Curves

Hakan Bilen
University of Edinburgh

Computer Graphics
Fall 2017

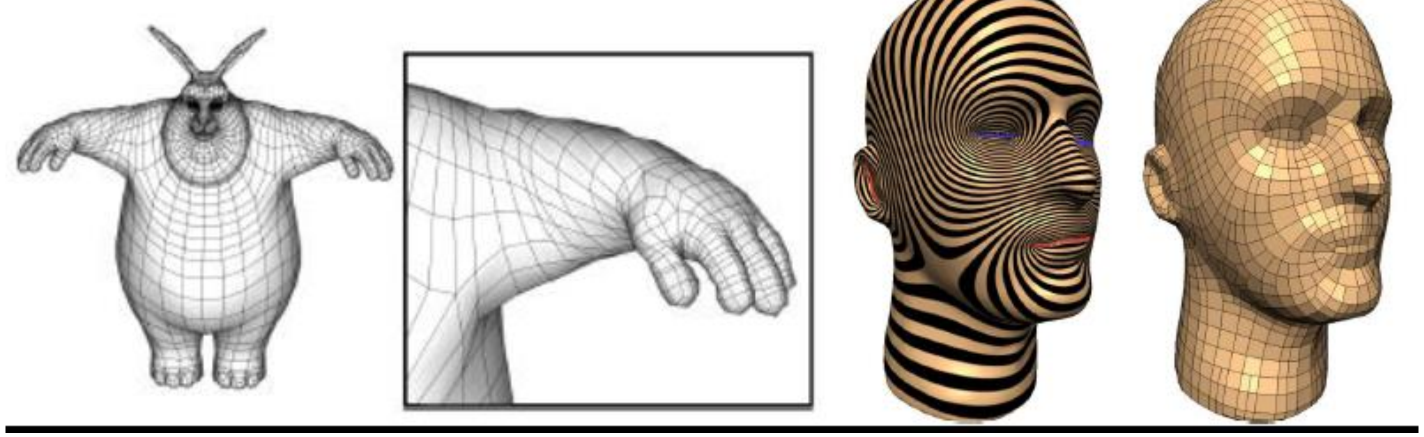
Some slides are courtesy of Steve Marschner and Taku Komura

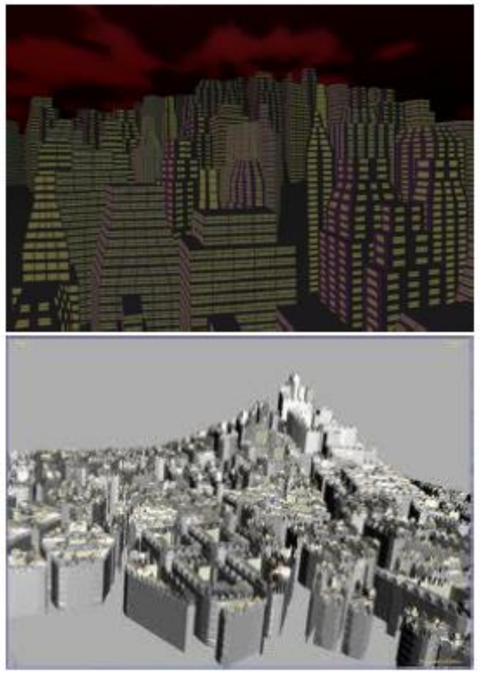
How to create a virtual world?

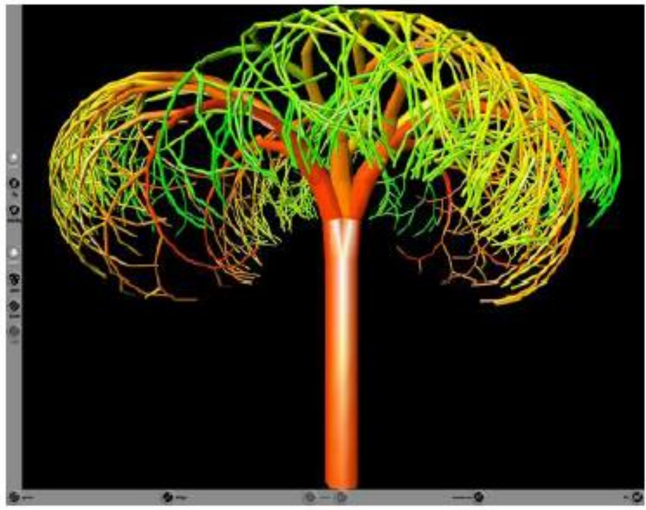
To compose scenes

We need to define objects

- Characters
- Terrains
- Objects (trees, furniture, buildings etc)

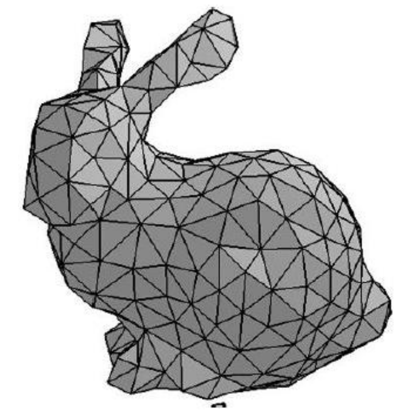


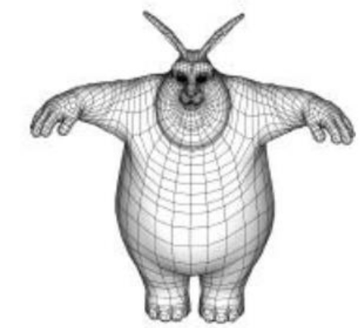




Geometric representations

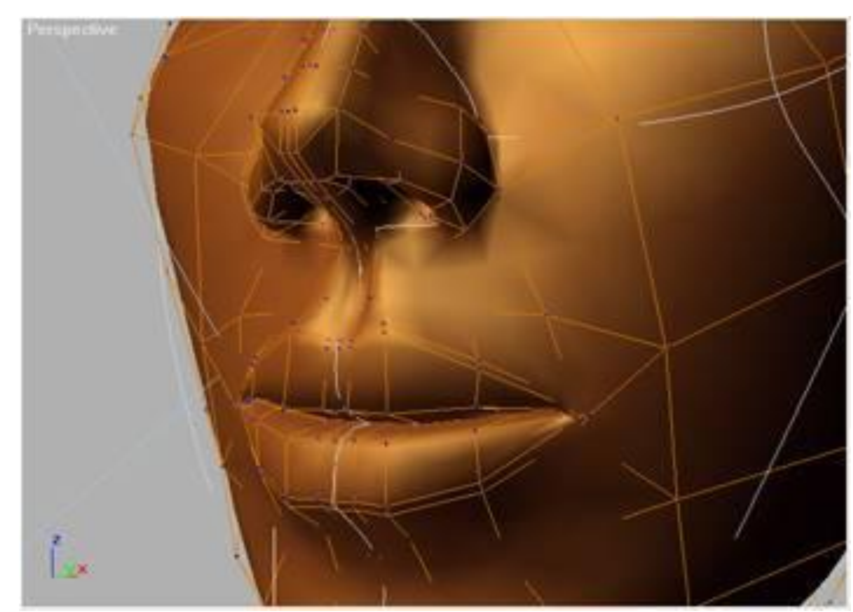
- Meshes
  - Triangle, quadrilateral, polygon





- Implicit surfaces
  - Blobs, metaballs

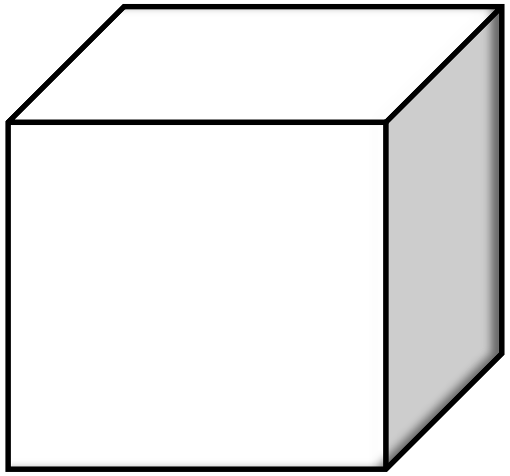
- Parametric surfaces / curves
  - Polynomials
  - Bezier curves, B-splines



Motivation

Smoothness

Many applications require smooth surfaces



[scene360.com]

Can produce smooth surfaces with less parameters

- Easier to design
- Can efficiently preserve complex structures

Original Spline

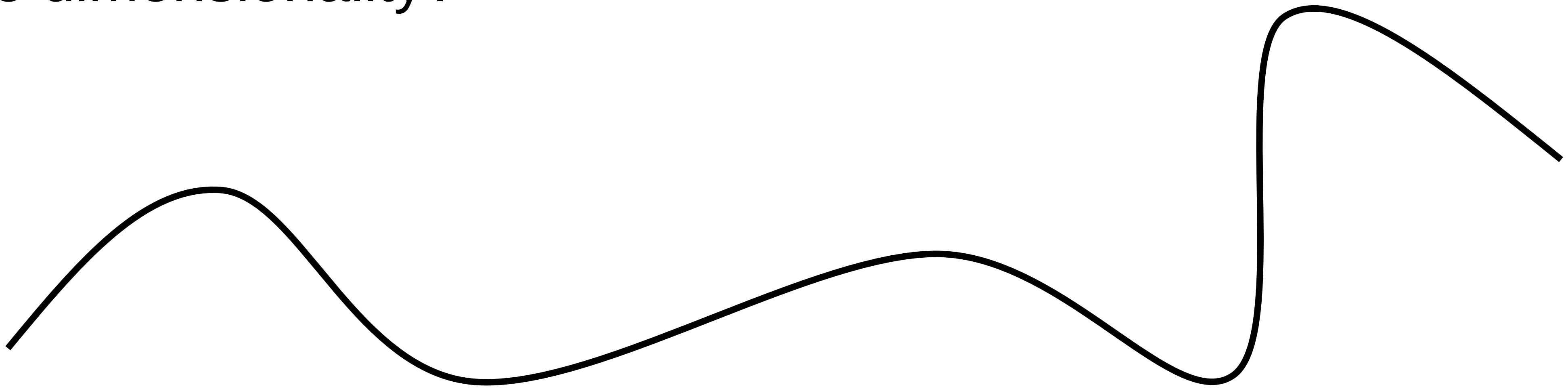
From draftsmanship to CG

- Control
  - user specified control points
  - analogy: ducks
- Smoothness
  - smooth functions
  - usually low order polynomials
  - analogy: physical constraints, optimization

What is a curve?

A set of points that the pen traces over an interval of time

What is the dimensionality?



Implicit form: $\mathrm{f}(\mathrm{x}, \mathrm{y}) = x^2 + y^2 - 1 = 0$

- Find the points that satisfy the equation

Parametric form: $(\mathrm{x}, \mathrm{y}) = \mathrm{f}(\mathrm{t}) = (cost, sint), \quad t \in [0,2\pi)$

- Easier to draw

What is a spline curve?

