

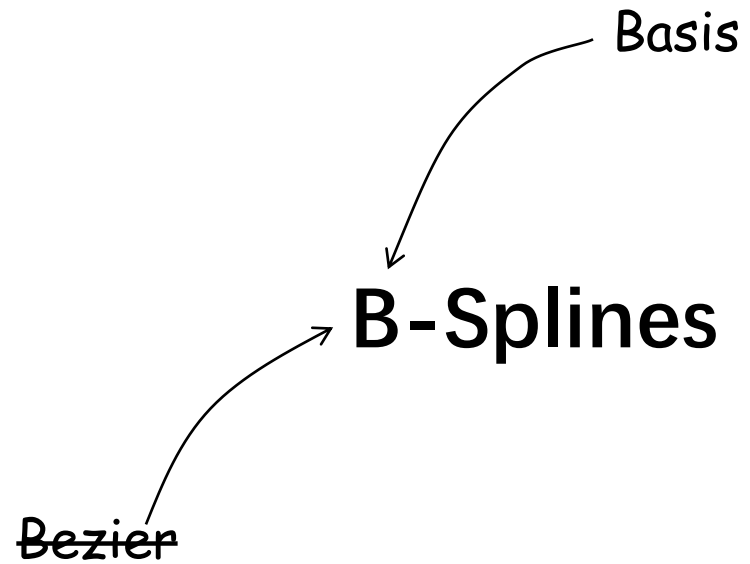
# 计算机辅助几何设计

## 2021秋学期

### B-Splines

陈仁杰

中国科学技术大学



Mathematical view: spline functions

Graphics view: spline curves (created using spline functions)

# Motivation

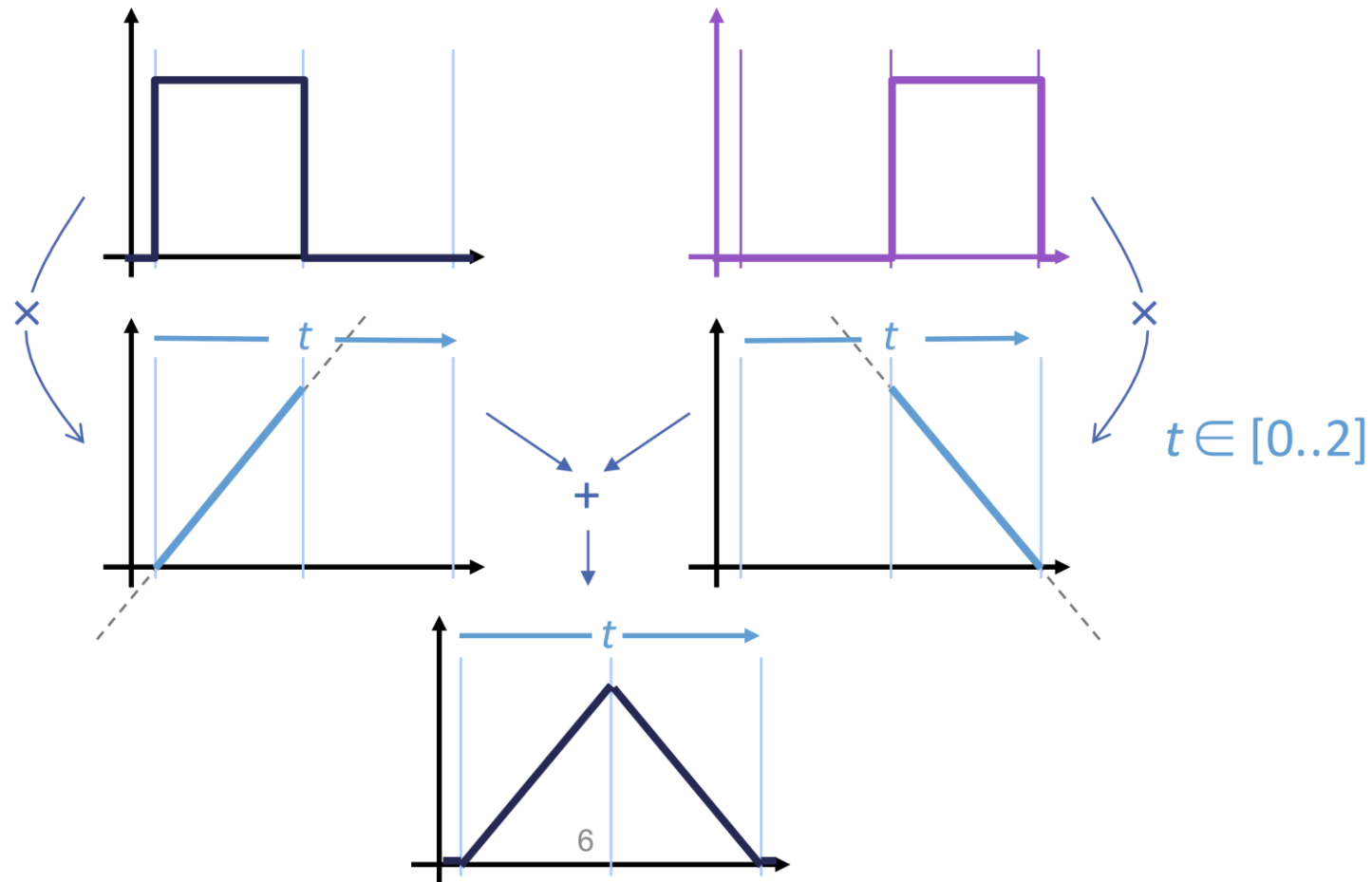
- Back to the algebraic approach for Bezier curves  
→ Bernstein polynomials
- Problem: global influence of the Bezier points
- Introduction of new basis function  
→ B-spline functions

# Some history

- **Early use of splines on computers for data interpolation**
  - Ferguson at Boeing, 1963
  - Gordon and de Boor at General Motors
  - B-splines, de Boor 1972
- **Free form curve design**
  - Gordon and Riesenfeld, 1974 → B-splines as a generalization of Bezier curves

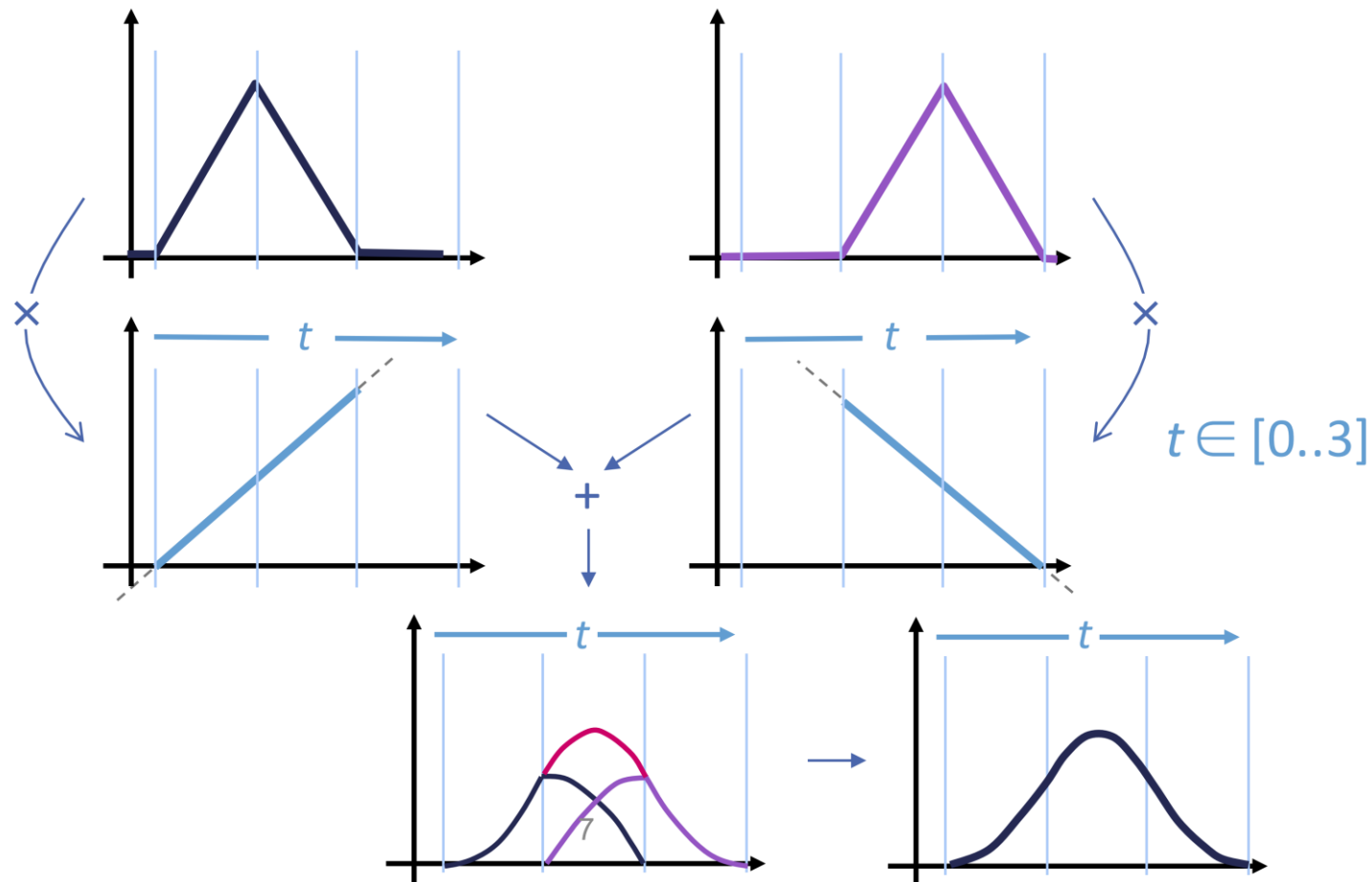
# Repeated linear interpolation

Another way to increase smoothness:



