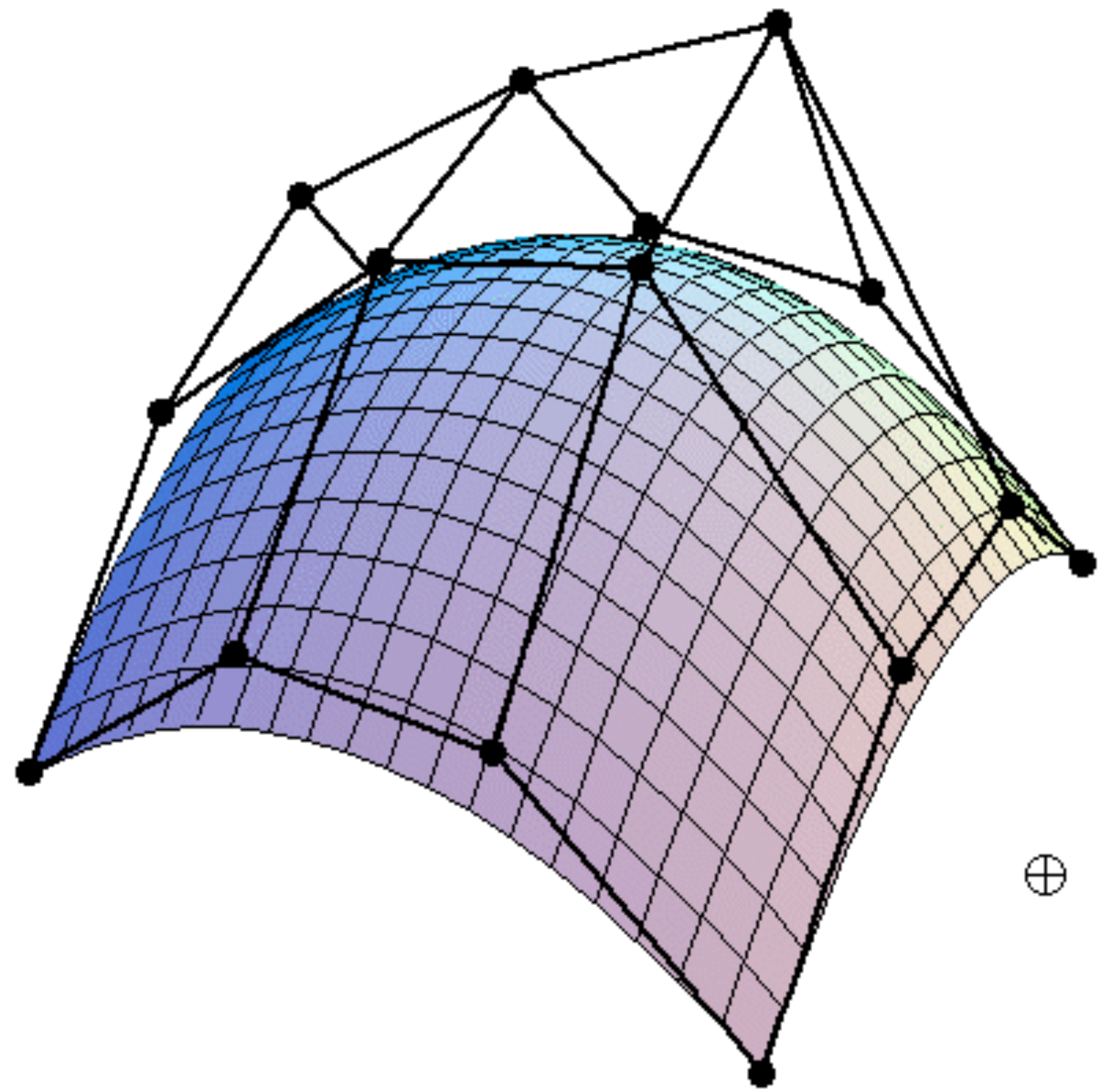
P. Rideout

Ed Catmull's "Gumbo" model

renderspirit.com

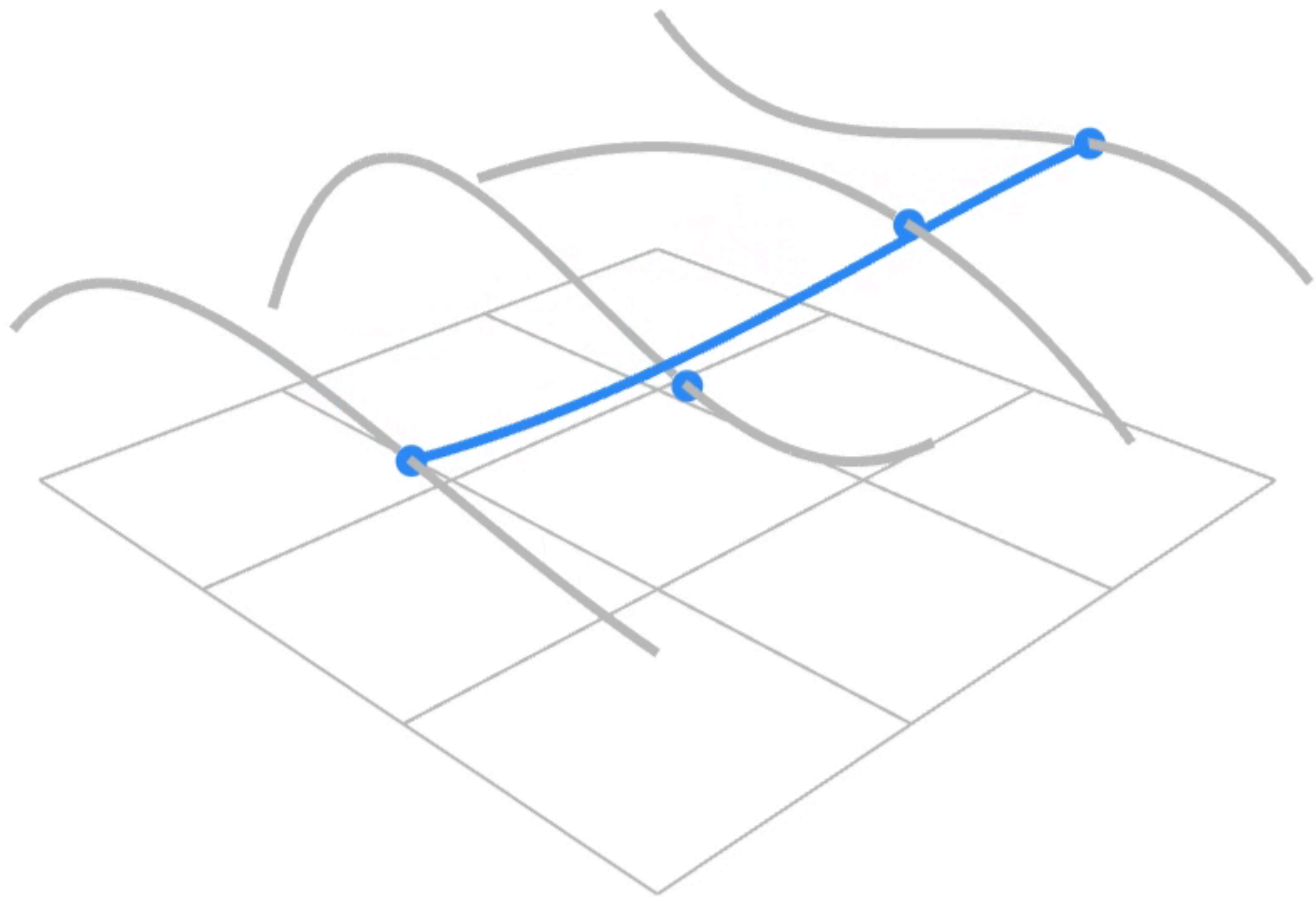
Utah Teapot

Bicubic Bézier Surface Patch



