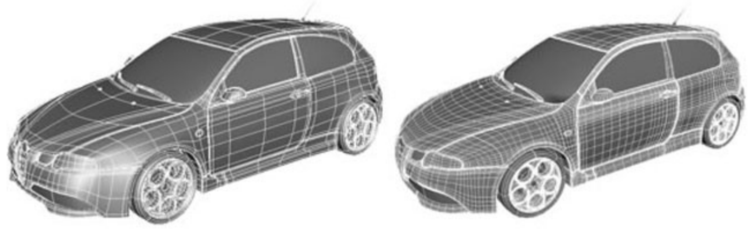
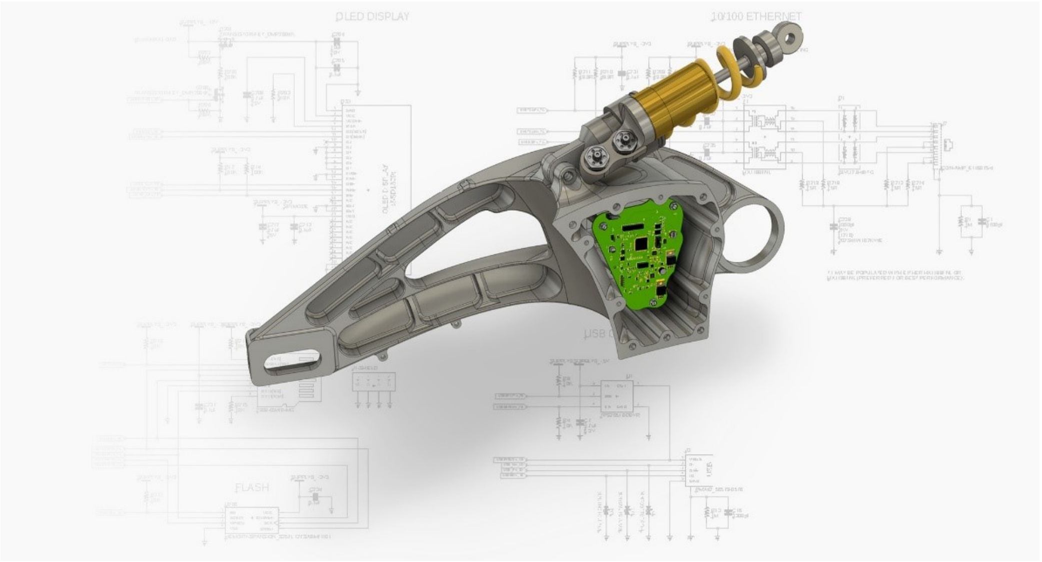
2D Spline Curves

Motivation: smoothness

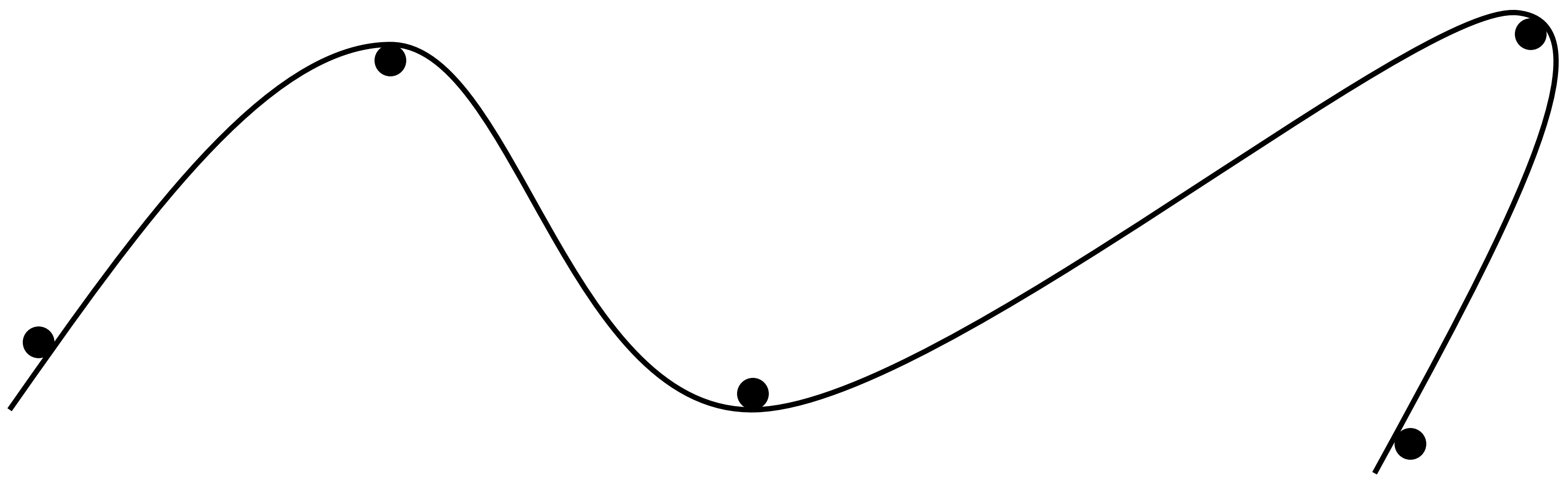
- In many applications we need smooth shapes
  - that is, without discontinuities
- More complicated than simple shapes…
  - things with corners (lines, squares, rectangles, …)
  - circles and ellipses (only get you so far!)





Classical approach

- Pencil-and-paper draftsmen also needed smooth curves
- Origin of “spline:” strip of flexible metal
  - held in place by pegs or weights to constrain shape
  - traced to produce smooth contour



Translating into usable math

- Smoothness
  - in drafting spline, comes from physical curvature minimization
  - in CG spline, comes from choosing smooth functions
    - usually low-order polynomials
- Control
  - in drafting spline, comes from fixed pegs
  - in CG spline, comes from user-specified *control points*

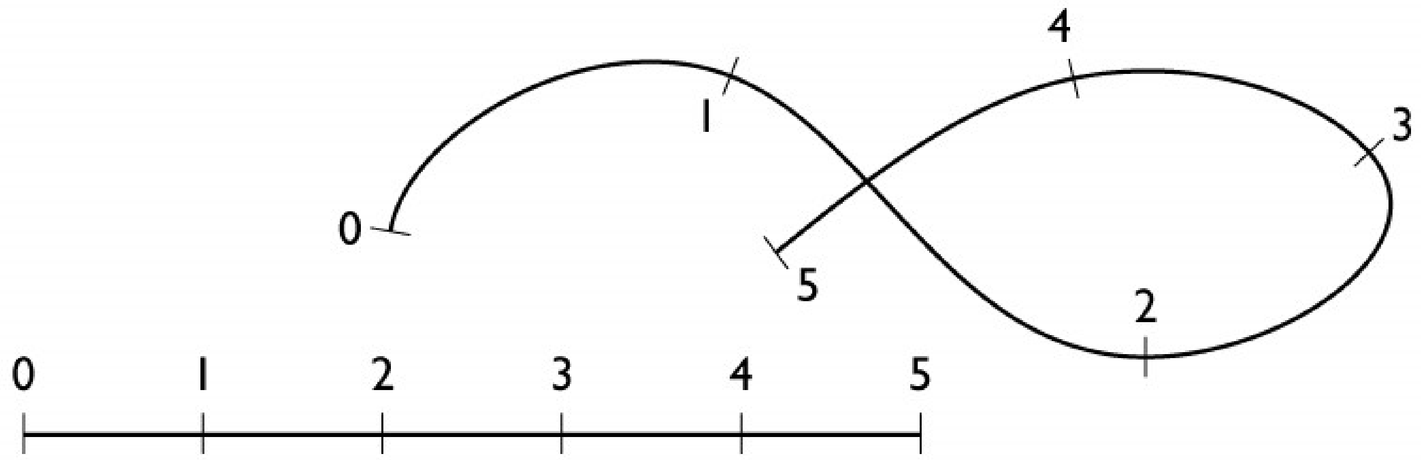
Defining spline curves

- At the most general they are parametric curves

$$S = \{\mathbf{p}(t)\,|\,t \in [0, N]\}$$

- Generally *f*(*t*) is a piecewise polynomial
    - For this course, the *discontinuities** are at the integers

      * i.e., discontinuity of the curve at k^{th} derivative for some k



Defining spline curves

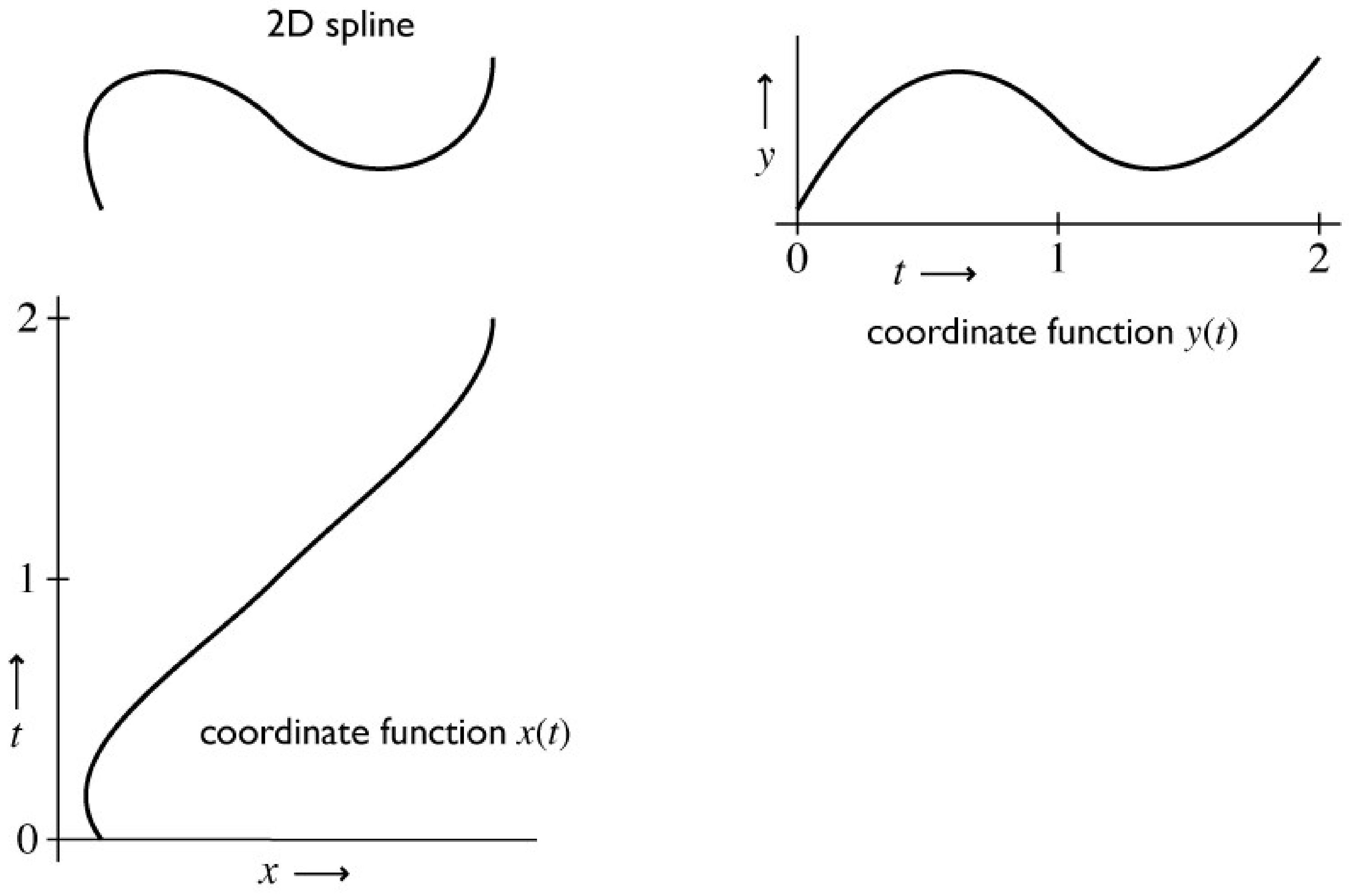
- Generally $f(t)$ is a piecewise polynomial
  - For this course, discontinuities are at the integers (but need not be in general)
  - e.g., a cubic spline has the following form over $[k, k + 1]$