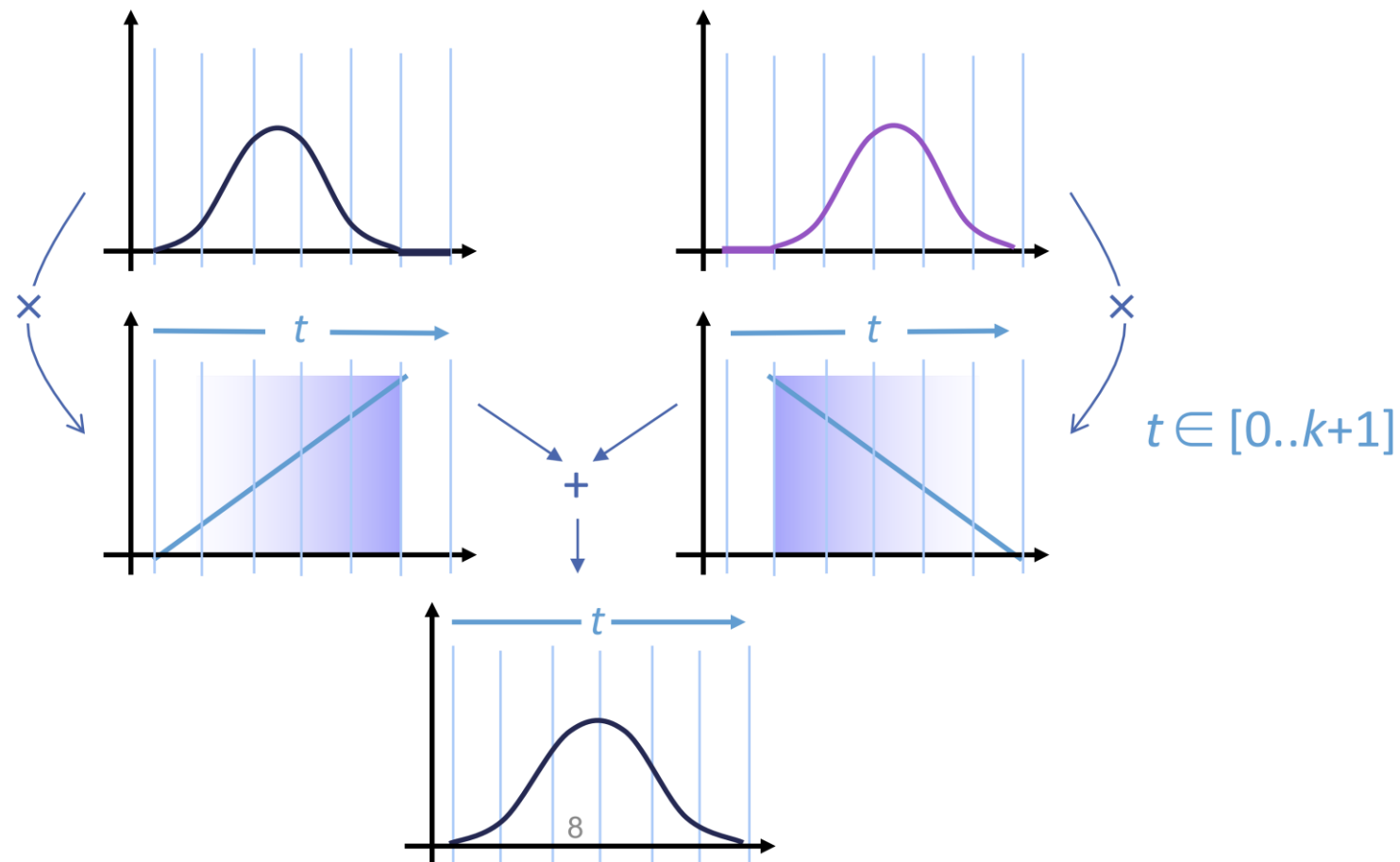
# Repeated linear interpolation

- Another way to increase smoothness:



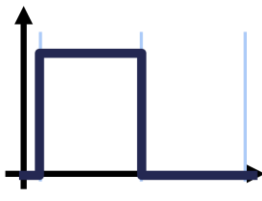
# Repeated linear interpolation

- Another way to increase smoothness

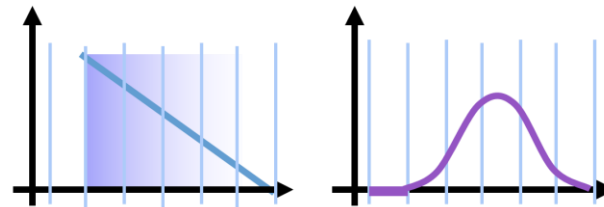
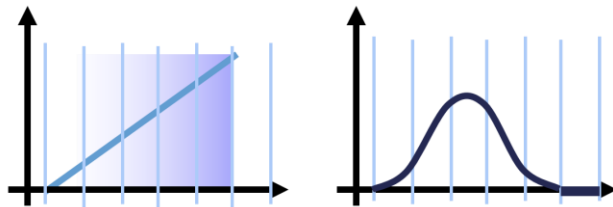


# De Boor Recursion: uniform case

- The **uniform** B-spline basis of order  $k$  (degree  $k - 1$ ) is given as

$$N_i^1(t) = \begin{cases} 1, & \text{if } i \leq t < i+1 \\ 0, & \text{otherwise} \end{cases}$$


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



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



# B-spline curves: general case

- Given: knot sequence  $t_0 < t_1 < \dots < t_n < \dots < t_{n+k}$   
( $(t_0, t_1, \dots, t_{n+k})$  is called knot vector)
- Normalized B-spline functions  $N_{i,k}$  of the order  $k$  (degree  $k - 1$ ) are defined as:

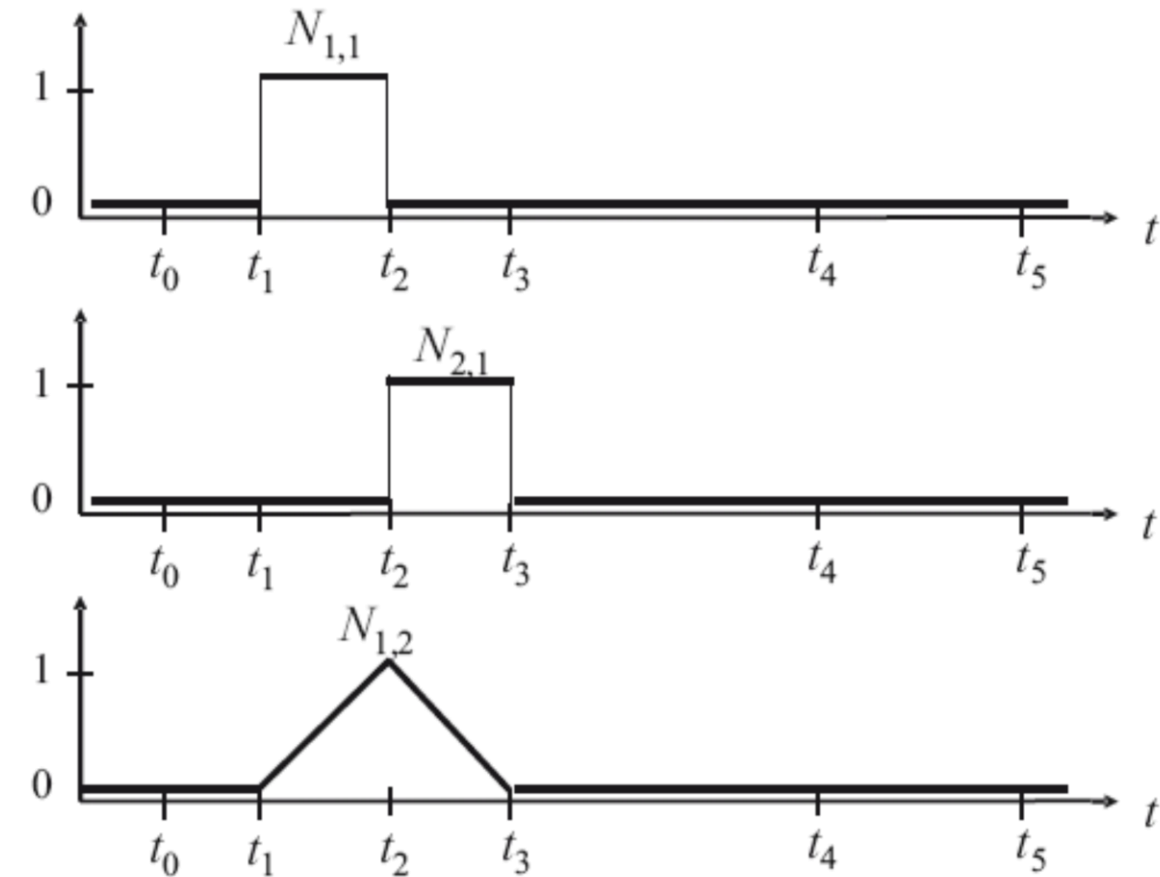
$$N_{i,1}(t) = \begin{cases} 1, & t_i \leq t < t_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t)$$

for  $k > 1$  and  $i = 0, \dots, n$

- **Remark:**
  - If a knot value is repeated  $k$  times, the denominator may vanish
  - In this case: The fraction is treated as a zero

# Example

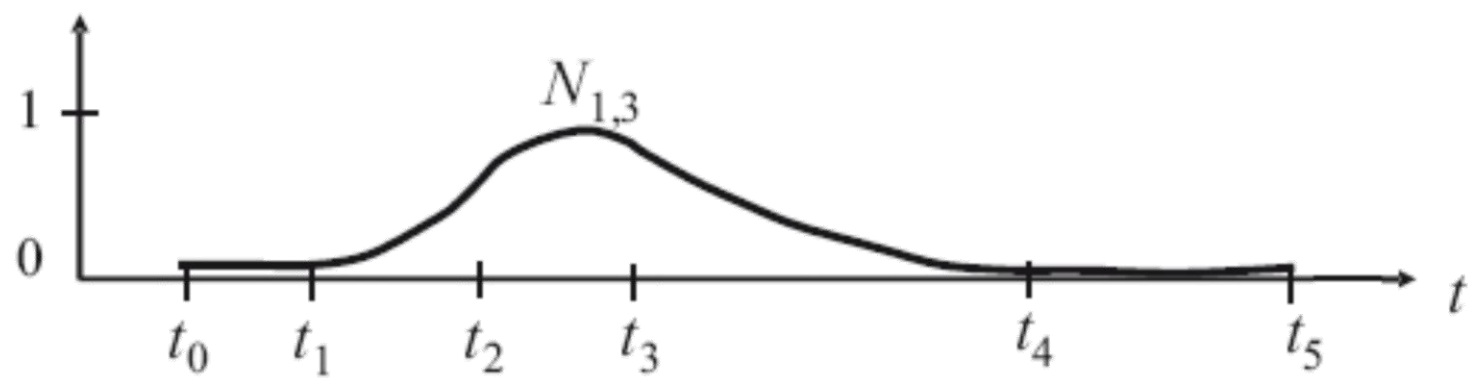
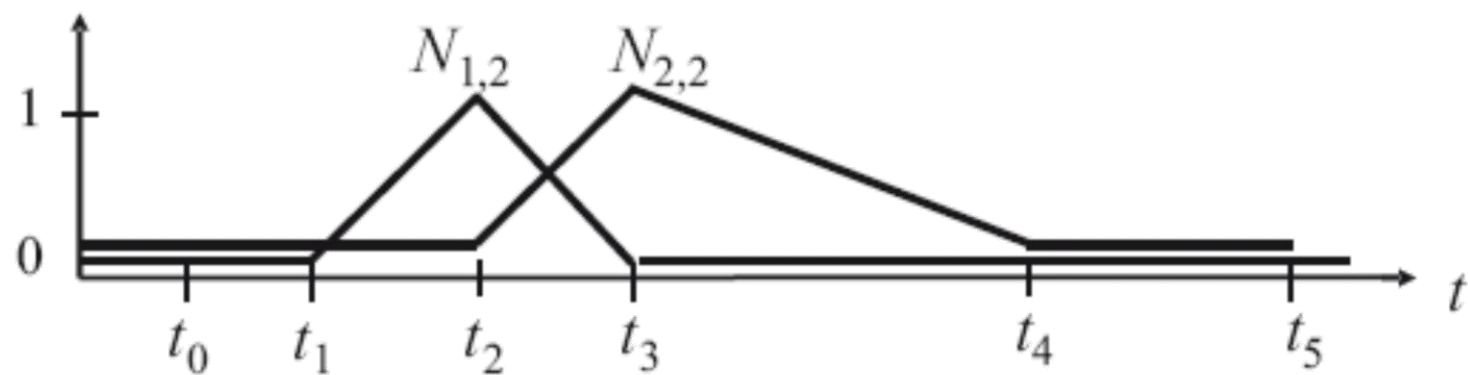


$$N_{i,1}(t) = \begin{cases} 1, & t_i \leq t < t_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

