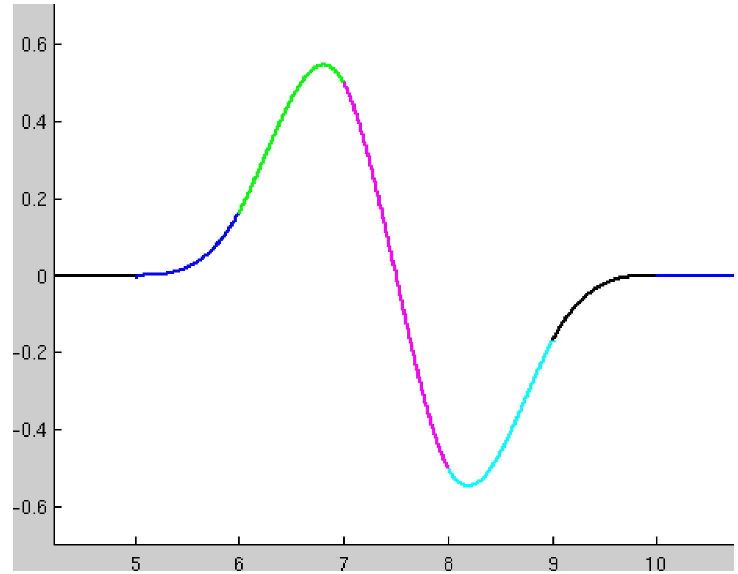
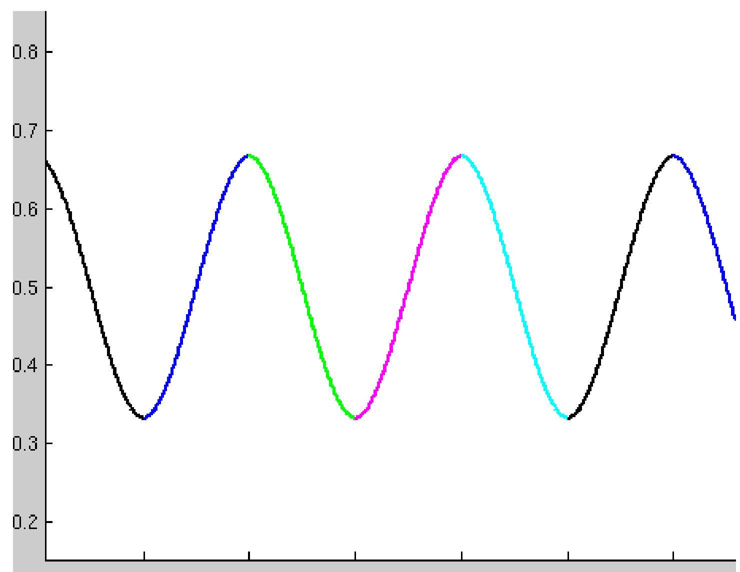
B(asis)-Splines

Ashish Myles
CISE, UF

Splines

- Piecewise polynomial
- More flexible than single polynomials
  - can have finite support
  - can be periodic
- Degree *d* splines – typically C^{d1} continuity