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x$$

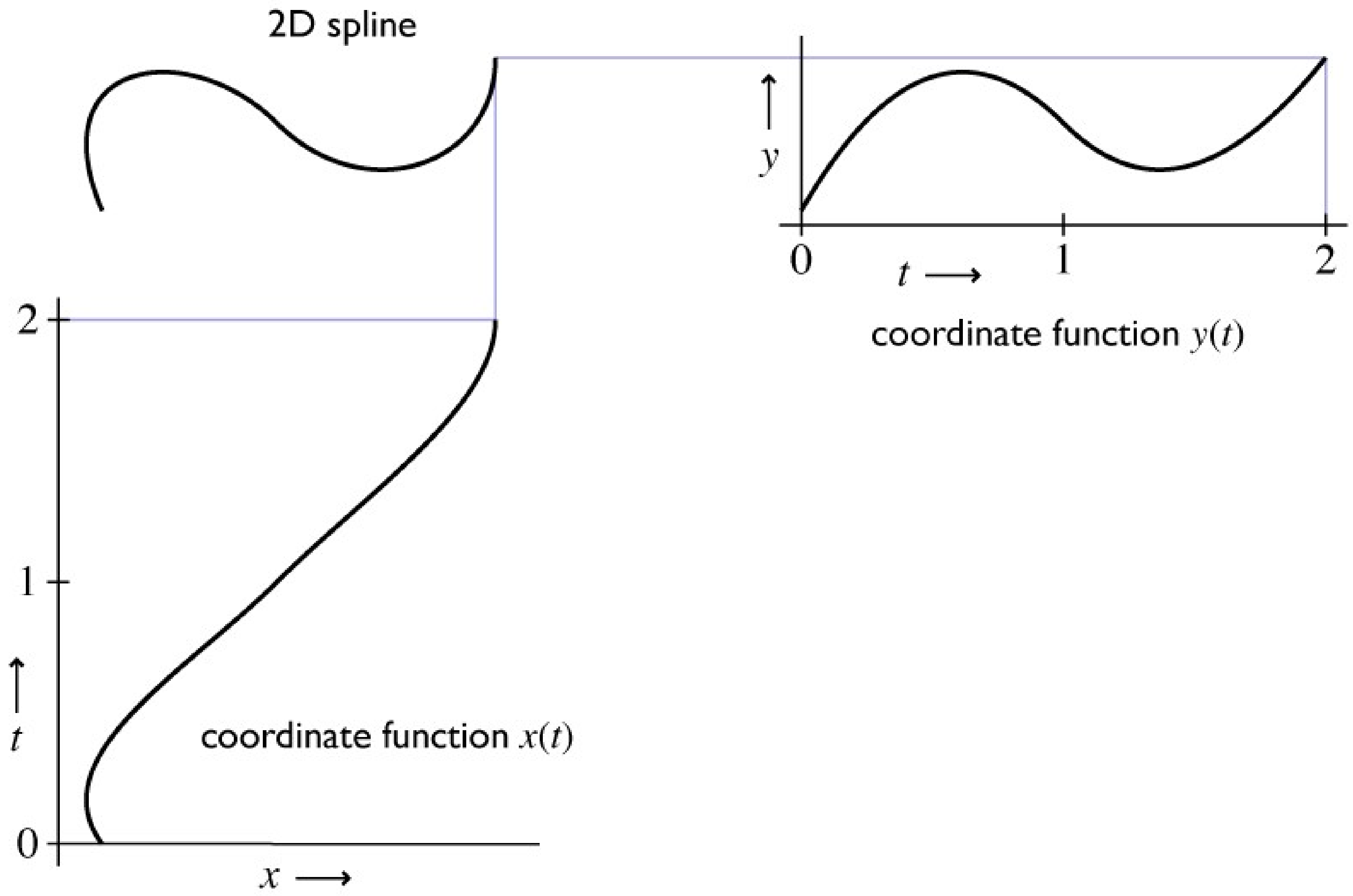
$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y$$

  - Coefficients are different for every interval

Coordinate functions

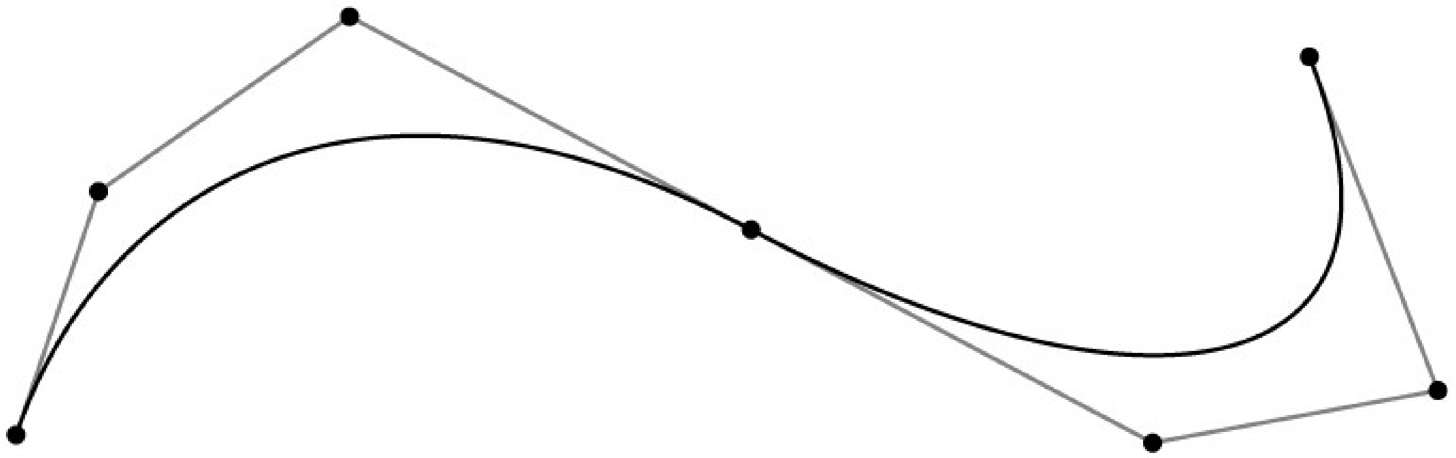


Coordinate functions



Control of spline curves

- Specified by a sequence of control points
- Shape is guided by control points (also called the control polygon)… many types of curves
  - interpolating: passes through points
  - approximating: merely guided by points



How splines depend on their controls

- Each coordinate is separate
  - Function $x(t)$ is determined solely by the x coordinates of the control points
  - This means 1D, 2D, 3D, … curves are all really the same
- Spline curves are **linear** functions of their controls
- Consider $x(t)$, for fixed t
  - Moving a control point 2 cm moves $x(t)$ twice as far as moving it by 1 cm
  - $x(t)$ is a linear combination (weighted sum) of the control point x coordinates
  - $\boldsymbol{p}(t)$, for fixed t, is a linear combination (weighted sum) of the control points

Trivial example: piecewise linear

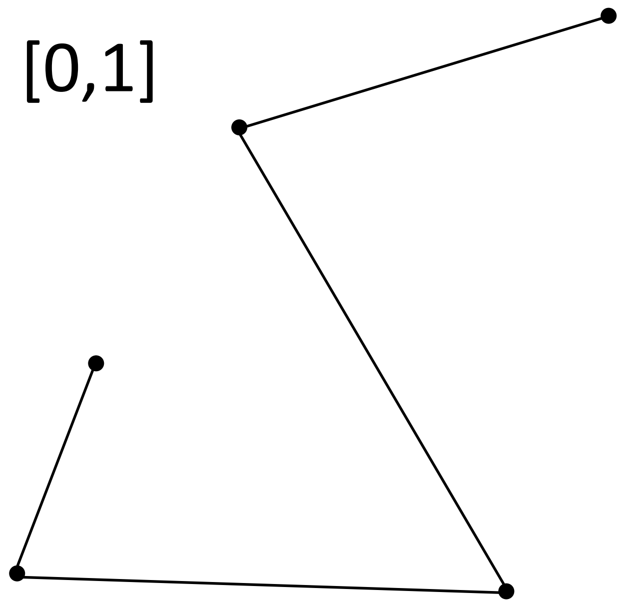
- This spline is just a polygon
  - Control points are the vertices
- But we can derive it anyway as an illustration
- Each interval will be a linear function of t in [0,1]

$$x(t) = at + b$$

  - Constraints are values at endpoints

$$b = x_0 ; a = x_1 - x_0$$

  - This is linear interpolation



Trivial example: piecewise linear

- Vector formulation

$$x(t) = (x_1 - x_0)t + x_0$$
$$x(t) = (y_1 - y_0)t + y_0$$
$$\boldsymbol{p}(t) = (\boldsymbol{p}_1 - \boldsymbol{p}_0)t + \boldsymbol{p}_{\mathbf{0}}$$

- Matrix formulation

$$\boldsymbol{p}(t) = \left(\begin{bmatrix} t & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}\right) \begin{bmatrix} \boldsymbol{p}_0 \\ \boldsymbol{p}_1 \end{bmatrix}$$

- ***Understand this abuse of notation in the matrix formulation!***

Trivial example: piecewise linear

- Basis function formulation
  - regroup expression by $\boldsymbol{p}$ rather than t

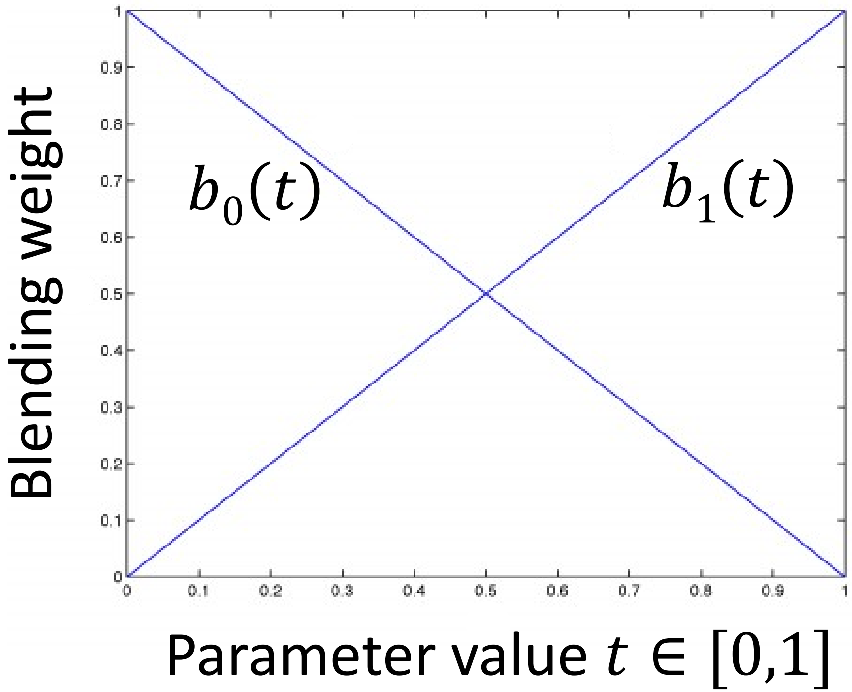
$$\begin{aligned}\boldsymbol{p}(t) &= (\boldsymbol{p}_1 - \boldsymbol{p}_0)t + \boldsymbol{p}_\mathbf{0} \\ &= (1-t)\boldsymbol{p}_0 + t\boldsymbol{p}_1\end{aligned}$$

  - interpretation in matrix viewpoint

$$\boldsymbol{p}(t) = \left(\underbrace{\begin{bmatrix} t & 1 \end{bmatrix}}_{\text{Power basis}} \underbrace{\begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}}_{\text{Change of basis}} \right) \underbrace{\begin{bmatrix} \boldsymbol{p}_0 \\ \boldsymbol{p}_1 \end{bmatrix}}_{\text{Geometry}}$$