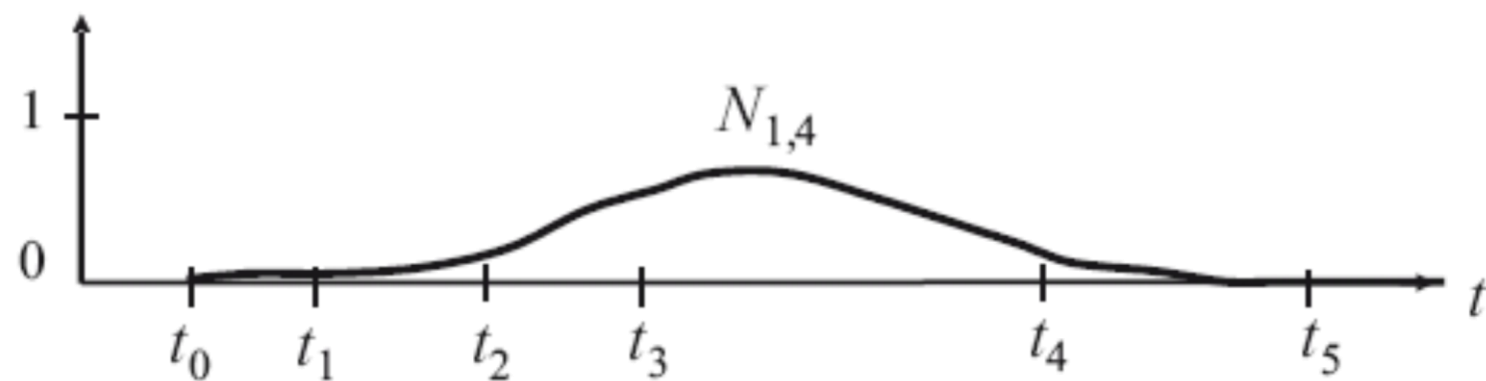
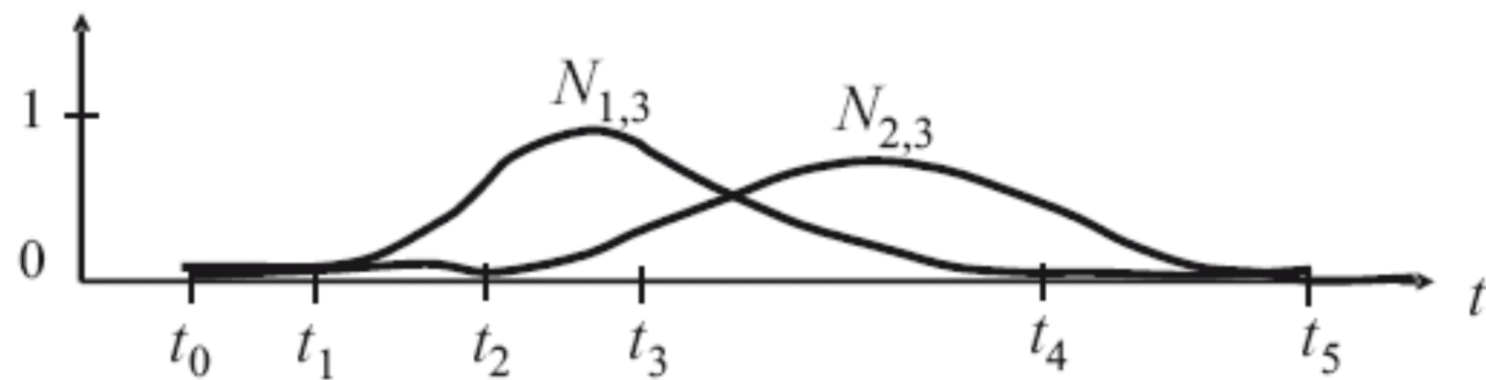
$$N_{i,k}(t) = \frac{t-t_i}{t_{i+k-1}-t_i} N_{i,k-1}(t) + \frac{t_{i+k}-t}{t_{i+k}-t_{i+1}} N_{i+1,k-1}(t)$$

for  $k > 1$  and  $i = 0, \dots, n$

# Example

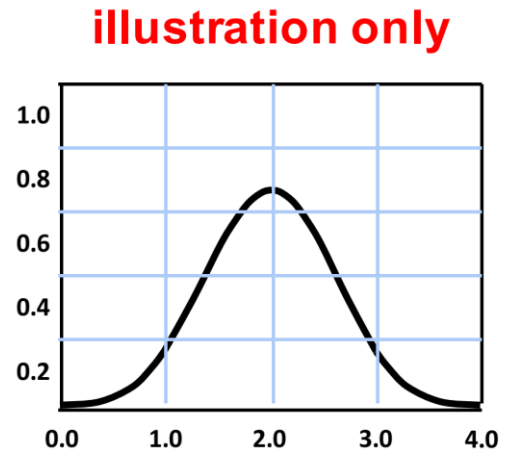


# Example



# Key Ideas

- We design one basis function  $b(t)$
- Properties:
  - $b(t)$  is  $C^2$  continuous
  - $b(t)$  is piecewise polynomial, degree 3 (cubic)
  - $b(t)$  has local support
  - Overlaying shifted  $b(t + i)$  forms a partition of unity
  - $b(t) \geq 0$  for all  $t$
- In short:
  - All desirable properties build into the basis
  - Linear combinations will inherit these



# Shifted Basis Functions

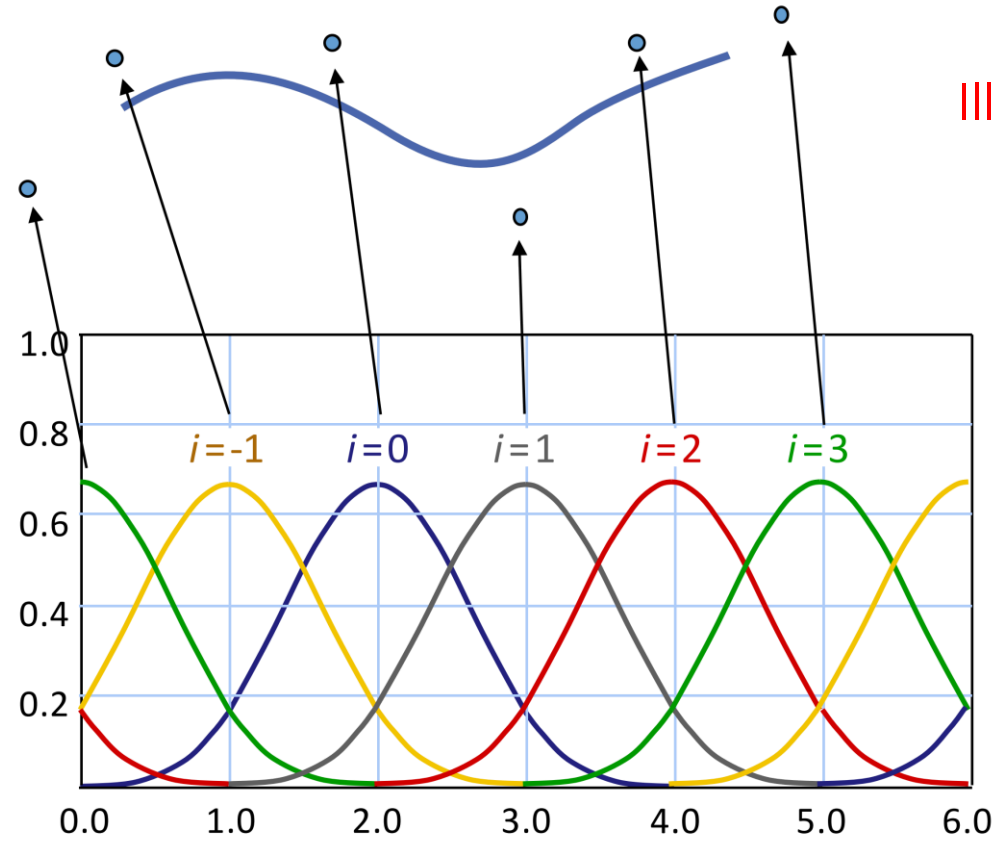


Illustration only

Shifted basis function  $b(t)$

# Basis properties

- For the so defined basis functions, the following properties can be shown:
  - $N_{i,k}(t) > 0$  for  $t_i < t < t_{i+k}$
  - $N_{i,k}(t) = 0$  for  $t_0 < t < t_i$  or  $t_{i+k} < t < t_{n+k}$
  - $\sum_{i=0}^n N_{i,k}(t) = 1$  for  $t_{k-1} \leq t \leq t_{n+1}$
- For  $t_i \leq t_j \leq t_{i+k}$ , the basis functions  $N_{i,k}(t)$  are  $C^{k-2}$  at the knots  $t_j$
- The interval  $[t_i, t_{i+k}]$  is called support of  $N_{i,k}$

# B-spline curves

- Given:  $n + 1$  control points  $\mathbf{d}_0, \dots, \mathbf{d}_n \in \mathbb{R}^3$   
knot vector  $T = (t_0, \dots, t_n, \dots, t_{n+k})$
- Then, the B-spline curve  $\mathbf{x}(t)$  of the order  $k$  is defined as

$$\mathbf{x}(t) = \sum_{i=0}^n N_{i,k}(t) \cdot \mathbf{d}_i$$

- The points  $\mathbf{d}_i$  are called *de Boor points*

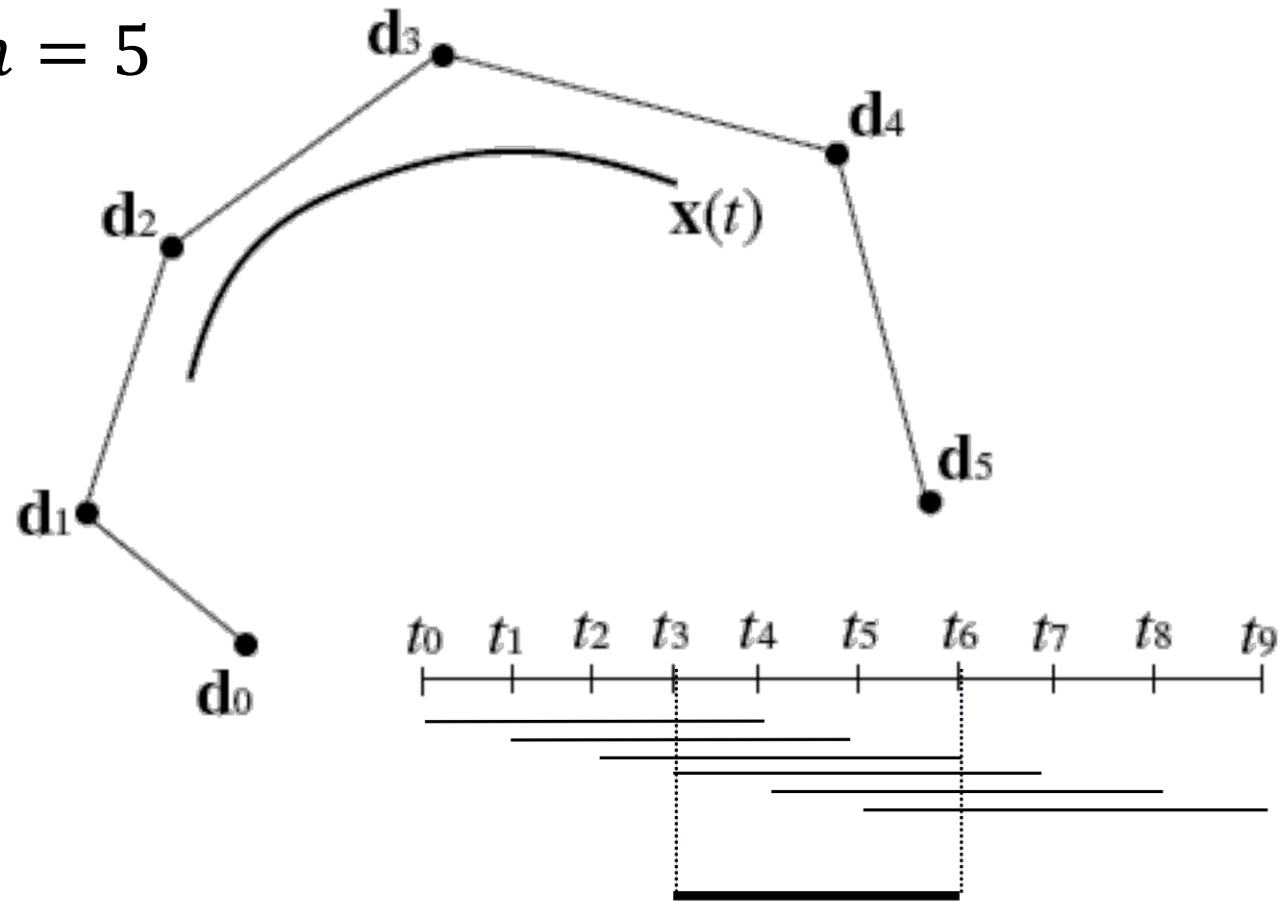
**Carl R. de Boor**

German-American mathematician  
University of Wisconsin-Madison



# Example

- $k = 4, n = 5$



Support intervals of  $N_{i,k}$

Curve defined in interval  $t_3 \leq t \leq t_6$

# B-spline curves

## Multiple weighted knot vectors

- So far:  $T = (t_0, \dots, t_n, \dots, t_{n+k})$  with  $t_0 < t_1 < \dots < t_{n+k}$
- Now: also multiple knots allowed, i.e. with  $t_0 \leq t_1 \leq \dots \leq t_{n+k}$
- The recursive definition of the B spline function  $N_{i,k}$  ( $i = 0, \dots, n$ ) works nonetheless, as long as no more than  $k$  knots coincide

