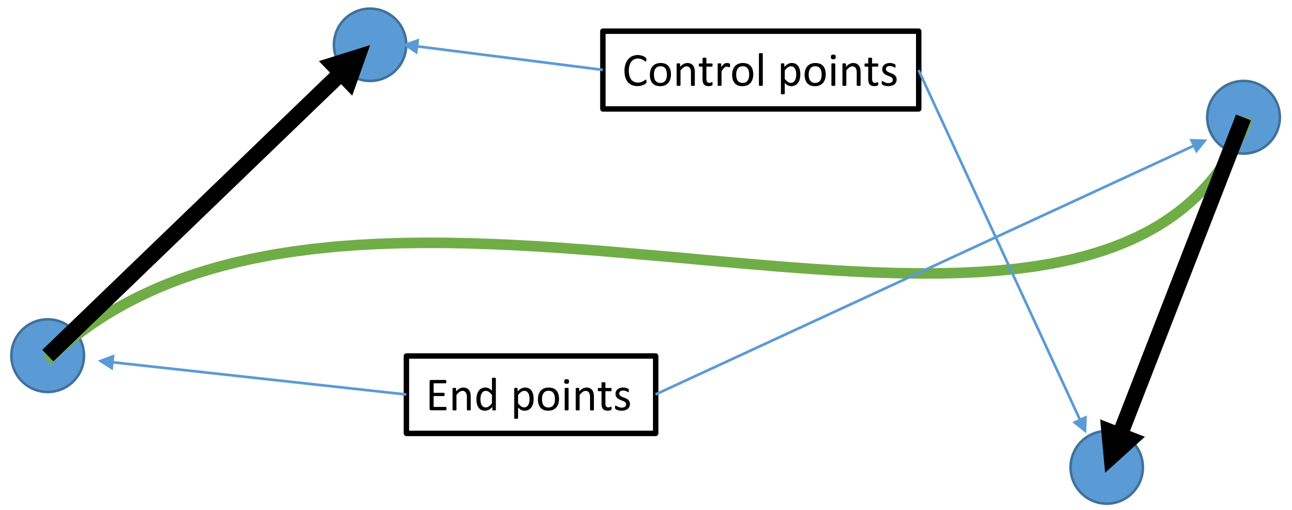
Interpolation – Bezier Curves

CS 370 - Lecture 9

May 20, 2016

Bezier Curves

This curve interpolates the end points. It is affected by the control points, but doesn't interpolate them.



The control points dictate the derivative at the nodes (roughly like in Hermite interpolation) but their *position* also affects the curve shape.

Bernstein Polynomials

1st tool we need is *Bernstein polynomials*.

For a given degree N, they have the form

$$B_{i,N}(t) = \binom{N}{i} t^i (1-t)^{N-i},$$

where i is in [0,N].

Reminder: Binomial coefficients

The notation $\binom{n}{k}$ indexes the binomial coefficients, so $\binom{n}{k} = \frac{n!}{k!(n-k)!}$.
A handy mnemonic (for low degrees) is "Pascal's triangle":

n=0	1
n=1	$1\ 1$
n=2	$1\ 2\ 1$
n=3	$1\ 3\ 3\ 1$
n=4	$1\ 4\ 6\ 4\ 1$
n=5	$1\ 5\ 10\ 10\ 5\ 1$

…

where the entries for a given row are the sum of the two nearest entries in the row above. E.g. $\binom{4}{2} = ?$, $\binom{3}{1} = ?$ A: 6 and 3, respectively.

Bernstein Polynomials – N=1

$$B_{i,N}(t) = \binom{N}{i} t^i (1-t)^{N-i}, \text{for i} = 0 \ldots \text{N.}$$

What are the Bernstein polynomials for N=1?

$$B_{0,1}(t) = \binom{1}{0} \mathrm{t}^0 (1-\mathrm{t})^1 = (1-\mathrm{t})$$

$$B_{1,1}(t) = \binom{1}{1} \mathrm{t}^1 (1-\mathrm{t})^0 = t$$

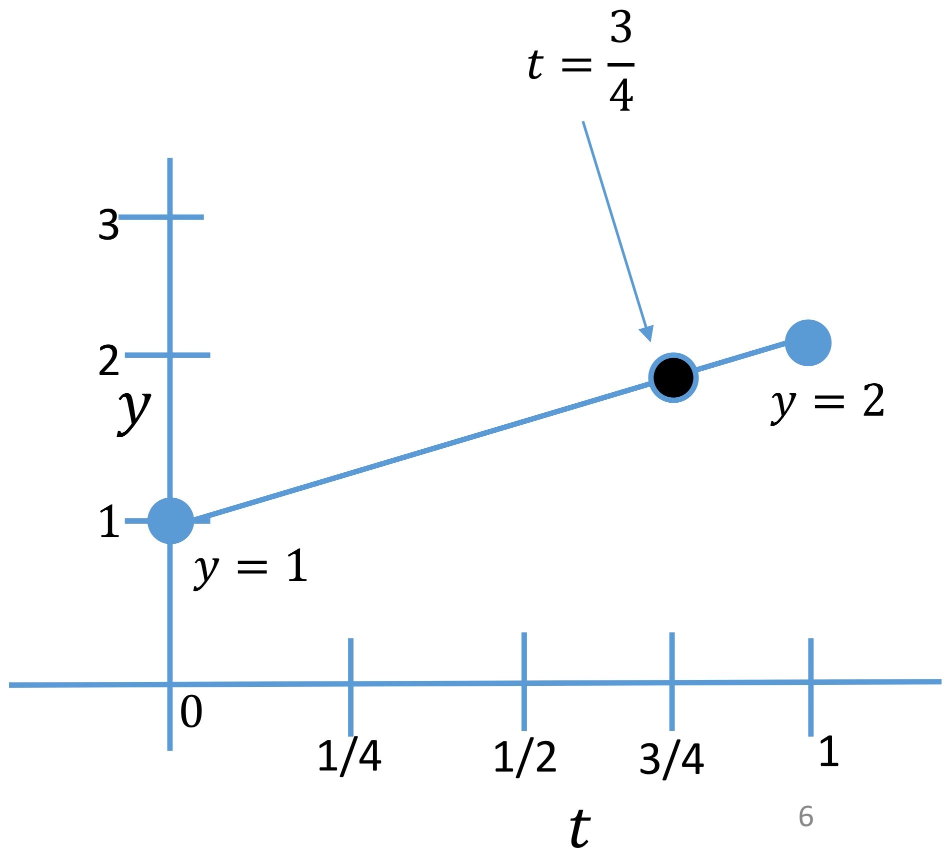
Notice: These are just the weights for *linear interpolation* on $[0,1]$.