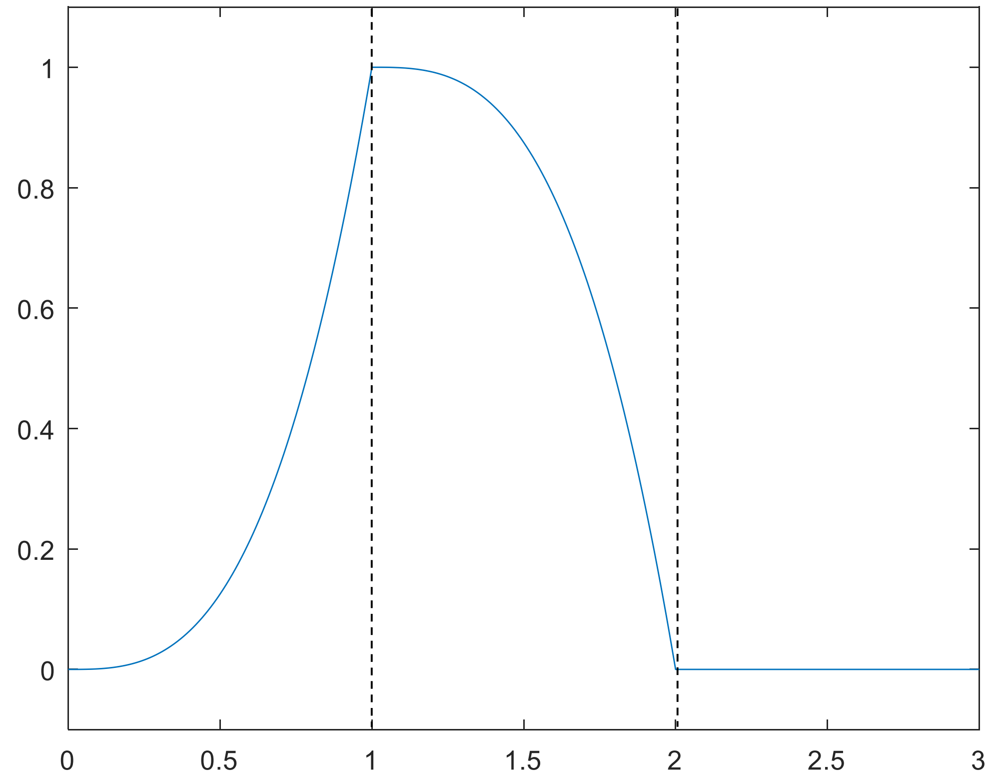
in this context

$f(t)$ is a

- parametric curve
- piecewise polynomial function that switches between different functions for different t intervals

Example:

$$f(t) = \begin{cases} t^3, & \text{if } 0 \leq t < 1 \\ 1 - (t-1)^3, & \text{if } 1 \leq t < 2 \\ 0, & \text{otherwise} \end{cases}$$

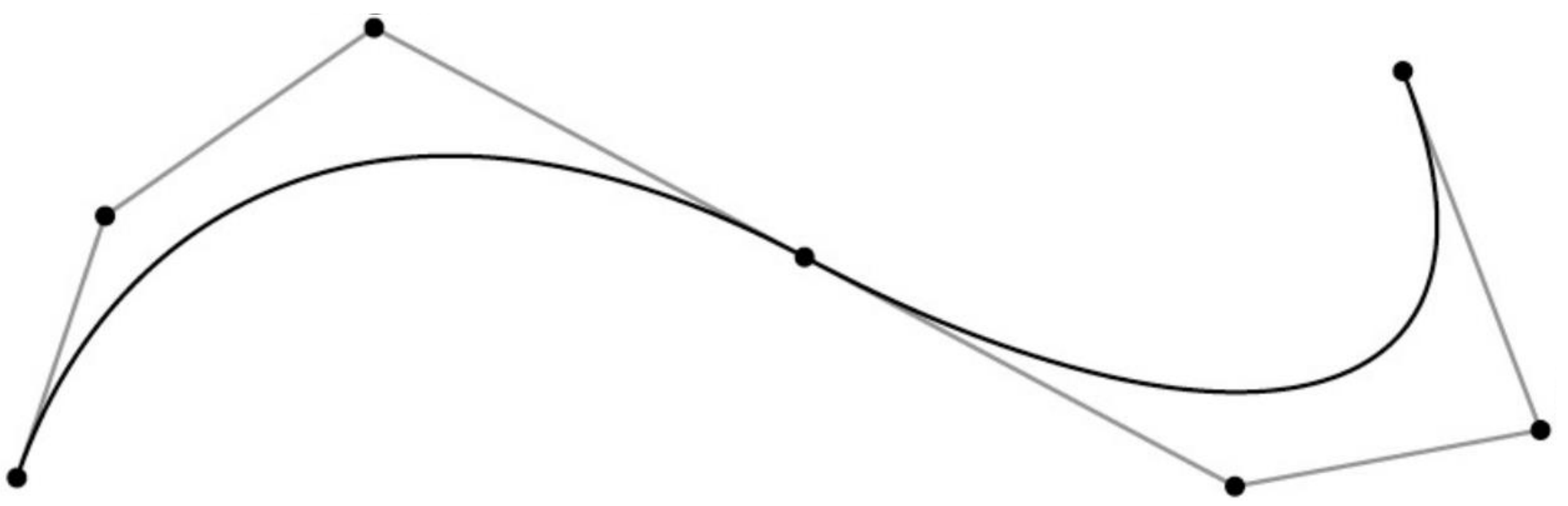


Defining spline curves

- Discontinuities at the integers [t=k]
- Each spline piece is defined over [k,k+1] (e.g. a cubic spline)

$$f(t) = at^3 + bt^2 + ct + d$$

- Different coefficients for every interval
- Control of spline curves
  - Interpolate
  - Approximate



Today

- Spline segments
  - Linear
  - Quadratic
  - Hermite
  - Bezier
- Chaining splines

- Notation
  - vectors bold and lowercase $\boldsymbol{v}$
  - points as column vector $\boldsymbol{p} = \begin{pmatrix} p_x & p_y \end{pmatrix}$
  - matrices bold and uppercase $\boldsymbol{M}$

Spline segments

Linear Segment

A line segment connecting point $\boldsymbol{p_o}$ to $\boldsymbol{p_1}$

Such that $f(0) = \boldsymbol{p_0}$ and $f(1) = \boldsymbol{p_1}$

$$f_x(t) = (1 - t)\boldsymbol{x_o} + t\boldsymbol{x_1}$$

$$f_y(t) = (1 - t)\boldsymbol{y_o} + t\boldsymbol{y_1}$$

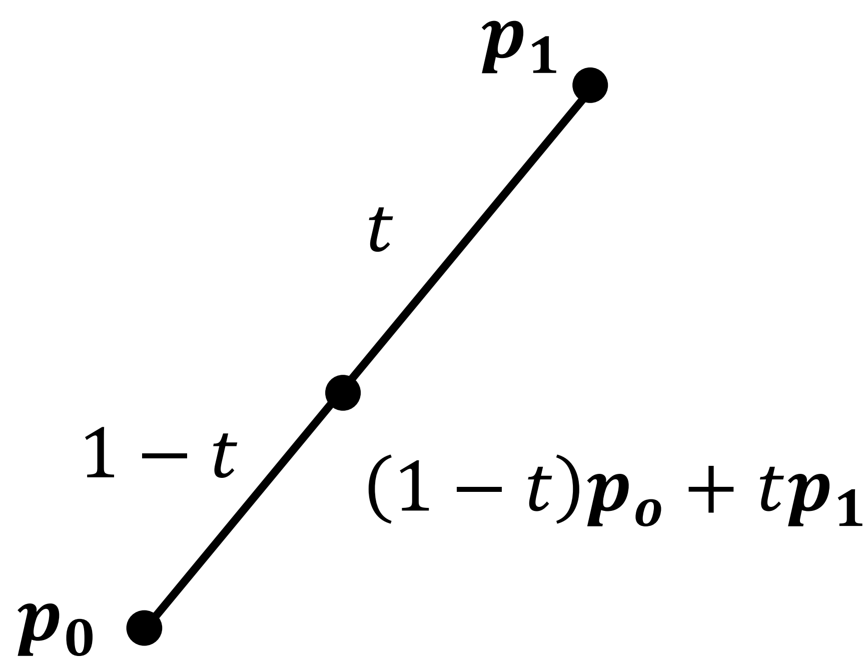
Vector formulation

$$f(t) = (1 - t)\boldsymbol{p_o} + t\boldsymbol{p_1}$$

Matrix formulation

$$f(t) = (t \quad 1)\begin{pmatrix} ? & ? \\ ? & ? \end{pmatrix}\begin{pmatrix} \boldsymbol{p_0} \\ \boldsymbol{p_1} \end{pmatrix}$$

$$f(t) = (t \quad 1)\begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} \boldsymbol{p_0} \\ \boldsymbol{p_1} \end{pmatrix}$$



Matrix form of spline

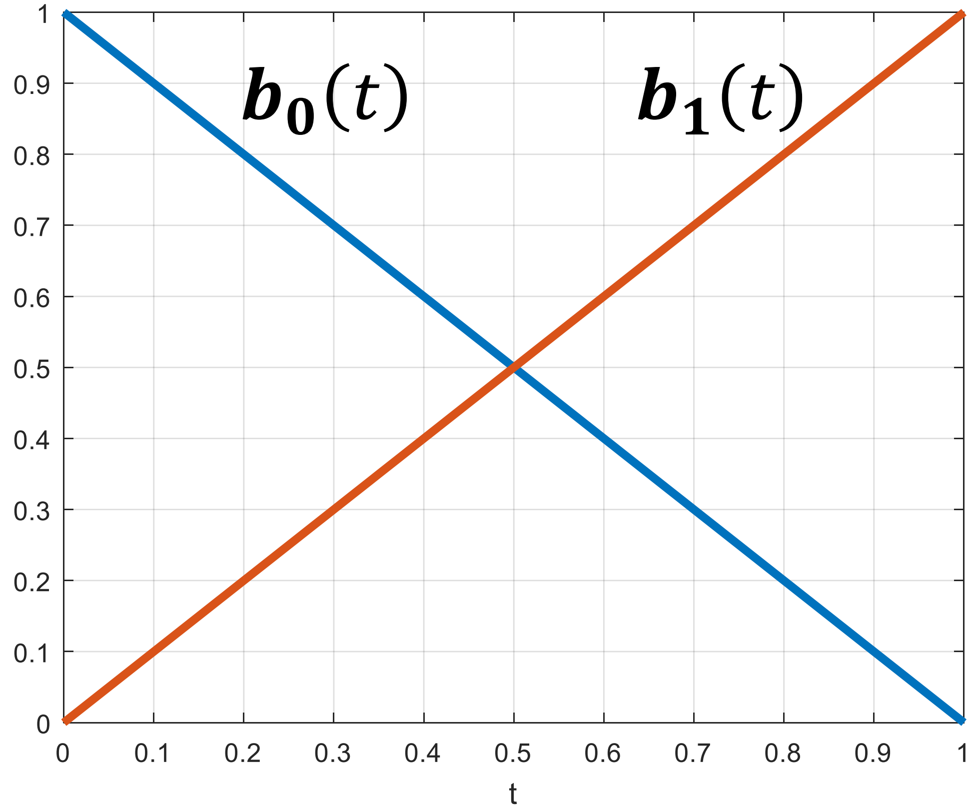
$$f(t) = (t \quad 1)\begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} \boldsymbol{p_0} \\ \boldsymbol{p_1} \end{pmatrix}$$