# B-spline curves

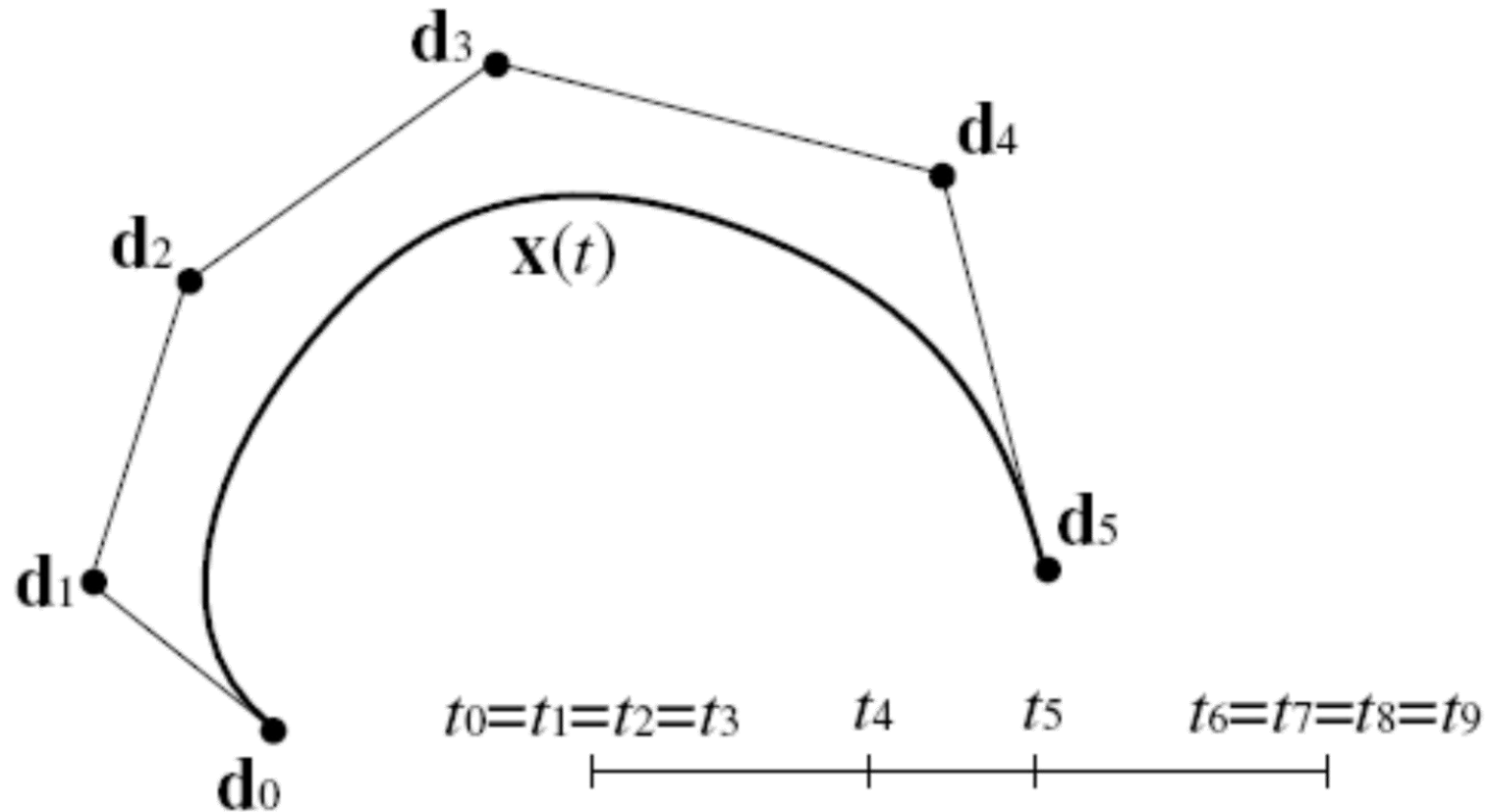
## Effect of multiple knots:

- set:  $t_0 = t_1 = \cdots = t_{k-1}$
- and  $t_{n+1} = t_{n+2} = \cdots = t_{n+k}$

$\mathbf{d}_0$  and  $\mathbf{d}_n$  are interpolated

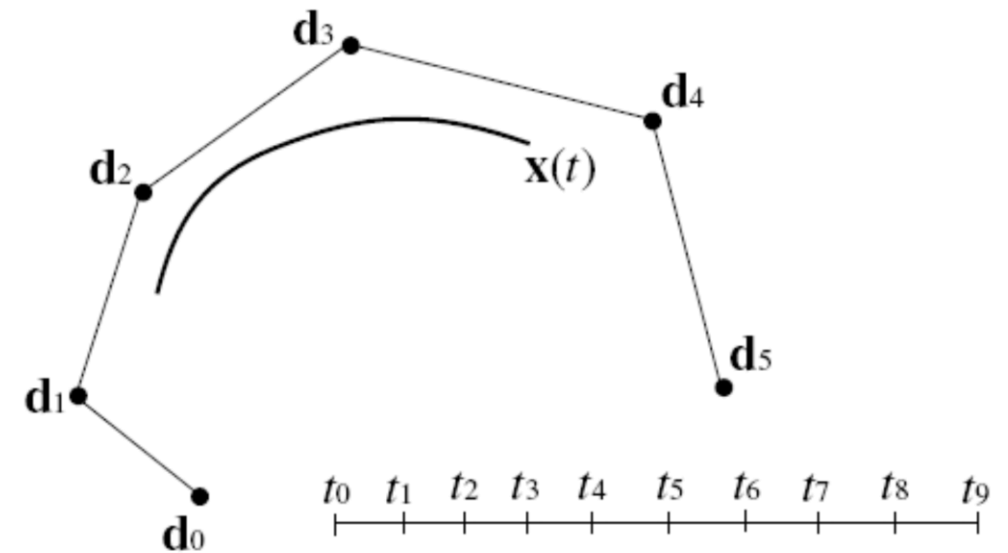
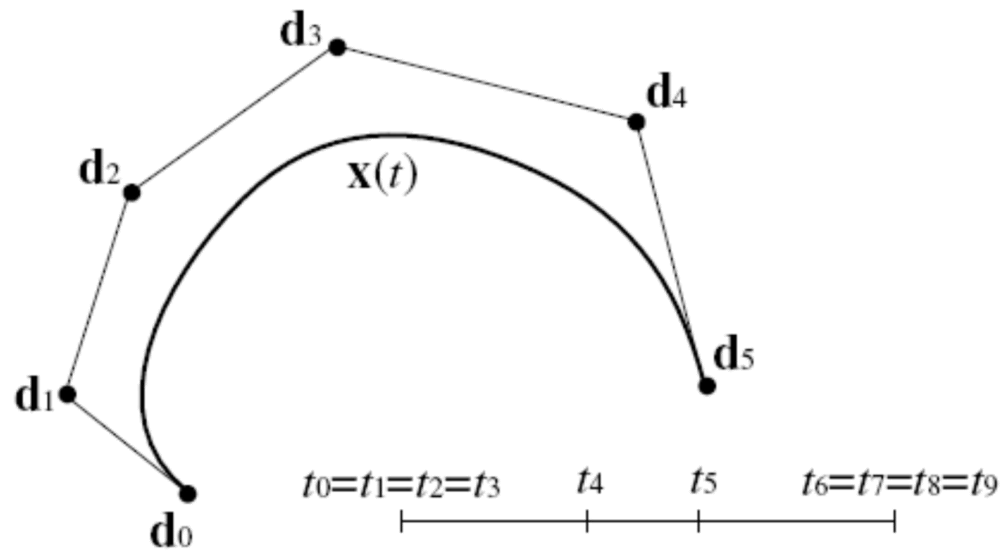
# B-spline curves

- Example:  $k = 4, n = 5$



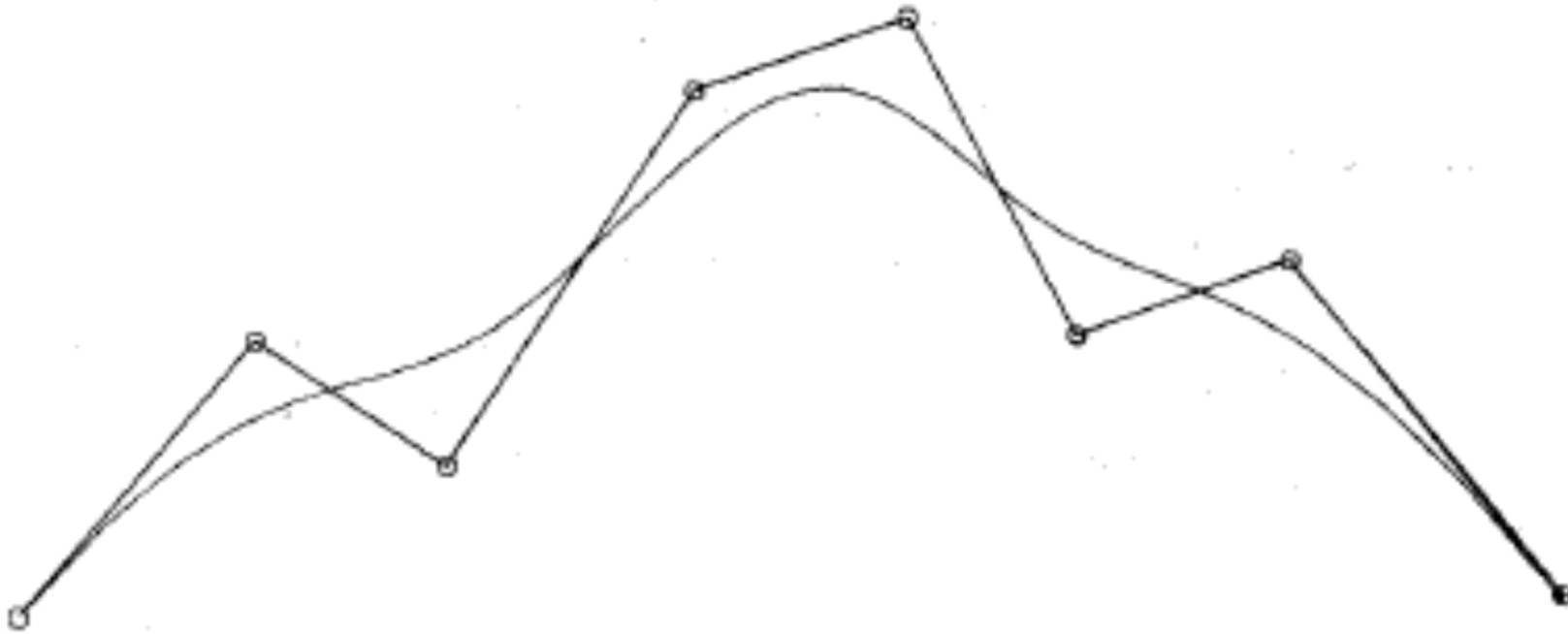
# B-spline curves

- Example:  $k = 4, n = 5$



# B-spline curves

- Further example



# B-spline curves

## Interesting property:

- B-spline functions  $N_{i,k}$  ( $i = 0, \dots, k - 1$ ) of the order  $k$  over the knot vector  $T = (t_0, t_1, \dots, t_{2k-1}) = (\underbrace{0, \dots, 0}_{k \text{ times}}, \underbrace{1, \dots, 1}_{k \text{ times}})$

