Computer Aided Geometric Design
Fall Semester 2024

B-Splines

陈仁杰
renjiec@ustc.edu.cn
http://staff.ustc.edu.cn/~renjiec

Basis

B-Splines

~~Bézier~~

Mathematical view: spline functions

Graphics view: spline curves (created using spline functions)

Motivation

- Back to the algebraic approach for Bézier curves

  →Bernstein polynomials

- Problem: global influence of the Bézier points

- Introduction of new basis function

  →B-spline functions

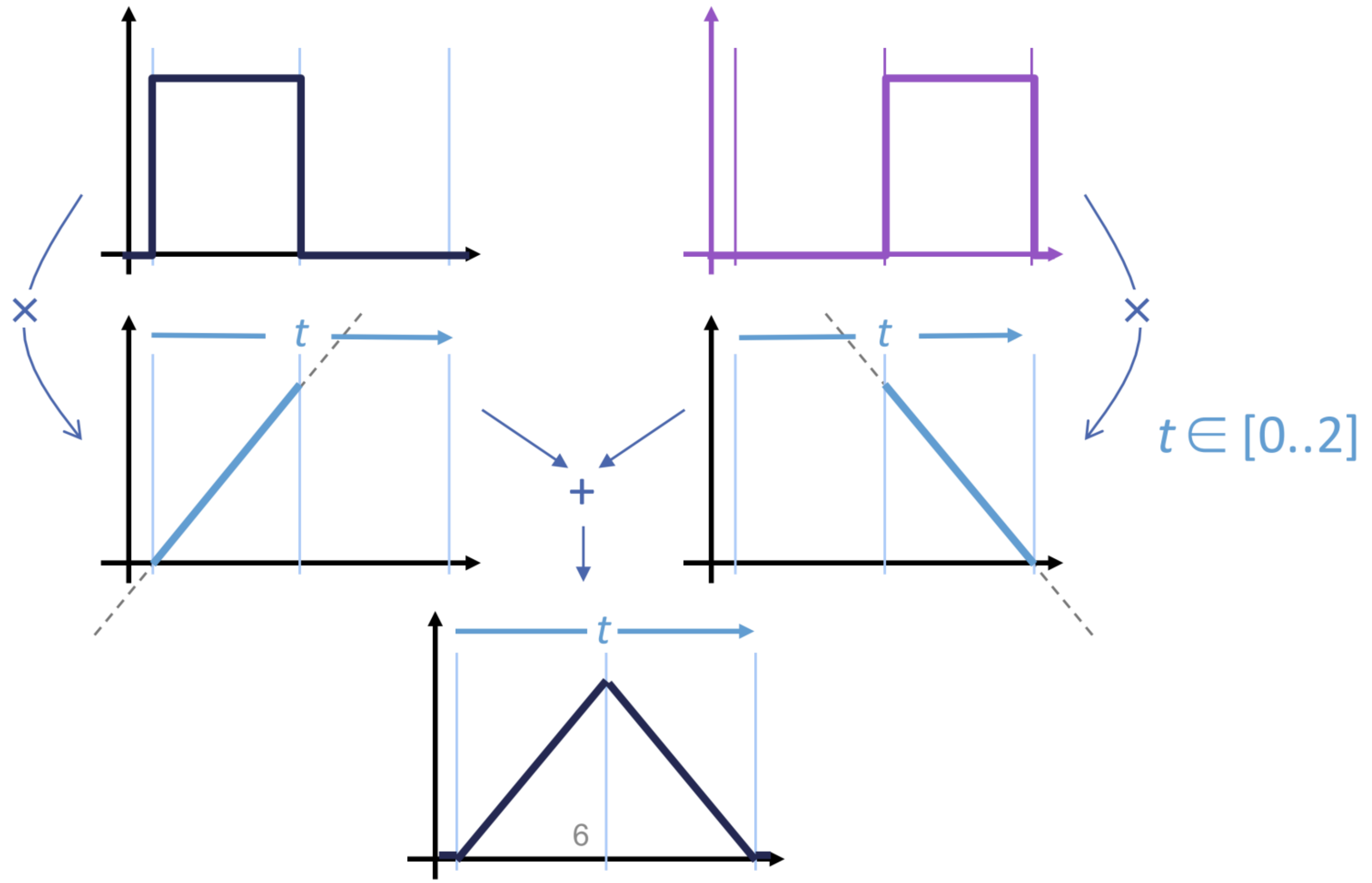
Some history

- **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 Bézier curves

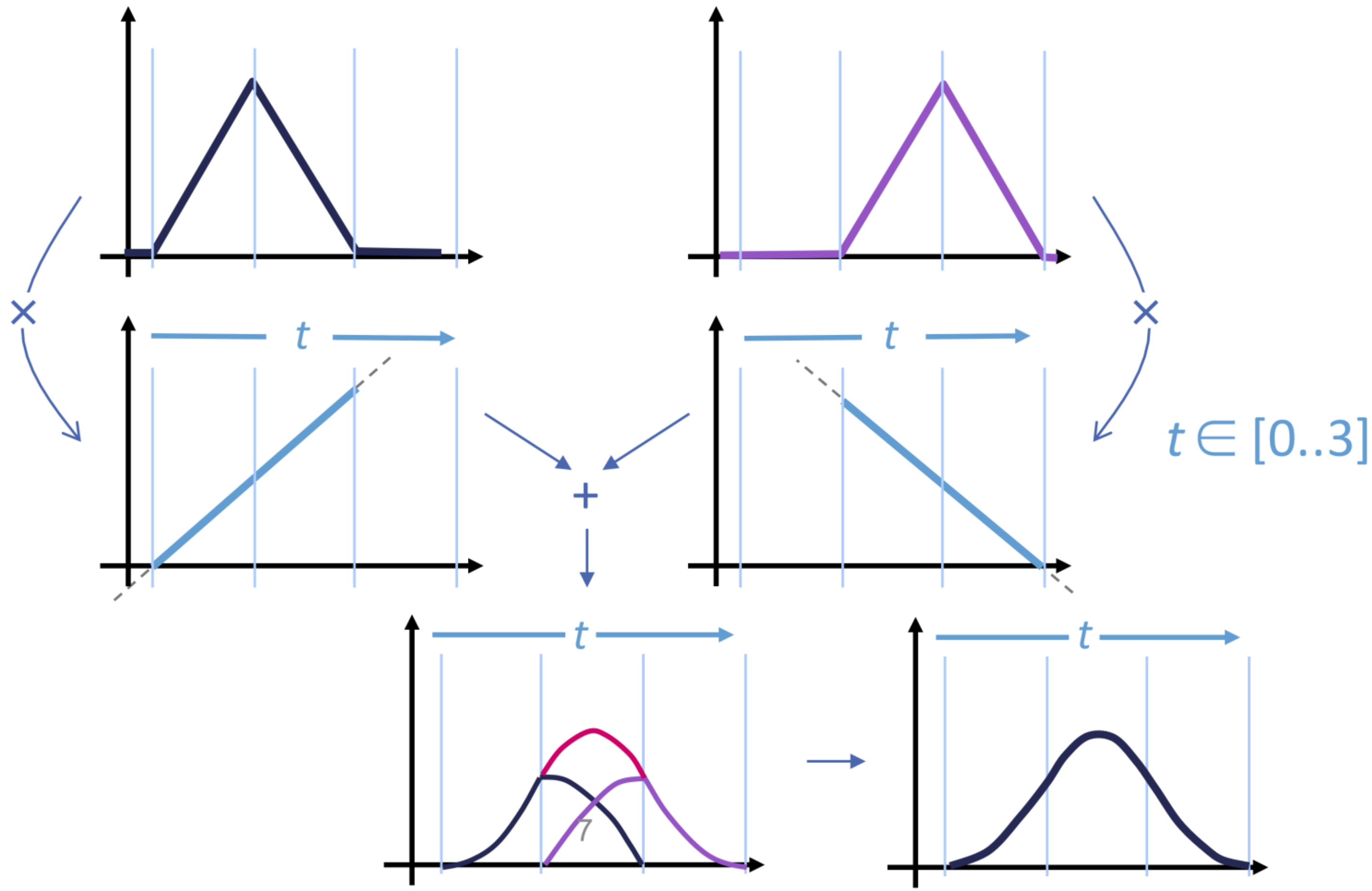
Repeated linear interpolation

Another way to increase smoothness:



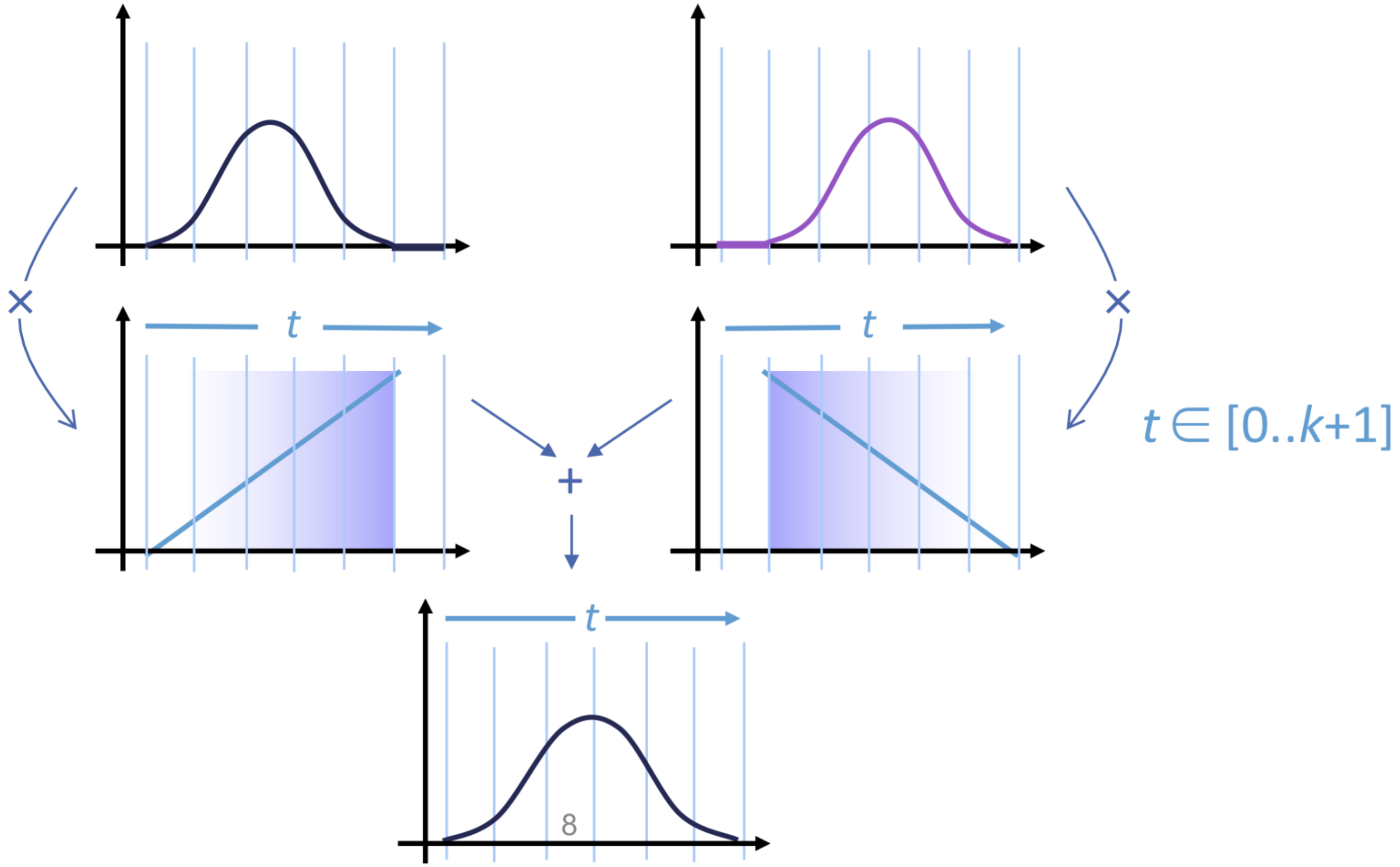
Repeated linear interpolation

- Another way to increase smoothness:



Repeated linear interpolation

- **Another way to increase smoothness**

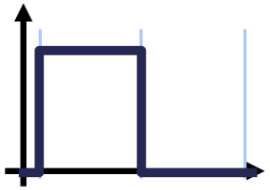


N.B.: we will use N_i^k or $N_{i,k}$ interchangeably

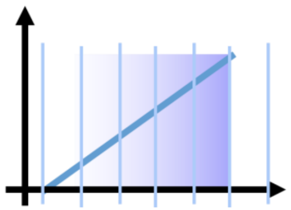
De Boor Recursion: uniform case

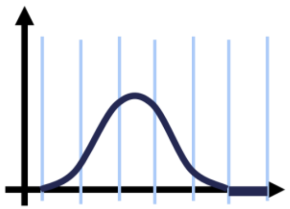
- **The uniform B-spline basis of order $\boldsymbol{k}$ (degree $\boldsymbol{k-1}$) is given as**

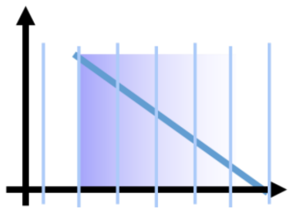
$$N_i^1(t) = \begin{cases} 1, \text{if } i \le t < i+1 \\ 0, \quad \text{otherwise} \end{cases}$$

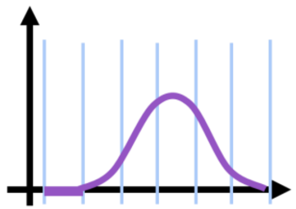


$$N_i^k(t) = \frac{t-i}{(i+k-1)-i} N_i^{k-1}(t) \qquad + \qquad \frac{(i+k)-t}{(i+k)-(i+1)} N_{i+1}^{k-1}(t)$$