Linear Interpolation - Example

e.g., Linearly interpolating between $(t_0 = 0, y_0 = 1)$ and $(\mathrm{t}_1 = 1, y_1 = 2)$ at $t = \frac{3}{4}$:

$$p(t) = y_0(1 - t) + y_1 t$$
$$= y_0 B_{0,1}(t) + y_1 B_{1,1}(t)$$

For $t = 3/4$, $p\left(\frac{3}{4}\right) = (1)\left(\frac{1}{4}\right) + (2)\left(\frac{3}{4}\right) = \frac{7}{4}$.



Bernstein Polynomials, N=2

$$B_{i,N}(t) = \binom{N}{i} t^i (1-t)^{N-i}, \text{for i} = 0 \ldots \text{N.}$$

e.g., What are the Bernstein polynomials for N=2?

$$B_{0,2}(t) = \binom{2}{0} t^0 (1-t)^2 = (1-t)^2$$

$$B_{1,2}(t) = \binom{2}{1} t^1 (1-t)^1 = 2t(1-t)$$

$$B_{2,2}(t) = \binom{2}{2} t^2 (1-t)^0 = t^2$$

Bernstein Polynomials, N=3

$$B_{i,N}(t) = \binom{N}{i} t^i (1-t)^{N-i}, \text{for i} = 0 \ldots \text{N}.$$

e.g., What are the Bernstein polynomials for N=3?

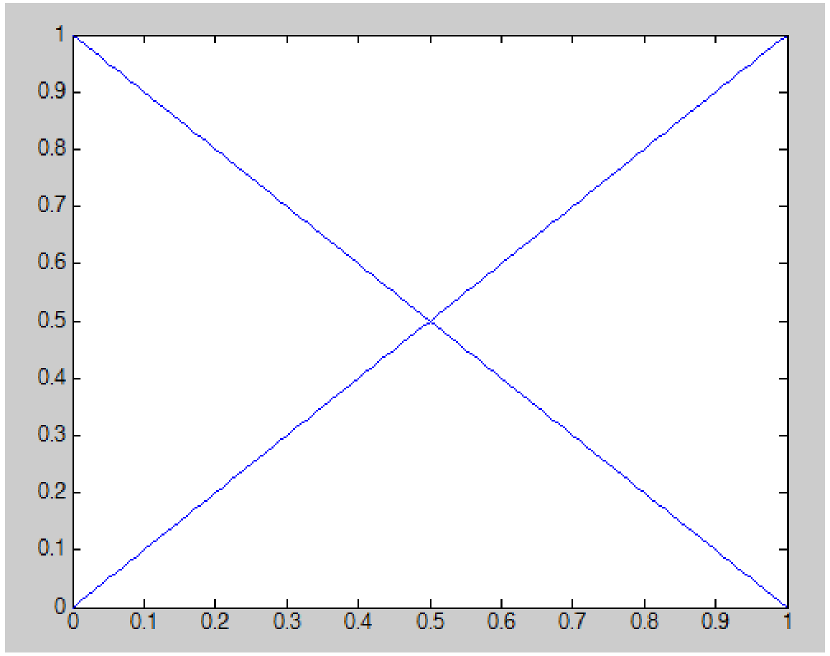
$$B_{0,3}(t) = \binom{3}{0} t^0 (1-t)^3 = (1-t)^3$$

$$B_{1,3}(t) = \binom{3}{1} t^1 (1-t)^2 = 3t(1-t)^2$$

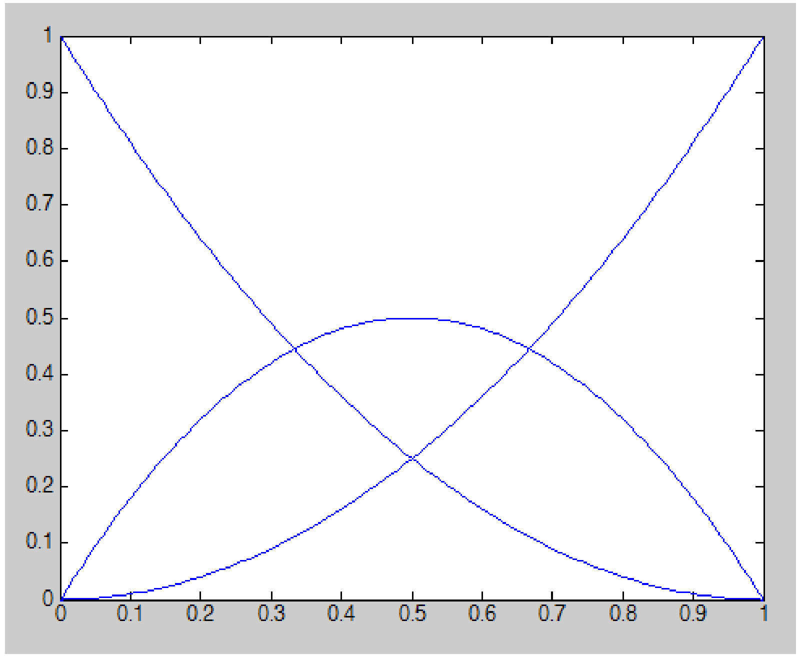
$$B_{2,3}(t) = \binom{3}{2} t^2 (1-t)^1 = 3t^2(1-t)$$