Bezier surface and 4 x 4 array of control points

Visualizing Bicubic Bézier Surface Patch

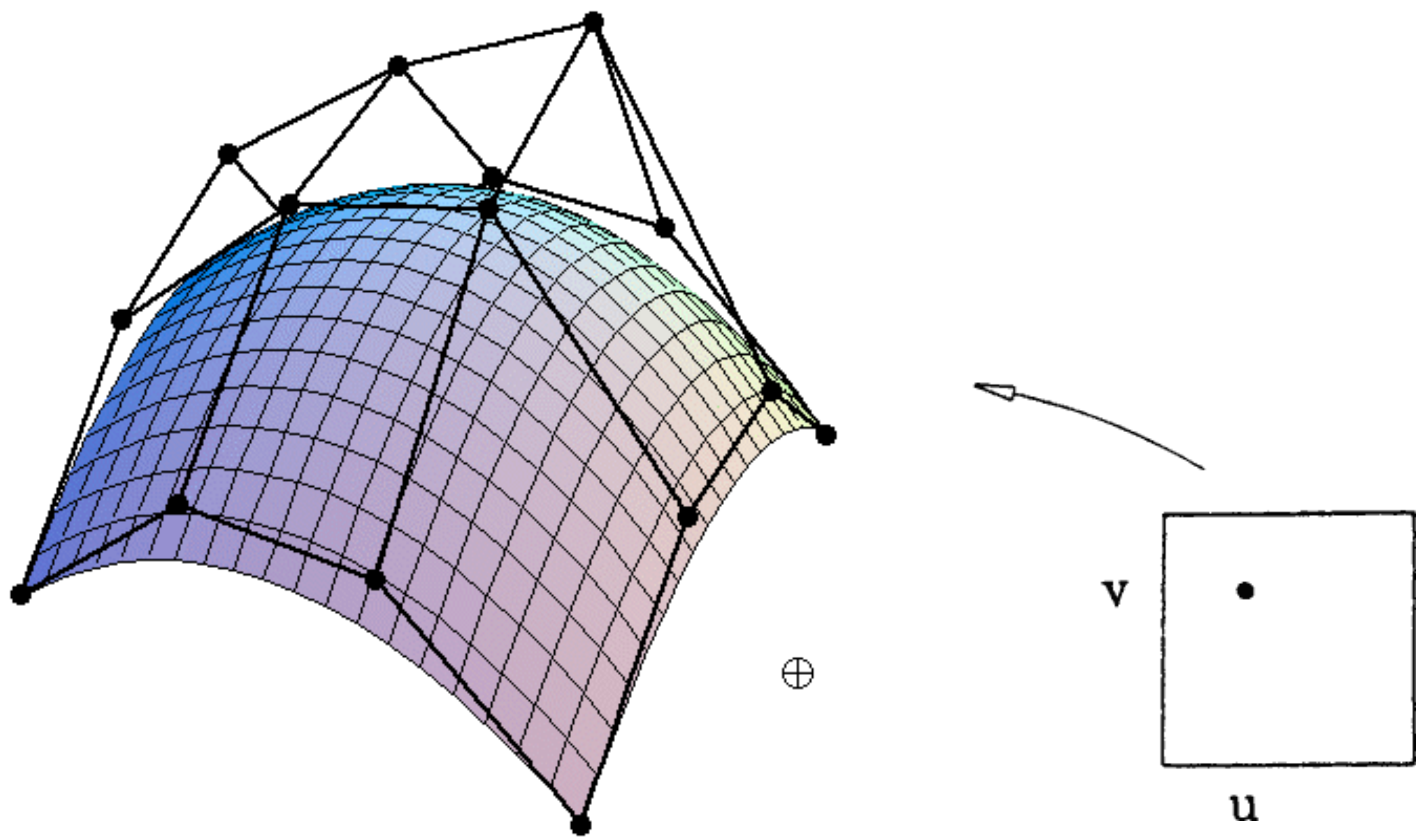


Animation: Steven Wittens, Making Things with Maths, http://acko.net