Blending functions $\boldsymbol{b}(t)$ specify how to blend the values of the control point vector

$$\boldsymbol{f}(t) = \boldsymbol{b_0}(t)\boldsymbol{p_0} + \boldsymbol{b_1}(t)\boldsymbol{p_1}$$

$$\boldsymbol{b_0}(t) = 1 - t$$

$$\boldsymbol{b_1}(t) = t$$



Beyond line segment

Quadratic

A quadratic ($\boldsymbol{f}(t) = a_0 + a_1 t + a_2 t^2$) passes through $\boldsymbol{p_0}$, $\boldsymbol{p_1}$, $\boldsymbol{p_2}$ s.t.

$$\boldsymbol{p_0} = f(0) \quad = a_0 + 0 \quad a_1 + 0^2 \quad a_2$$

$$\boldsymbol{p_1} = f(0.5) = a_0 + 0.5\, a_1 + 0.5^2\, a_2$$

$$\boldsymbol{p_2} = f(1) \quad = a_0 + 1 \quad a_1 + 1^2 \quad a_2$$

Points can be written in terms of constraint matrix $\boldsymbol{C}$

$$\begin{pmatrix} \boldsymbol{p_0} \\ \boldsymbol{p_1} \\ \boldsymbol{p_2} \end{pmatrix} = \boldsymbol{Ca} \Rightarrow \begin{pmatrix} \boldsymbol{p_0} \\ \boldsymbol{p_1} \\ \boldsymbol{p_2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0.5 & 0.25 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} \Rightarrow \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \boldsymbol{C^{-1}} \begin{pmatrix} \boldsymbol{p_0} \\ \boldsymbol{p_1} \\ \boldsymbol{p_2} \end{pmatrix}$$

$\boldsymbol{f}(t)$ can be written in terms of basis matrix $\boldsymbol{B} = \boldsymbol{C^{-1}}$ and points $\boldsymbol{p}$

$$\boldsymbol{f}(t) = \boldsymbol{tBp} = \boldsymbol{tC}^{-1}\boldsymbol{p} = (t^2 \;\; t \;\; 1) \begin{pmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{p_0} \\ \boldsymbol{p_1} \\ \boldsymbol{p_2} \end{pmatrix}$$

Matrix form of spline

Blending functions

$$\boldsymbol{f}(t) = (t^2 \;\; t \;\; 1)\begin{pmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} \boldsymbol{p_0} \\ \boldsymbol{p_1} \\ \boldsymbol{p_2} \end{pmatrix}$$

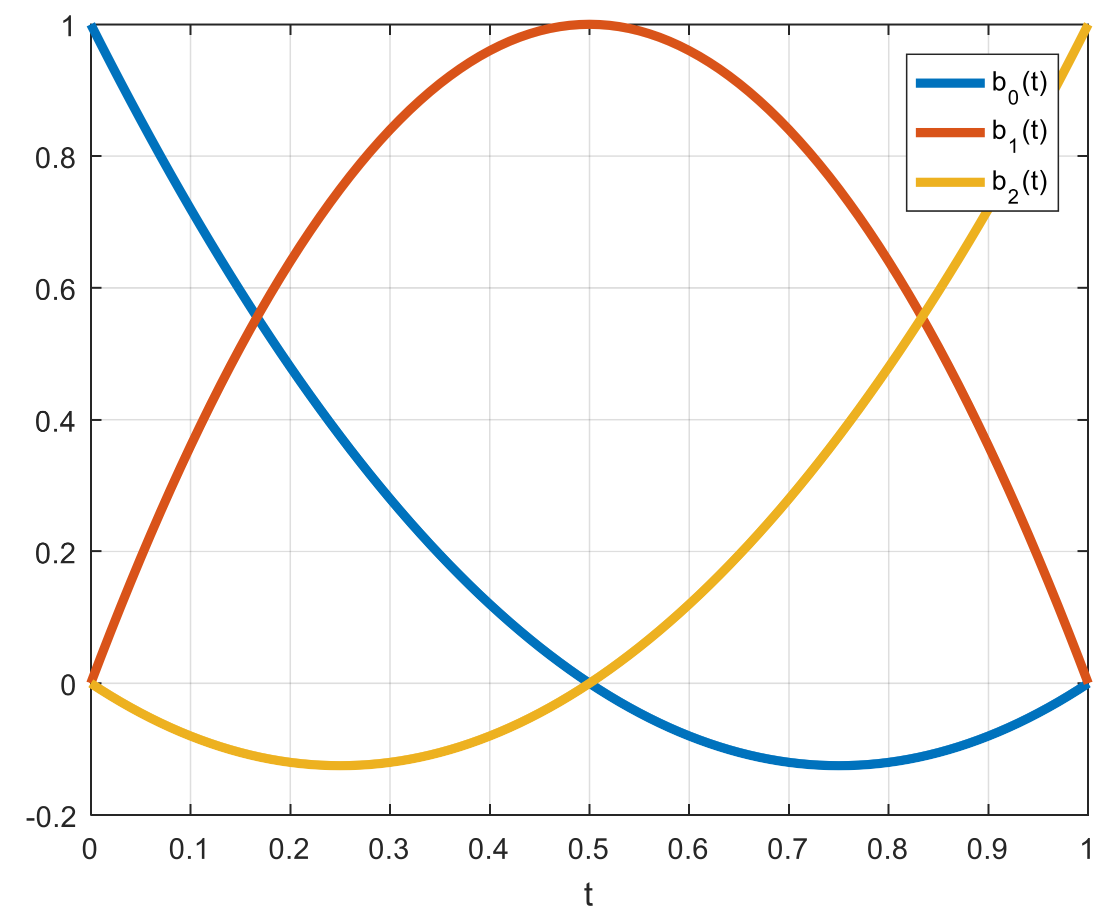
Blending functions $\boldsymbol{b}(t)$ specify how to blend the values of the control point vector

$$\boldsymbol{f}(t) = \boldsymbol{b_0}(t)\boldsymbol{p_0} + \boldsymbol{b_1}(t)\boldsymbol{p_1} + \boldsymbol{b_2}(t)\boldsymbol{p_2}$$

$$\boldsymbol{b_0}(t) = 2t^2 - 3t + 1$$

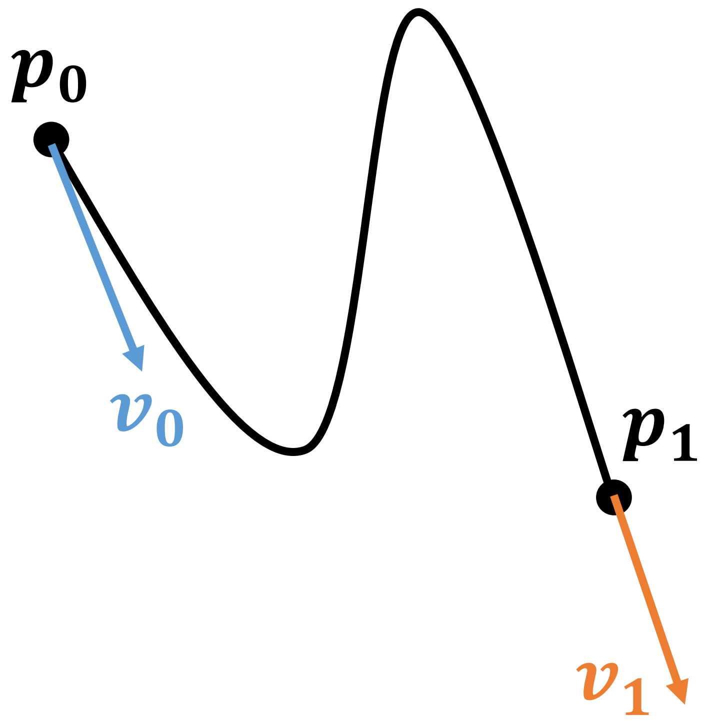
$$\boldsymbol{b_1}(t) = -4t^2 - 4t$$

$$\boldsymbol{b_2}(t) = 2t^2 - 1$$



Hermite spline

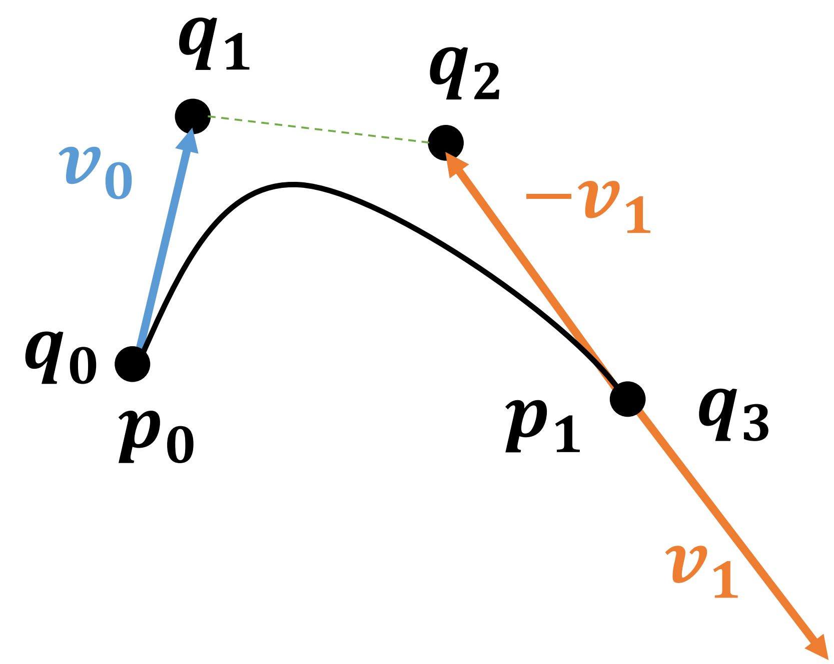
- Piecewise cubic ($\boldsymbol{f}(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$)
- Additional constraint on tangents (derivatives)
  - $\boldsymbol{f}(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$
  - $\boldsymbol{f}'^{(t)} = a_1 + 2a_2 t + 3a_3 t^2$
  - $\boldsymbol{p_0} = f(0) \quad = a_0$
  - $\boldsymbol{p_1} = f(1) \quad = a_0 + a_1 + a_2 + a_3$
  - $\boldsymbol{v}_1 = f'(0) \quad = a_1$
  - $\boldsymbol{v}_2 = f'(1) \quad = a_1 + 2a_2 + \ 3a_3$