are Bernstein polynomials  $B_i^{k-1}$  of degree  $k - 1$

# B-spline curves properties

- Given:
  - $T = (\underbrace{t_0, \dots, t_0}_{k \text{ times}}, t_k, \dots, t_n, \underbrace{t_{n+1}, \dots, t_{n+1}}_{k \text{ times}})$
  - de Boor polygon  $\mathbf{d}_0, \dots, \mathbf{d}_n$
- Then, the following applies for the related B-spline curve  $\mathbf{x}(t)$ :



# B-spline curves properties

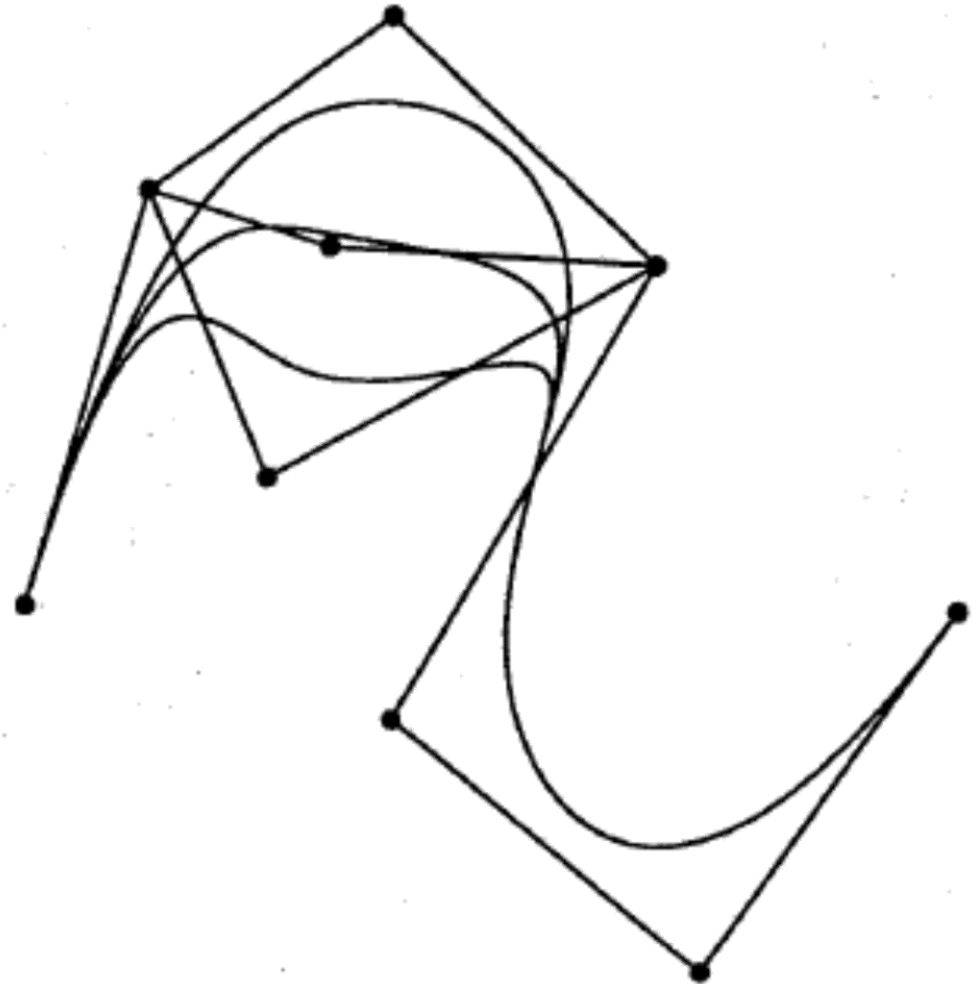
- $\mathbf{x}(t_0) = \mathbf{d}_0, \mathbf{x}(t_{n+1}) = \mathbf{d}_n$  (end point interpolation)
- $\mathbf{x}'(t_0) = \frac{k-1}{t_k - t_0} (\mathbf{d}_1 - \mathbf{d}_0)$  (tangent direction at  $\mathbf{d}_0$ , similar in  $\mathbf{d}_n$ )
- $\mathbf{x}(t)$  consists of  $n - k + 2$  polynomial curve segments of degree  $k - 1$  (assuming no multiple inner knots)

# B-spline curves properties

- Multiple inner knots  $\Rightarrow$  reduction of continuity of  $x(t)$ .  
 $l$ -times inner knot ( $1 \leq l < k$ ) means  
 $C^{k-l-1}$ -continuity
- Local impact of the de Boor points: moving of  $d_i$  only changes the curve in the region  $[t_i, t_{i+k}]$
- The insertion of new de Boor points does not change the polynomial degree of the curve segments

# B-spline curves properties

Locality of B-spline curves



# B-spline curves

## Evaluation of B-spline curves

- Using B-spline functions
- Using the de Boor algorithm  
Similar algorithm to the de Casteljau algorithm for Bezier curves;  
consists of a number of linear interpolations on the de Boor polygon

# The de Boor algorithm

- Given:

$\mathbf{d}_0, \dots, \mathbf{d}_n$ : de Boor points

$(t_0, \dots, t_{k-1} = t_0, t_k, t_{k+1}, \dots, t_n, t_{n+1}, \dots, t_{n+k} = t_{n+1})$ :

Knot vector

- wanted:

Curve point  $\mathbf{x}(t)$  of the B-spline curve of the order  $k$

# The de Boor algorithm

1. Search index  $r$  with  $t_r \leq t < t_{r+1}$

2. for  $i = r - k + 1, \dots, r$

$$d_i^0 = d_i$$

- for  $j = 1, \dots, k - 1$

for  $i = r - k + 1 + j, \dots, r$

$$d_i^j = \left(1 - \alpha_i^j\right) \cdot d_{i-1}^{j-1} + \alpha_i^j \cdot d_i^{j-1}$$

$$\text{with } \alpha_i^j = \frac{t - t_i}{t_{i+k-j} - t_i}$$

Then:  $d_r^{k-1} = x(t)$

# B-spline curves

- The intermediate coefficients  $d_i^j(t)$  can be placed into a triangular shaped matrix of points – the de Boor scheme:

$$d_{r-k+1} = d_{r-k+1}^0$$

$$d_{r-k+2} = d_{r-k+2}^0 \quad d_{r-k+2}^1$$

...

$$d_{r-1} = d_{r-1}^0 \quad d_{r-1}^1 \quad \dots \quad d_{r-1}^{k-2}$$

$$d_r = d_r^0 \quad d_r^1 \quad \dots \quad d_r^{k-2} \quad d_r^{k-1} = x(t)$$

# B-spline curves: interpolation

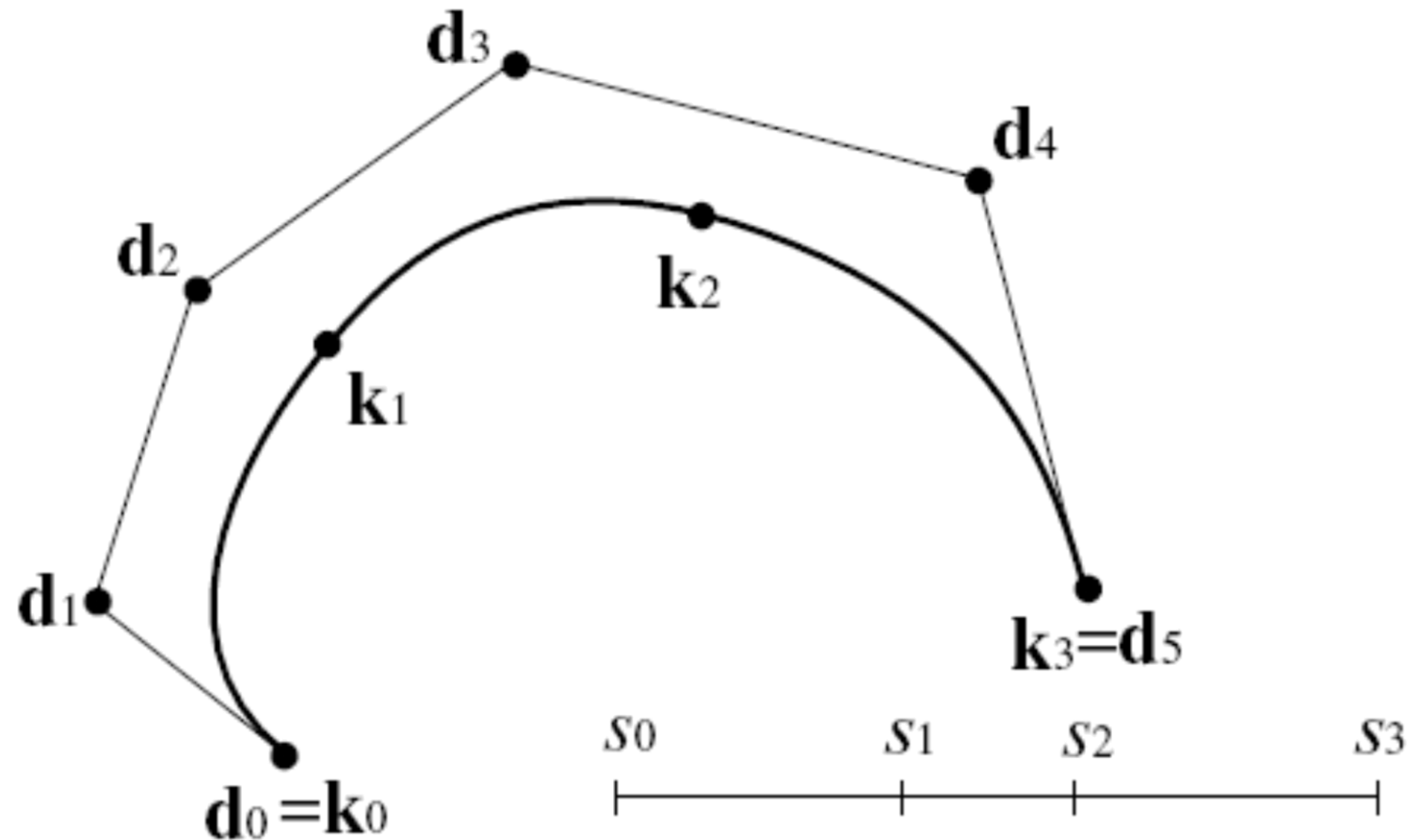
## Interpolating B-spline curves

- Given:  $n + 1$  control points  $\mathbf{k}_0, \dots, \mathbf{k}_n$   
knot sequence  $s_0, \dots, s_n$
- Wanted: piecewise cubic interpolating B-spline curve  $\mathbf{x}$   
i.e.,  $\mathbf{x}(s_i) = \mathbf{k}_i$  for  $i = 0, \dots, n$
- Approach: piecewise cubic  $\Rightarrow k = 4$ 
  - $\mathbf{x}(t)$  consists of  $n$  segments  $\Rightarrow n + 3$  de Boor points