Evaluating Bézier Surfaces

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 $[0,1]^2$

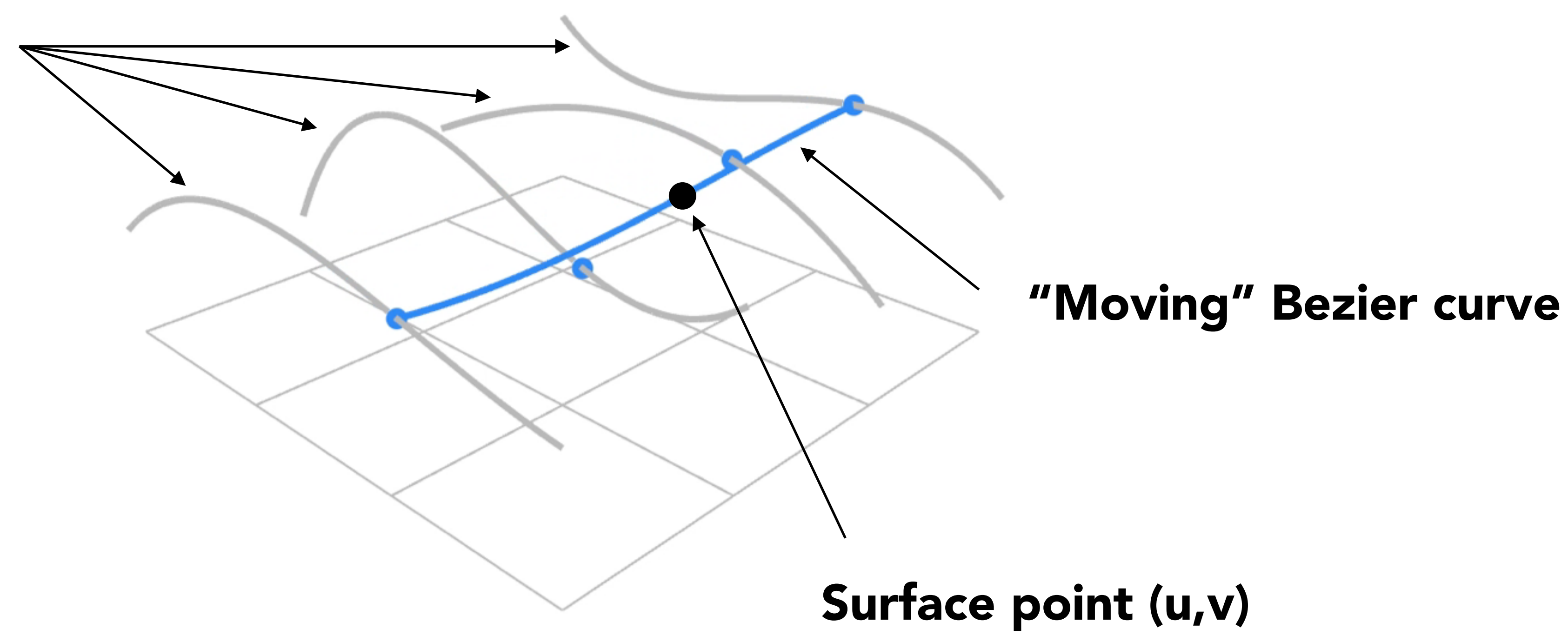


Method: Separable 1D de Casteljau Algorithm

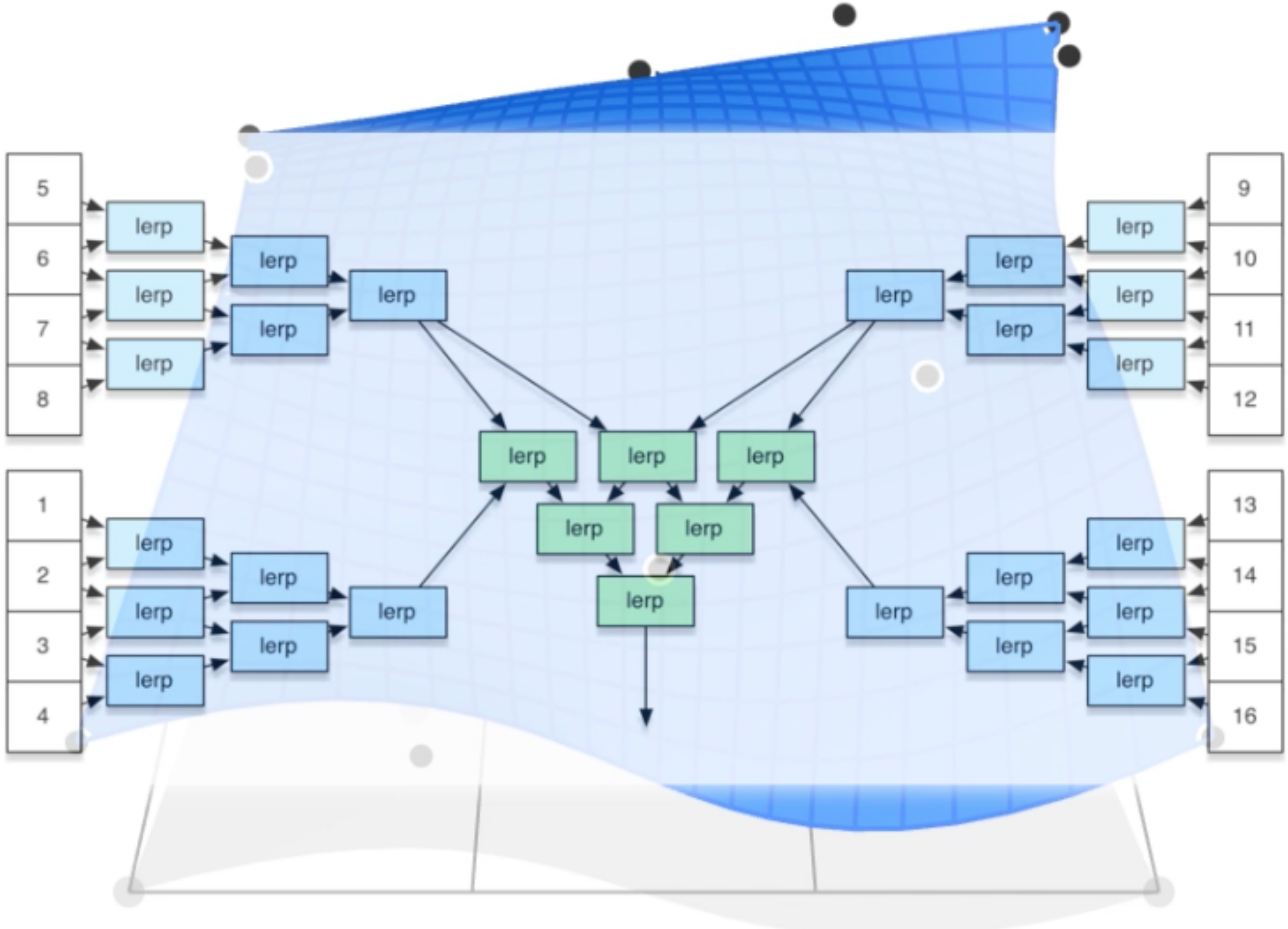