- Simpler matrix form

$$\boldsymbol{f}(t) = (t^3 \ t^2 \ \ t \ \ 1)\begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 2 \end{pmatrix}\begin{pmatrix} \boldsymbol{p_0} \\ \boldsymbol{p_1} \\ \boldsymbol{v_1} \\ \boldsymbol{v_2} \end{pmatrix}$$

Hermite to Bézier



Specify tangents as points

- $p_0 = q_0, p_1 = q_3, v_0 = 3(q_1 - q_0), v_1 = 3(q_3 - q_2)$
- $\begin{pmatrix} \boldsymbol{p_0} \\ \boldsymbol{p_1} \\ \boldsymbol{v_1} \\ \boldsymbol{v_2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 1 & 0 \\ 0 & 0 & -3 & 3 \end{pmatrix} \begin{pmatrix} \boldsymbol{q_0} \\ \boldsymbol{q_1} \\ \boldsymbol{q_2} \\ \boldsymbol{q_3} \end{pmatrix}$
- Update Hermite eq. (from previous slide)

$$\boldsymbol{f}(t) = (t^3\ t^2\ \ t\ \ 1) \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 1 & 0 \\ 0 & 0 & -3 & 3 \end{pmatrix} \begin{pmatrix} \boldsymbol{q_0} \\ \boldsymbol{q_1} \\ \boldsymbol{q_2} \\ \boldsymbol{q_3} \end{pmatrix}$$

Bézier matrix

$$\boldsymbol{f}(t) = (t^3\ t^2\ \ t\ \ 1)\begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 2 & 0 & -6 & 6 \end{pmatrix}\begin{pmatrix} \boldsymbol{q_0} \\ \boldsymbol{q_1} \\ \boldsymbol{q_2} \\ \boldsymbol{q_3} \end{pmatrix}$$

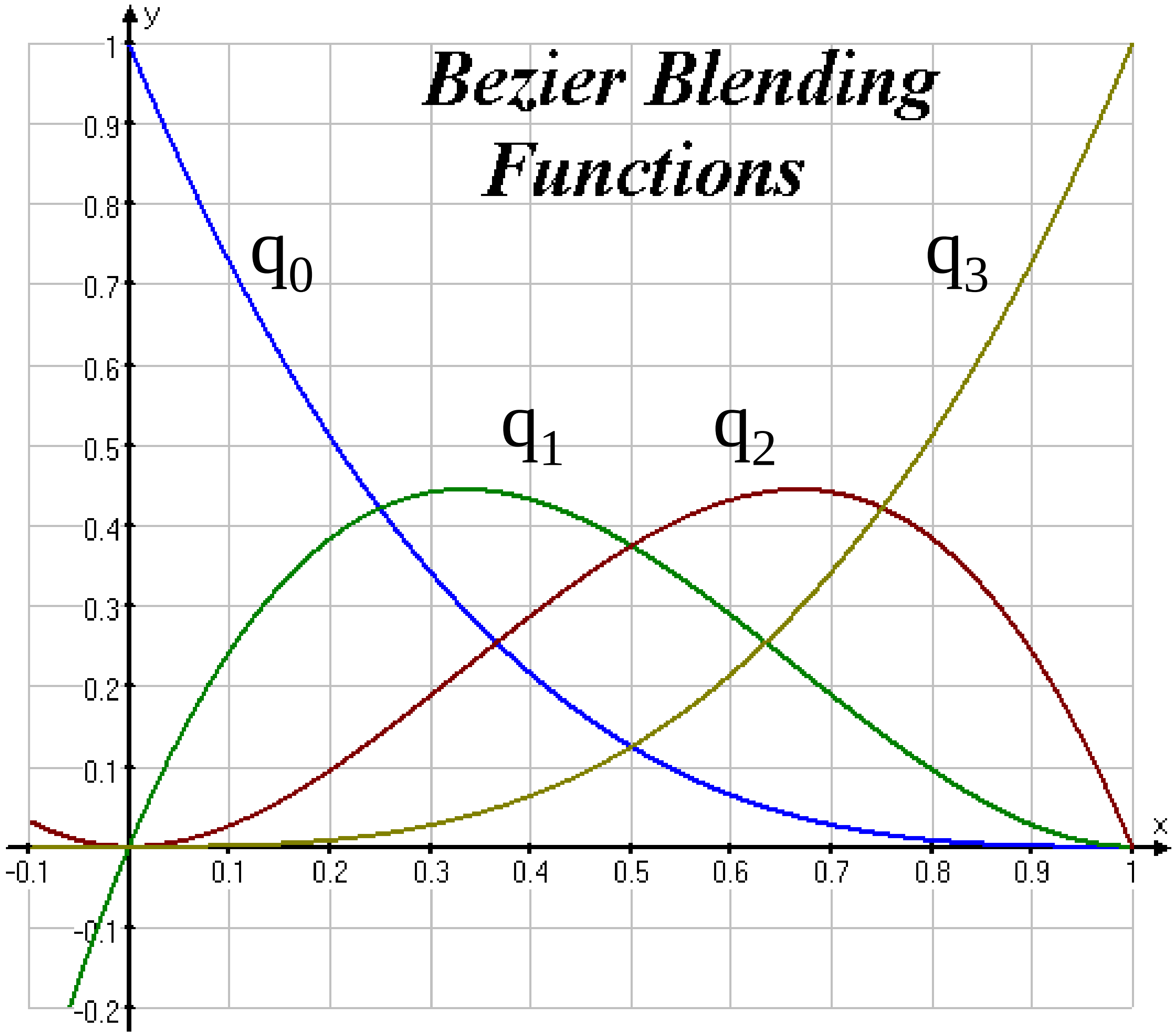
- $\boldsymbol{f}(t) = \sum_{n=0}^{d} \boldsymbol{b_{n,3}} \boldsymbol{q_n}$
- Blending functions $\boldsymbol{b}(t)$ has a special name in this case:
- Bernstein polynomials

$$b_{n,k} = \binom{n}{k} t^k (1-t)^{n-k}$$

and that defines Bézier curves for any degree

Bézier blending functions

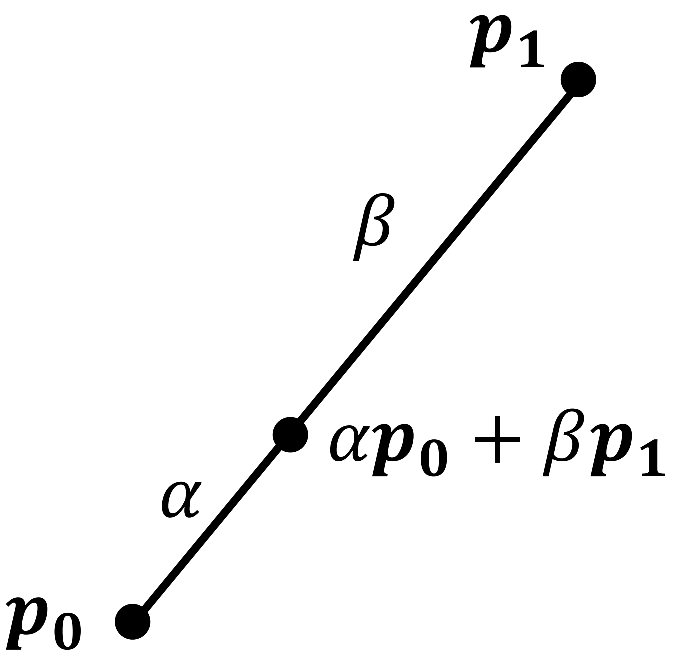
- The functions sum to 1 at any point along the curve.
- Endpoints have full weight