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

B-spline curves: general case

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

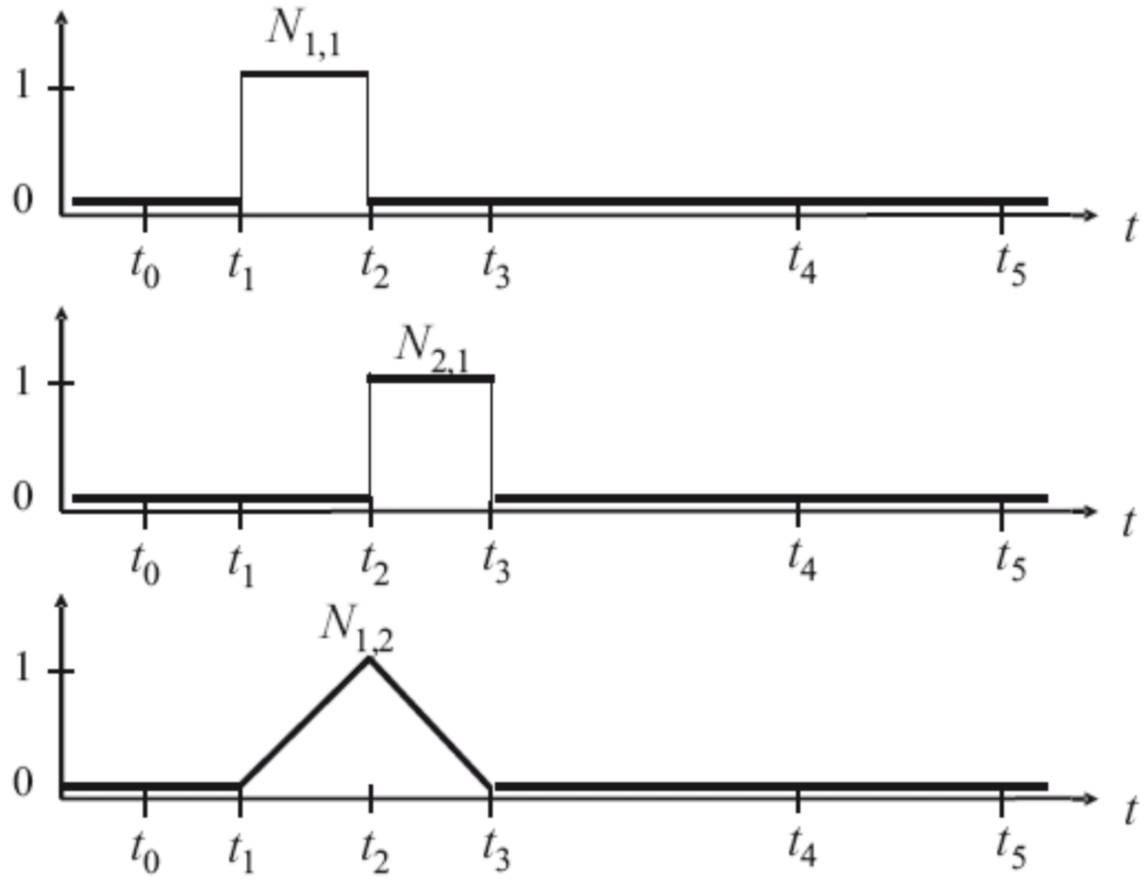
$$N_{i,1}(t) = \begin{cases} 1, & t_i \leq t < t_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