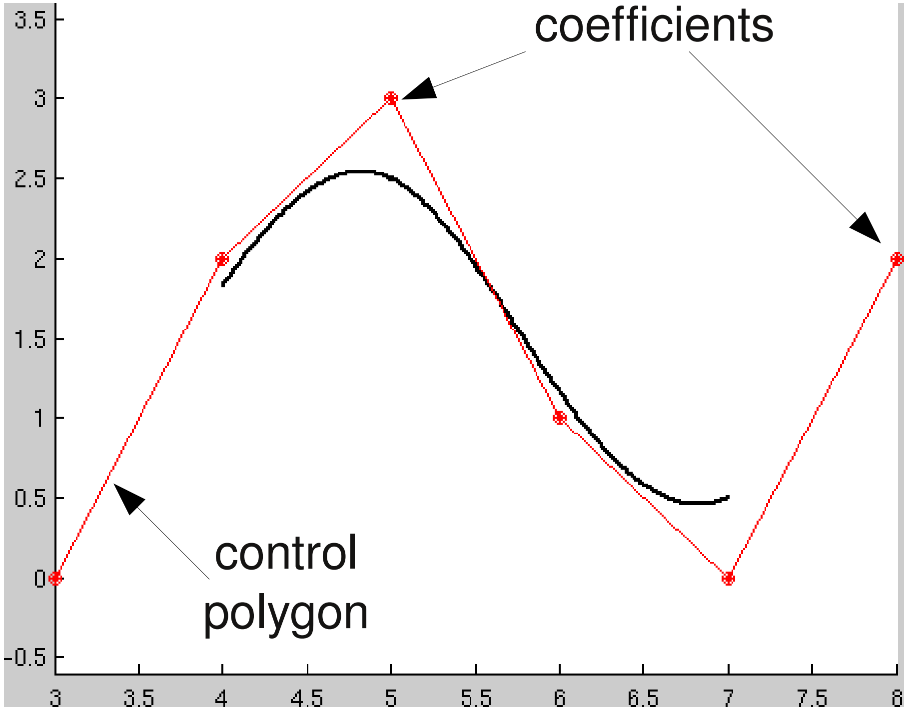


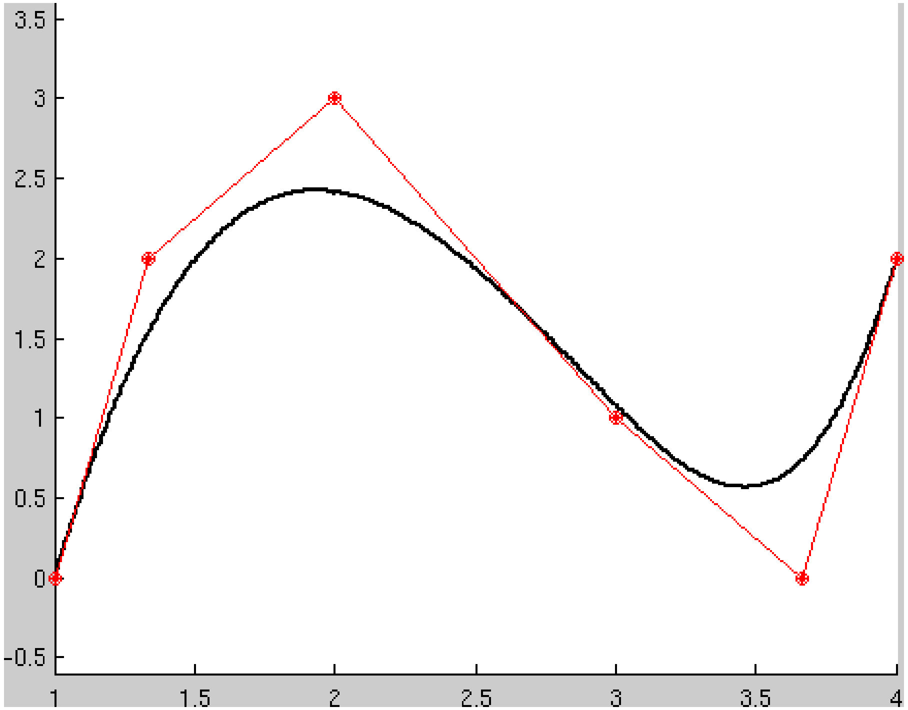
Some polynomial representations

- Polynomials
  - Power / Taylor series
  - Newton polynomials
  - Lagrange polynomials
  - Hermite polynomials
  - **Bézier** (very special case of B-spline)
- Splines
  - Box spline (for curves, same as uniform B-spline)
  - **B-spline**

B-spline examples

- http://www.cs.technion.ac.il/~cs234325/Applets/applets/bspline/GermanApplet.html
- http://www.engin.umd.umich.edu/CIS/course.des/cis577/projects/BSP/welcome.html
- http://www.cs.uwaterloo.ca/~r3fraser/splines/bspline.html
- http://www.doc.ic.ac.uk/~dfg/AndysSplineTutorial/BSplines.html





Polynomials as linear combinations

- Power basis $\{1, t, t^2, \ldots, t^d\}$

  $p(t) = \sum_{i=0}^{d} c_i t^i$

- Taylor basis $\{1, (t-t_0), (t-t_0)^2, \ldots, (t-t_0)^d\}$

  $p(t) = \sum_{i=0}^{d} c_i (t-t_0)^i$

- Bézier basis $\underbrace{\left\{(1-t)^d, d(1-t)^{d-1} t, \ldots, \binom{d}{i}(1-t)^{d-i} t^i, \ldots, t^d\right\}}_{\text{terms in the binomial expansion of } ((1-t)+t)^d \quad (=1)}$

  $p(t) = \sum_{i=0}^{d} c_i \binom{d}{i} (1-t)^{d-i} t^i$

Splines as linear combinations

- A linear combination of **spline** basis functions

- Defined by
  - k knots t_i
    - non-decreasing sequence specifying domain
    - determines basis functions (hence continuity and ranges)
  - n coefficients c_i
    - coefficients with which the basis functions are multiplied
  - degree d automatically determined: $k = n + d + 1$

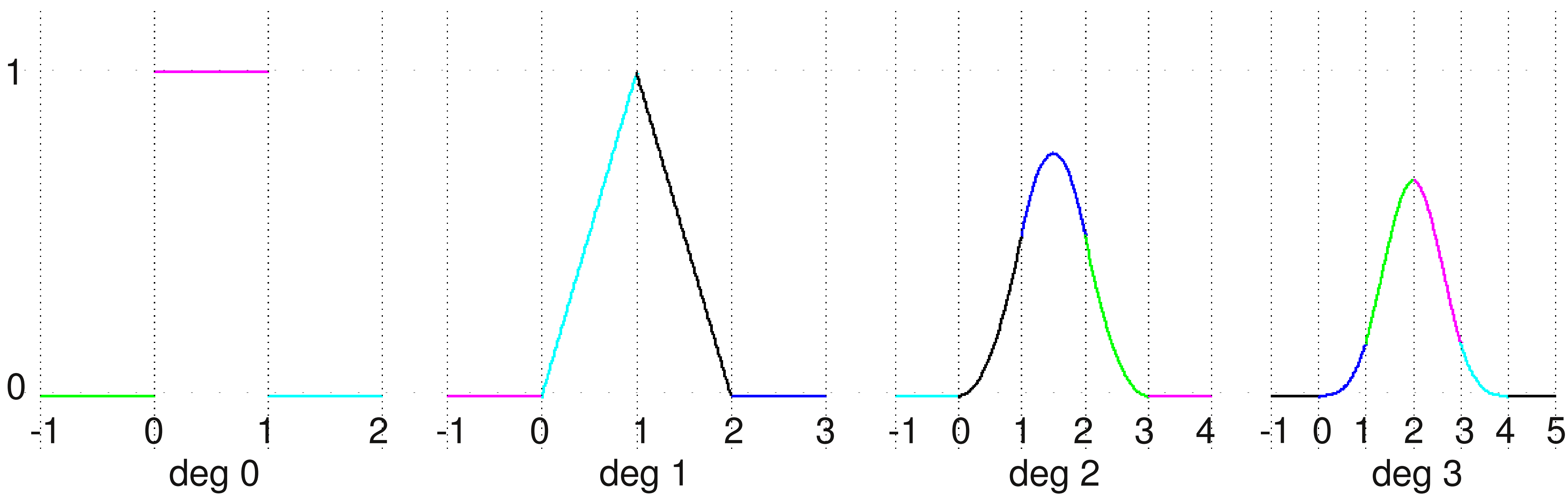
B-spline basis

- Basis function:

$$N_i^0(t) = \begin{cases} 1 & \text{if } t \in [t_i, t_{i+1}] \\ 0 & \text{otherwise} \end{cases}$$

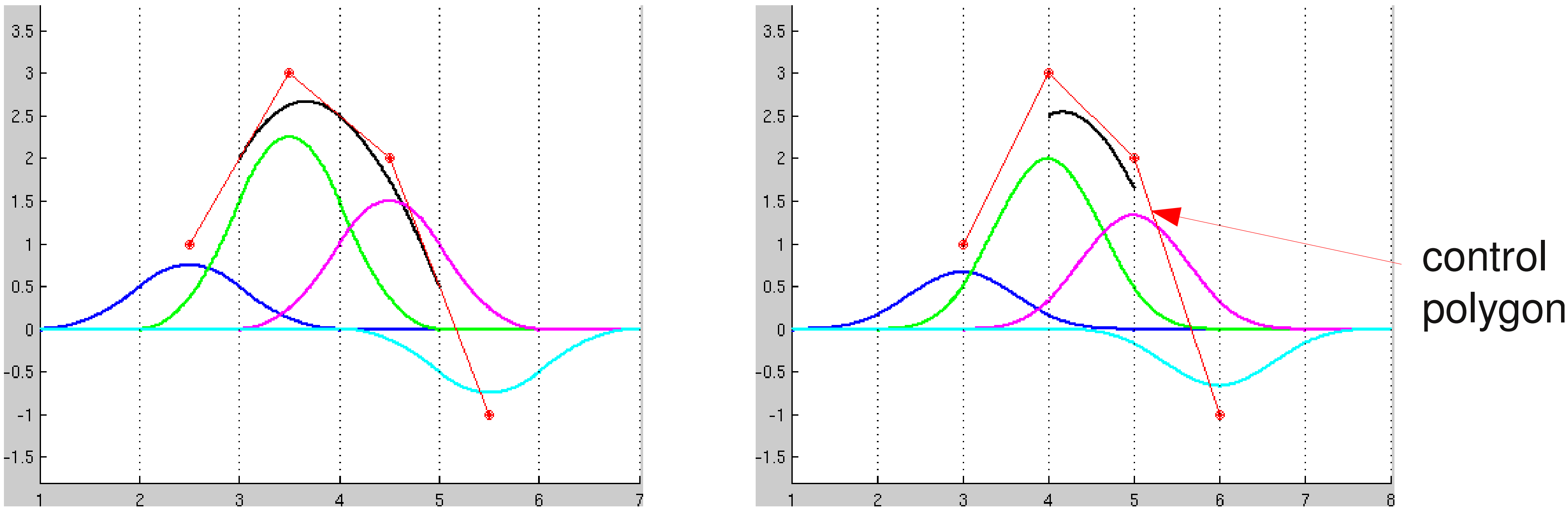
$$N_i^d(t) = \frac{t - t_i}{t_{i+d} - t_i} N_i^{d-1}(t) + \frac{t_{i+d+1} - t}{t_{i+d+1} - t_{i+1}} N_{i+1}^{d-1}(t)$$

- Non-neg. & finite support
- Shifts add up to 1 in their overlap
- (Uniform B-splines <=> **uniformly spaced knots**)



Spline in B-spline form

- Curve (degree *d*): $p(t) = \sum_i c_i N_i^d(t)$



deg 2 deg 3

- Example: $c_i = [1, 3, 2, -1]$ (y-coord only)