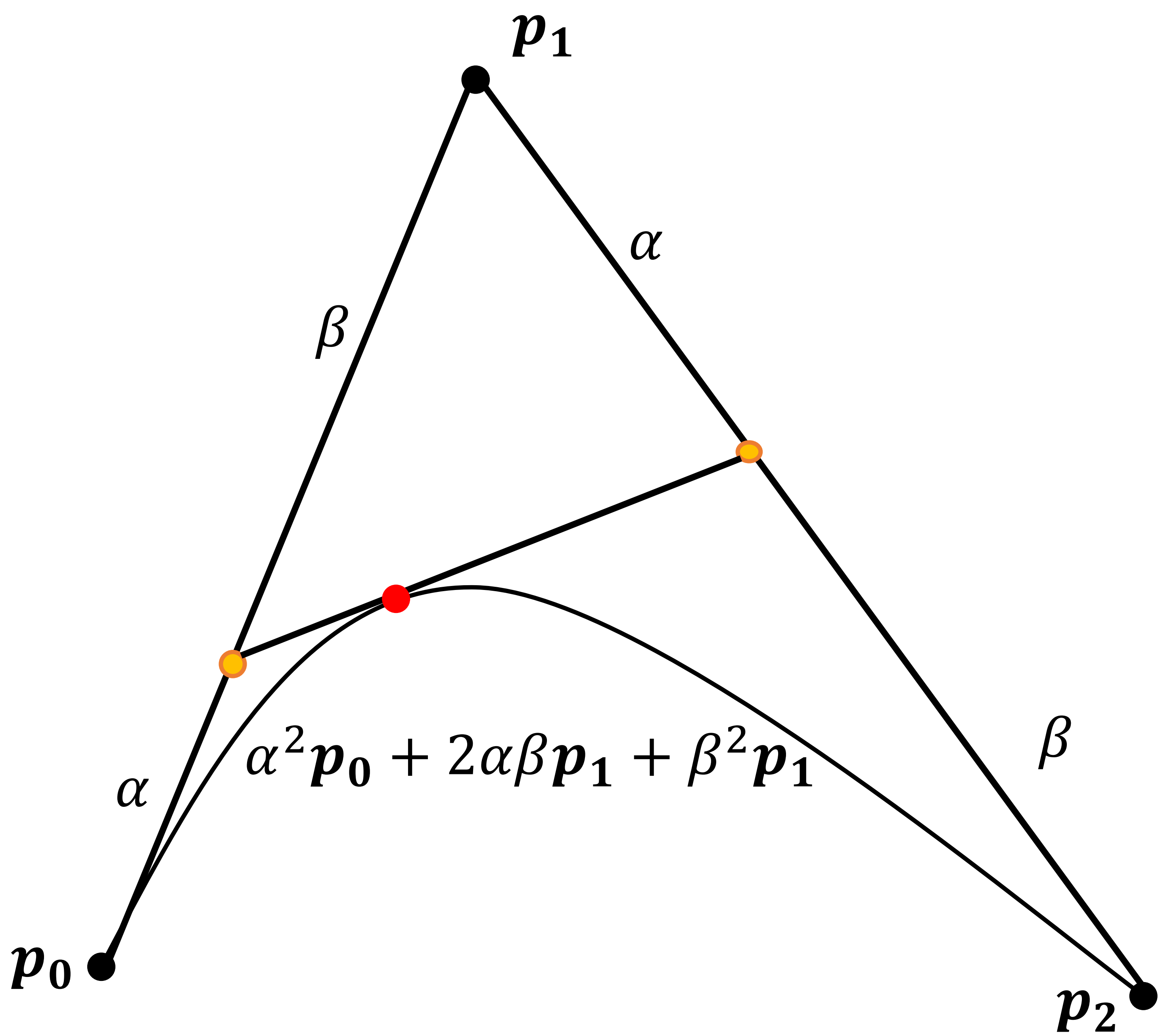


Another view to Bézier segments

de Casteljau algorithm

Blend each linear spline with α and $\beta = 1 - \alpha$





Review

http://www.inf.ed.ac.uk/teaching/courses/cg/d3/hermite.html

http://www.inf.ed.ac.uk/teaching/courses/cg/d3/bezier.html

http://www.inf.ed.ac.uk/teaching/courses/cg/d3/Casteljau.html

Today

- Spline segments
  - Linear
  - Quadratic
  - Hermite
  - Bezier
- Chaining splines
  - Continuity and local control
  - Hermite curves
  - Bezier curves
  - Catmull-Rom curves
  - B-splines

Putting segments together

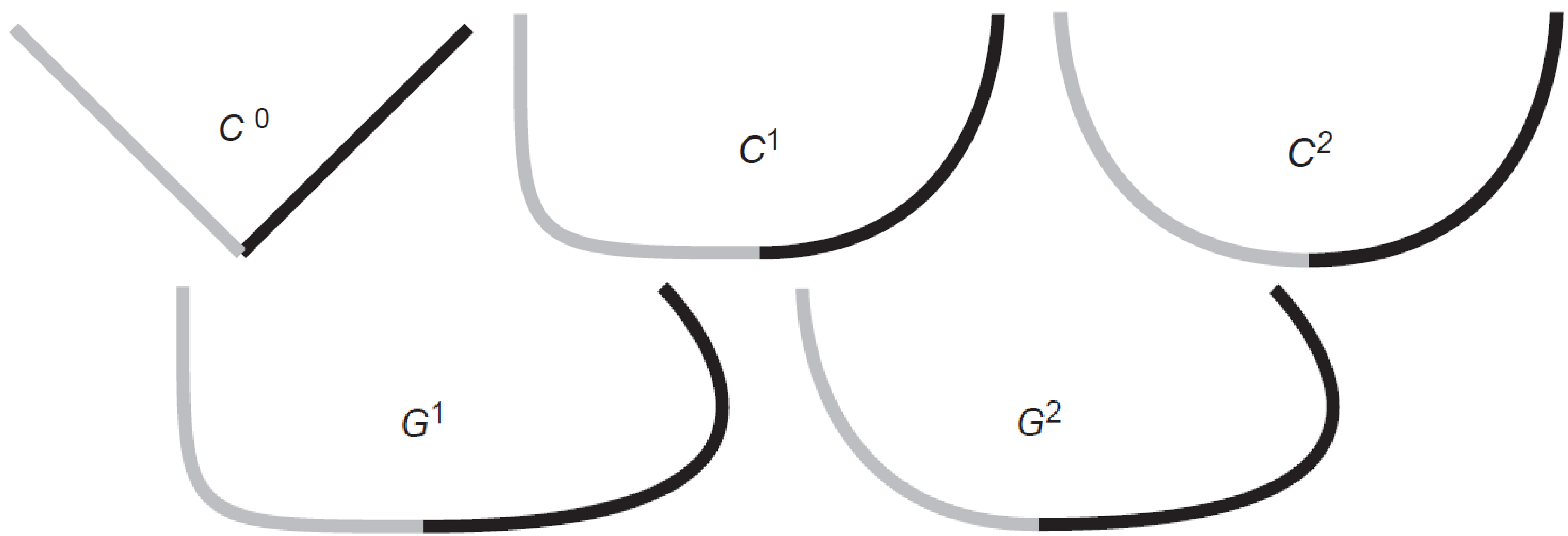
- Limited degree of freedom with a single polynomial
- Will it be smooth enough?

Measuring smoothness

Continuity

Smoothness as degree of continuity

- zero-order (C^0): position match
- first-order (C^1): tangent match
- second-order (C^2): curvature match
- C^N vs G^N

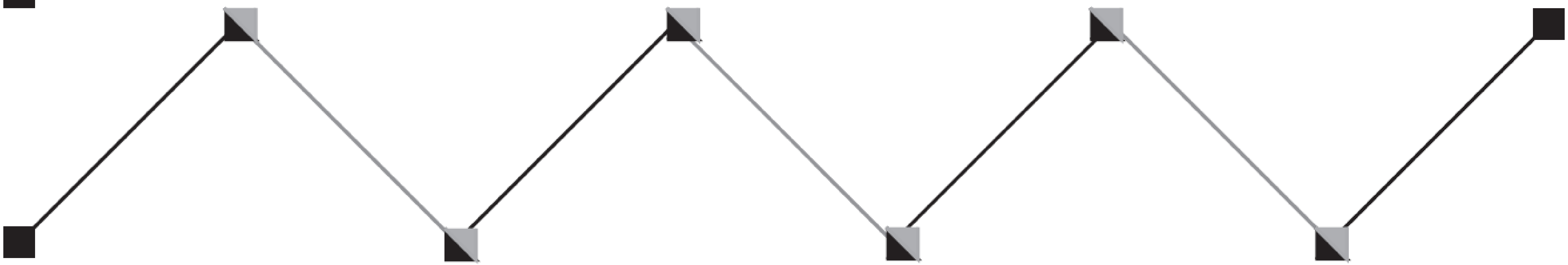


Putting segments together

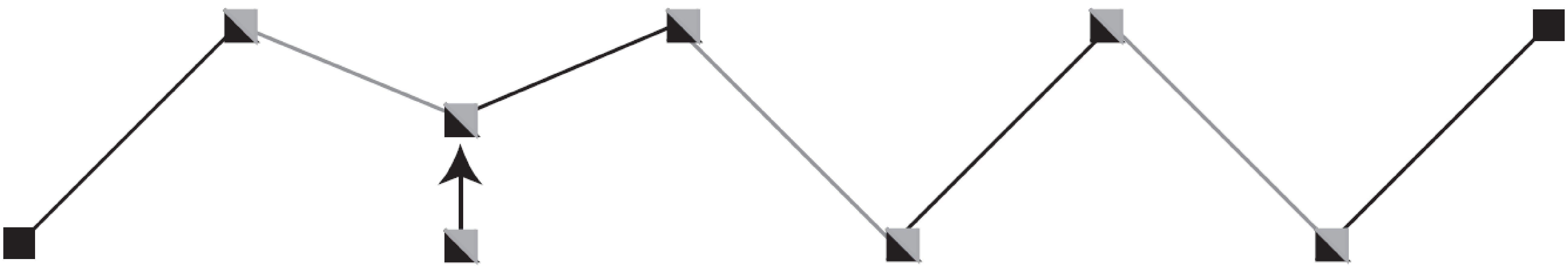
- Limited degree of freedom with a single polynomial
- Will it be smooth enough?
- Local control

Local control

Each line segment is parameterized by its endpoints.



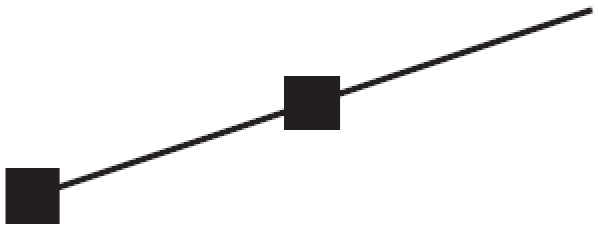
The end of one segment is shared with the beginning endpoint of the next segment.



Moving a control point causes a change only in a localized region.

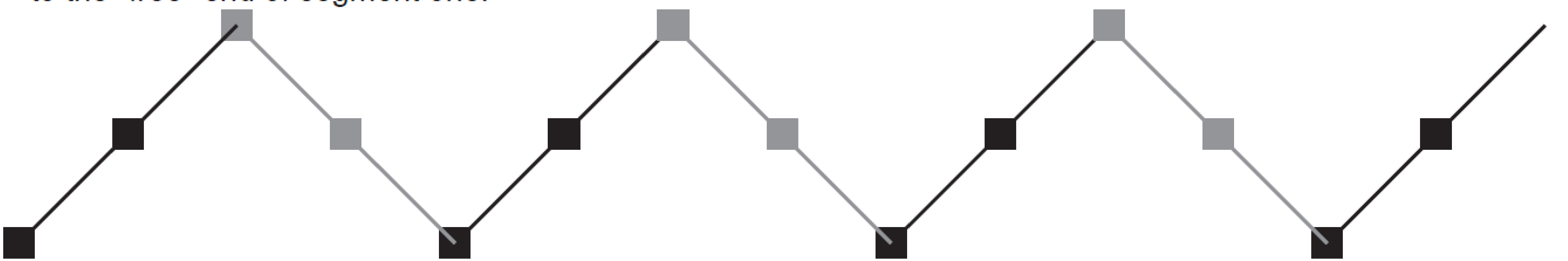
How many knots for m segments with 2 control points each?

Local control



Each line segment is parameterized by its endpoint and its centerpoint.

The endpoint of segment two is equated to the "free" end of segment one.



The endpoint of segment three is equated to the "free" end of segment two, etc.

?