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

  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

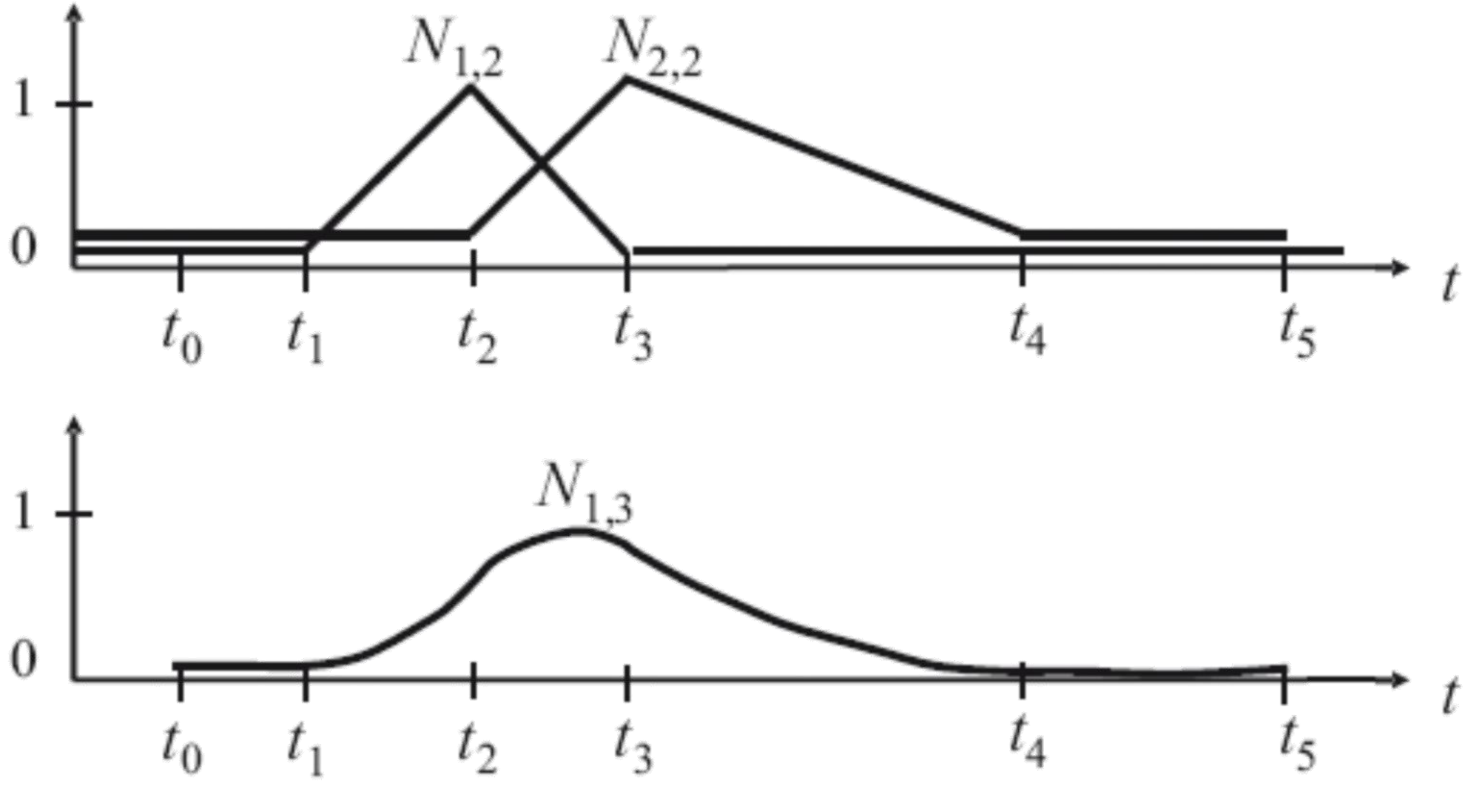


$$N_{i,1}(t) = \begin{cases} 1, & t_i \leq t < t_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

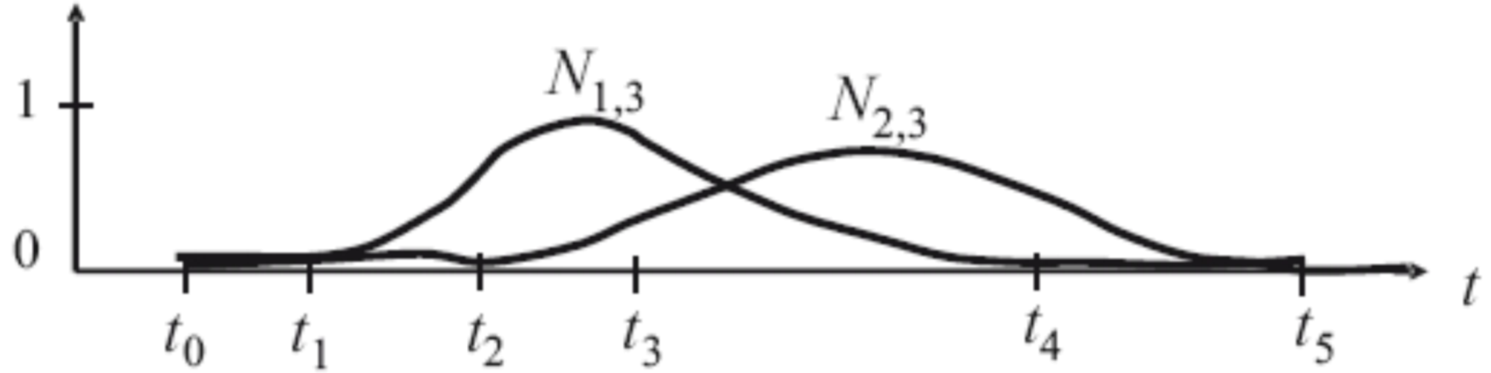
$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t)$$