$$B_{3,3}(t) = \binom{3}{3} t^3 (1-t)^0 = t^3$$

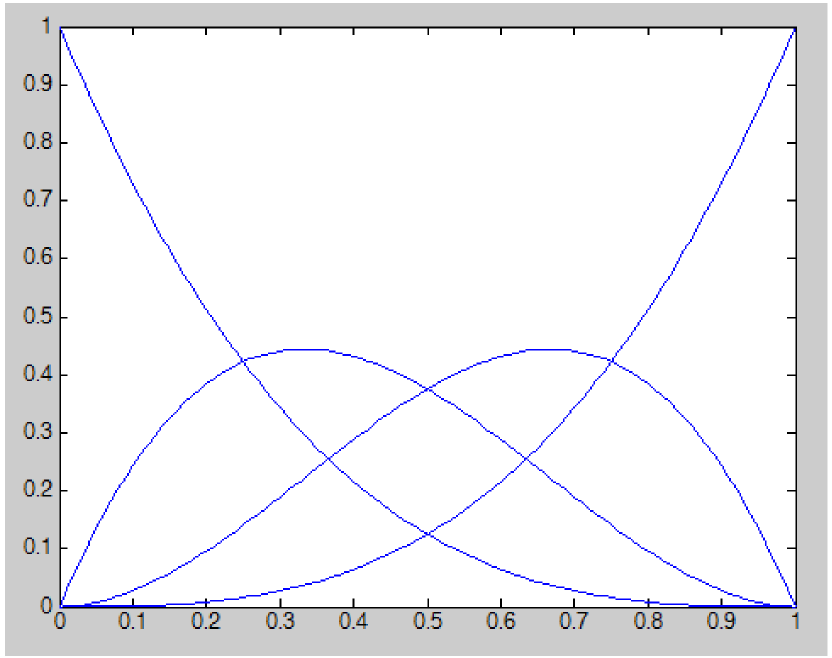
Bernstein Polynomials Visualized



Bernstein polynomials, N=1



Bernstein polynomials, N=2



Bernstein polynomials, N=3

Bernstein Polynomials - Recurrence

Property #1: Bernstein polynomials satisfy a recurrence relation – *i.e.*, they can be built up from lower degree Bernstein polynomials

$$B_{i,N} = (1-t) \cdot B_{i,N-1}\,(t) + t \cdot B_{i-1,N-1}(t).$$

assuming base and boundary cases of

$B_{0,0}(t) = 1$, and $B_{i,N}(t) = 0$ for $i < 0 \text{ and } i > N$.

The next level is just a linear interpolation/blend of the polynomials one degree lower!

Recurrence Example

Bernstein polynomial form: $B_{i,N}(t) = \binom{N}{i} t^i (1-t)^{N-i}$

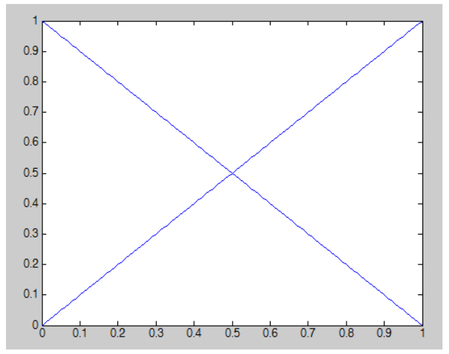
Recurrence: $B_{i,N}(t) = (1-t) \cdot B_{i,N-1}(t) + t \cdot B_{i-1,N-1}(t)$.

Base/boundary: $B_{0,0}(t) = 1$, and $B_{i,N}(t) = 0$.

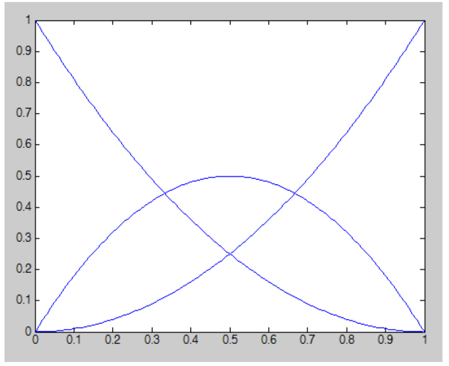
Verify the recurrence for $B_{2,3}(t) = 3t^2(1-t)$:

$$\begin{aligned} B_{2,3}(t) &= (1-t) \cdot B_{2,2}(t) + t \cdot B_{1,2}(t) \\ &= (1-t)t^2 + t\big(2t(1-t)\big) \\ &= (1-t)t^2 + 2t^2(1-t) \\ &= 3t^2(1-t) \end{aligned}$$

Bernstein polynomials – Non-negativity & derivatives



Property #2: They are non-negative on the interval [0,1].



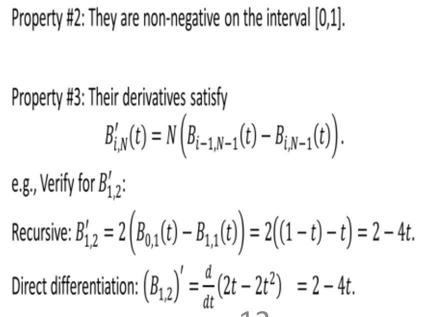
Property #3: Their derivatives satisfy

$$B'_{i,N}(t) = N\left(B_{i-1,N-1}(t) - B_{i,N-1}(t)\right).$$

e.g., Verify for $B'_{1,2}$:

Recursive: $B'_{1,2} = 2\left(B_{0,1}(t) - B_{1,1}(t)\right) = 2\big((1-t)-t\big) = 2-4t.$

Direct differentiation: $\left(B_{1,2}\right)' = \frac{d}{dt}(2t-2t^2) \quad = 2-4t.$



Property #2: They are non-negative on the interval [0,1].

Property #3: Their derivatives satisfy

$$B'_{i,N}(t) = N\left(B_{i-1,N-1}(t) - B_{i,N-1}(t)\right).$$

e.g., Verify for $B'_{1,2}$:

Recursive: $B'_{1,2} = 2\left(B_{0,1}(t) - B_{1,1}(t)\right) = 2\big((1-t)-t\big) = 2-4t.$