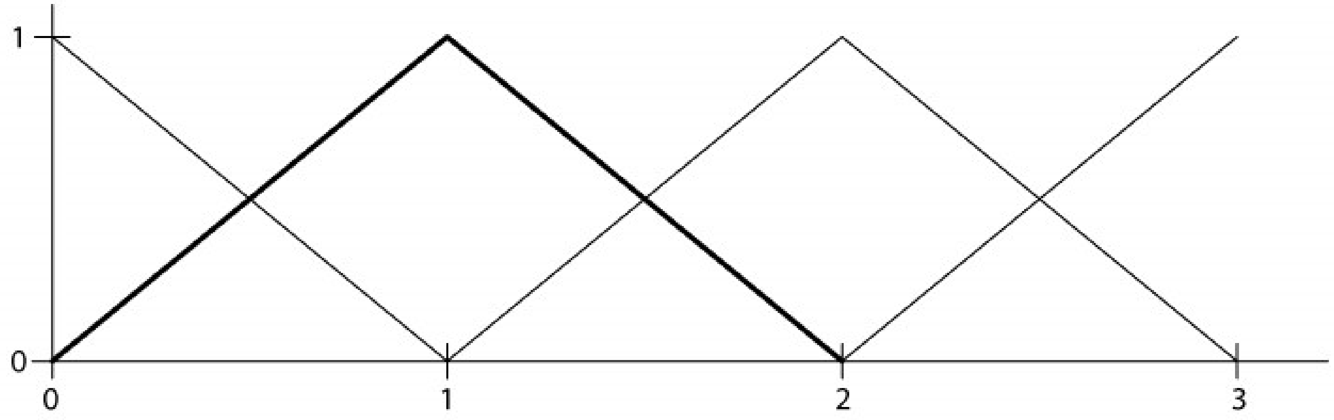
Trivial example: piecewise linear

- Vector blending formulation: “average of points”
  - blending functions: contribution of each point as t changes



Trivial example: piecewise linear

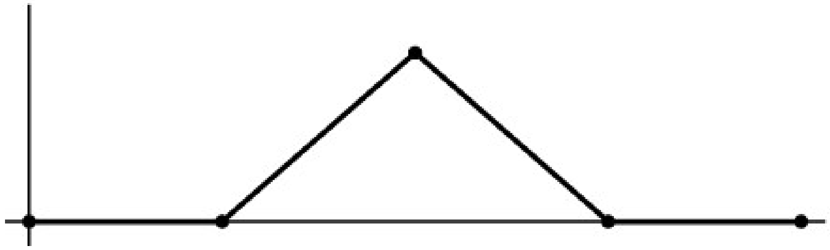
- Basis function formulation: "function times point"
  - Basis functions: contribution of each point as *t* changes



  - Can think of them as blending functions glued together
  - This is just like a reconstruction filter!

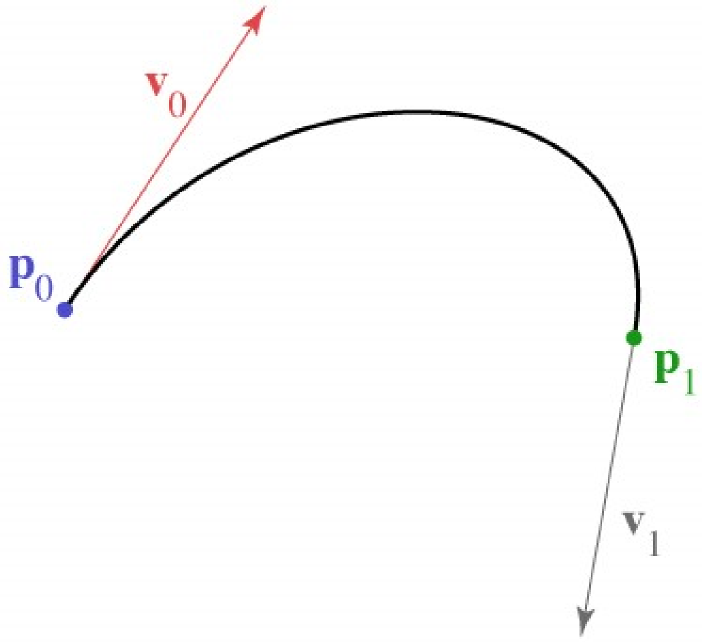
Seeing the basis functions

- Basis functions of a spline are revealed by how the curve changes in response to a change in one control
  - to get a graph of the basis function, start with the curve laid out in a straight, constant-speed line
    - what are $x(t)$ and $y(t)$?
  - then move one control straight up



Hermite splines

- Less trivial example
- Form of curve: piecewise cubic
- Constraints: endpoints and tangents (derivatives)



Hermite splines

- Solve constraints to find coefficients

$$x(t) = at^3 + bt^2 + ct + d$$
$$x'(t) = 3at^2 + 2bt + c$$

Cubic polynomial, and derivative

Evaluated at 0 and 1 (Hermite controls)

Power coefficients wrt Hermite controls

$$x(0) = x_0 = d$$
$$x(1) = x_1 = a + b + c + d$$
$$x'(0) = x'_0 = c$$
$$x'(1) = x'_1 = 3a + 2b + c$$

$$d = x_0$$
$$c = x'_0$$
$$a = 2x_0 - 2x_1 + x'_0 + x'_1$$
$$b = -3x_0 + 3x_1 - 2x'_0 - x'_1$$

$$M = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}$$

Matrix M converts coefficients $[a\ b\ c\ d]^T$ of the power basis $[t^3\ t^2\ t\ 1]$ to Hermite basis controls. Inverting this matrix provides the solution for the power basis coefficients above.

Hermite splines

- Matrix form is much simpler

$$\boldsymbol{p}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_0 \\ \boldsymbol{p}_1 \\ \boldsymbol{v}_0 \\ \boldsymbol{v}_1 \end{bmatrix}$$

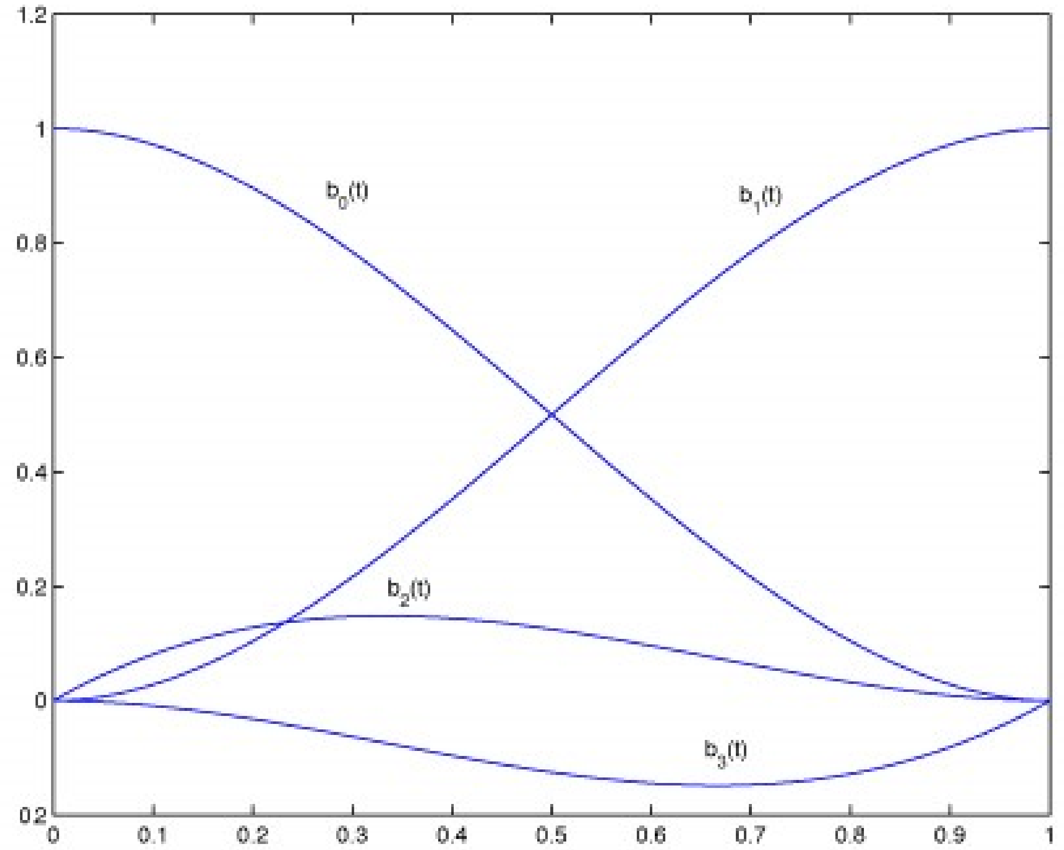
  - Coefficients are rows
  - Basis functions are columns
    - Basis functions that weight points produce an **affine combination** of points
    - Basis functions that weight vectors sum to zero, i.e., an **interpolation** of vectors

Longer Hermite splines

- Can only do so much with one Hermite spline
- Can use these splines as segments of a longer curve
  - curve from t = 0 to t = 1 defined by first segment
  - curve from t = 1 to t = 2 defined by second segment
- To avoid discontinuity, match derivatives at junctions
  - this produces a C^1 curve

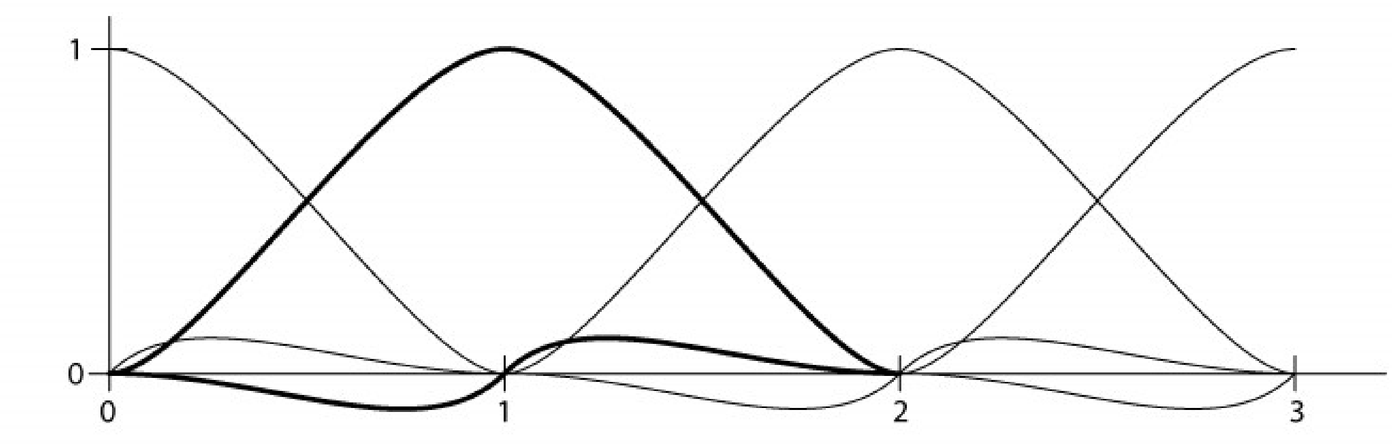
Hermite splines

- Hermite blending functions



Hermite splines

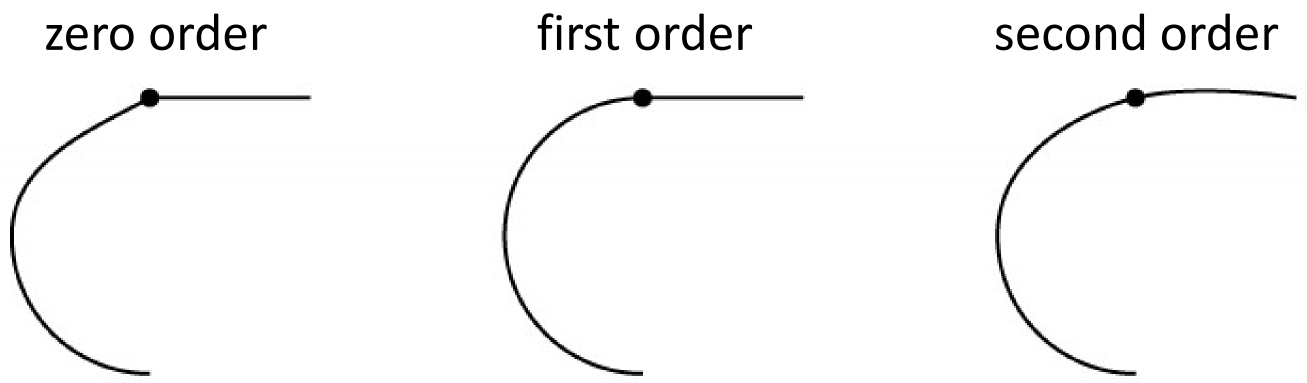
- Hermite basis functions



Continuity

- Smoothness is described by degree of continuity
  - Parametric continuity
    - C^0 continuity: position matches
    - C^1 continuity: first derivatives match
    - C^2 continuity: second derivatives match
  - Geometric continuity
    - zero-order (G^0): position matches from both sides
    - first-order (G^1): tangent matches from both sides
    - second-order (G^2): curvature matches from both sides