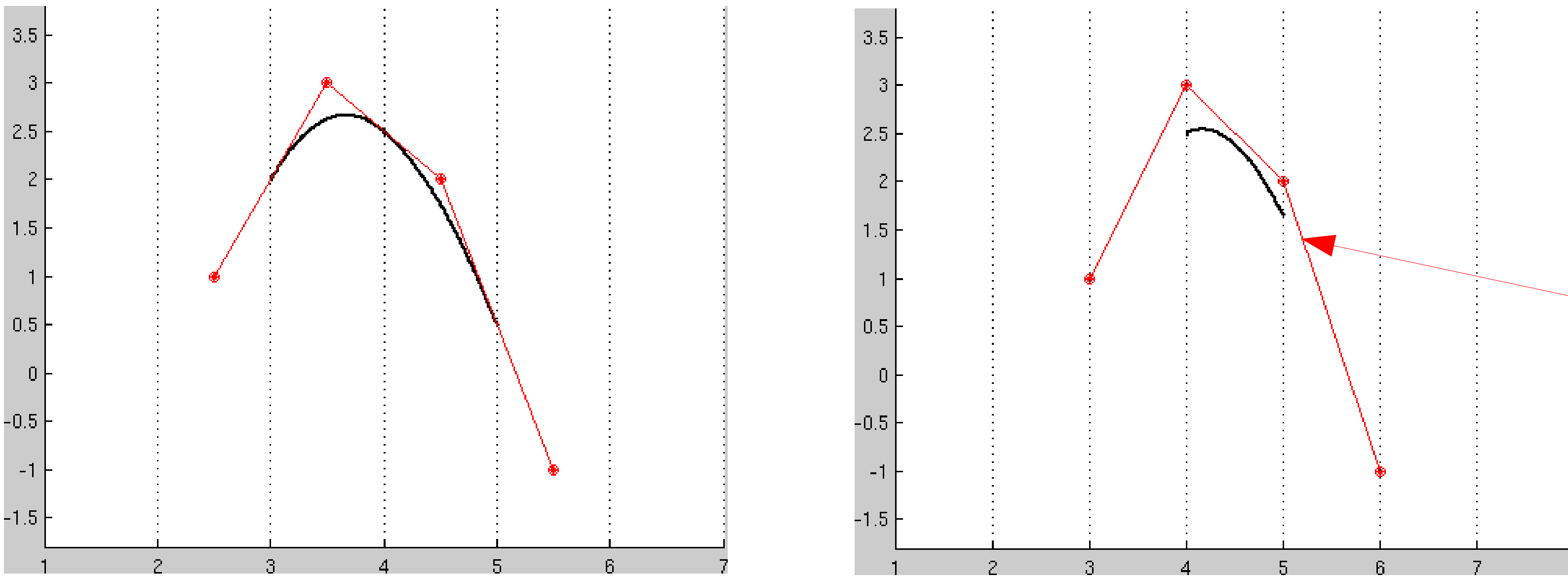
- **Greville abscissa:**
  natural x-coord for c_i

$$t_i^* = \frac{1}{d} \sum_{j=i+1}^{i+d} t_j$$

- $k = n + d + 1$ knots

Geometric Properties

- Curve (degree *d*): $p(t) = \sum_i c_i N_i^d(t)$ ← sum to 1, non-negative



deg 2   deg 3

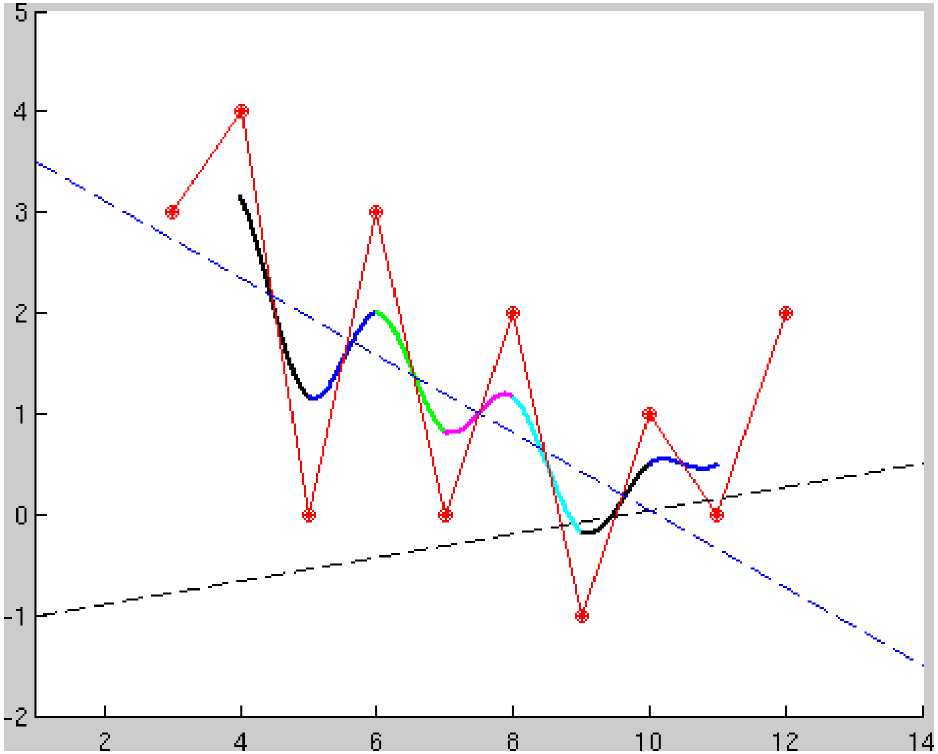
control polygon

- Affine invariance: $c_i' = A(c_i)$ <=> $p'(t) = A(p(t))$
- Convex hull property: curve lies within $CH(c_i)$

Variation diminishing property

- No line intersects the spline more times than it intersects the control polygon.