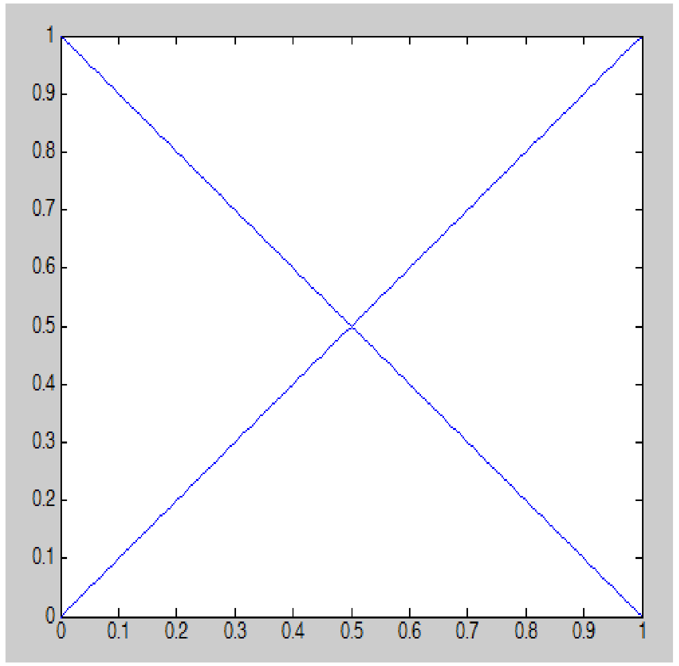
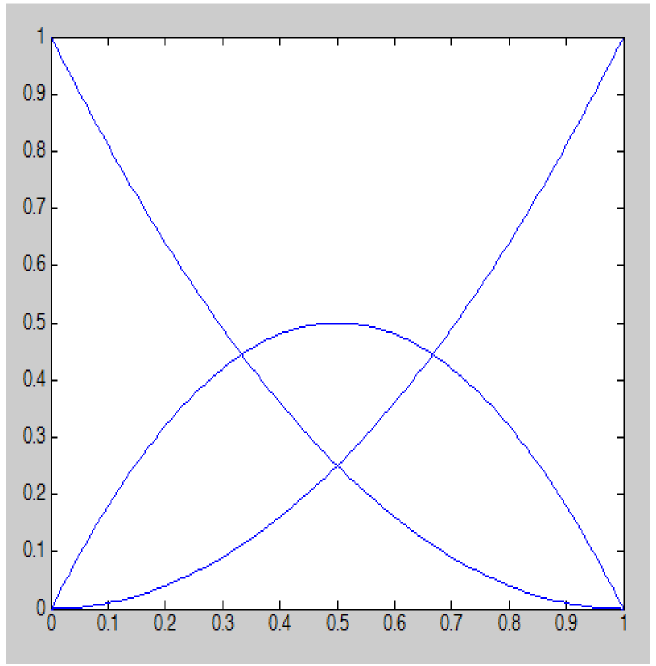
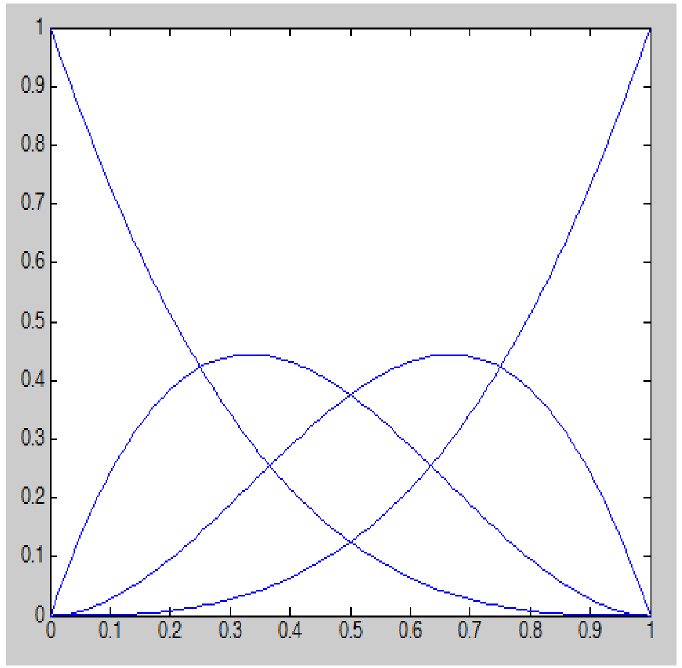
Direct differentiation: $\left(B_{1,2}\right)' = \frac{d}{dt}(2t-2t^2) \quad = 2-4t.$

Bernstein Polynomials – Partition of Unity

Property #4: Partition of unity:

The values of the Bernstein polynomials of a given degree all sum to 1.

$$\sum_{i=0}^{N} B_{i,N}(t) = 1$$

Bernstein Polynomials – Partition of Unity

Example: Verify that Bernstein polynomials partition unity for N=3.

$$\sum_{i=0}^{N} B_{i,3}(t) = t^3 + 3(1-t)t^2 + 3(1-t)^2t + (1-t)^3$$

$$= \cancel{t^3} + \cancel{3t^2} - \cancel{3t^3} + \cancel{3t} - \cancel{6t^2} + \cancel{3t^3} + 1 - \cancel{3t} + \cancel{3t^2} - \cancel{t^3}$$

$$= 1$$

Bernstein Polynomials - Properties

Property #5: The Bernstein polynomials form a ***polynomial basis***.

Any degree N polynomial can be written as some linear combination of the Bernstein polynomials of degree N, $B_{i,N}(t)$.

$$p(x) = \sum_{i=0}^{N} \alpha_i \, B_{i,N}(t)$$

Bezier Curves

A new kind of interpolant combining parametric curves with the Bernstein polynomial basis.

Given points (x_i, y_i) for $i = 0 \dots N$, the Bezier curve is

$$P(t) = \sum_{i=0}^{N} B_{i,N}(t) \cdot (x_i, y_i)$$

i.e., a linear (actually affine) combination of points where coefficients are determined by evaluating Bernstein polynomials.

Bezier Curves - Interpolation

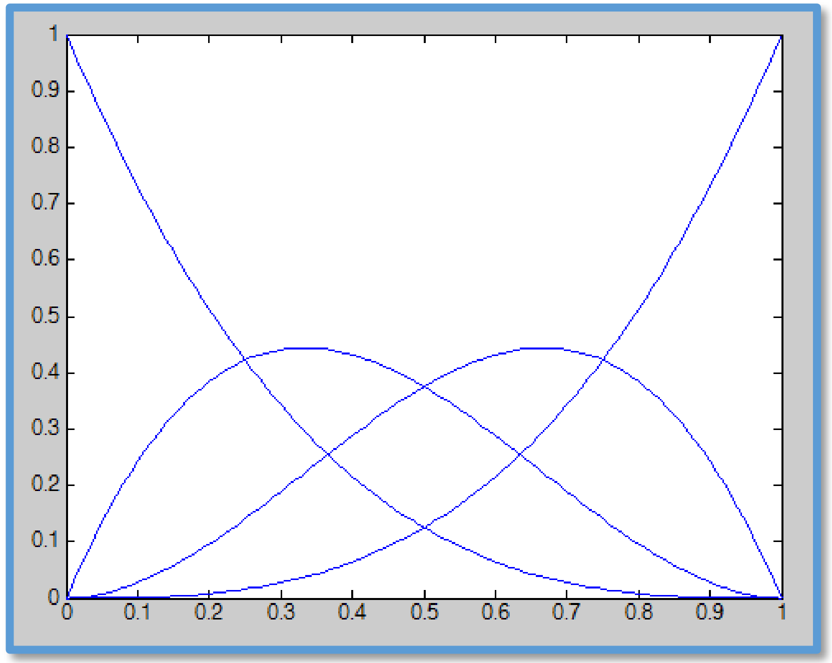
$$P(t) = \sum_{i=0}^{N} B_{i,N}(t)\,(x_i, y_i)$$

Various properties of Bezier curves come from the use of Bernstein polynomials.

a) They interpolate the **first** and **last** point.