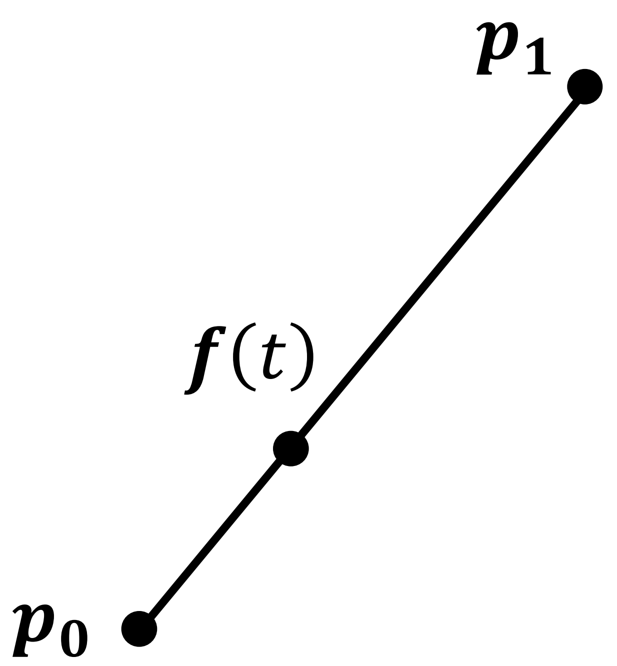
Local control

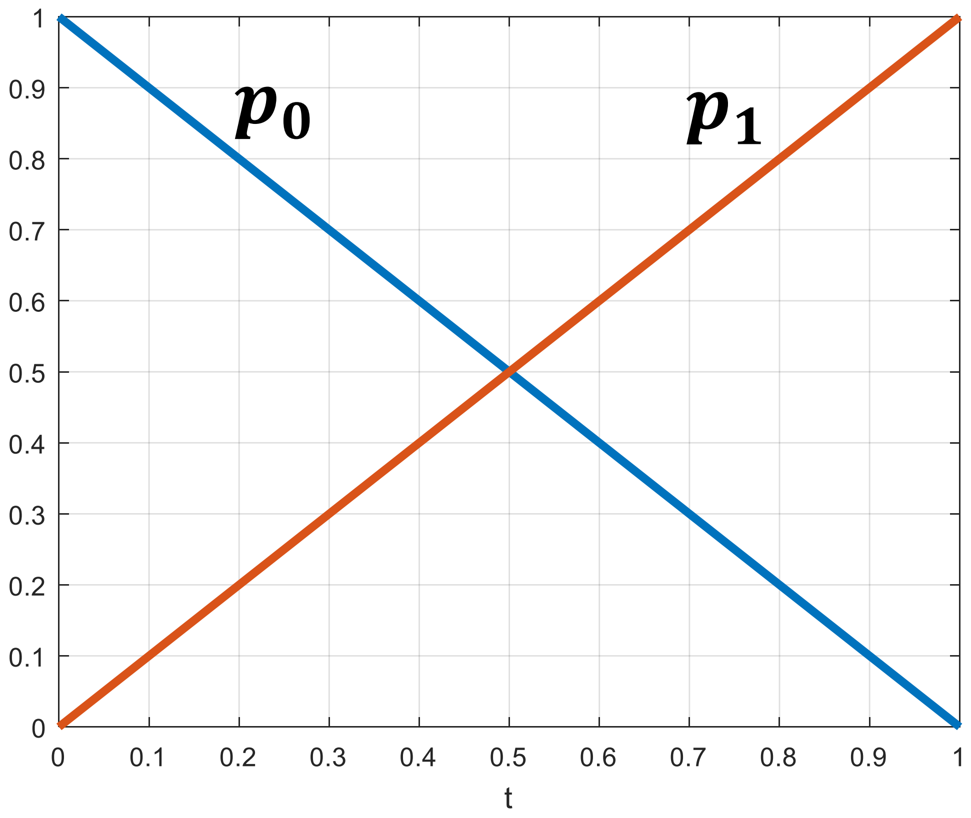
- changing control point only affects a limited part of spline
  - without this, splines are very difficult to use
    - overshooting
    - fixed computation
  - many likely formulations lack this
    - natural spline
    - polynomial fits (matlab demo)

Putting segments together

Piecewise linear

Blending functions for a linear segment (0<t<1)

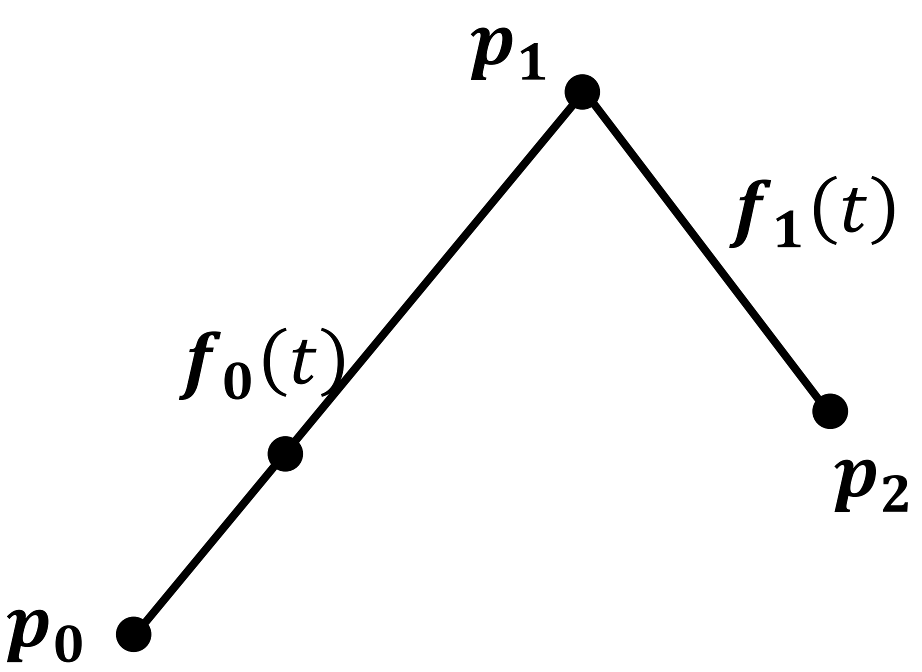


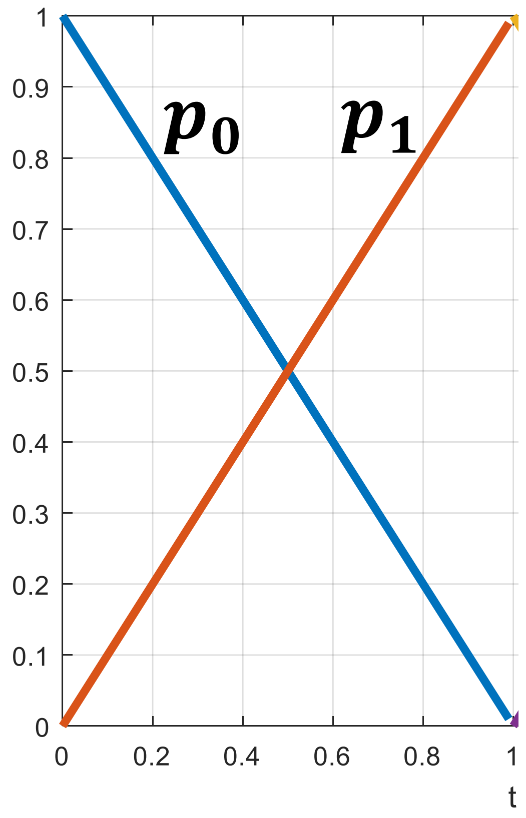


$$f(t) = (1-t)p_0 + tp_1$$

Putting segments together

Piecewise linear





$$\boldsymbol{f_0}(t) = (1-t)\boldsymbol{p_0} + t\boldsymbol{p_1} \quad (0 < t < 1)$$

$$\boldsymbol{f_1}(t) = (?)\boldsymbol{p_1} + (?)\boldsymbol{p_2} \quad (1 < t < 2)$$

$$\boldsymbol{f_1}(t) = (2-t)\boldsymbol{p_1} + (t-1)\boldsymbol{p_2} \quad (1 < t < 2)$$

Putting segments together

How can we chain these segments to a longer curve?