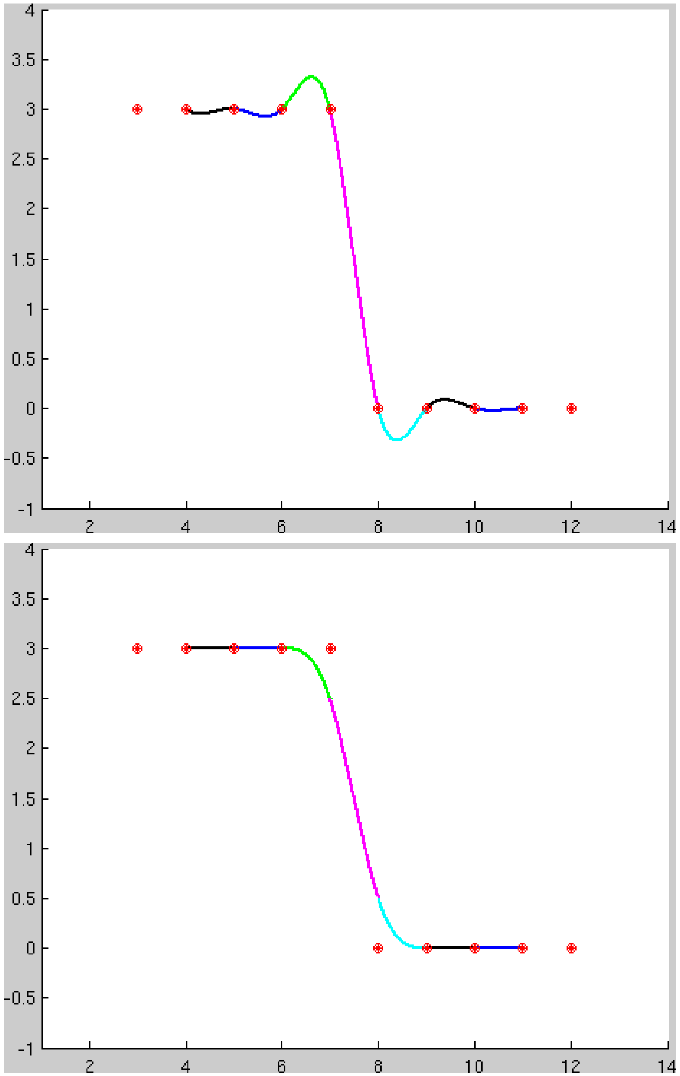
- i.e. The curve will not wiggle more than the control polygon.



An alternative to interpolation

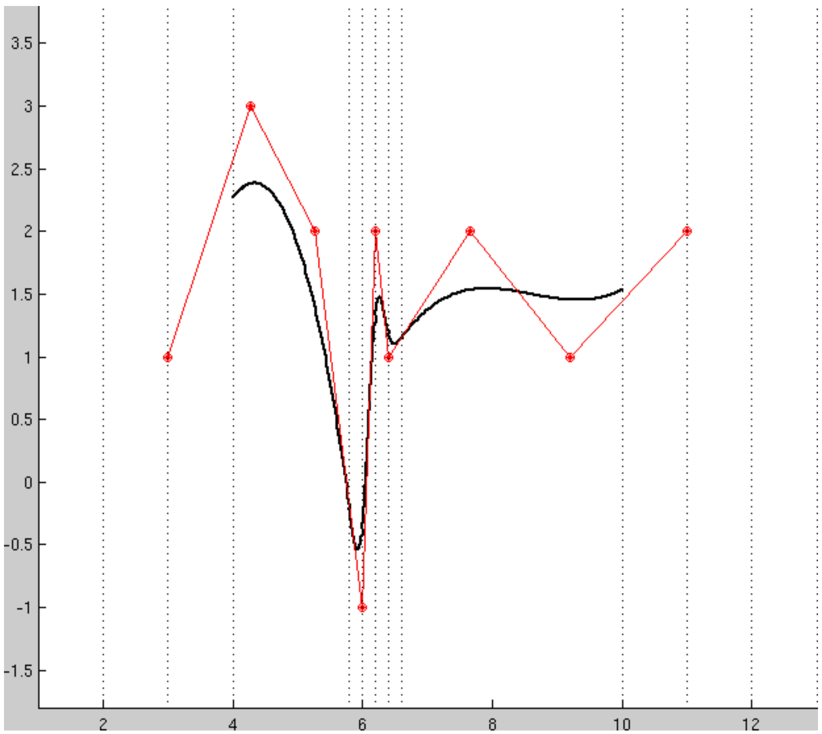
- Interpolating samples suffers from the Gibb's phenomenon

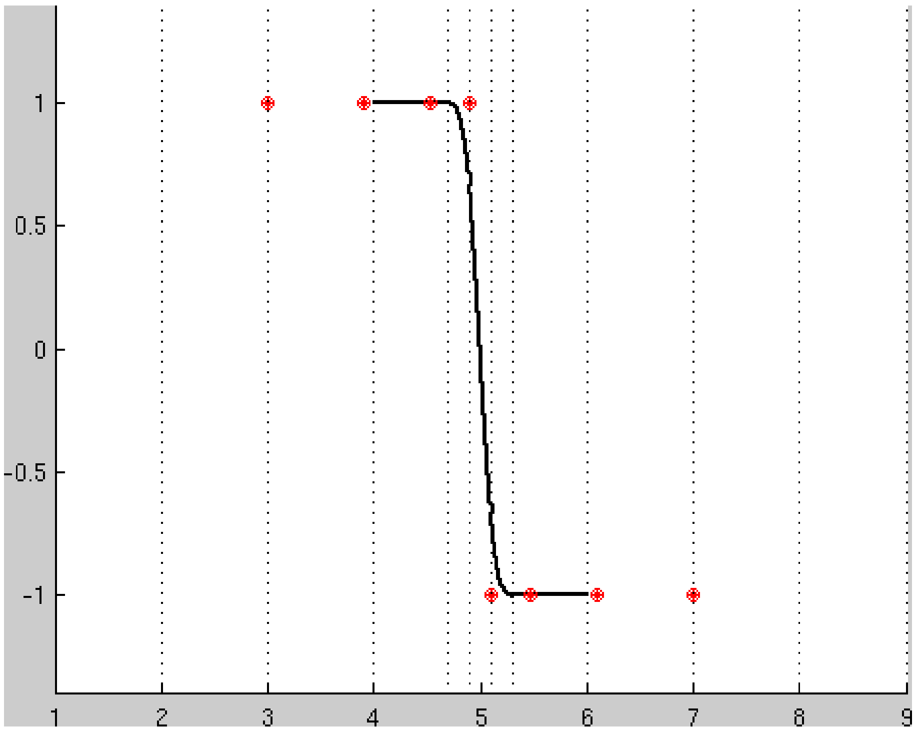
- Treating samples as coefficients has no such problems – curve can't wiggle more than coefficients.