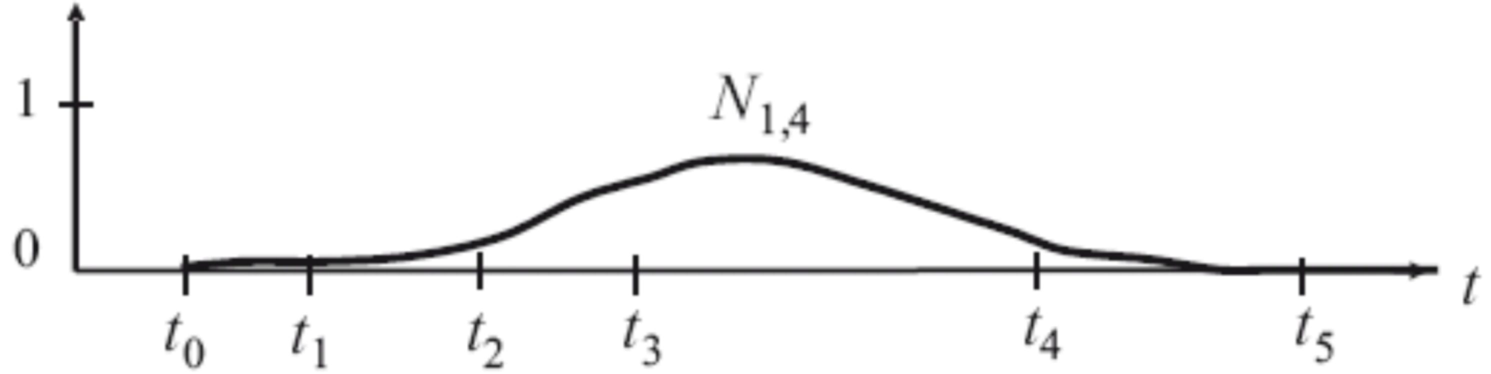
for $k > 1$ and $i = 0, \dots, n$

Example



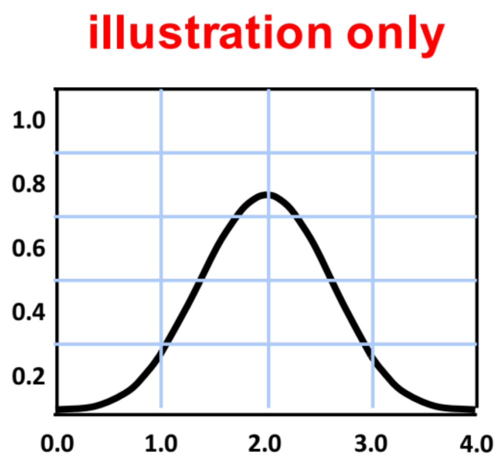
Example





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

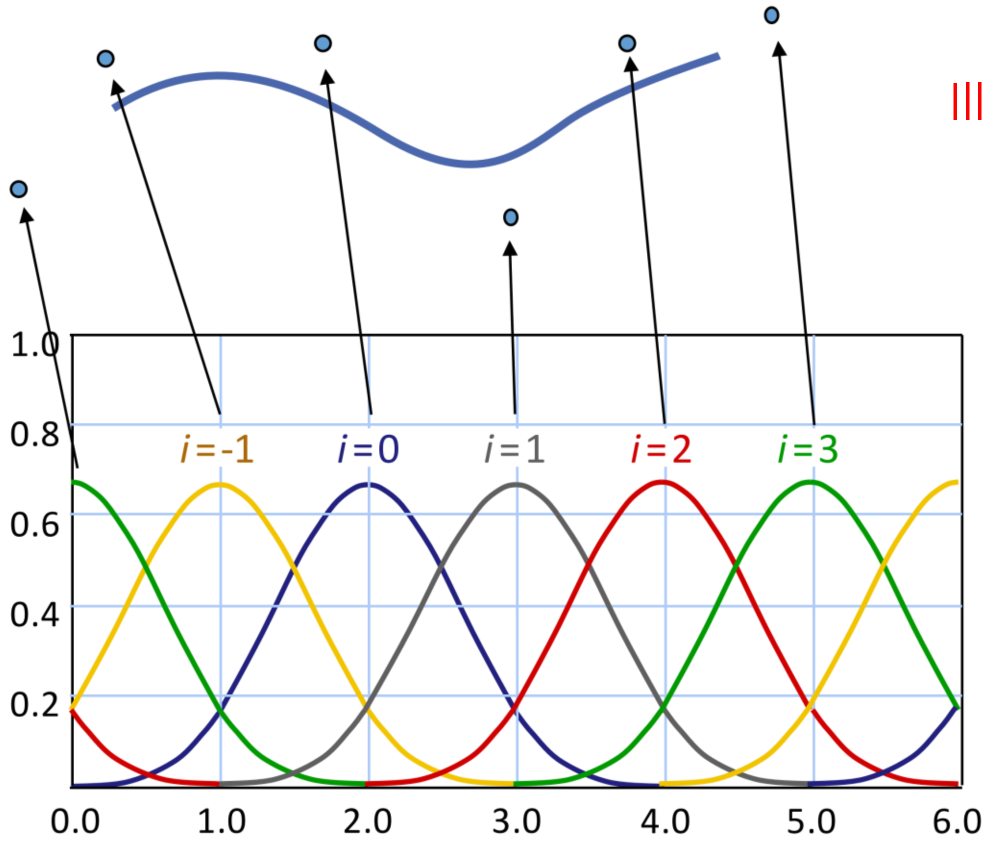


Illustration only

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 $t_{k-1} \le t \le t_{n+1}$
- For $t_i \le t_j \le t_{i+k}$, 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 $\boldsymbol{d}_0, \dots, \boldsymbol{d}_n \in \mathbb{R}^3$
  knot vector $T = (t_0, \dots, t_n, \dots t_{n+k})$