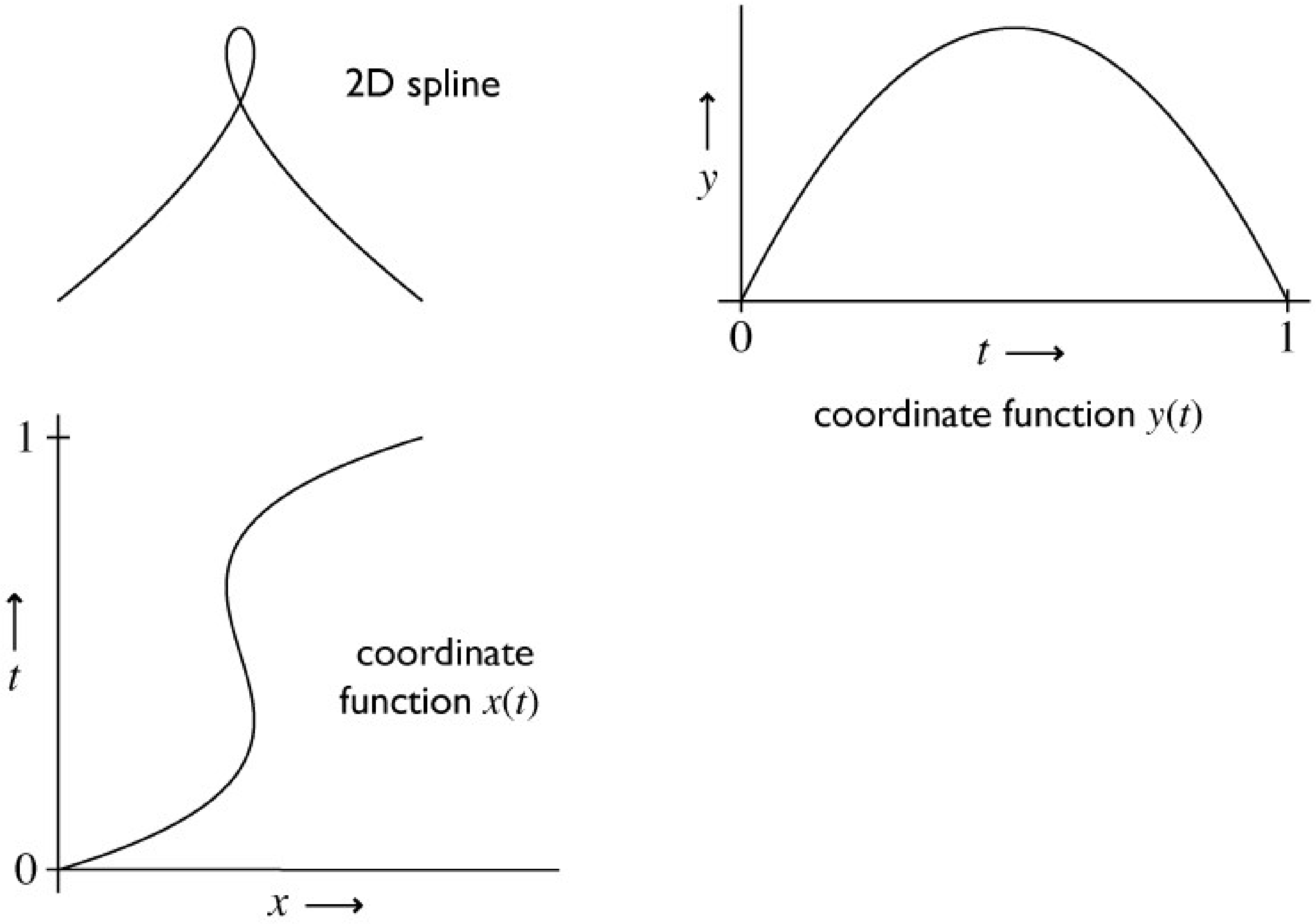
Continuity



Continuity

- Parametric continuity (C) of spline is continuity of coordinate functions
- Geometric continuity (G) is continuity of the curve itself
- ***Neither form of continuity is guaranteed by the other***
  - Can be C1 but not G1 when $\boldsymbol{p}(t)$ comes to a halt (next slide)
  - Can be G1 but not C1 when the tangent vector changes length abruptly

Geometric vs. parametric continuity

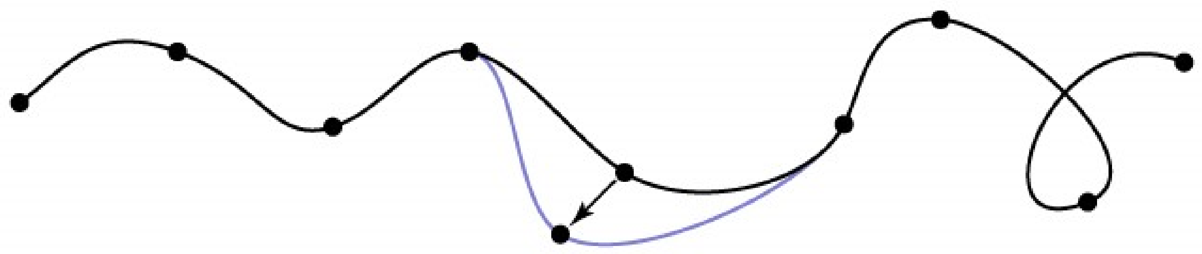


Nice properties of splines…

- Local control
- Convex hull property
- Affine invariance

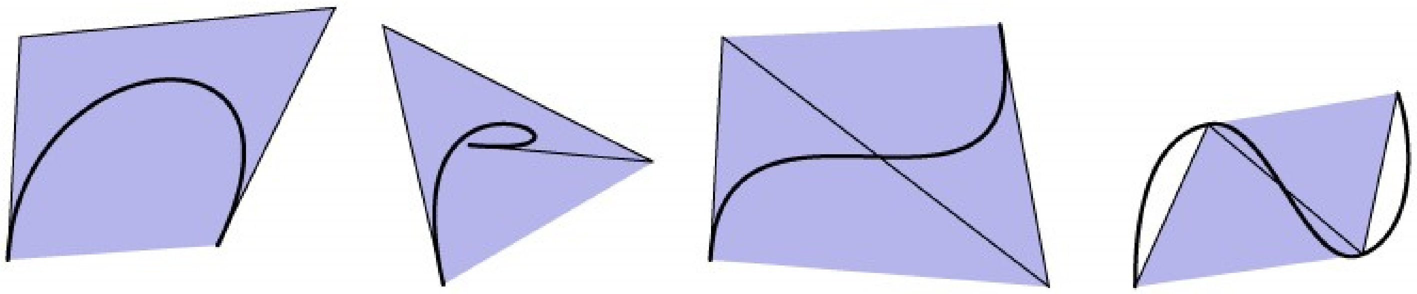
Control

- Local control
  - changing control point only affects a limited part of spline
  - without this, splines are very difficult to use
  - many formulations lack this property
    - natural spline
    - polynomial fits



Control

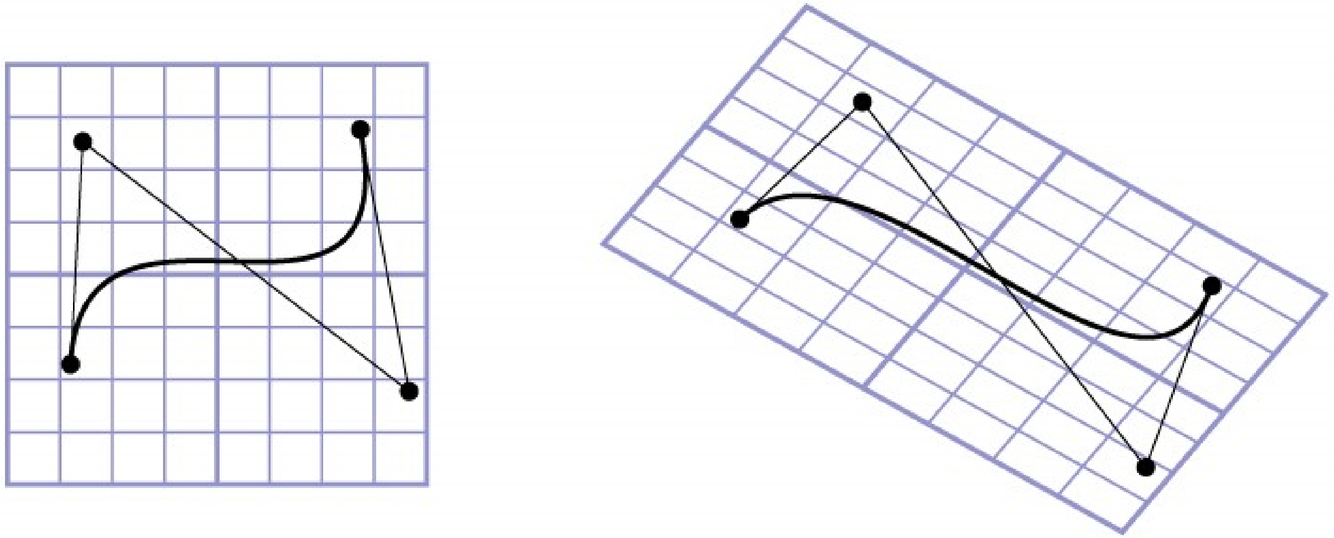
- Convex hull property
  - convex hull = smallest convex region containing points
    - think of a rubber band around some pins
  - some splines stay inside convex hull of control points
    - make clipping, culling, picking, etc. simpler



YES YES YES NO

Affine invariance

- Transforming the control points is the same as transforming the curve
  - true for all commonly used splines
  - extremely convenient in practice…



Matrix form of spline

$$\mathbf{p}(t) = \mathbf{a}t^3 + \mathbf{b}t^2 + \mathbf{c}t + \mathbf{d}$$

$$\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$$

$$\mathbf{p}(t) = b_0(t)\mathbf{p}_0 + b_1(t)\mathbf{p}_1 + b_2(t)\mathbf{p}_2 + b_3(t)\mathbf{p}_3$$

Bézier matrix

$$\boldsymbol{p}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_0 \\ \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \end{bmatrix}$$