Examples of non-uniform splines

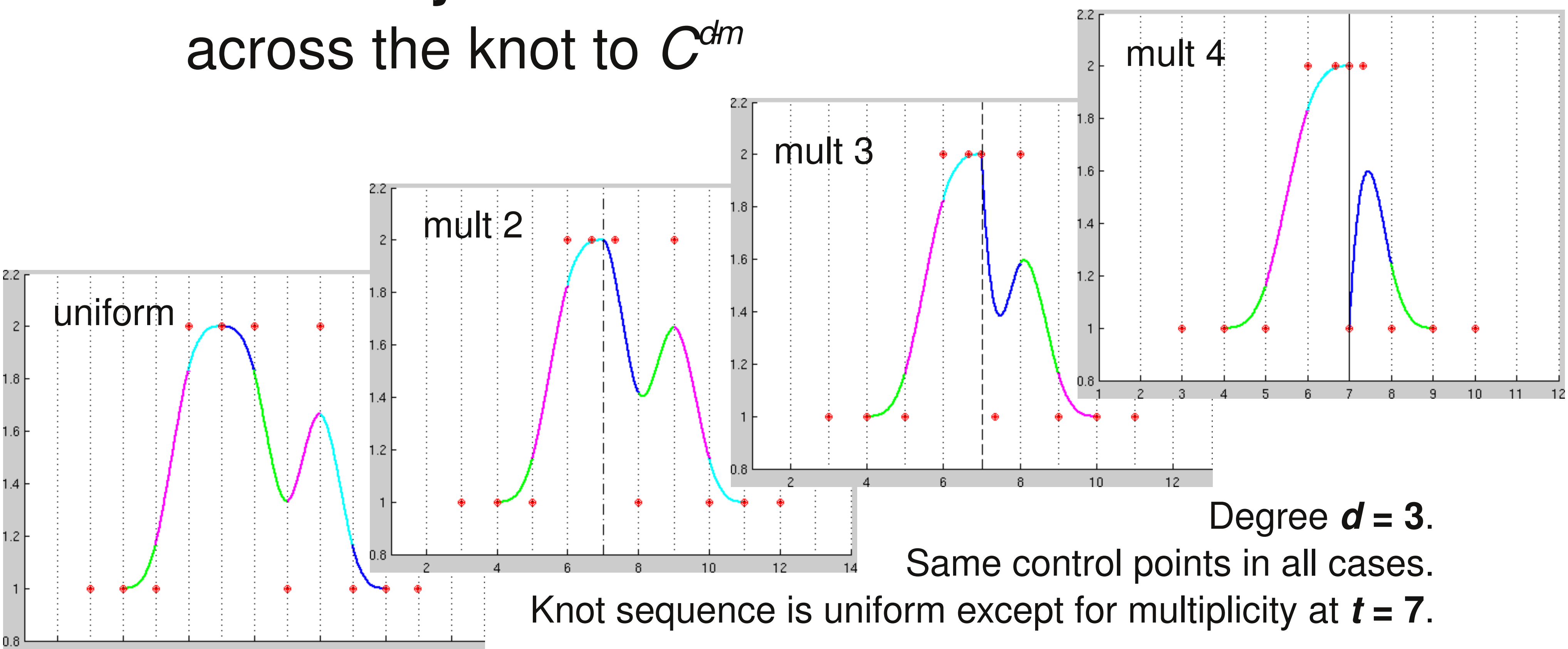
- Knot sequence can be denser in areas needing more degrees of freedom.