At left, $B_{0,N}(0) = 1$, and $B_{i,N}(0) = 0$ for $i \neq 0$.

At right, $B_{N,N}(1) = 1$, and $B_{i,N}(1) = 0$ for $i \neq N$.



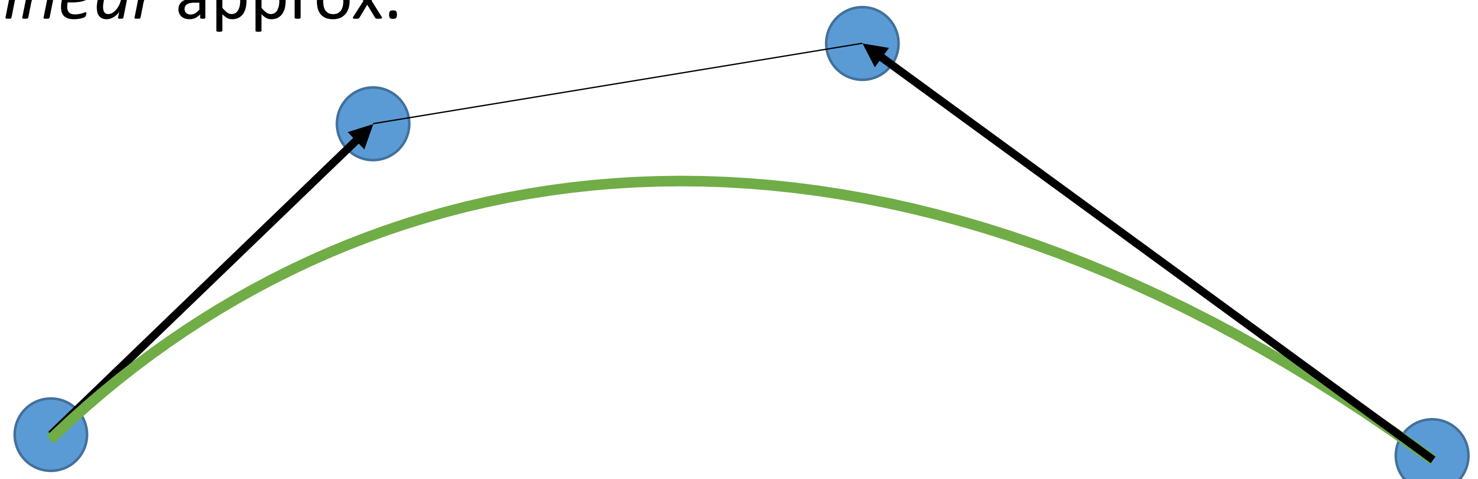
Bernstein polynomials, N=3

Bezier Curves – Endpoint Derivatives

b) Bernstein polynomial derivative formula leads to derivatives (tangent directions) at end points being:

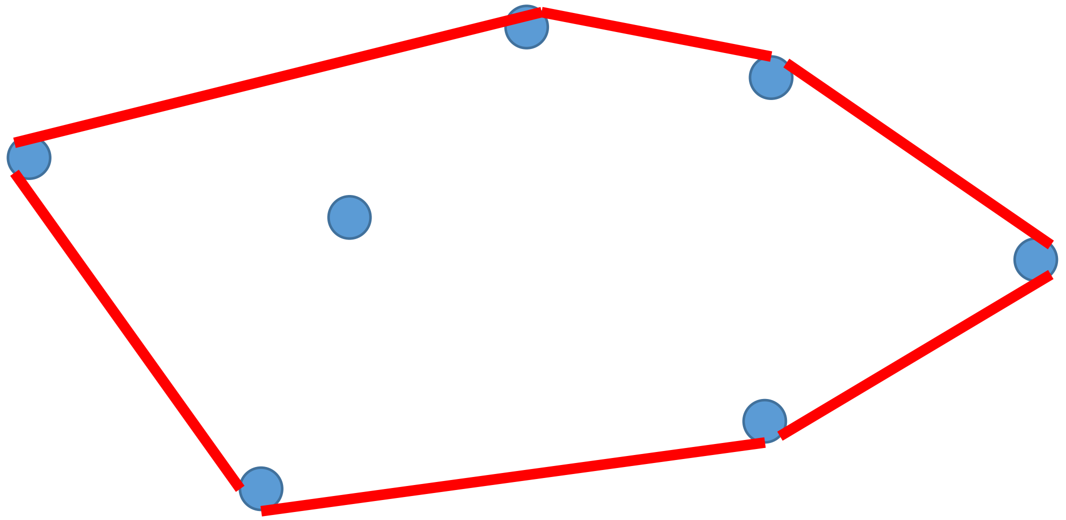
$$P'(0) = N(x_1 - x_0, y_1 - y_0)$$
$$P'(1) = N(x_n - x_{n-1}, y_n - y_{n-1})$$

So the tangents at the endpoints align with the first/last segment of the piecewise *linear* approx.

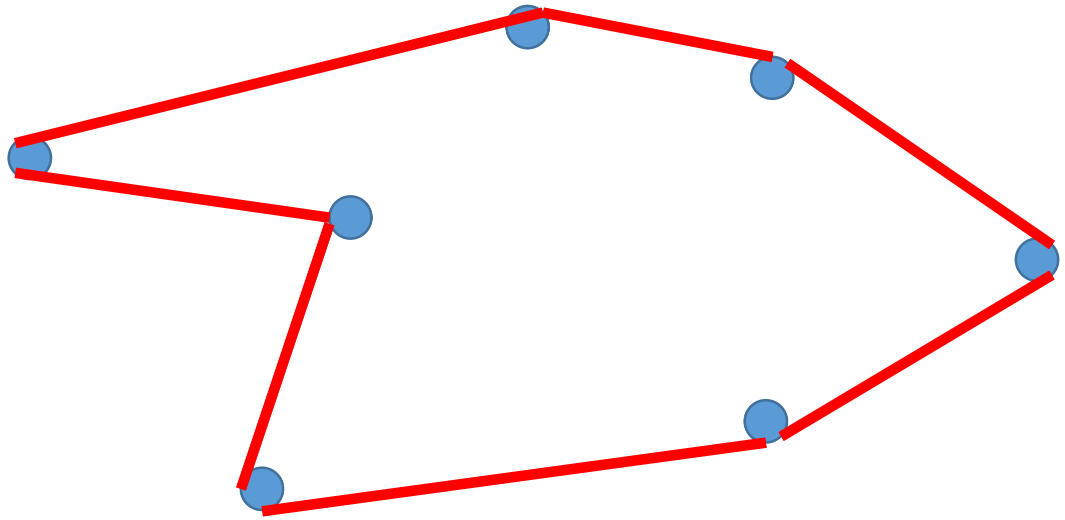


Convex Hull

Definition: The *convex hull* of a set of points is the smallest convex region that contains all the points.