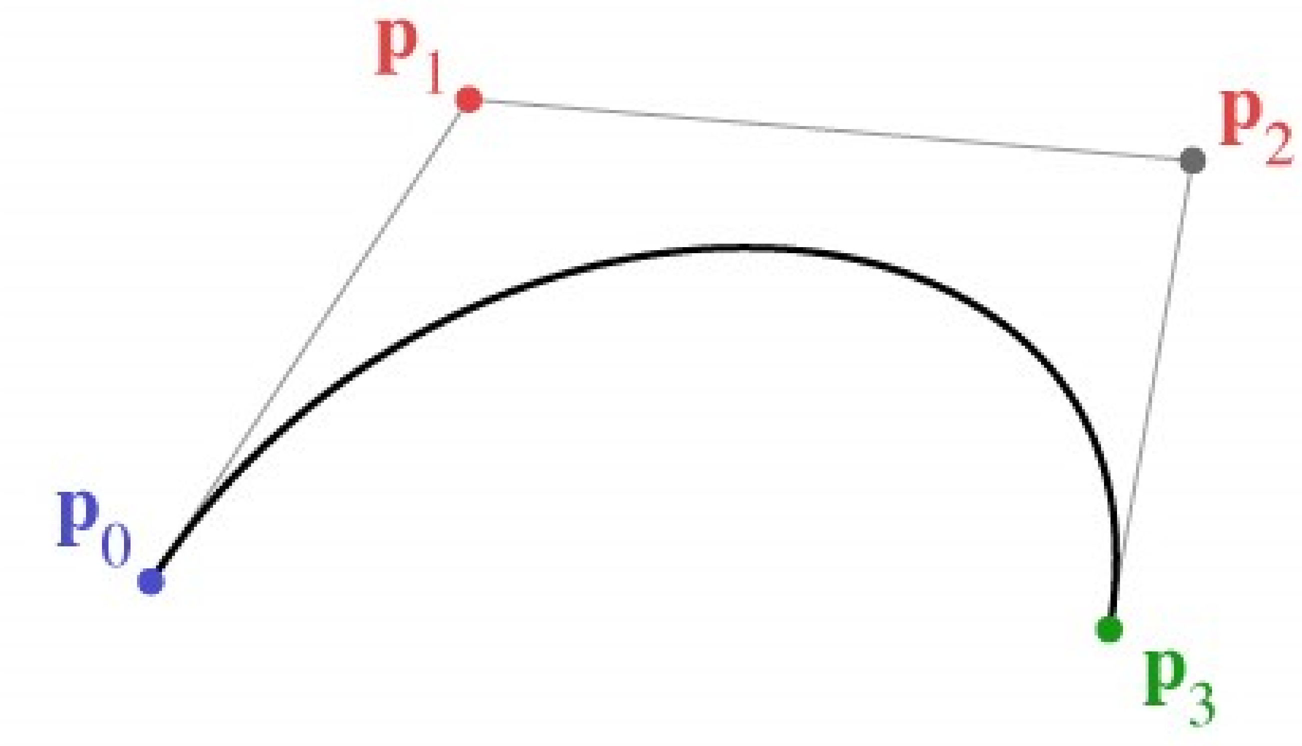
- Note that these are the Bernstein polynomials (a division of unity, more soon)

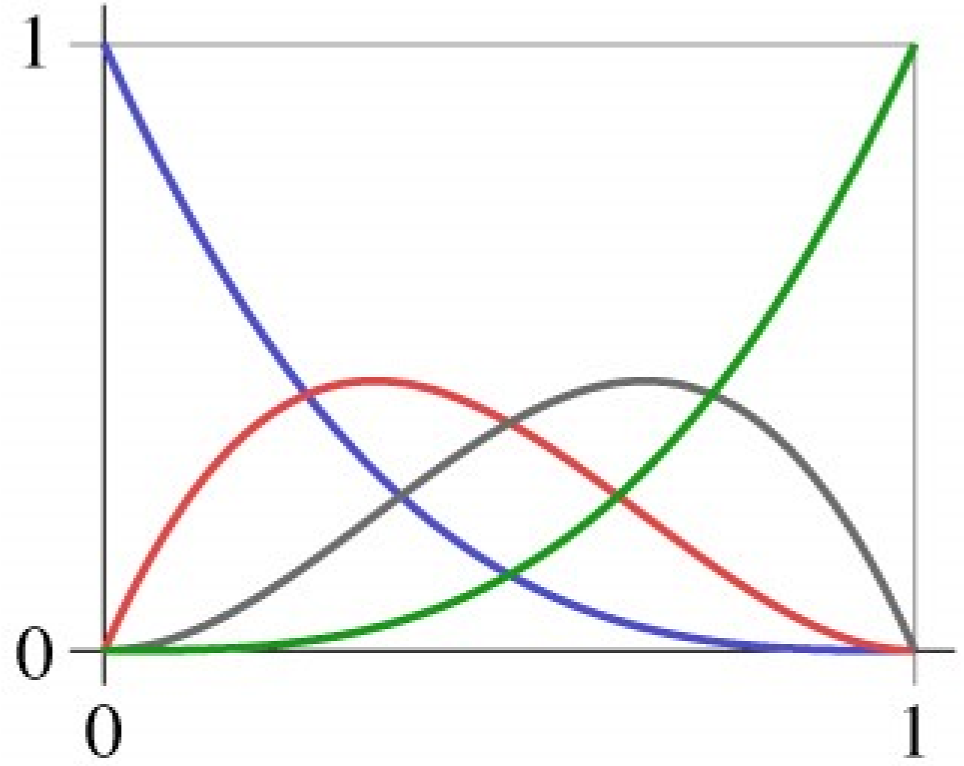
$$Bin(t) = C(n,i)\, t^i (1 - t)^{n-i}$$

where $C(n,i)$ is n choose i, $\binom{n}{i} = \frac{n!}{i!(n-i)!}$

- This defines Bézier curves for any degree

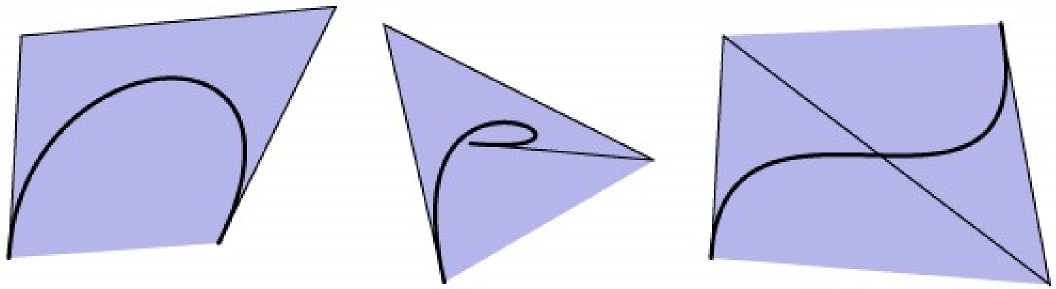
Bézier basis





Convex hull

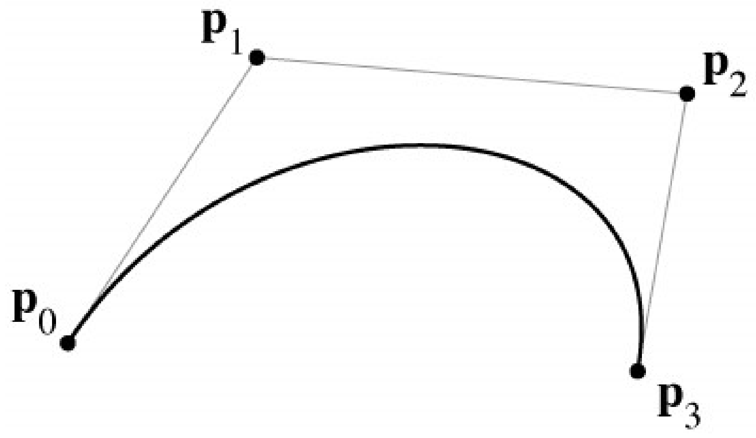
- Spline has the ***convex hull property i***f basis functions are all positive
  - Basis functions must also still be an ***affine combination***, i.e., sum to 1



  - If any basis function is ever negative, no convex hull property
    - proof: take the other three points at the same place

Hermite to cubic Bézier

- Mixture of points and vectors might be awkward, e.g., does not let us use the convex hull property
- Specify tangents as differences of points



- Note derivative for cubic Bezier is 3 times the offset
- Can easily prove by taking the derivative (try this for a degree n Bezier curve!)

Bézier Curves

- See Bernstein basis as binomial expansion (i.e., division of unity)

$$1 = (1 - t) + t$$

$$1^n = \left((1 - t) + t\right)^n$$

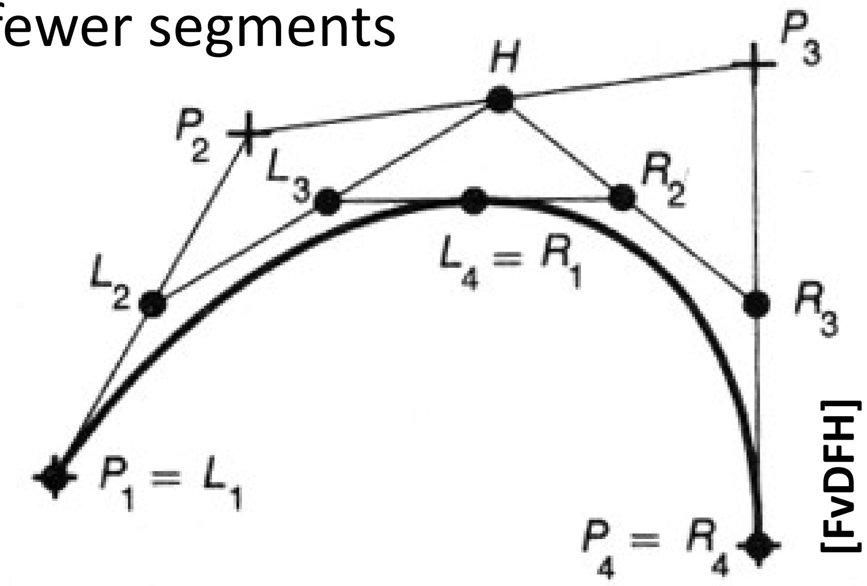
$$1 = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} \quad \text{or} \quad \sum_{i=0}^{n} B_{in}(t)$$

- Bernstein basis funcs $B_{in}(t)$ have nice properties
  - Convex combination: $B_{in}(t) \geq 0$ for $t \in [0,1]$
  - Symmetry: $B_{in}(1-t) = B_{(n-i)n}(t)$
  - Derivatives: $\frac{\mathrm{d}}{\mathrm{dt}} B_{in}(t) = n\left(B_{(i-1)(n-1)}(t) - B_{i(n-1)}(t)\right)$
  - And many other properties!

Let $B_{(-1)n}(t) = 0$
and $B_{n(n-1)}(t) = 0$

Bézier curve subdivision: Casteljau's algorithm

- Splits a Bézier spline segment into a two-segment curve
  1. Subdivide control polygon with ratio $t : (1 - t)$
  2. Connect these points to get control poly with 1 fewer segments
  3. Repeat until you have only a point
  4. Intermediate points generated form the new control polygons
- Often used with $t = 0.5$
- Can also be achieved with a change of basis, consider $t = 0.5s$ and $t = 0.5 + 0.5s$
- Permits local refinement, divide and conquer algorithms
  - Intersection of ray with curve or surface?