Decreasing inherent continuity

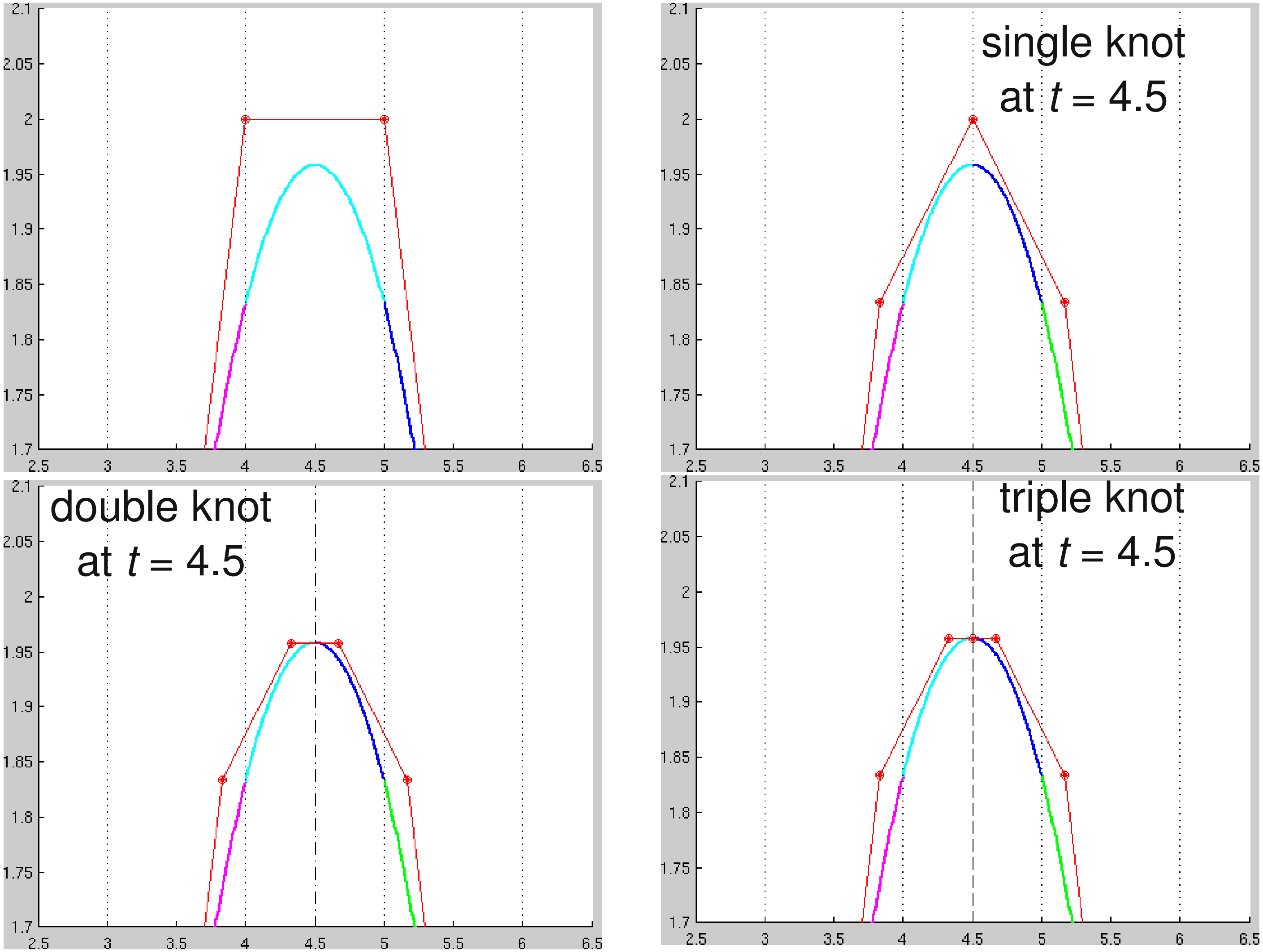
- Knot **multiplicity**
  - Repeating a knot *m* times decreases the **inherent continuity** of the basis functions across the knot to C^{d-m}



Degree ***d*** **= 3**.
Same control points in all cases.
Knot sequence is uniform except for multiplicity at ***t*** **= 7**.

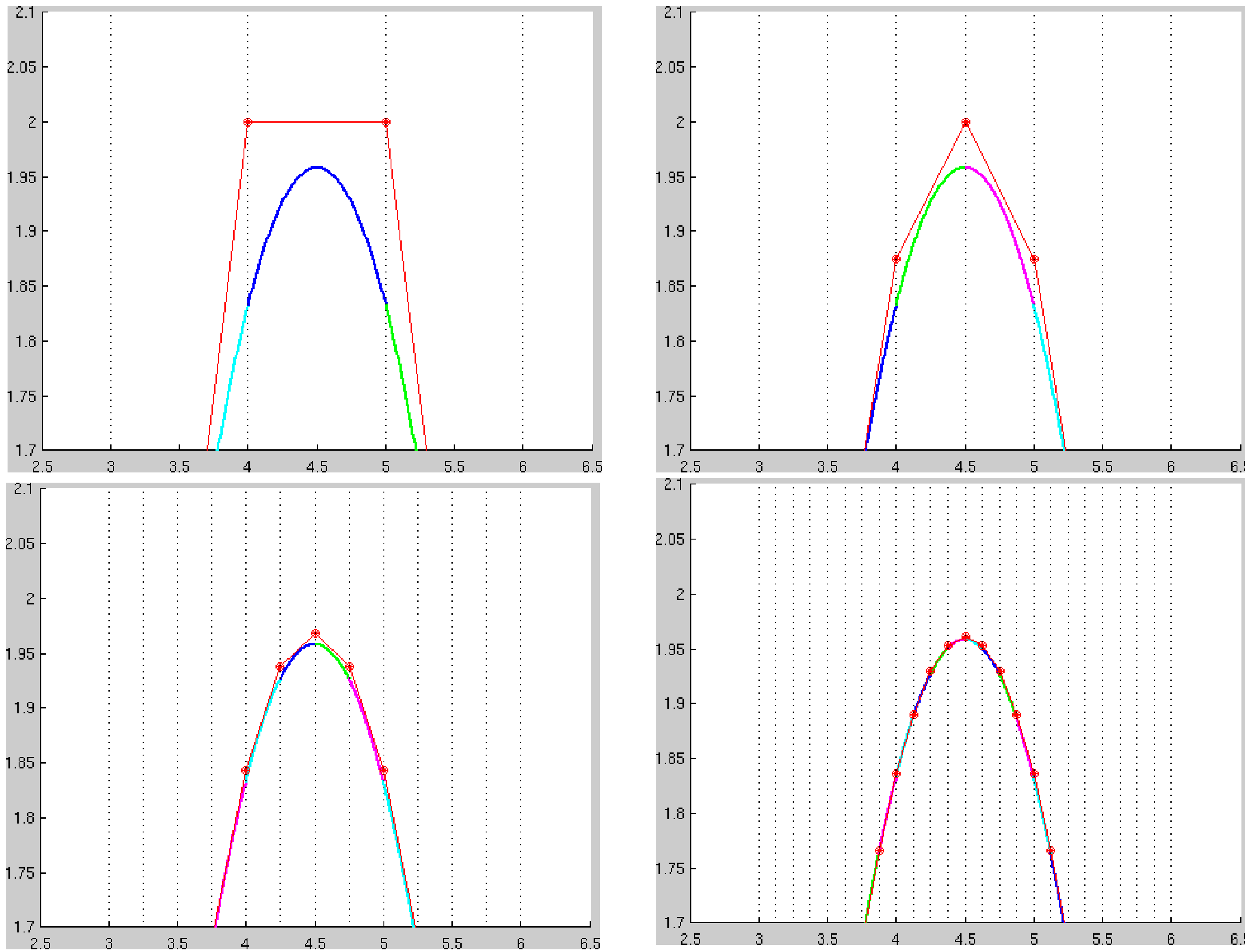
Evaluation – deBoor's Algorithm

- Evaluate at ***t* = 4.5** by repeated knot insertion without changing the underlying function.