Goal: Evaluate surface position corresponding to (u,v)

(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

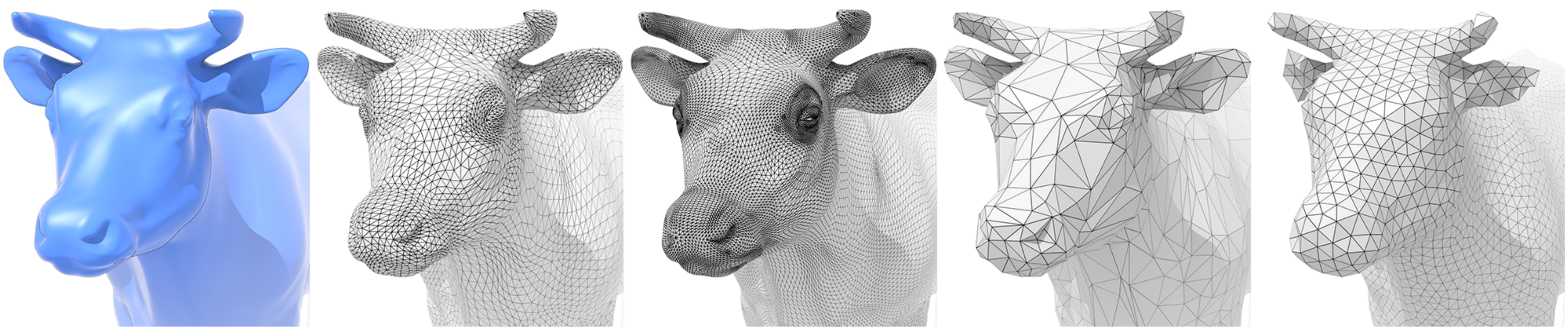


Method: Separable 1D de Casteljau Algorithm



Mesh Operations: Geometry Processing

- Mesh subdivision
- Mesh simplification
- Mesh regularization



Thank you!

(And thank Prof. Ravi Ramamoorthi and Prof. Ren Ng for many of the slides!)