Convex Hull



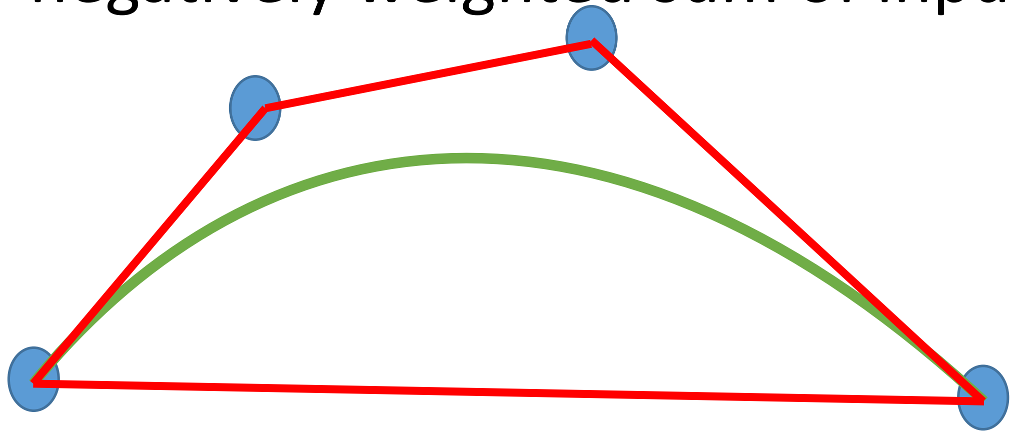
Not a Convex Hull

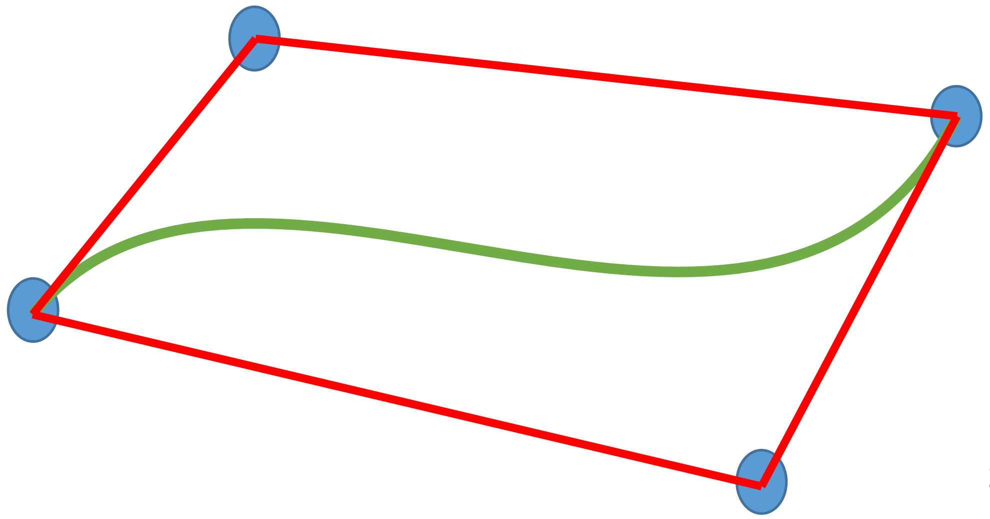
Bezier curves - Convexity

$$P(t) = \sum_{i=0}^{N} B_{i,N}(t)\,(x_i, y_i)$$

c) A Bezier curve is always bounded by the *convex hull* of its input/control points. Why? Due to which Bernstein poly property?

Bernstein polys *partition unity*, so every point on the curve is a non-negatively weighted sum of input points, with weights summing to 1.





Bezier Curves – Recursive Construction (de Casteljau's algorithm)

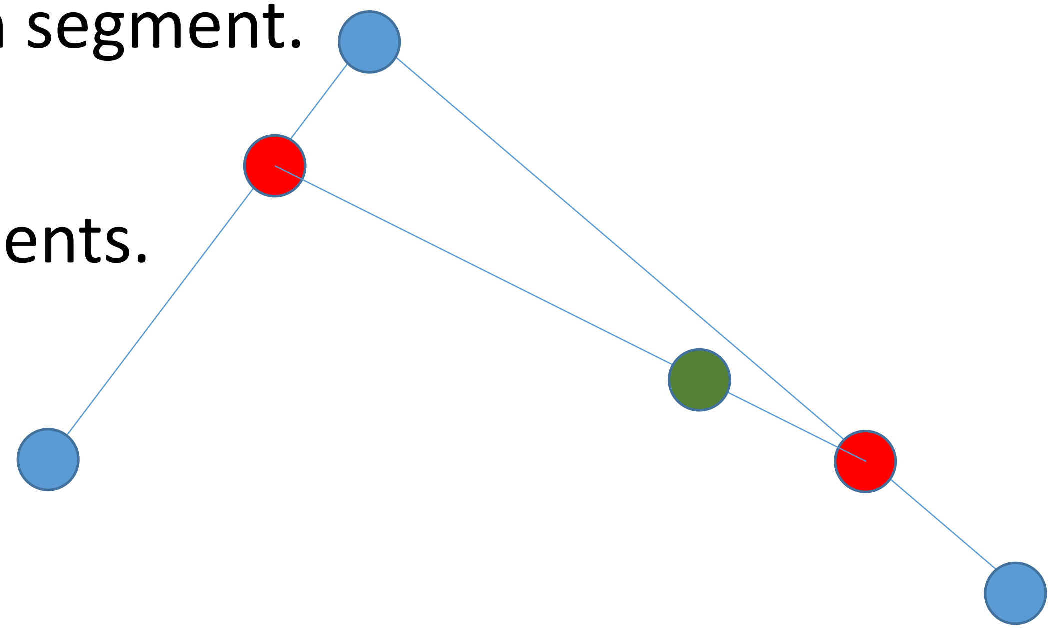
The recurrence relation of Bernstein polynomials suggests an interesting way to build up Bezier curves.