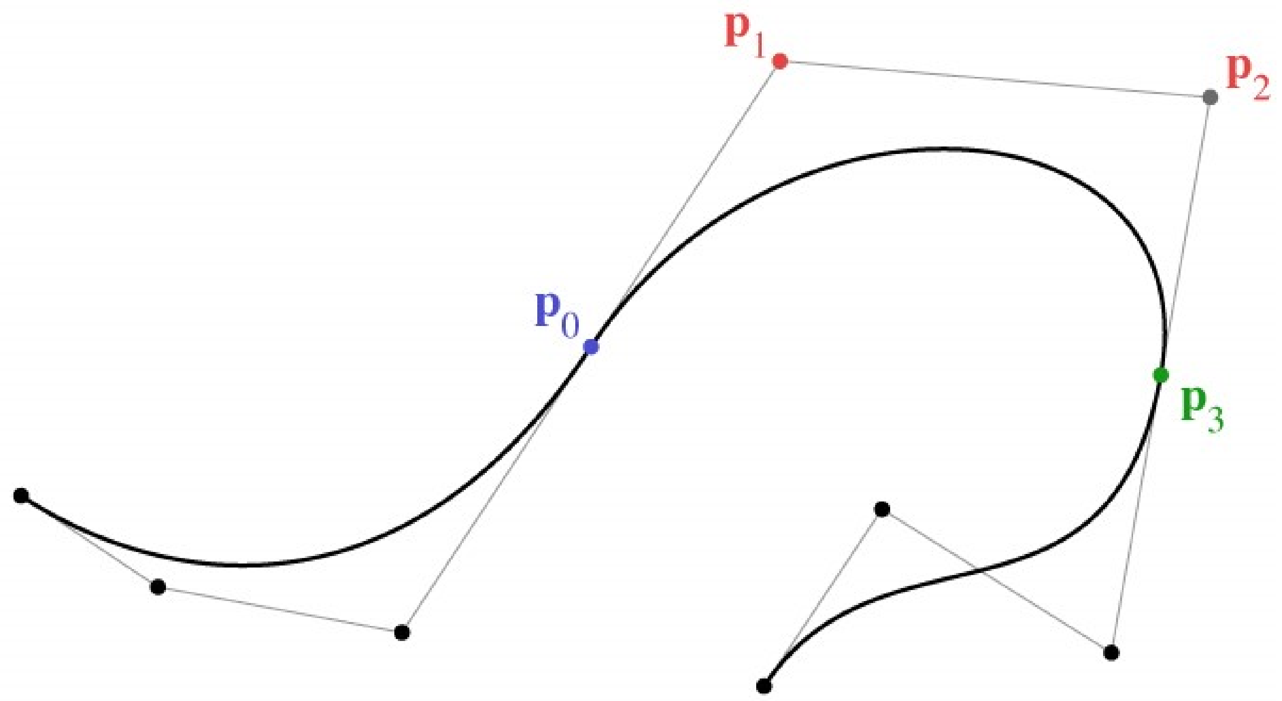
[FvDFH]

Chaining spline segments

- Hermite curves are convenient because they can be made long easily
- Bézier curves are convenient because their controls are all points and they have nice properties
  - For cubic Bézier, they interpolate every 4th point, which is a little strange
  - (Note that we are not covering B splines)
- Can get Hermite from Bézier by defining tangents from control points
  - A similar construction leads to the interpolating Catmull-Rom spline

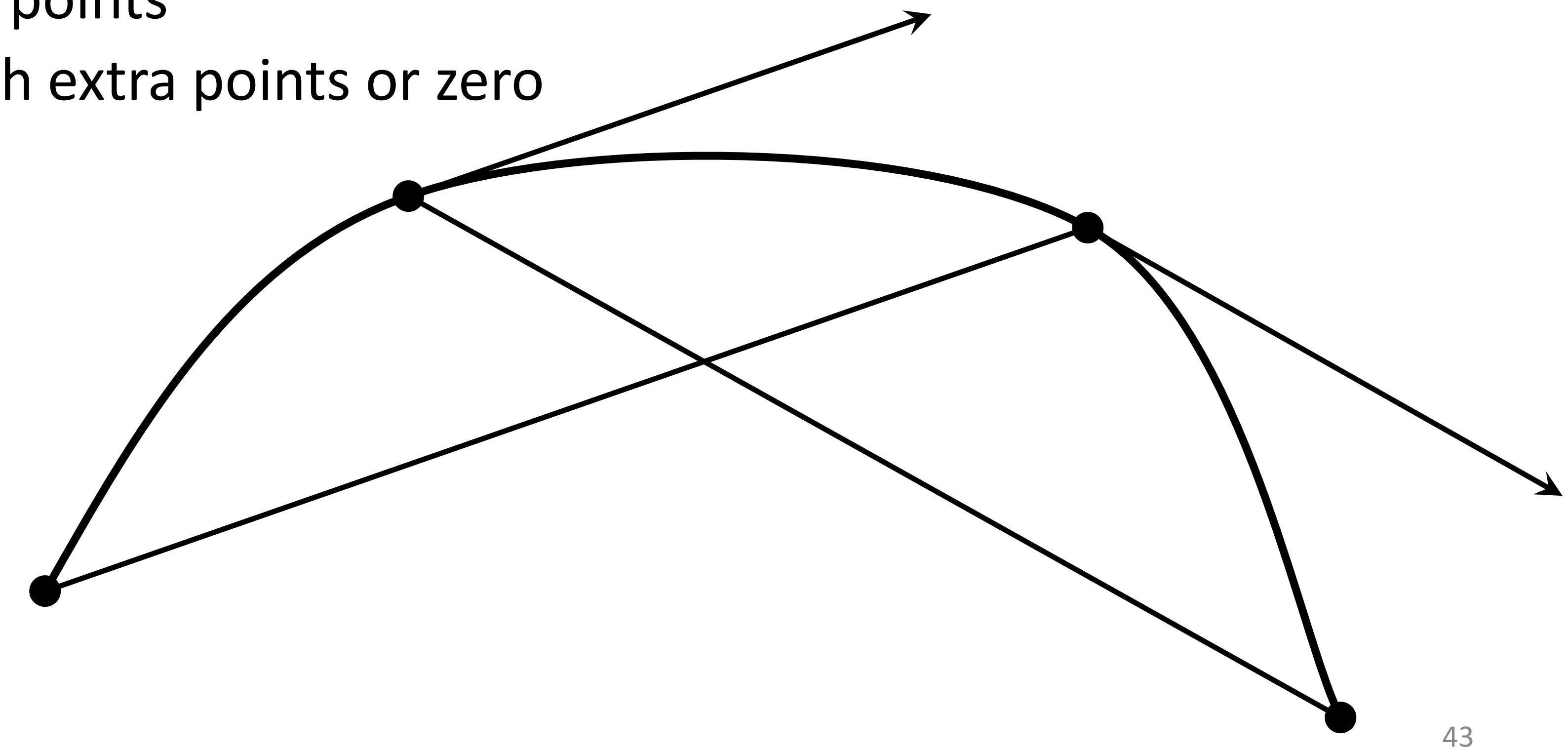
Chaining Bézier splines

- No continuity built in
- Achieve G1 using collinear control points (with care)

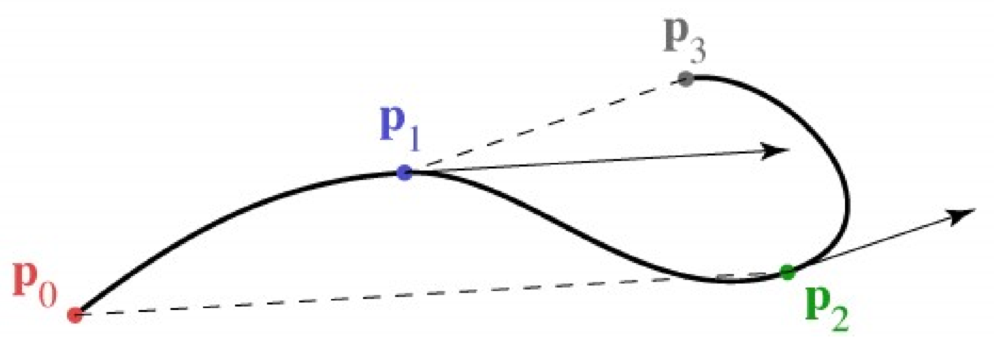


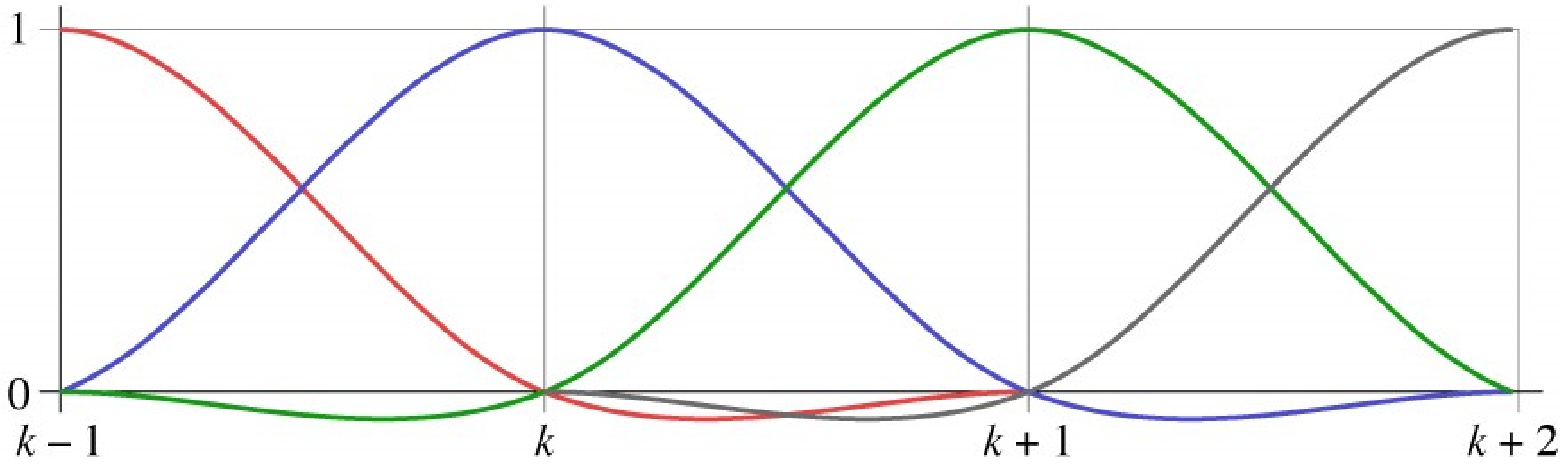
Hermite from Catmull-Rom

- Have not yet seen any interpolating splines
- Would like to define tangents automatically
  - use adjacent control points
  - Set end tangents with extra points or zero



Catmull-Rom basis





Catmull-Rom splines

- Our first example of an interpolating spline
- Like Bézier, equivalent to Hermite
    - All splines of this form are equivalent
- First example of a spline based on just a control point sequence
- Does not have convex hull property

Cubic Bézier splines

- Very widely used type, especially in 2D
  - e.g., it is a primitive in PostScript/PDF
- Can represent C^1 and/or G^1 curves with corners
- Can easily add points at any position

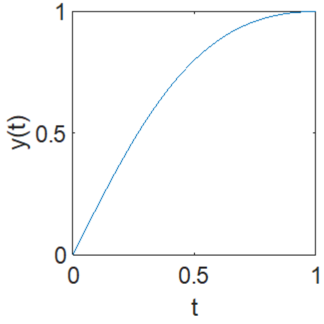
Interpolating points

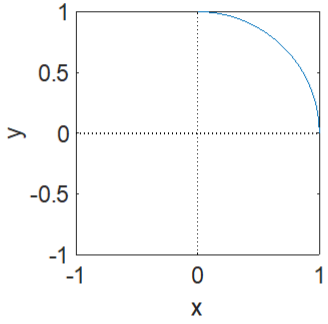
- What about just solving for basis that gives p_0 p_1 p_2 p_3 at $t = 0,1,2,3$?
- Can you easily explain why this different from Catmull-Rom?

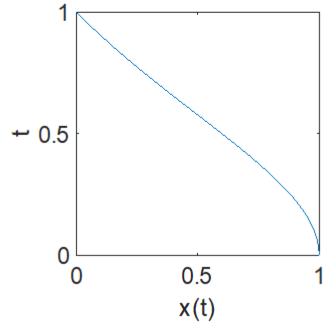
Rational Curves

- Can not represent a circle with a polynomial (consider the derivatives)
- Can with rational polynomials, A circular arc for t in $[0,1]$

$$x(t) = \frac{1 - t^2}{1 + t^2} \qquad y(t) = \frac{2t}{1 + t^2}$$







  - Can see equivalence with substitution of $t = \tan(u/2)$ and application of trig identity $\tan^2 u = \frac{1-\cos(2u)}{1+\cos(2u)}$

- Rational Bezier degree n, control points g_i and weights w_i, has the form

$$p(t) = \frac{\sum B_{in}(t) w_i g_i}{\sum B_{in}(t) w_i}$$

Rational Curves

- Think of control points as being in homogeneous coordinates [wx,wy,wz,w]
- What control points to use with a quadratic rational curve to produce a circular arc for t in $[0,1]$

$$x(t) = \frac{1 - t^2}{1 + t^2} \quad y(t) = \frac{2t}{1 + t^2}$$

Rational Curves

- What is the tangent of a rational curve?

(a) Given the quadratic Bezier curve $p(t) = \sum_{i=0}^{2} B_{i,2}(t) g_i$, compute an expression for the point at the center of the curve, and its derivative (i.e., $p(t)$ and $p'(t)$ at $t = 1/2$) with respect to its control points.