- Use first segment between t=0 to t=1
- Use second segment between t=1 to t=2

$$f(t) = f_i(t-i) \text{ for } i \le t \le i+1$$

- Shift blending functions

$$f_0(t) = b_0(t)\boldsymbol{p_0} + b_1(t)\boldsymbol{p_1}$$

$$f_1(t) = b_0(t-1)\boldsymbol{p_1} + b_1(t-1)\boldsymbol{p_2}$$

- Match derivatives at end points to avoid discontinuity

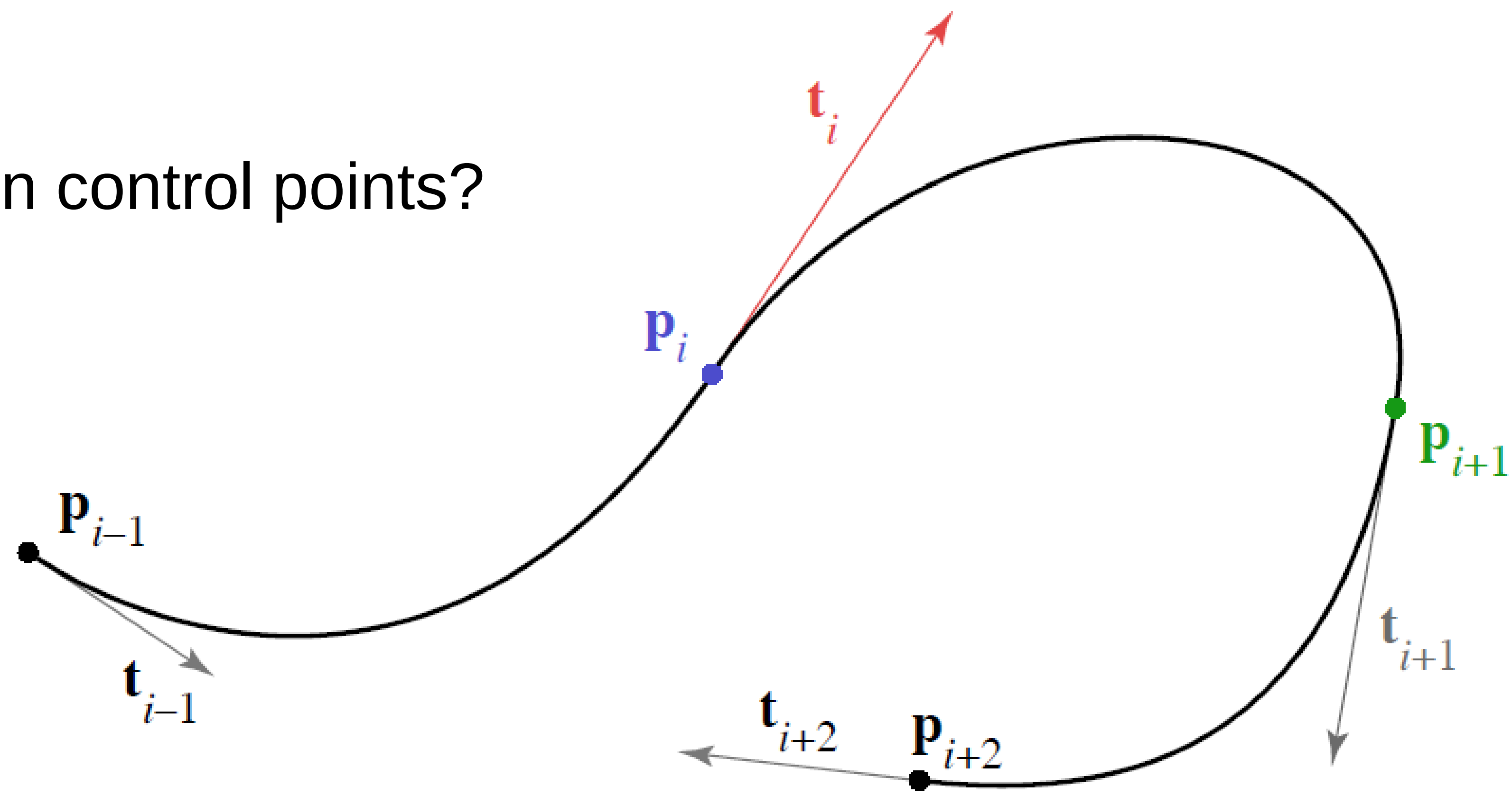
Hermite basis

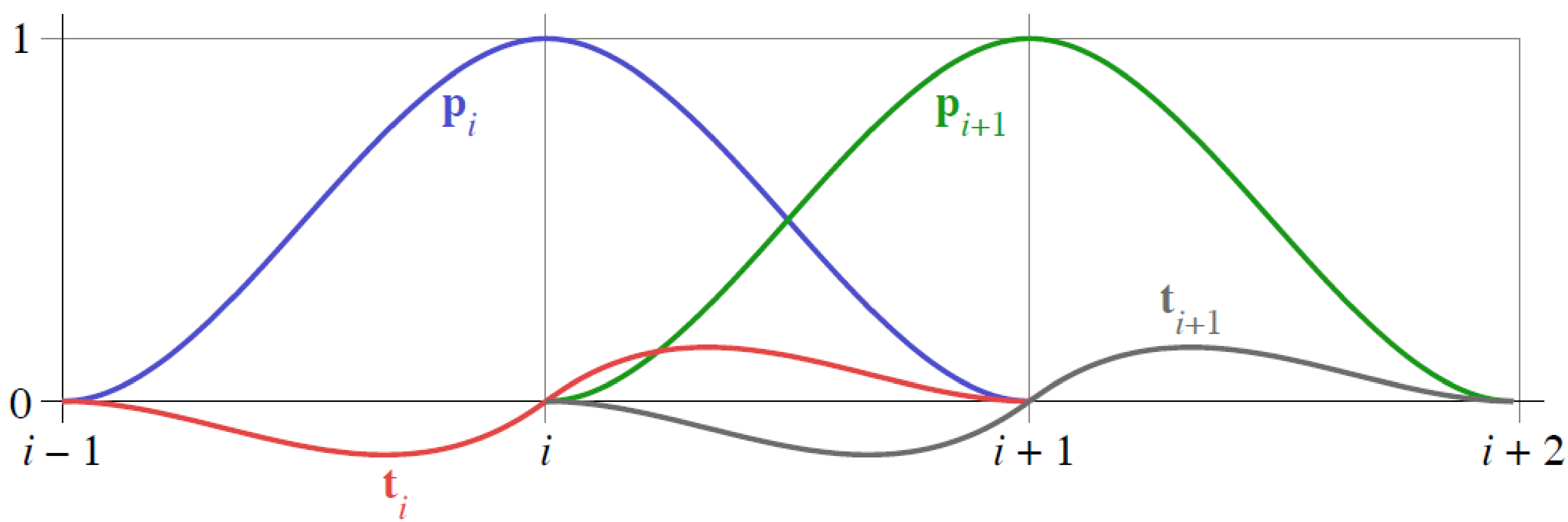
- How many segments for n control points?

  (n-2)/2

- Continuity?

  C^1



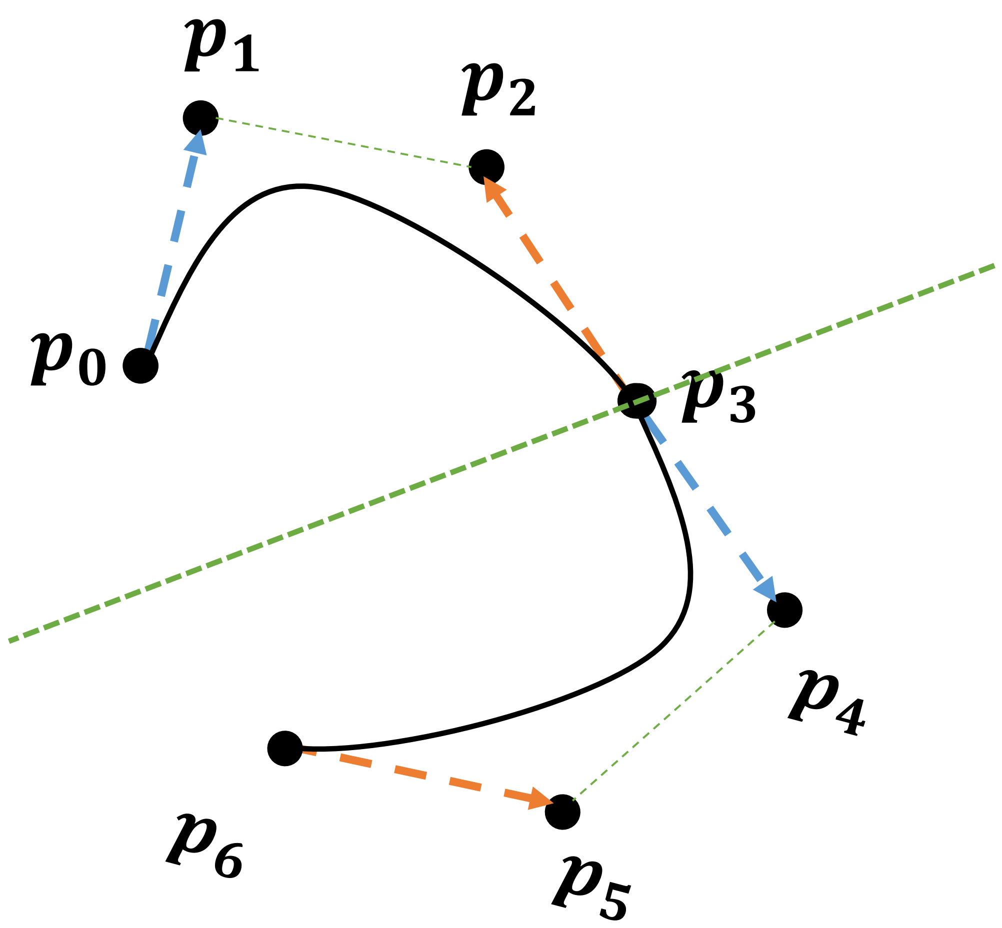


Slide credit: S. Marschner

Chaining Bézier Splines

Can we add blending functions as in Hermite?

- If $(p_3 - p_2)$ is collinear with $(p_4 - p_3)$, it is G^1 continuous
  - $p_3 - p_2 = k(p_4 - p_3),\ k > 0$
- If tangents match, it is C^1 continuous
  - $p_3 - p_2 = k(p_4 - p_3)$, $\mathrm{k} = 1$



Chaining segments

- Hermite curves are convenient because they can be made long easily
- Bézier curves are convenient because their controls are all points
  - but it is fussy to maintain continuity constraints
  - and they interpolate every 3rd point, which is a little odd
- We derived Bézier from Hermite by defining tangents from control points

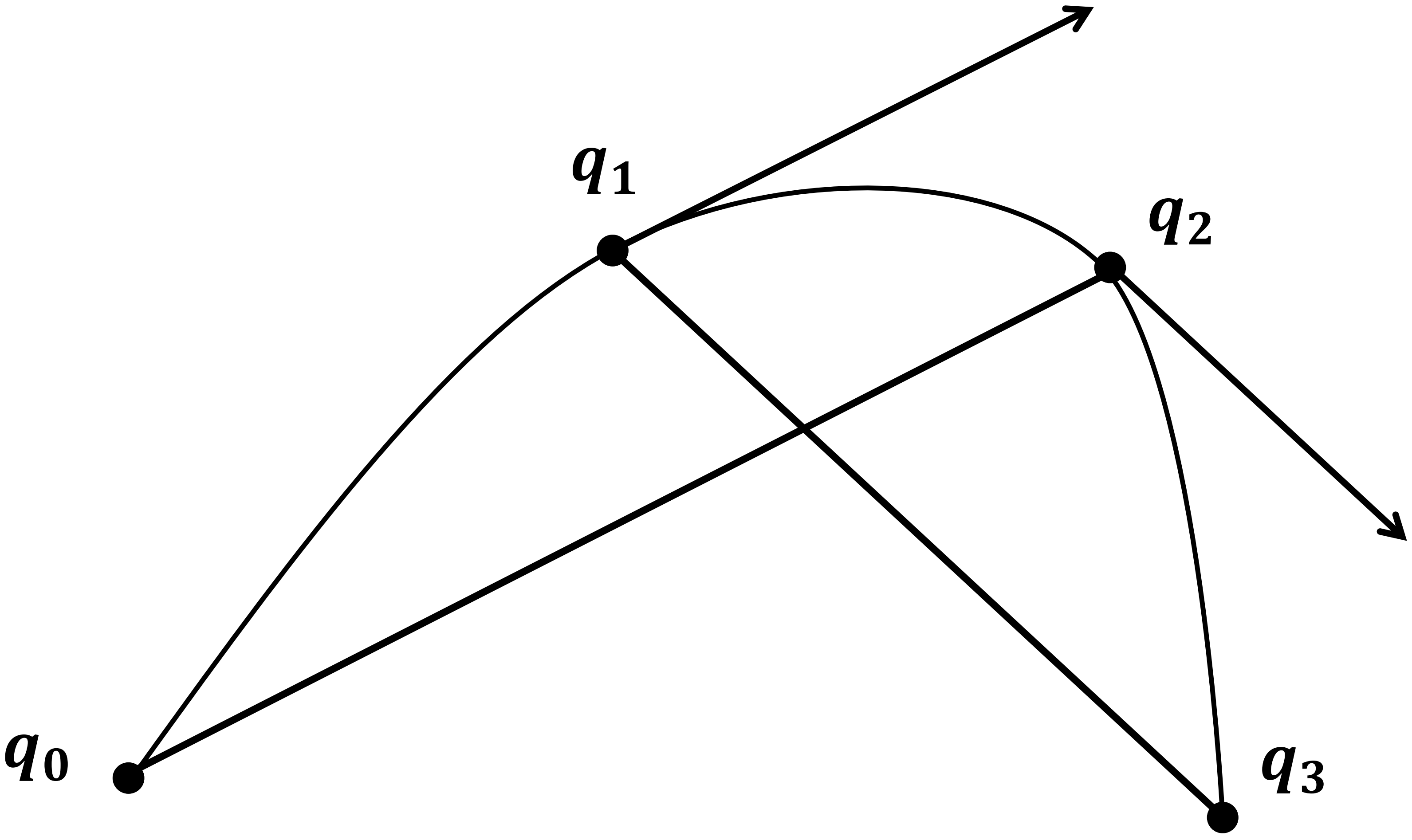
– a similar construction leads to the interpolating Catmull-Rom