Compute linear interpolant at along each segment.

Draw segment(s) between those points.

Linearly interpolate points on those segments.

Repeat to the desired degree…



Try This Interactive Example / Visualization:
http://www.jasondavies.com/animated-bezier/

Bezier Curves - Usage

Sequences of Bezier curves are usually connected to produce a piecewise continuous curve.

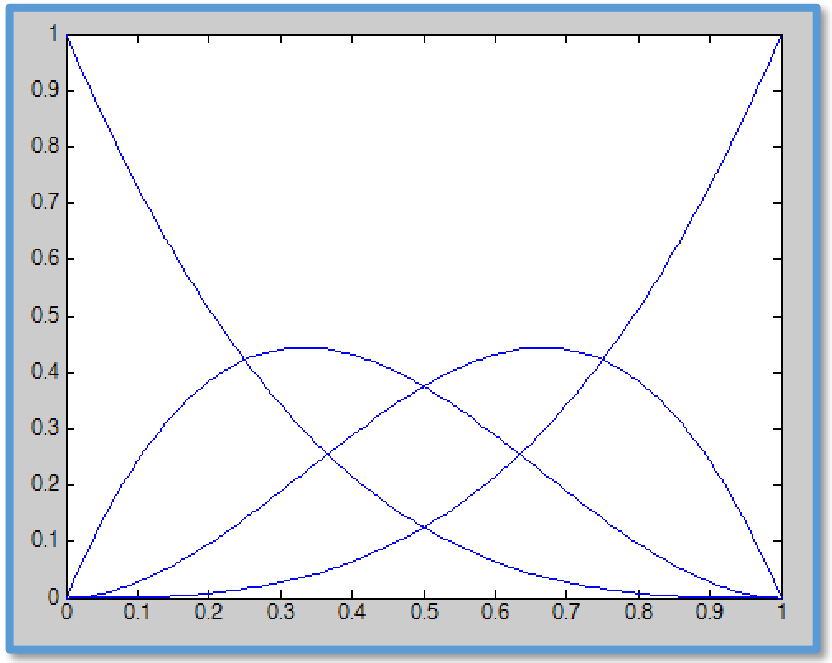
e.g., Powerpoint curves are (cubic) Bezier curves.

The “handles” used for local manipulation are just the 2nd and 3rd control points of the (4-point) Bezier curve.

Bezier Curves – Locality of Edits

Within a single Bezier curve, moving a control point tends to "mostly" affect the region of the curve closest to it.

Since each Bernstein polynomial has a local maximum at $t = k/N$, the largest Influence is for points near $P\left(\frac{k}{N}\right)$.



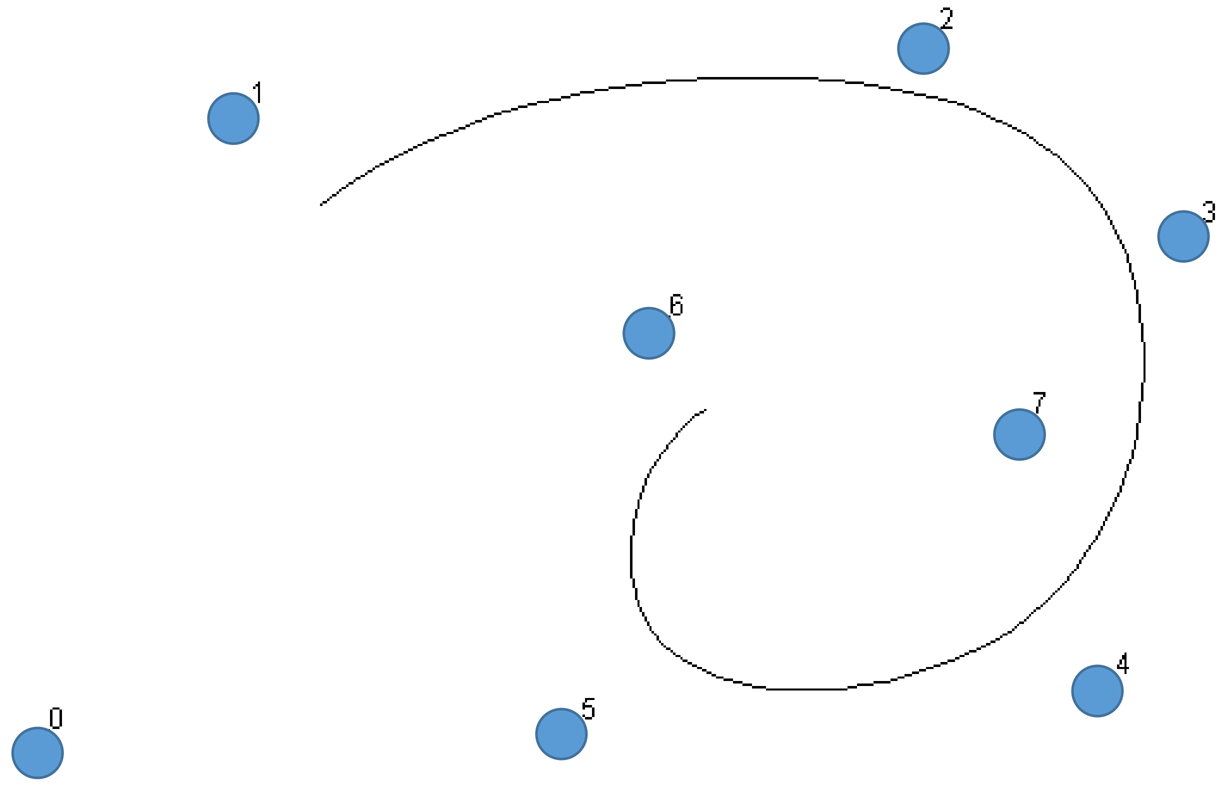
B-Spline Curves

B-spline curves are another related type of curve, with a higher degree of smoothness, compared to the Bezier curves we've seen.

But! Basic B-splines don't necessarily guarantee interpolation: the curve need not pass through (any of) the input points!

In fact, Bezier curves are a special case of B-splines.

B-Spline Curves



High degree of smoothness, but without interpolating the control points.