(b) Suppose we have the rational quadratic Bezier curve

$$p(t) = \frac{N(t)}{D(t)} \qquad \text{where} \qquad N(t) = \sum_{i=0}^{2} B_{i,2}(t) w_i g_i \qquad D(t) = \sum_{i=0}^{2} B_{i,2}(t) w_i$$

Let $g_0 = (1,0)$, $g_1 = (1,1)$, $g_2 = (0,1)$ and $w_0 = 1$, $w_1 = 1$, $w_2 = 2$ be the nonhomogeneous control points and weights of this rational quadratic Bezier curve.

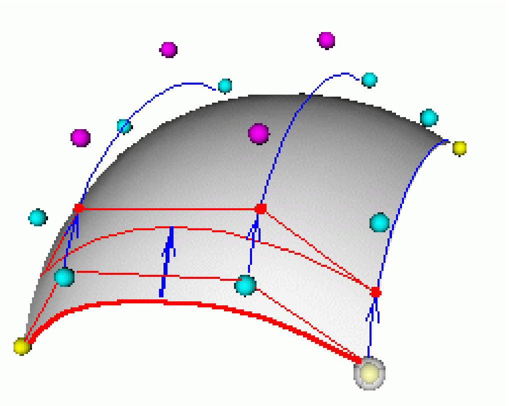
Use the formula from the previous part of this question to compute the value and derivative of the numerator and the denominator at the center of the curve, that is, $N(t), N'(t), D(t), D'(t)$, for $t = 1/2$. Keep your answer in the form of a reduced fraction.

3D spline surfaces

Tensor Product Patch

- Can make surfaces by using a grid of control points
  - In the example below, you can think of it as evaluating 4 curves in one direction to find the control points of a curve in the other direction.

$$\boldsymbol{s}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{in}(u)\, B_{jm}(v)\, \boldsymbol{P}_{ij} = \sum_{i=0}^{n} B_{in}(u) \sum_{j=0}^{m} B_{jm}(v)\, \boldsymbol{P}_{ij}$$



Bi-cubic Bezier patch

What is surface normal at (u,v) ?

$$\frac{\partial s}{\partial u} \times \frac{\partial s}{\partial v}$$

Can you make a sphere with Bezier tensor product patches? Why not?

Need to use rational Bezier tensor product patches

Teapot!

```
32
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
4,17,18,19,8,20,21,22,12,23,24,25,16,26,27,28
19,29,30,31,22,32,33,34,25,35,36,37,28,38,39,40
31,41,42,1,34,43,44,5,37,45,46,9,40,47,48,13
13,14,15,16,49,50,51,52,53,54,55,56,57,58,59,60
16,26,27,28,52,61,62,63,56,64,65,66,60,67,68,69
28,38,39,40,63,70,71,72,66,73,74,75,69,76,77,78
40,47,48,13,72,79,80,49,75,81,82,53,78,83,84,57
57,58,59,60,85,86,87,88,89,90,91,92,93,94,95,96
60,67,68,69,88,97,98,99,92,100,101,102,96,103,104,105
69,76,77,78,99,106,107,108,102,109,110,111,105,112,113,114
78,83,84,57,108,115,116,85,111,117,118,89,114,119,120,93
121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136
124,137,138,121,128,139,140,125,132,141,142,129,136,143,144,133
133,134,135,136,145,146,147,148,149,150,151,152,69,153,154,155
136,143,144,133,148,156,157,145,152,158,159,149,155,160,161,69
162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177
165,178,179,162,169,180,181,166,173,182,183,170,177,184,185,174
174,175,176,177,186,187,188,189,190,191,192,193,194,195,196,197
177,184,185,174,189,198,199,186,193,200,201,190,197,202,203,194
204,204,204,204,207,208,209,210,211,211,211,211,212,213,214,215
204,204,204,204,210,217,218,219,211,211,211,211,215,220,221,222
204,204,204,204,219,224,225,226,211,211,211,211,222,227,228,229
204,204,204,204,226,230,231,207,211,211,211,211,229,232,233,212
212,213,214,215,234,235,236,237,238,239,240,241,242,243,244,245
215,220,221,222,237,246,247,248,241,249,250,251,245,252,253,254
222,227,228,229,248,255,256,257,251,258,259,260,254,261,262,263
229,232,233,212,257,264,265,234,260,266,267,238,263,268,269,242
270,270,270,270,279,280,281,282,275,276,277,278,271,272,273,274
270,270,270,270,282,289,290,291,278,286,287,288,274,283,284,285
270,270,270,270,291,298,299,300,288,295,296,297,285,292,293,294
270,270,270,270,300,305,306,279,297,303,304,275,294,301,302,271
```

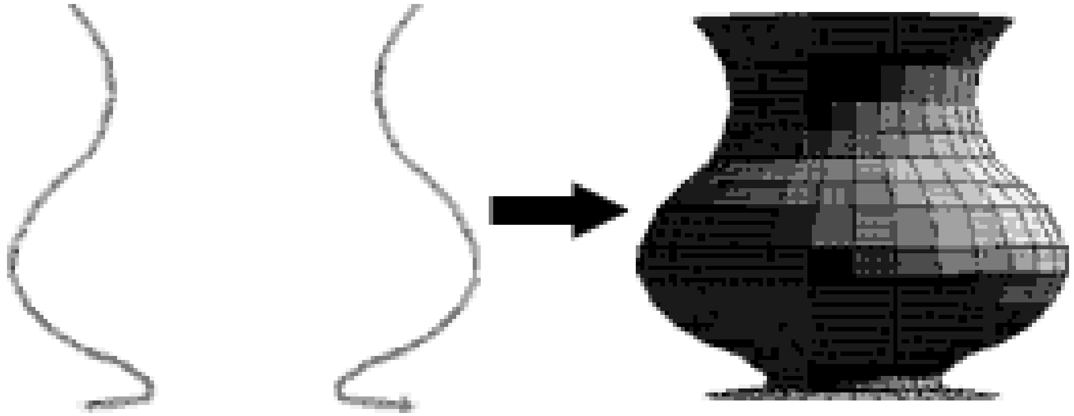
```
306
1.4,0.0,2.4
1.4,-0.784,2.4
0.784,-1.4,2.4
0.0,-1.4,2.4
1.3375,0.0,2.53125
1.3375,-0.749,2.53125
0.749,-1.3375,2.53125
0.0,-1.3375,2.53125
1.4375,0.0,2.53125
1.4375,-0.805,2.53125
0.805,-1.4375,2.53125
0.0,-1.4375,2.53125
1.5,0.0,2.4
1.5,-0.84,2.4
0.84,-1.5,2.4
0.0,-1.5,2.4
-0.784,-1.4,2.4
-1.4,-0.784,2.4
-1.4,0.0,2.4
-0.749,-1.3375,2.53125
-1.3375,-0.749,2.53125
-1.3375,0.0,2.53125
-0.805,-1.4375,2.53125
-1.4375,-0.805,2.53125
-1.4375,0.0,2.53125
-0.84,-1.5,2.4
-1.5,-0.84,2.4
-1.5,0.0,2.4
-1.4,0.784,2.4
-0.784,1.4,2.4
0.0,1.4,2.4
-1.3375,0.749,2.53125
-0.749,1.3375,2.53125
0.0,1.3375,2.53125
-1.4375,0.805,2.53125
-0.805,1.4375,2.53125
0.0,1.4375,2.53125
-1.5,0.84,2.4
-0.84,1.5,2.4
0.0,1.5,2.4
0.784,1.4,2.4
1.4,0.784,2.4
0.749,1.3375,2.53125
1.3375,0.749,2.53125
0.805,1.4375,2.53125
1.4375,0.805,2.53125
0.84,1.5,2.4
1.5,0.84,2.4
1.75,0.0,1.875
1.75,-0.98,1.875
0.98,-1.75,1.875
0.0,-1.75,1.875
2.0,0.0,1.35
2.0,-1.12,1.35
1.12,-2.0,1.35
0.0,-2.0,1.35
2.0,0.0,0.9
2.0,-1.12,0.9
1.12,-2.0,0.9
0.0,-2.0,0.9
-0.98,-1.75,1.875
-1.75,-0.98,1.875
-1.75,0.0,1.875
-1.12,-2.0,1.35
-2.0,-1.12,1.35
-2.0,0.0,1.35
-1.12,-2.0,0.9
-2.0,-1.12,0.9
-2.0,0.0,0.9
-1.75,0.98,1.875
-0.98,1.75,1.875
0.0,1.75,1.875
-2.0,1.12,1.35
-1.12,2.0,1.35
0.0,2.0,1.35
-2.0,1.12,0.9
-1.12,2.0,0.9
0.0,2.0,0.9
0.98,1.75,1.875
1.75,0.98,1.875
1.12,2.0,1.35
2.0,1.12,1.35
1.12,2.0,0.9
2.0,1.12,0.9
2.0,0.0,0.45
2.0,-1.12,0.45
1.12,-2.0,0.45
0.0,-2.0,0.45
1.5,0.0,0.225
1.5,-0.84,0.225
0.84,-1.5,0.225
0.0,-1.5,0.225
1.5,0.0,0.15
1.5,-0.84,0.15
0.84,-1.5,0.15
0.0,-1.5,0.15
-1.12,-2.0,0.45
-2.0,-1.12,0.45
-2.0,0.0,0.45
-0.84,-1.5,0.225
-1.5,-0.84,0.225
-1.5,0.0,0.225
-0.84,-1.5,0.15
-1.5,-0.84,0.15
-1.5,0.0,0.15
-2.0,1.12,0.45
-1.12,2.0,0.45
0.0,2.0,0.45
-1.5,0.84,0.225
-0.84,1.5,0.225
0.0,1.5,0.225
-1.5,0.84,0.15
-0.84,1.5,0.15
0.0,1.5,0.15
1.12,2.0,0.45
2.0,1.12,0.45
0.84,1.5,0.225
1.5,0.84,0.225
0.84,1.5,0.15
1.5,0.84,0.15
-1.6,0.0,2.025
-1.6,-0.3,2.025
-1.5,-0.3,2.25
-1.5,0.0,2.25
-2.3,0.0,2.025
-2.3,-0.3,2.025
-2.5,-0.3,2.25
-2.5,0.0,2.25
-2.7,0.0,2.025
-2.7,-0.3,2.025
-3.0,-0.3,2.25
-3.0,0.0,2.25
-2.7,0.0,1.8
-2.7,-0.3,1.8
-3.0,-0.3,1.8
-3.0,0.0,1.8
-1.5,0.3,2.25
-1.6,0.3,2.025
-2.5,0.3,2.25
-2.3,0.3,2.025
-3.0,0.3,2.25
-2.7,0.3,2.025
-3.0,0.3,1.8
-2.7,0.3,1.8
-2.7,0.0,1.575
-2.7,-0.3,1.575
-3.0,-0.3,1.35
-3.0,0.0,1.35
-2.5,0.0,1.125
-2.5,-0.3,1.125
-2.65,-0.3,0.9375
-2.65,0.0,0.9375
-2.0,-0.3,0.9
-1.9,-0.3,0.6
-1.9,0.0,0.6
-3.0,0.3,1.35
-2.7,0.3,1.575
-2.65,0.3,0.9375
-2.5,0.3,1.125
-1.9,0.3,0.6
-2.0,0.3,0.9
1.7,0.0,1.425
1.7,-0.66,1.425
1.7,-0.66,0.6
1.7,0.0,0.6
2.6,0.0,1.425
2.6,-0.66,1.425
3.1,-0.66,0.825
3.1,0.0,0.825
2.3,0.0,2.1
2.3,-0.25,2.1
2.4,-0.25,2.025
2.4,0.0,2.025
2.7,0.0,2.4
2.7,-0.25,2.4
3.3,-0.25,2.4
3.3,0.0,2.4
1.7,0.66,0.6
1.7,0.66,1.425
3.1,0.66,0.825
2.6,0.66,1.425
2.4,0.25,2.025
2.3,0.25,2.1
3.3,0.25,2.4
2.7,0.25,2.4
2.8,0.0,2.475
2.8,-0.25,2.475
3.525,-0.25,2.49375
3.525,0.0,2.49375
2.9,0.0,2.475
2.9,-0.15,2.475
3.45,-0.15,2.5125
3.45,0.0,2.5125
2.8,0.0,2.4
2.8,-0.15,2.4
3.2,-0.15,2.4
3.2,0.0,2.4
3.525,0.25,2.49375
2.8,0.25,2.475
3.45,0.15,2.5125
2.9,0.15,2.475
3.2,0.15,2.4
2.8,0.15,2.4
0.0,0.0,3.15
0.0,-0.002,3.15
0.002,0.0,3.15
0.8,0.0,3.15
0.8,-0.45,3.15
0.45,-0.8,3.15
0.0,-0.8,3.15
0.0,0.0,2.85
0.2,0.0,2.7
0.2,-0.112,2.7
0.112,-0.2,2.7
0.0,-0.2,2.7
-0.002,0.0,3.15
-0.45,-0.8,3.15
-0.8,-0.45,3.15
-0.8,0.0,3.15
-0.112,-0.2,2.7
-0.2,-0.112,2.7
-0.2,0.0,2.7
0.0,0.002,3.15
-0.8,0.45,3.15
-0.45,0.8,3.15
0.0,0.8,3.15
-0.2,0.112,2.7
-0.112,0.2,2.7
0.0,0.2,2.7
0.45,0.8,3.15
0.8,0.45,3.15
0.112,0.2,2.7
0.2,0.112,2.7
0.4,0.0,2.55
0.4,-0.224,2.55
0.224,-0.4,2.55
0.0,-0.4,2.55
1.3,0.0,2.55
1.3,-0.728,2.55
0.728,-1.3,2.55
0.0,-1.3,2.55
1.3,0.0,2.4
1.3,-0.728,2.4
0.728,-1.3,2.4
0.0,-1.3,2.4
-0.224,-0.4,2.55
-0.4,-0.224,2.55
-0.4,0.0,2.55
-0.728,-1.3,2.55
-1.3,-0.728,2.55
-1.3,0.0,2.55
-0.728,-1.3,2.4
-1.3,-0.728,2.4
-1.3,0.0,2.4
-0.4,0.224,2.55
-0.224,0.4,2.55
0.0,0.4,2.55
-1.3,0.728,2.55
-0.728,1.3,2.55
0.0,1.3,2.55
-1.3,0.728,2.4
-0.728,1.3,2.4
0.0,1.3,2.4
0.224,0.4,2.55
0.4,0.224,2.55
0.728,1.3,2.55
1.3,0.728,2.55
0.728,1.3,2.4
1.3,0.728,2.4
0.0,0.0,0.0
1.5,0.0,0.15
1.5,0.84,0.15
0.84,1.5,0.15
0.0,1.5,0.15
1.5,0.0,0.075
1.5,0.84,0.075
0.84,1.5,0.075
0.0,1.5,0.075
1.425,0.0,0.0
1.425,0.798,0.0
0.798,1.425,0.0
0.0,1.425,0.0
-0.84,1.5,0.15
-1.5,0.84,0.15
-1.5,0.0,0.15
-0.84,1.5,0.075
-1.5,0.84,0.075
-1.5,0.0,0.075
-0.798,1.425,0.0
-1.425,0.798,0.0
-1.425,0.0,0.0
-1.5,-0.84,0.15
-0.84,-1.5,0.15
0.0,-1.5,0.15
-1.5,-0.84,0.075
-0.84,-1.5,0.075
0.0,-1.5,0.075
-1.425,-0.798,0.0
-0.798,-1.425,0.0
0.0,-1.425,0.0
0.84,-1.5,0.15
1.5,-0.84,0.15
0.84,-1.5,0.075
1.5,-0.84,0.075
0.798,-1.425,0.0
1.425,-0.798,0.0
```