Convergence under knot insertion

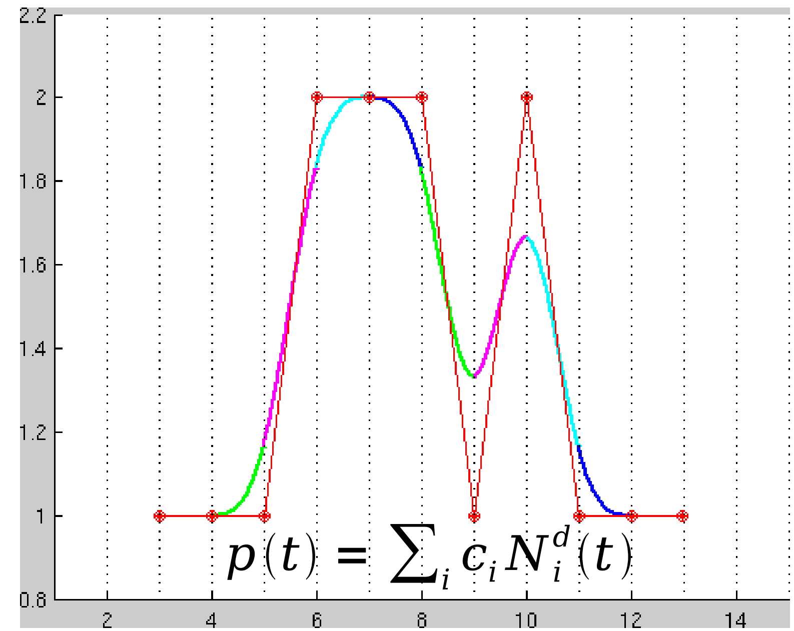
- Repeated uniform knot insertion converges to function as fast as $O(h^2)$, with h = knot width.

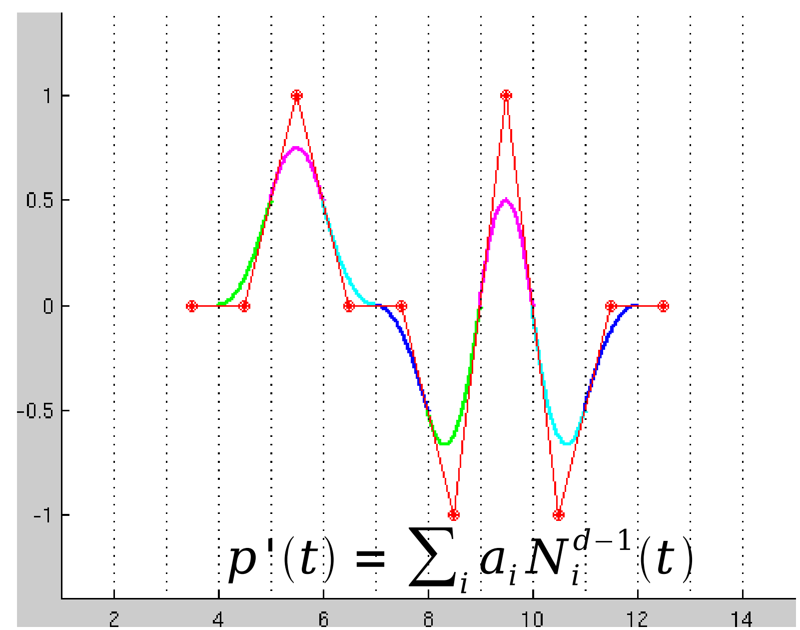


Derivatives

- Compute using **divided differences**
  - deg 1 lower
  - continuity 1 lower
  - domain the same

$$a_i = \frac{d}{t_{i+d+1} - t_{i+1}} \left(c_{i+1} - c_i \right)$$





Matlab spline toolbox

- Written by deBoor himself
- I used for my figures:
  - spmak, spapi – create/interpolate a B-spline
  - fnplt – plot the B-spline
  - fnrfn – do knot insertion
  - fnder, fnint – differentiation and integration
- It's well documented and comes with tutorials and demos

Summary

- The B-spline form is
  - geometrically intuitive
  - numerically robust
  - easy to differentiate
  - easy to make discontinuous
  - very, very knotty

- Matlab spline toolbox

More terms to look up

- Tensor-product B-splines
  - for surfaces, volumes
- Bézier
  - tensor-product (special case of B-splines)
  - total-degree (triangular) – no good B-spline equivalent
- Blossoms
  - Excellent theoretical tool
  - Inefficient for implementation, though

References

- Curves and Surfaces for CAGD
  - Gerald Farin
- Bézier and B-Spline Techniques
  - Hartmut Prautzsch, Wolfgang Boehm, and Marco Paluszny
- Matlab spline toolbox documentation & demos