spline

Catmull-Rom

Would like to define tangents automatically

– use adjacent control points



Hermite to Catmull-Rom

- $p_0 = f(0)$
- $p_1 = f(1)$
- $v_1 = f'(0)$
- $v_2 = f'(1)$

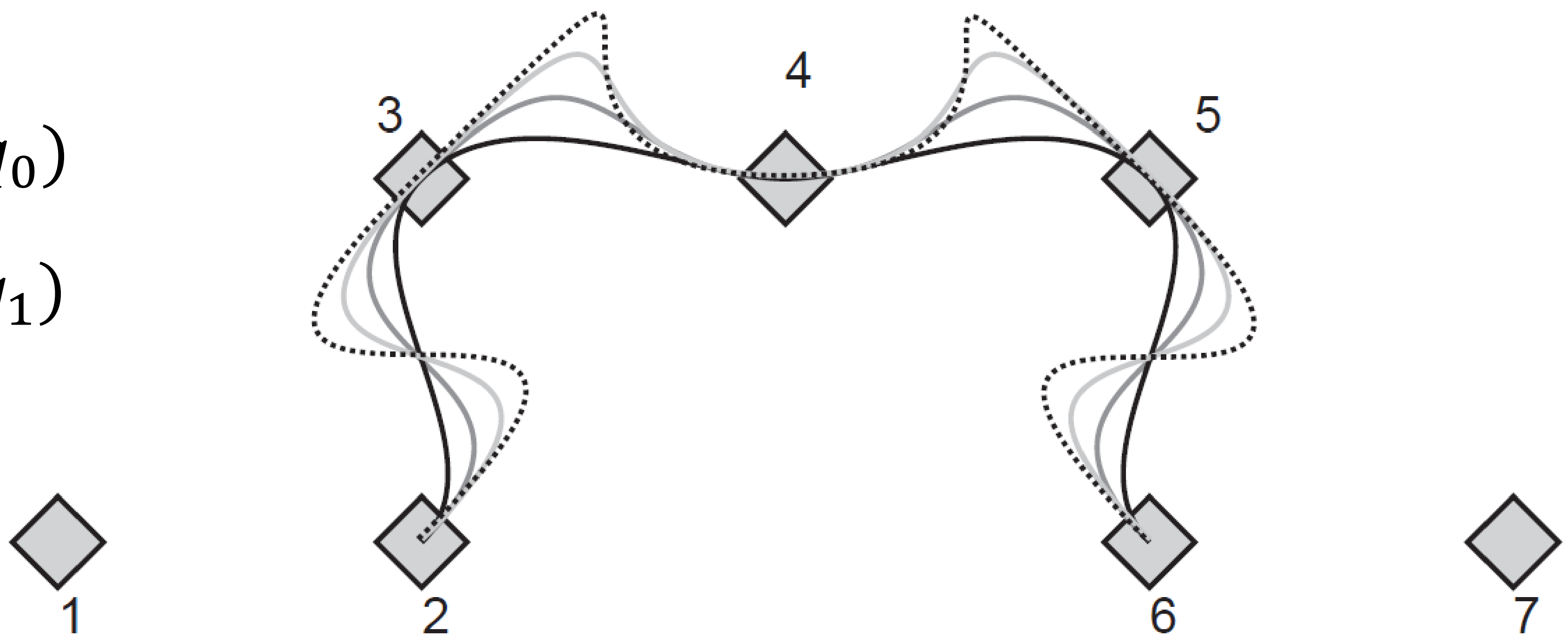
- $p_0 = q_1$
- $p_1 = q_2$
- $v_0 = \frac{1}{2}(q_2 - q_0)$
- $v_1 = \frac{1}{2}(q_3 - q_1)$

- $\boldsymbol{f}(t) =$
$$(t^3\ t^2\ \ t\ \ 1)\begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 2 \end{pmatrix}\begin{pmatrix} \boldsymbol{p_0} \\ \boldsymbol{p_1} \\ \boldsymbol{v_0} \\ \boldsymbol{v_1} \end{pmatrix}$$

$$\boldsymbol{f}(t) = (t^3\ t^2\ \ t\ \ 1)\begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 2 \end{pmatrix}\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -0.5 & 0 & 0.5 & 0 \\ 0 & -0.5 & 0 & 0.5 \end{pmatrix}\begin{pmatrix} \boldsymbol{q_0} \\ \boldsymbol{q_1} \\ \boldsymbol{q_2} \\ \boldsymbol{q_3} \end{pmatrix}$$

Catmull-Rom

- Interpolating curve
- Like Bézier, equivalent to Hermite
- Continuity ?
- Local control ?
- Add tension
  - $p_0 = q_1$
  - $p_1 = q_2$
  - $v_0 = \frac{1}{2}(1-t)(q_2 - q_0)$
  - $v_1 = \frac{1}{2}(1-t)(q_3 - q_1)$



B-splines

• We may want more continuity than C^1

• $f(t) = \sum_{i=1}^{n} \boldsymbol{p_i}\, b_i(t)$

– parameterized by k control points

– made of polynomials of degree k-1

– is $C^{(k-2)}$

• B-splines are a clean, flexible way of making long splines with arbitrary order of continuity

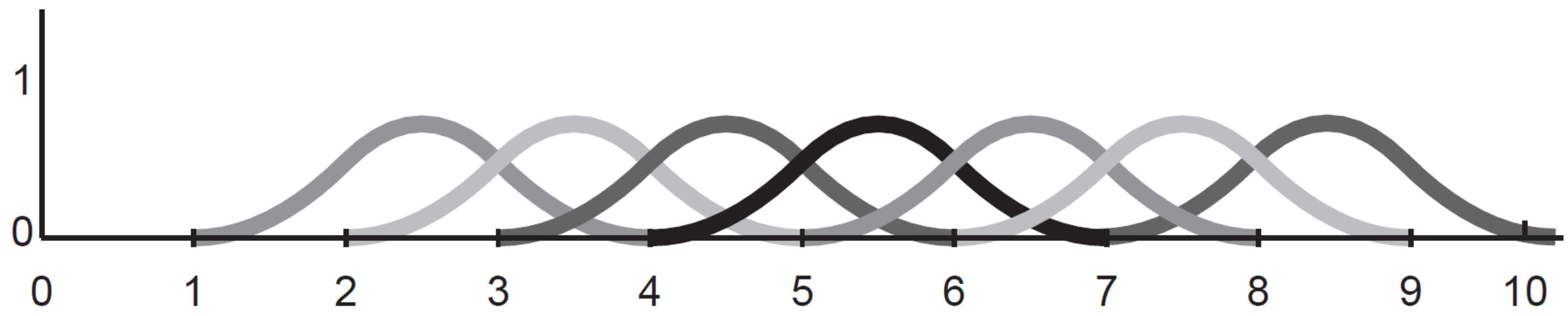
• Various ways to think of construction

– a simple one is convolution

– relationship to sampling and reconstruction

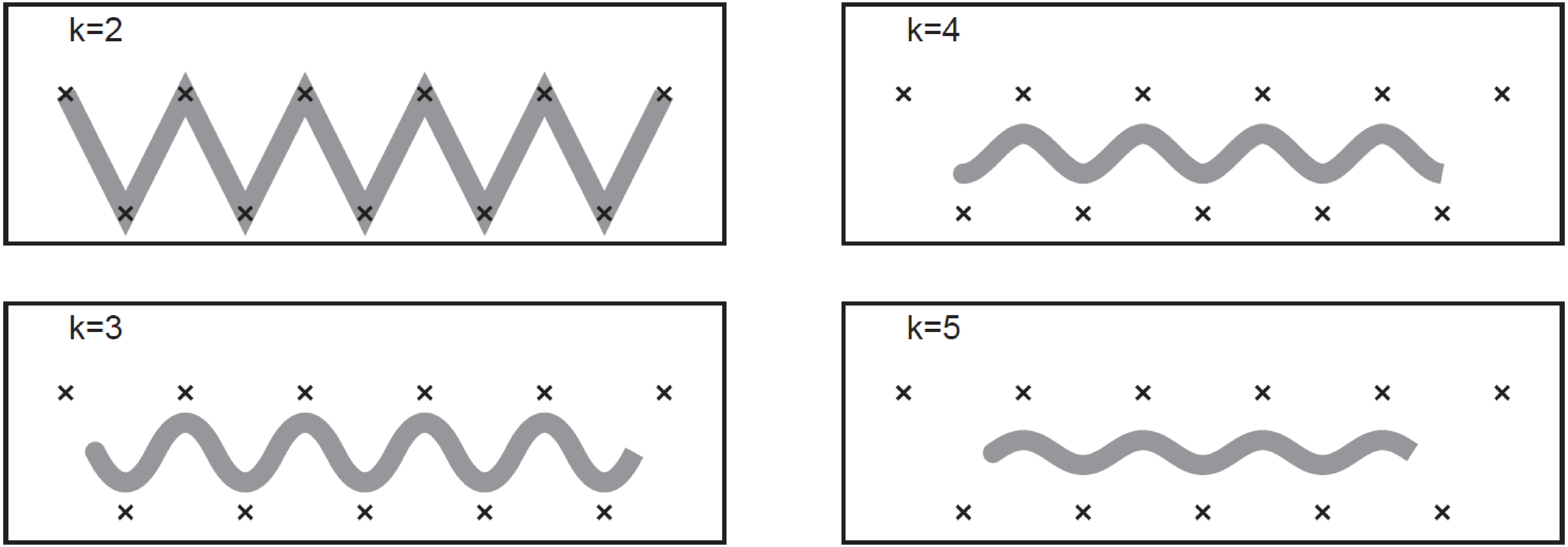
Quadratic B-spline

$$b_{i,3}(t) = \begin{cases} \dfrac{1}{2}u^2, & \text{if } i \le t < i+1, & u = t - i \\ -u^2 + u + \dfrac{1}{2}, & \text{if } i+1 \le t < i+1, & u = t - (i+1) \\ \dfrac{1}{2}(1-u)^2, & \text{if } i+2 \le t < i+3, & u = t - (i+2) \\ \dfrac{1}{2}u^2, & \text{otherwise.} & \end{cases}$$



B-Spline

Smoothing effect



Summary

- Spline segments
  - Linear
  - Quadratic
  - Hermite
  - Bezier
- Chaining splines
- Catmull Rom curve and B-splines
- Suggested reading: B1 Chapter 15