Question

How would you find the intersection between a ray and a tensor product patch?

Surface of Revolution

$$c(u) = \begin{pmatrix} c_x(u) \\ c_y(u) \\ c_z(u) \end{pmatrix} \qquad s(u,\theta) = R_z(\theta)p(u) = \begin{pmatrix} cos(\theta)c_x(u) - sin(\theta)c_y(u) \\ sin(\theta)c_x(u) + cos(\theta)c_y(u) \\ c_z(u) \end{pmatrix}$$



- Note that this is **possible** with ***rational*** tensor product patches

Swept Surfaces

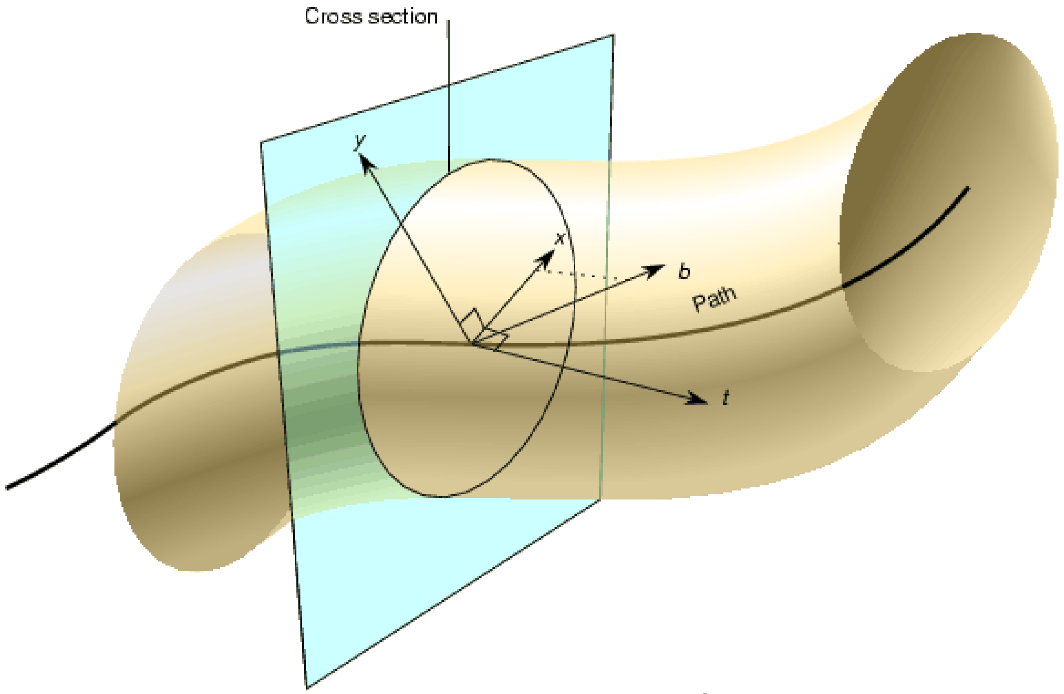
- Sweep curve along a path

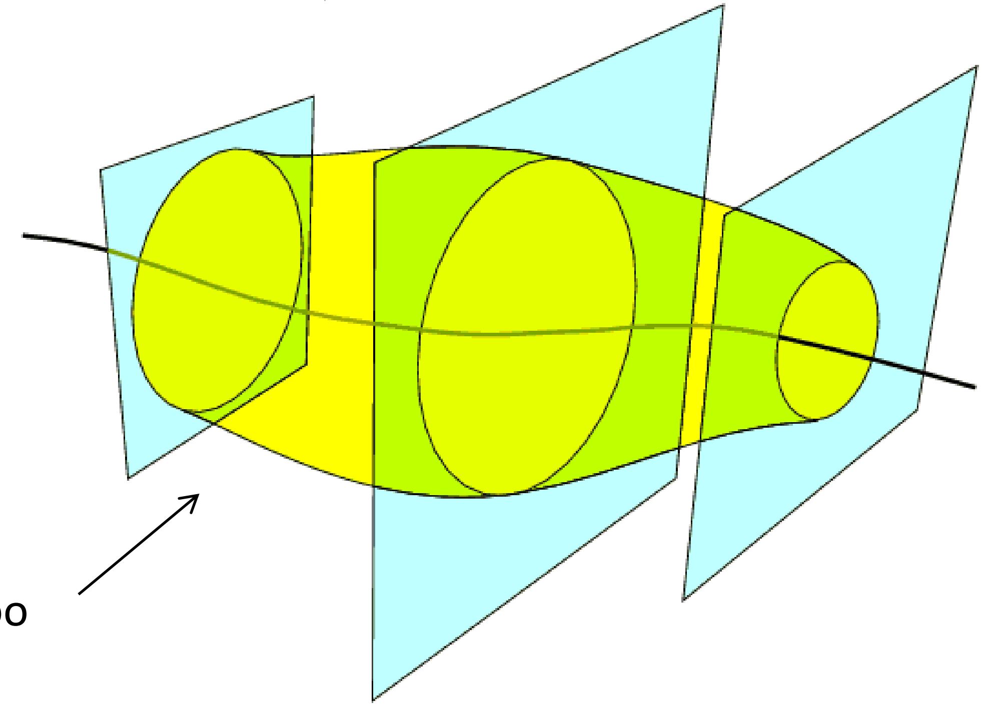
$$s(u, v) = c(u) + p(v)$$

- Cure can rotate to follow path too

$$s(u, v) = R(v)c(u) + p(v)$$

What to choose for R?
Frenet Frame?
Parallel Transport frame?





Scale can change along path too

Frenet Frame

- Can define frame along curve

$$T(v) = \frac{p'(v)}{||p'(v)||}$$

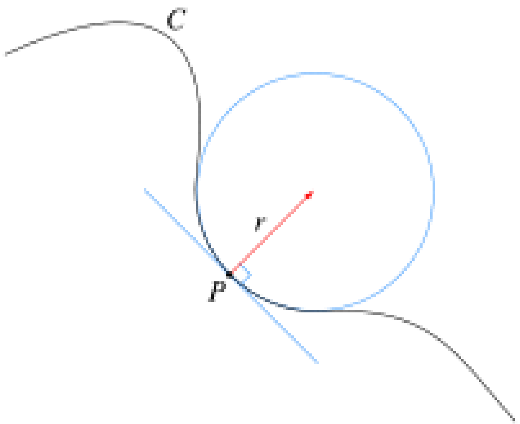
$$B(v) = \frac{p'(v) \times p''(v)}{||p'(v) \times p''(v)||}$$

$$N(v) = B(v) \times T(v)$$

This provides a coordinate frame, or rotation R for sweeping

  - tangent, binormal, and normal
  - normal points in direction of osculating circle with radius equal to 1 over the curvature kappa

$$\kappa = \frac{||p'(v) \times p''(v)||}{||p(v)||^3}$$



Parallel Transport Frame / Bishop Frame

- Frenet frame problems
  - Not always defined. **Give an example!**
  - Discontinuous. **Give an example!**

- Parallel Transport
  - Given initial frame U0 V0 move this frame along the curve using turns that are as small as possible
    - Consider the change in tangent T for a small step h along curve
    - Compute axis of turn
    - Compute angle of turn
    - Turn the frame

$$\mathrm{A} = T(v) \times T(v+h)$$
$$\alpha = \cos^{-1}\big(T(v) \cdot T(v+h)\big)$$
$$V(v+h) = R(A,\alpha)V(v)$$
$$U(v+h) = R(A,\alpha)U(v)$$

Parallel Transport

- Advantages
  - Frames are defined everywhere
  - Typically more useful for shape design and animation because of minimum turning (minimum twist) property
- Disadvantage
  - Do not know the frame without walking along the curve.
    - Frenet frame is an intrinsic property, can be computed anywhere with local information
  - There is no continuity if we close the curve
    - Frenet frame will always match when closed, while Parallel transport will not
    - The angle difference is related to total turning/torsion

Summary and other topics

- Tricks with rational splines
  - 2D circles, 3D conics, spheres, cylinders, surfaces of revolution
- B-Splines
  - Convenient alternative to Bezier for longer curves
- Non-uniform Ration B-Splines (NURBS)
  - More degrees of freedom, allows better control of velocity on curve
- Other surfaces
  - Many other ways to define surfaces!
- Trimming
  - Trim surface with curves in parameter space

Questions

- When can you write a cubic Hermite curve as a Quadratic Bezier?
- How many real scalar values are necessary to describe the shape of a rational Bezier curve?
- Show that a Bernstein basis polynomial $B_{in}(t)$ attains only one maximum on [0, 1] and does so at $t = i/n$.

Review and More Information

- FCG Chapter 15 Curves
- See also supplemental notes in course contents