# B-spline curves: interpolation

- Example:  $n = 3$



# B-spline curves: interpolation

- We choose the knot vector
  - $T = (t_0, t_1, t_2, t_3, t_4, \dots, t_{n+2}, t_{n+3}, t_{n+4}, t_{n+5}, t_{n+6})$   
 $= (s_0, s_0, s_0, s_0, s_1, \dots, s_{n-1}, s_n, s_n, s_n, s_n)$
- Then, the following conditions arise:
  - $\mathbf{x}(s_0) = \mathbf{k}_0 = \mathbf{d}_0$
  - $\mathbf{x}(s_i) = \mathbf{k}_i = N_{i,4}(s_i)\mathbf{d}_i + N_{i+1,4}(s_i)\mathbf{d}_{i+1} + N_{i+2,4}(s_i)\mathbf{d}_{i+2}$   
for  $i = 1, \dots, n - 1$
  - $\mathbf{x}(s_n) = \mathbf{k}_n = \mathbf{d}_{n+2}$
- Total:  $n + 1$  conditions for  $n + 3$  unknown de Boor points  
→ 2 end conditions

# B-spline curves: interpolation

- Here as example: natural end conditions

$$x''(s_0) = 0 \Leftrightarrow \frac{d_2 - d_1}{s_2 - s_0} = \frac{d_1 - d_0}{s_1 - s_0}$$

$$x''(s_n) = 0 \Leftrightarrow \frac{d_{n+2} - d_{n+1}}{s_n - s_{n-1}} = \frac{d_{n+1} - d_n}{s_n - s_{n-2}}$$

# B-spline curves: interpolation

- This results in the following tridiagonal system of equations:

$$\begin{pmatrix} 1 & & & & & & \\ \alpha_0 & \beta_0 & \gamma_0 & & & & \\ & \alpha_1 & \beta_1 & \gamma_1 & & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & & \ddots & \\ & & & & & & \ddots \\ & & & & & & & \ddots \\ & & & & & & & \alpha_{n-1} & \beta_{n-1} & \gamma_{n-1} \\ & & & & & & & & \alpha_n & \beta_n & \gamma_n \\ & & & & & & & & & & 1 \end{pmatrix} \begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ \vdots \\ d_n \\ d_{n+1} \\ d_{n+2} \end{pmatrix} = \begin{pmatrix} k_0 \\ 0 \\ k_1 \\ \vdots \\ \vdots \\ \vdots \\ k_{n-1} \\ 0 \\ k_n \end{pmatrix}$$

# B-spline curves: interpolation

- with

$$\alpha_0 = s_2 - s_0$$

$$\beta_0 = -(s_2 - s_0) - (s_1 - s_0)$$

$$\gamma_0 = s_1 - s_0$$

$$\alpha_n = s_n - s_{n-1}$$

$$\beta_n = -(s_n - s_{n-1}) - (s_n - s_{n-2})$$

$$\gamma_n = s_n - s_{n-2}$$

$$\alpha_i = N_{i,4}(s_i)$$

$$\beta_i = N_{i+1,4}(s_i)$$

$$\gamma_i = N_{i+2,4}(s_i)$$

for  $i = 1, \dots, n - 1$

Natural end conditions



# B-spline curves: interpolation

- Solving a tridiagonal system of equations: Thomas-algorithm!
- $O(n)$
- Only for diagonally dominant matrices

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \cdot & \\ & & \cdot & \cdot & c_{n-1} \\ 0 & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \cdot \\ \cdot \\ d_n \end{bmatrix}$$

# B-spline curves: interpolation

- Solving a tridiagonal system of equation: Thomas-algorithm!

Forward elimination phase

for  $k = 2:n$

$$m = \frac{a_k}{b_{k-1}}$$

$$b_k = b_k - mc_{k-1}$$

$$d_k = d_k - md_{k-1}$$

end

Backward substitution phase

$$x_n = \frac{d_n}{b_n}$$

for  $k = n - 1:-1:1$

$$x_k = \frac{d_k - c_k x_{k+1}}{b_k}$$

end

# Bezier curves to B-splines

## Conversion between cubic Bezier and B-spline curves

- Given:

$\mathbf{k}_0, \dots, \mathbf{k}_n$ : control points

$t_0, \dots, t_n$ : knot sequence

2 end conditions

$b_0, \dots, b_{3n}$ : Bezier points for  $C^2$ -continuous interpolating cubic Bezier spline curve

- Wanted: same curve in B-spline form



# Bezier curves to B-splines

- Knot vector  $T = (t_0, t_0, t_0, t_0, t_1, \dots, t_{n-1}, t_n, t_n, t_n, t_n)$

- $\mathbf{d}_0, \dots, \mathbf{d}_{n+2}$  are determined by

$$d_0 = b_0$$

$$d_1 = b_1$$

$$d_i = b_{3i-4} + \frac{\Delta_{i-1}}{\Delta_{i-2}} (b_{3i-4} - b_{3i-5}) \text{ for } i = 2, \dots, n$$

$$d_{n+1} = b_{3n-1}$$

$$d_{n+2} = b_{3n}$$

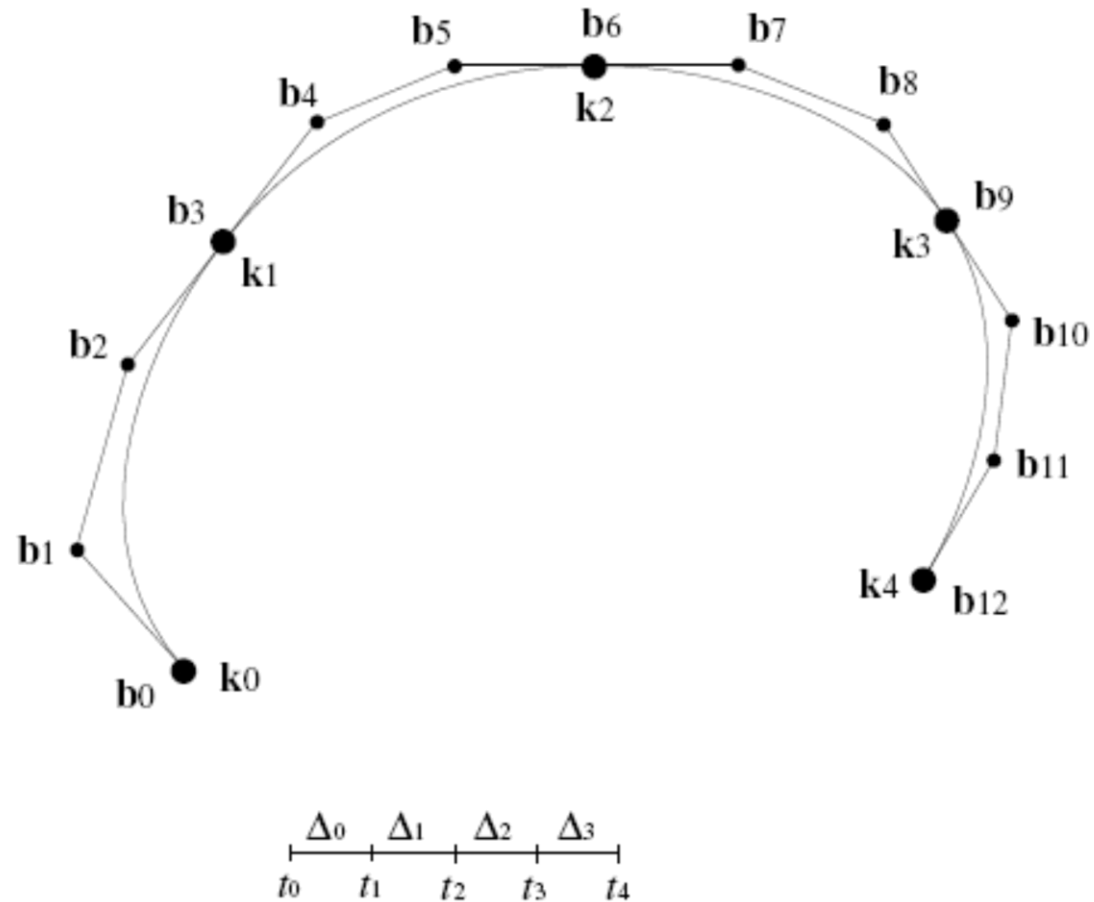
where  $\Delta_i = t_{i+1} - t_i$  for  $i = 0, \dots, n-1$

- The inverse problem is solvable as well

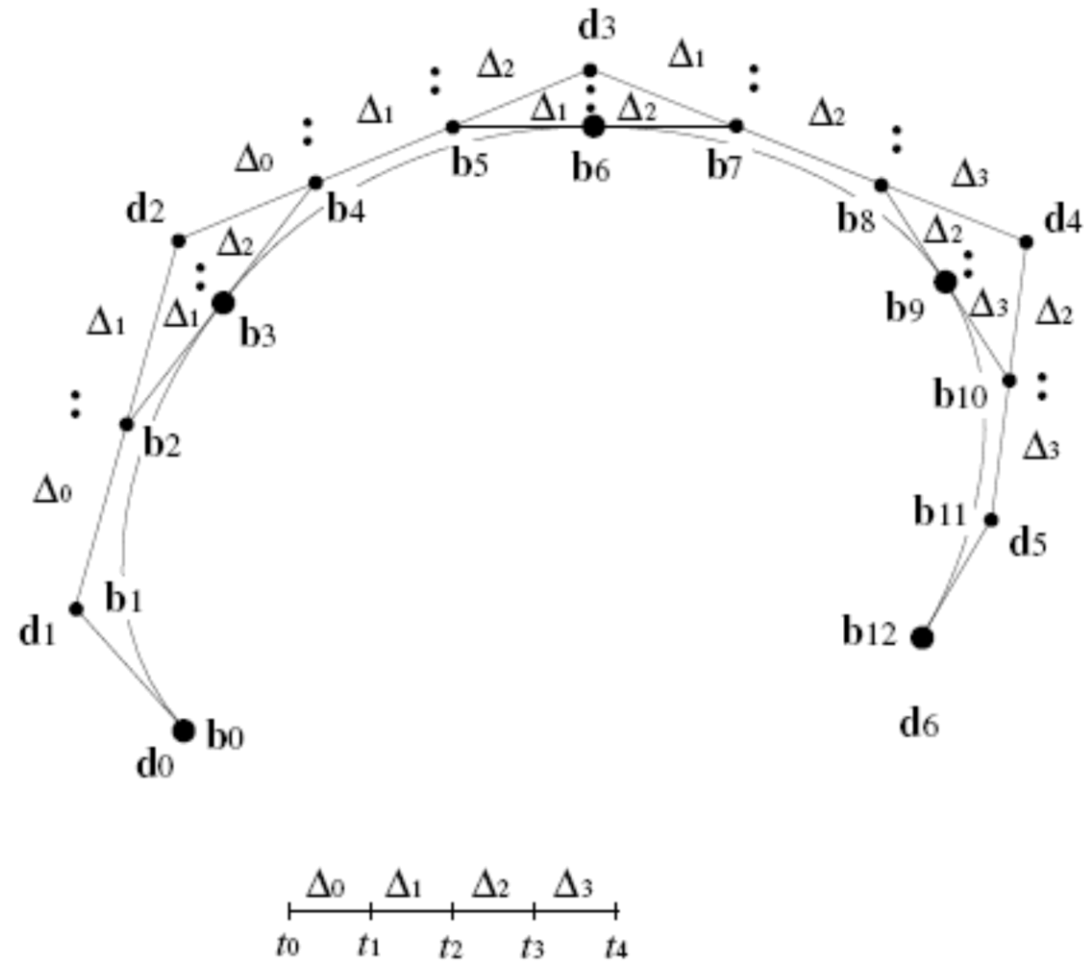
Remember the condition on  $d^-$  and  $d^+$  for  $C^2$  continuity of Bezier splines