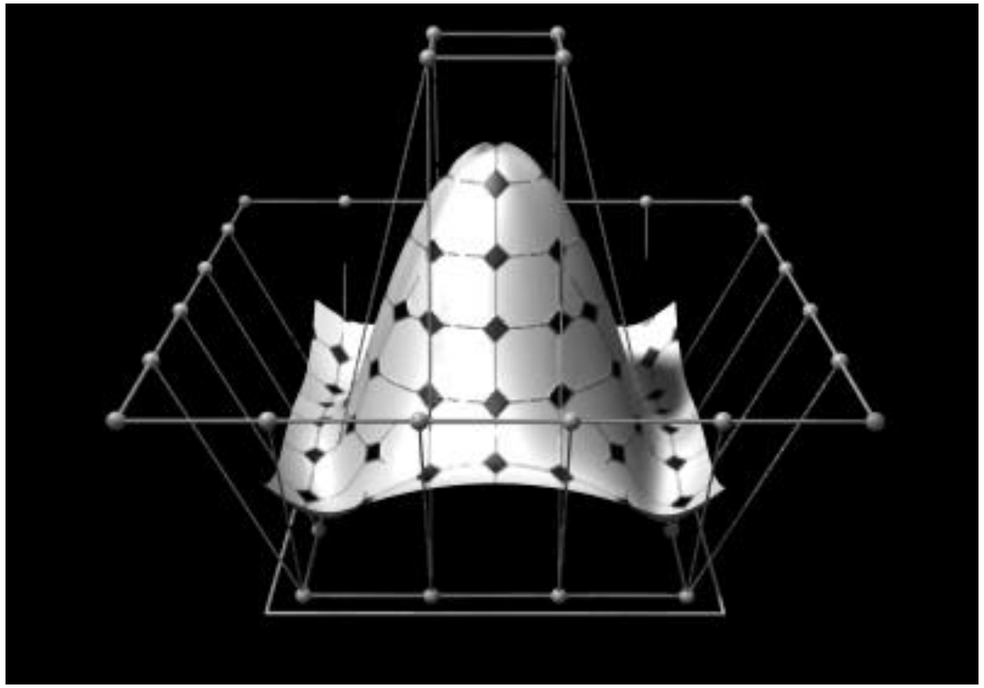
Moving from Curves to Surfaces

Smooth curve ideas can also be generalized to *surfaces* in 3D.

Common surface types in computer-aided design (CAD) and graphics:

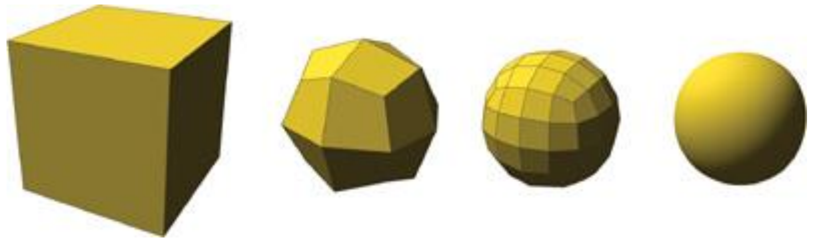
1) NURBS ("non-uniform rational B-splines").

2) Subdivision surfaces.



A NURB surface, with its control mesh.

Subdivision Surfaces



An initial cube mesh, subdivided recursively to form a smooth “subdivision surface”.

Geri from the 1997 Pixar short “Geri’s Game”.
It was among the first films to use subdivision surfaces.

Jos Stam, from Toronto’s Autodesk, received a scientific and technical Academy Award for work on this topic!

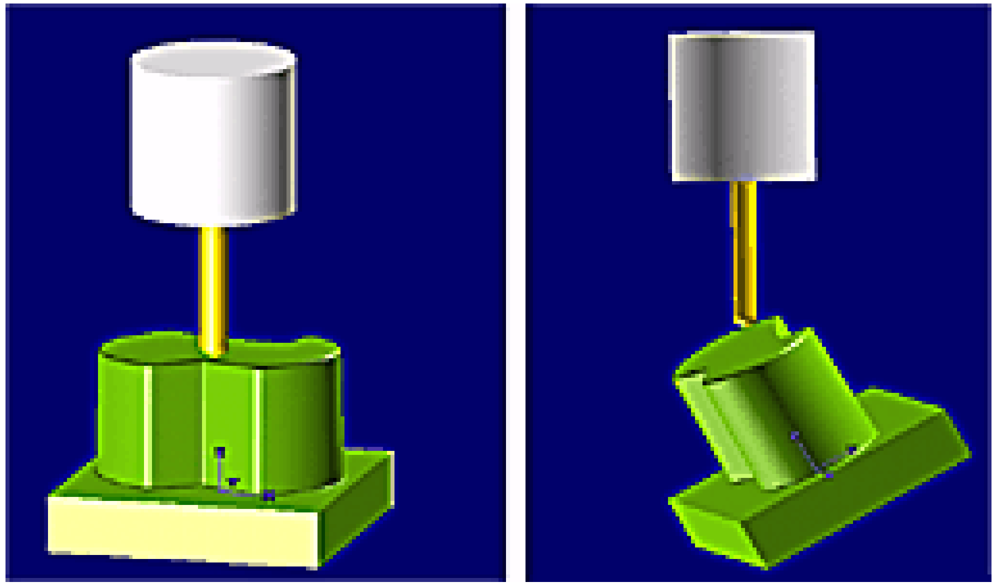
Unit Recap – Interpolation and Splines

Core concepts:

- Fitting basic polynomials via Vandermonde matrix or Lagrange polys.
- Fitting *piecewise* polynomials with Hermite interpolation and cubic splines, by matching values and derivatives between intervals.
- Using Hermite interpolation formulas to compute cubic splines more efficiently.
- Using parametric curves for more general curve shapes.
- Using Bezier curves for more flexible local control, via extra control points.

Splines at Waterloo

- Waterloo’s own Prof. Steve Mann is an expert on splines and interpolation, for applications in computer aided design and 5-axis machining, among others.
- Take the grad course “CS 779: Splines” to learn (**lots**) more!
- There’s also a nice interactive online book (not from Waterloo): https://pomax.github.io/bezierinfo/



Next Time: Intro to Ordinary Differential Equations

A common use of numerical methods for differential equations is simulating physics, for applications in physics, engineering, biology, computer animation, and more.

[Zheng and James, 2011]
http://www.cs.cornell.edu/projects/Sound/mc/