# Bezier curves to B-splines

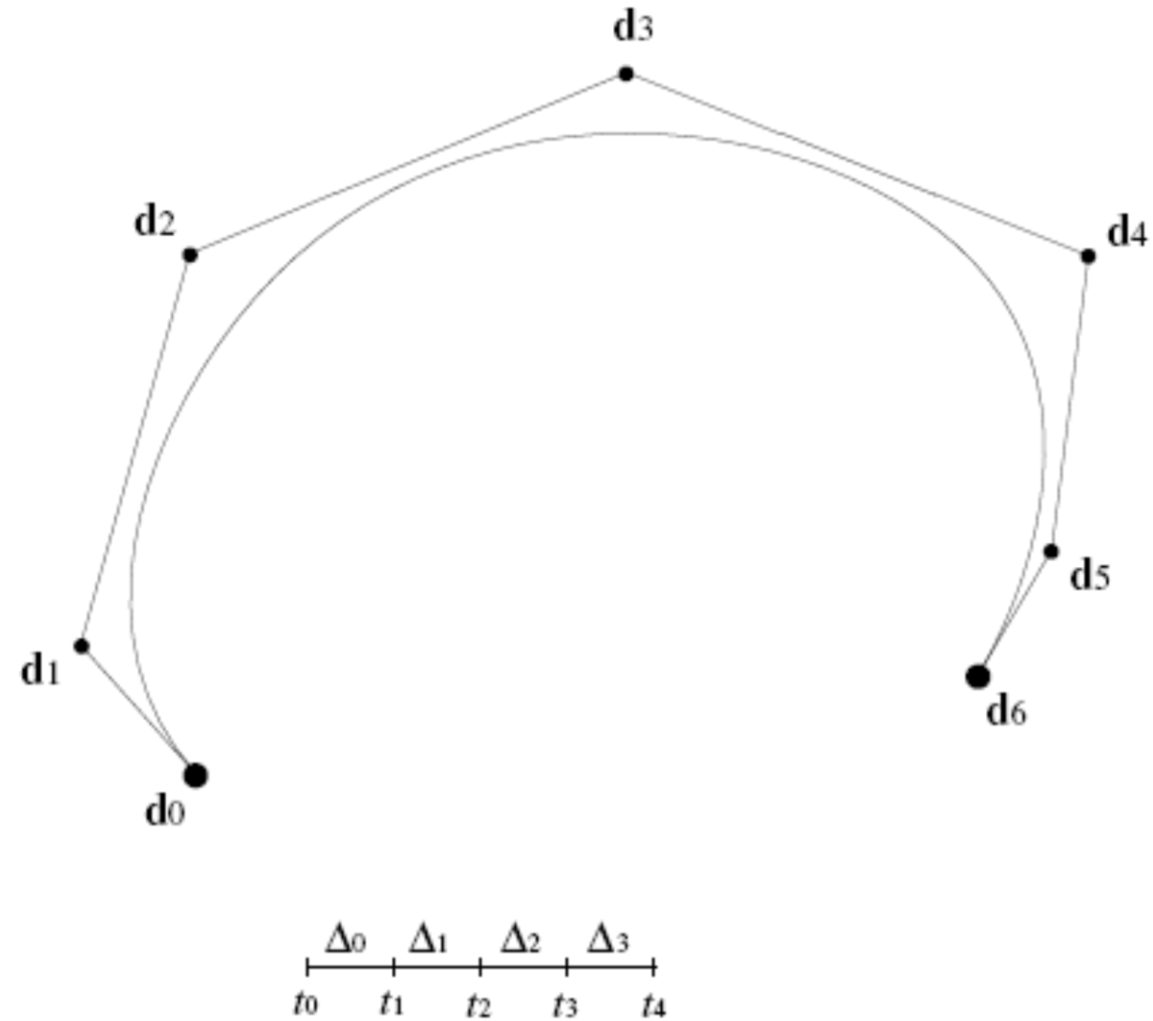
- Examples:  $n = 4$



# Bezier curves to B-splines



# Bezier curves to B-splines



# Summary of Bezier and B-spline curves

1. Bezier curve for  $n + 1$  control points  $b_0, \dots, b_n$ :
  - Polynomial curve of degree  $n$
  - Uniquely defined by control points
  - End point interpolation, remaining points are approximated
  - Pseudo-local impact of control points

# Summary of Bezier and B-spline curves

2. Interpolating cubic Bezier-spline curves by the control points  $k_0, \dots, k_n$
- Consists of  $n$  piecewise cubic curve segments
  - $C^2$ -continuous at the control points
  - Uniquely defined by parameterization (i.e. knot sequence) and two end conditions
  - Interpolates all control points
  - Pseudo-local impact of the control points

# Summary of Bezier and B-spline curves

3. Piecewise cubic B-spline curve for control points  $d_0, \dots, d_n$  and knot vector  $T = (t_0, t_0, t_0, t_0, t_1, \dots, t_{n-1}, t_n, t_n, t_n, t_n)$
- Consists of  $n - 2$  piecewise cubic curve segments which are  $C^2$ -conditions at the knots
  - Uniquely defined by  $d_i$  and  $T$
  - End point interpolation, the remaining points are approximated
  - Local impact of the de Boor points

# Summary of Bezier and B-spline curves

4. Interpolating cubic B-spline through the control points  $k_0, \dots, k_n$ 
  - Possible to formulate like (3) using 2 end conditions and solution of a tridiagonal system of equations for each  $x, y$ - and  $z$ - component
  - Identical curve to (2)



# B-splines

detailed examples

# B-spline curves: general case (reminder)

- Given: knot sequence  $t_0 < t_1 < \dots < t_n < \dots < t_{n+k}$   
( $(t_0, t_1, \dots, t_{n+k})$  is called knot vector)
- Normalized B-spline functions  $N_{i,k}$  of the order  $k$  (degree  $k - 1$ ) are defined as:

$$N_{i,1}(t) = \begin{cases} 1, & t_i \leq t < t_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t)$$

for  $k > 1$  and  $i = 0, \dots, n$

- **Remark:**
  - If a knot value is repeated  $k$  times, the denominator may vanish
  - In this case: The fraction is treated as a zero

# B-spline basis evaluation: ex. 1

- For order 4 and knot sequence

$$T = [t_0 \ t_1 \ t_2 \ t_3 \ t_4 \ t_5 \ t_6 \ t_7] = [0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1]$$

Evaluate the B-spline function  $N_{0,4}(t)$ ,  $N_{1,4}(t)$ ,  $N_{2,4}(t)$ ,  $N_{3,4}(t)$

# B-spline basis evaluation: ex. 1

$$N_{0,1}(t) = N_{1,1}(t) = N_{2,1}(t) = N_{4,1}(t) = N_{6,1}(t) = 0$$

$$N_{3,1}(t) = 1 \quad (0 \leq t < 1)$$

$$N_{0,2}(t) = \frac{t-t_0}{t_1-t_0} N_{0,1}(t) + \frac{t_2-t}{t_2-t_1} N_{1,1}(t) = 0$$

$$N_{1,2}(t) = \frac{t-t_1}{t_2-t_1} N_{1,1}(t) + \frac{t_3-t}{t_3-t_2} N_{2,1}(t) = 0$$

$$N_{2,2}(t) = \frac{t-t_2}{t_3-t_2} N_{2,1}(t) + \frac{t_4-t}{t_4-t_3} N_{3,1}(t) = (1-t)N_{3,1}(t)$$

$$N_{3,2}(t) = \frac{t-t_3}{t_4-t_3} N_{3,1}(t) + \frac{t_5-t}{t_5-t_4} N_{4,1}(t) = tN_{3,1}(t)$$

$$N_{4,2}(t) = \frac{t-t_4}{t_5-t_4} N_{4,1}(t) + \frac{t_6-t}{t_6-t_5} N_{5,1}(t) = 0$$

$$N_{5,2}(t) = \frac{t-t_5}{t_6-t_5} N_{5,1}(t) + \frac{t_7-t}{t_7-t_6} N_{6,1}(t) = 0$$

# B-spline basis evaluation: ex. 1

$$N_{0,3}(t) = \frac{t-t_0}{t_2-t_0} N_{0,2}(t) + \frac{t_3-t}{t_3-t_1} N_{1,2}(t) = 0$$

$$N_{1,3}(t) = \frac{t-t_1}{t_3-t_1} N_{1,2}(t) + \frac{t_4-t}{t_4-t_2} N_{2,2}(t) = (1-t)^2 N_{3,1}(t)$$

$$N_{2,3}(t) = \frac{t-t_2}{t_4-t_2} N_{2,2}(t) + \frac{t_5-t}{t_5-t_3} N_{3,2}(t) = 2t(1-t) N_{3,1}(t)$$

$$N_{3,3}(t) = \frac{t-t_3}{t_5-t_3} N_{3,2}(t) + \frac{t_6-t}{t_6-t_4} N_{4,2}(t) = t^2 N_{3,1}(t)$$

$$N_{4,3}(t) = \frac{t-t_4}{t_6-t_4} N_{4,2}(t) + \frac{t_7-t}{t_7-t_5} N_{5,2}(t) = 0$$

# B-spline basis evaluation: ex. 1

- Finally
$$N_{0,4}(t) = \frac{t-t_0}{t_3-t_0} N_{0,3}(t) + \frac{t_4-t}{t_4-t_1} N_{1,3}(t) = (1-t)^3 N_{3,1}(t)$$
$$N_{1,4}(t) = \frac{t-t_1}{t_4-t_1} N_{1,3}(t) + \frac{t_5-t}{t_5-t_2} N_{2,3}(t) = 3(1-t)^2 t N_{3,1}(t)$$
$$N_{2,4}(t) = \frac{t-t_2}{t_5-t_2} N_{2,3}(t) + \frac{t_6-t}{t_6-t_3} N_{3,3}(t) = 3(1-t) t^2 N_{3,1}(t)$$
$$N_{3,4}(t) = \frac{t-t_3}{t_6-t_3} N_{3,3}(t) + \frac{t_7-t}{t_7-t_4} N_{4,3}(t) = t^3 N_{3,1}(t)$$

# B-spline basis evaluation: ex. 1

- Finally
$$N_{0,4}(t) = \frac{t-t_0}{t_3-t_0} N_{0,3}(t) + \frac{t_4-t}{t_4-t_1} N_{1,3}(t) = (1-t)^3 N_{3,1}(t)$$
$$N_{1,4}(t) = \frac{t-t_1}{t_4-t_1} N_{1,3}(t) + \frac{t_5-t}{t_5-t_2} N_{2,3}(t) = 3(1-t)^2 t N_{3,1}(t)$$
$$N_{2,4}(t) = \frac{t-t_2}{t_5-t_2} N_{2,3}(t) + \frac{t_6-t}{t_6-t_3} N_{3,3}(t) = 3(1-t) t^2 N_{3,1}(t)$$
$$N_{3,4}(t) = \frac{t-t_3}{t_6-t_3} N_{3,3}(t) + \frac{t_7-t}{t_7-t_4} N_{4,3}(t) = t^3 N_{3,1}(t)$$
- We clearly get the Bernstein basis function as mentioned earlier

# B-spline basis evaluation: ex. 2

- For order 4 and not sequence

$$T = [t_0 \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5 \quad t_6 \quad t_7] = [-3 \quad -2 \quad -1 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4]$$

Evaluate the corresponding basis



## B-spline basis evaluation: ex. 2

$$N_{0,2}(t) = (t + 3)N_{0,1}(t) + (-1 - t)N_{1,1}(t)$$

$$N_{1,2}(t) = (t + 2)N_{1,1}(t) + (-t)N_{2,1}(t)$$

$$N_{2,2}(t) = (t + 1)N_{2,1}(t) + (1 - t)N_{3,1}(t)$$

$$N_{3,2}(t) = tN_{3,1}(t) + (2 - t)N_{4,1}(t)$$

$$N_{4,2}(t) = (t - 1)N_{4,1}(t) + (3 - t)N_{5,1}(t)$$

$$N_{5,2}(t) = (t - 2)N_{5,1}(t) + (4 - t)N_{6,1}(t)$$

## B-spline basis evaluation: ex. 2

$$N_{0,3}(t) = \frac{1}{2}(t+2)N_{0,2}(t) + \frac{1}{2}(0-t)N_{1,2}(t)$$

$$N_{1,3}(t) = \frac{1}{2}(t+1)N_{1,2}(t) + \frac{1}{2}(1-t)N_{2,2}(t)$$



$$N_{2,3}(t) = \frac{1}{2}(t+0)N_{2,2}(t) + \frac{1}{2}(2-t)N_{3,2}(t)$$

$$N_{3,3}(t) = \frac{1}{2}(t-1)N_{3,2}(t) + \frac{1}{2}(3-t)N_{4,2}(t)$$

# B-spline basis evaluation: ex. 2

- Finally

$$N_{0,4}(t) = \frac{1}{3}(t+3)N_{0,3}(t) + \frac{1}{3}(1-t)N_{1,3}(t)$$

$$N_{1,4}(t) = \frac{1}{3}(t+2)N_{1,3}(t) + \frac{1}{3}(2-t)N_{2,3}(t)$$

$$N_{2,4}(t) = \frac{1}{3}(t+1)N_{2,3}(t) + \frac{1}{3}(3-t)N_{3,3}(t)$$

$$N_{3,4}(t) = \frac{1}{3}tN_{3,3}(t) + \frac{1}{3}(4-t)N_{4,3}(t)$$

# B-spline basis evaluation: ex. 2

- Then substituting

$$N_{0,4}(t) = \frac{1}{6}(t+3)^3 N_{0,1}(t) + \left\{ -(t+1)^3 + \frac{2}{3}t^3 - \frac{1}{6}(t-1)^3 \right\} N_{1,1}(t) + \left\{ \frac{2}{3}t^3 - \frac{1}{6}(t-1)^3 \right\} N_{2,1}(t) - \frac{1}{6}(t-1)^3 N_{3,1}(t)$$

$$N_{1,4}(t) = \frac{1}{6}(t+2)^3 N_{1,1}(t) + \left\{ -t^3 + \frac{2}{3}(t-1)^3 - \frac{1}{6}(t-2)^3 \right\} N_{2,1}(t) + \left\{ \frac{2}{3}(t-1)^3 - \frac{1}{6}(t-2)^3 \right\} N_{3,1}(t) - \frac{1}{6}(t-2)^3 N_{4,1}(t)$$

$$N_{2,4}(t) = \frac{1}{6}(t+1)^3 N_{2,1}(t) + \left\{ -(t-1)^3 + \frac{2}{3}(t-2)^3 - \frac{1}{6}(t-3)^3 \right\} N_{3,1}(t) + \left\{ \frac{2}{3}(t-2)^3 - \frac{1}{6}(t-3)^3 \right\} N_{4,1}(t) - \frac{1}{6}(t-3)^3 N_{5,1}(t)$$

$$N_{3,4}(t) = \frac{1}{6}t^3 N_{3,1}(t) + \left\{ -(t-2)^3 + \frac{2}{3}(t-3)^3 - \frac{1}{6}(t-4)^3 \right\} N_{4,1}(t) + \left\{ \frac{2}{3}(t-3)^3 - \frac{1}{6}(t-4)^3 \right\} N_{5,1}(t) - \frac{1}{6}(t-4)^3 N_{6,1}(t)$$

# de Boor algorithm (reminder)

1. Search index  $r$  with  $t_r \leq t < t_{r+1}$

2. for  $i = r - k + 1, \dots, r$

$$d_i^0 = d_i \quad \text{sometimes noted as } d_i^0(t) = d_i$$

- for  $j = 1, \dots, k - 1$

for  $i = r - k + 1 + j, \dots, r$

$$d_i^j = \left(1 - \alpha_i^j\right) \cdot d_{i-1}^{j-1} + \alpha_i^j \cdot d_i^{j-1}$$

$$\text{with } \alpha_i^j = \frac{t - t_i}{t_{i+k-j} - t_i}$$

Then:  $d_r^{k-1} = x(t)$

Order  $k$   
 $n + 1$  points  
 $n + k + 1$  knots

# de boor algorithm: ex. 1

- For order 4, de Boor points  $Q_0, Q_1, \dots, Q_8$  and knot sequence  
$$T = [t_0 \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5 \quad t_6 \quad t_7 \quad t_8 \quad t_9 \quad t_{10} \quad t_{11} \quad t_{12}]$$
$$= [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 6 \quad 6 \quad 6]$$

Evaluate the B-spline curve at  $t = 4.75$

# de Boor algorithm: ex. 1

- Since  $t_7 \leq 4.75 < t_8$ ,  $r = 7$ , therefore  $i = 7 - 4 + 1 = 4$

$$\begin{aligned} Q_5^{[1]}(4.75) &= (1 - \lambda)Q_4^{[0]}(4.75) + \lambda Q_5^{[0]}(4.75) \\ &= (1 - \lambda)Q_4 + \lambda Q_5 = 0.083Q_4 + 0.917Q_5 \end{aligned} \quad \left( \lambda = \frac{4.75 - t_5}{t_8 - t_5} = 0.917 \right)$$

$$\begin{aligned} Q_6^{[1]}(4.75) &= (1 - \lambda)Q_5^{[0]}(4.75) + \lambda Q_6^{[0]}(4.75) \\ &= (1 - \lambda)Q_5 + \lambda Q_6 = 0.417Q_5 + 0.583Q_6 \end{aligned} \quad \left( \lambda = \frac{4.75 - t_6}{t_9 - t_6} = 0.583 \right)$$

$$\begin{aligned} Q_7^{[1]}(4.75) &= (1 - \lambda)Q_6^{[0]}(4.75) + \lambda Q_7^{[0]}(4.75) \\ &= (1 - \lambda)Q_6 + \lambda Q_7 = 0.625Q_6 + 0.375Q_7 \end{aligned} \quad \left( \lambda = \frac{4.75 - t_7}{t_{10} - t_7} = 0.375 \right)$$

# de Boor algorithm: ex. 1

- Then

$$\begin{aligned}Q_6^{[2]}(4.75) &= (1 - \lambda)Q_5^{[1]}(4.75) + \lambda Q_6^{[1]}(4.75) \\&= 0.125(0.083Q_4 + 0.917Q_5) + 0.875(0.417Q_5 + 0.583Q_6) \\&= 0.01Q_4 + 0.479Q_5 + 0.510Q_6\end{aligned}$$

$$\left(\lambda = \frac{4.75 - t_6}{t_8 - t_6} = 0.875\right)$$

$$\begin{aligned}Q_7^{[1]}(4.75) &= (1 - \lambda)Q_6^{[1]}(4.75) + \lambda Q_7^{[1]}(4.75) \\&= 0.625(0.417Q_5 + 0.583Q_6) + 0.375(0.625Q_6 + 0.375Q_7) \\&= 0.261Q_5 + 0.598Q_6 + 0.141Q_7\end{aligned}$$

$$\left(\lambda = \frac{4.75 - t_7}{t_9 - t_7} = 0.375\right)$$



# de Boor algorithm: ex. 1

- Then

$$\begin{aligned} Q_7^{[3]}(4.75) &= (1 - \lambda)Q_6^{[2]}(4.75) + \lambda Q_7^{[2]}(4.75) \\ &= 0.25(0.01Q_4 + 0.479Q_5 + 0.510Q_6) \\ &\quad + 0.75(0.261Q_5 + 0.598Q_6 + 0.141Q_7) \\ &= 0.0025Q_4 + 0.316Q_5 + 0.576Q_6 + 0.106Q_7 \end{aligned} \quad \left( \lambda = \frac{4.75 - t_7}{t_8 - t_7} = 0.75 \right)$$

Order  $k$   
 $n + 1$  points  
 $n + k + 1$  knots

## de boor algorithm: ex. 2

- For order 4, de Boor points  $Q_0, Q_1, \dots, Q_6$  and knot sequence  
$$T = [t_0 \quad t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5 \quad t_6 \quad t_7 \quad t_8 \quad t_9 \quad t_{10}]$$
$$= [-3 \quad -2 \quad -1 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7]$$

Evaluate the B-spline curve at  $t = 3.5$

# de Boor algorithm: ex. 1

- Since  $t_6 \leq 3.5 < t_7$ ,  $r = 7$ , therefore  $i = 6 - 4 + 1 = 3$

$$Q_4^{[1]}(3.5) = (1 - \lambda)Q_3^{[0]} + \lambda Q_4^{[0]} \quad \left( \lambda = \frac{3.5 - t_4}{4 - 1} = 0.833 \right)$$

$$= 0.167Q_3 + 0.833Q_0$$

$$Q_5^{[2]}(3.5) = (1 - \lambda)Q_4^{[1]} + \lambda Q_5^{[1]} \quad \left( \lambda = \frac{3.5 - t_5}{4 - 2} = 0.75 \right)$$

$$= 0.25(0.167Q_3 + 0.833Q_0) + 0.75(0.5Q_0 + 0.5Q_1)$$

$$= 0.583Q_0 + 0.375Q_1 + 0.042Q_3$$

$$Q_5^{[1]}(3.5) = (1 - \lambda)Q_4^{[0]} + \lambda Q_5^{[0]} \quad \left( \lambda = \frac{3.5 - t_5}{4 - 1} = 0.5 \right)$$

$$= 0.167Q_0 + 0.833Q_1$$

$$Q_5^{[1]}(3.5) = (1 - \lambda)Q_4^{[0]} + \lambda Q_5^{[0]} \quad \left( \lambda = \frac{3.5 - t_6}{4 - 2} = 0.25 \right)$$

$$= 0.75(0.5Q_0 + 0.5Q_1) + 0.25(0.833Q_1 + 0.167Q_2)$$

$$= 0.375Q_0 + 0.583Q_1 + 0.042Q_2$$

$$Q_6^{[1]}(3.5) = (1 - \lambda)Q_5^{[0]} + \lambda Q_6^{[0]} \quad \left( \lambda = \frac{3.5 - t_6}{4 - 1} = 0.167 \right)$$

$$= 0.167Q_1 + 0.833Q_2$$

$$Q_6^{[3]}(3.5) = (1 - \lambda)Q_5^{[2]} + \lambda Q_6^{[2]} \quad \left( \lambda = \frac{3.5 - t_6}{4 - 3} = 0.5 \right)$$

$$= 0.5(0.583Q_0 + 0.375Q_1 + 0.042Q_3) + 0.5(0.375Q_0 + 0.583Q_1 + 0.042Q_2)$$

$$= 0.479Q_0 + 0.479Q_1 + 0.021Q_2 + 0.021Q_3$$