- Then, the B-spline curve $\boldsymbol{x}(t)$ of the order k is defined as

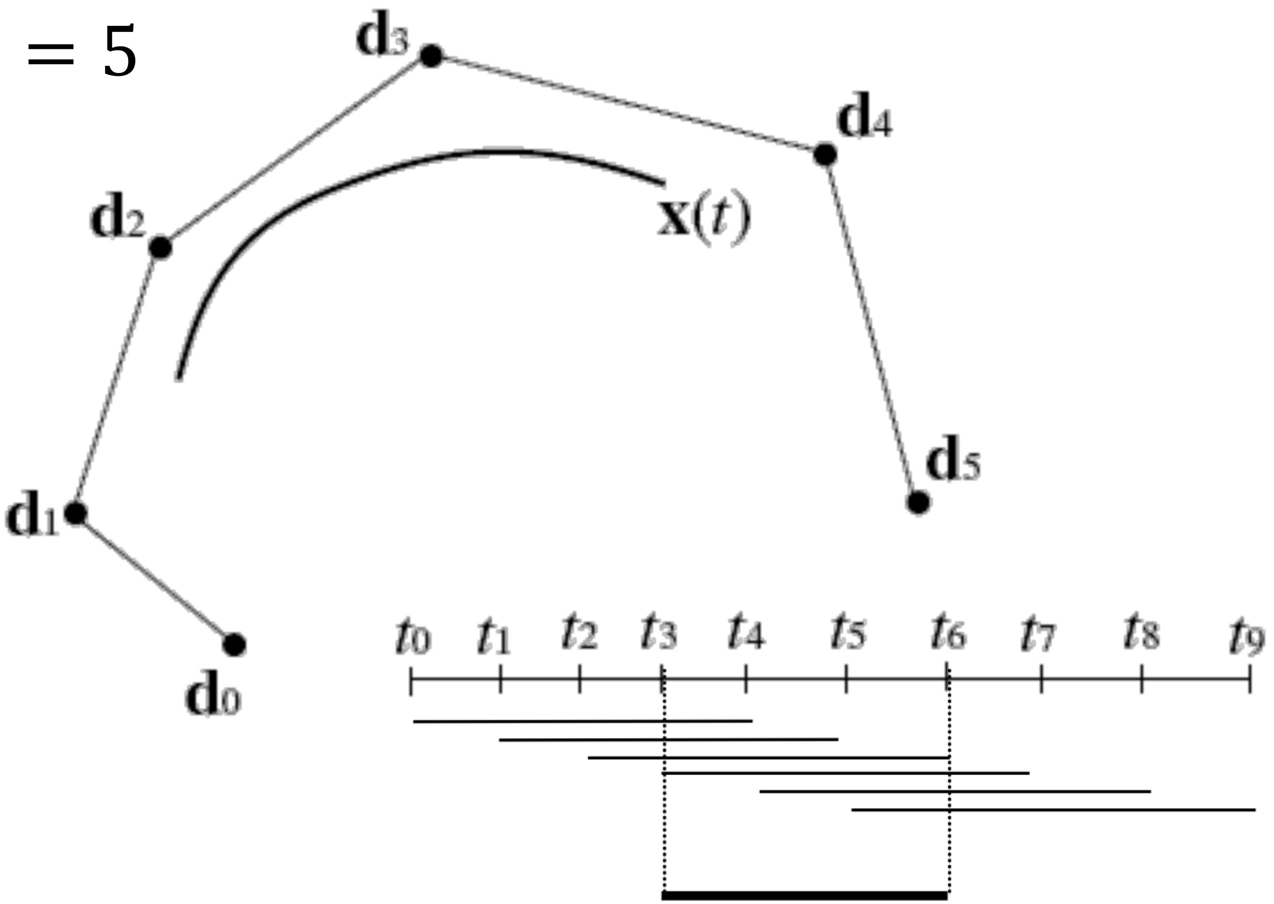
$$\boldsymbol{x}(t) = \sum_{i=0}^{n} N_{i,k}(t) \cdot \boldsymbol{d}_i$$

- The points $\boldsymbol{d}_i$ 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 $t_0 < t_1 < \cdots < t_{n+k}$
- Now: also multiple knots allowed, i.e. with $t_0 \leq t_1 \leq \cdots \leq t_{n+k}$
- 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

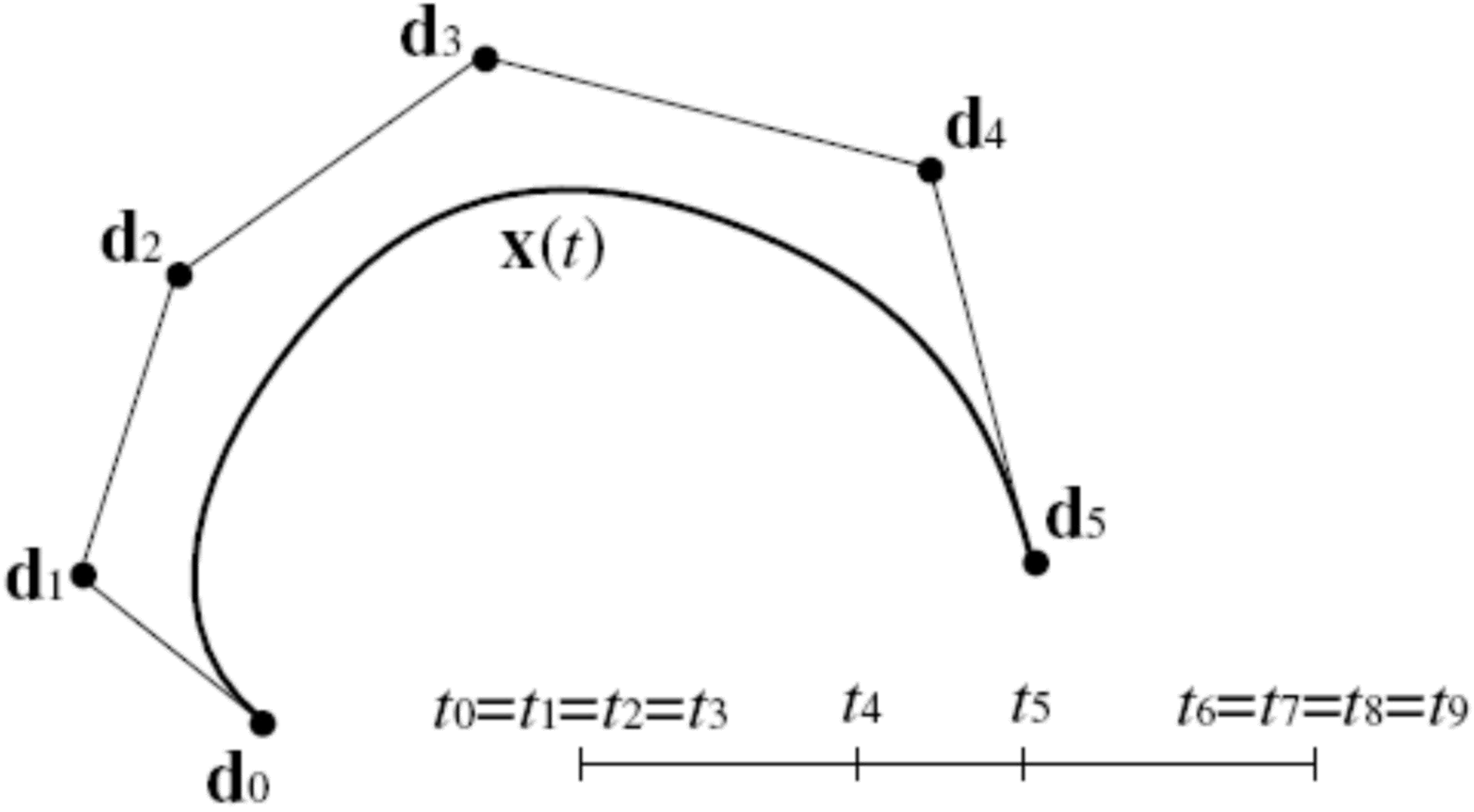
Effect of multiple knots:

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

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

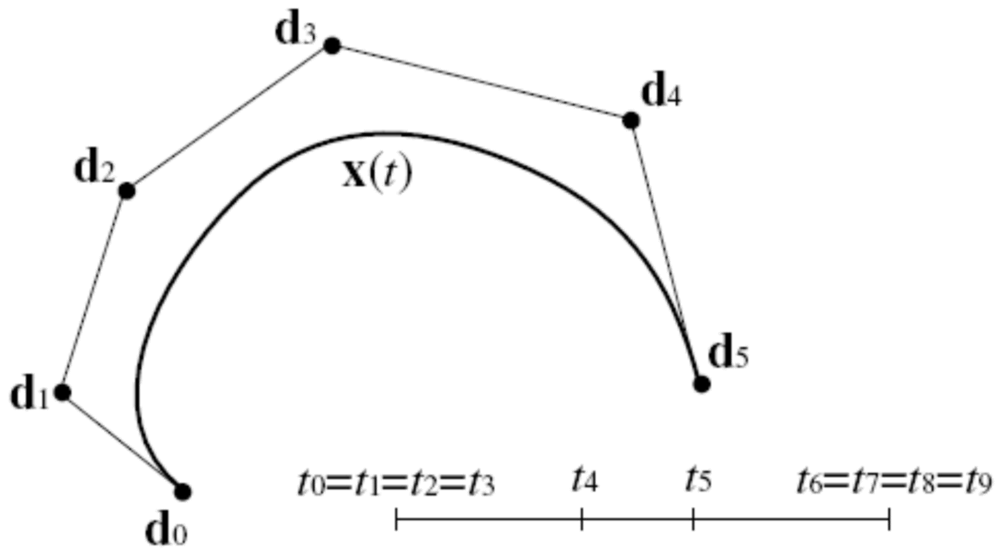
B-spline curves

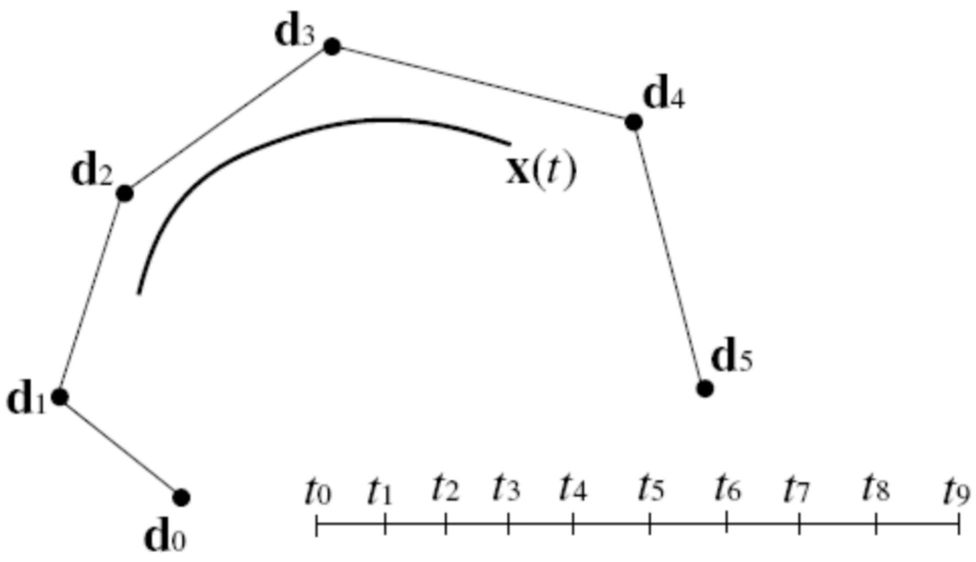
- Example: $k = 4, n = 5$



B-spline curves

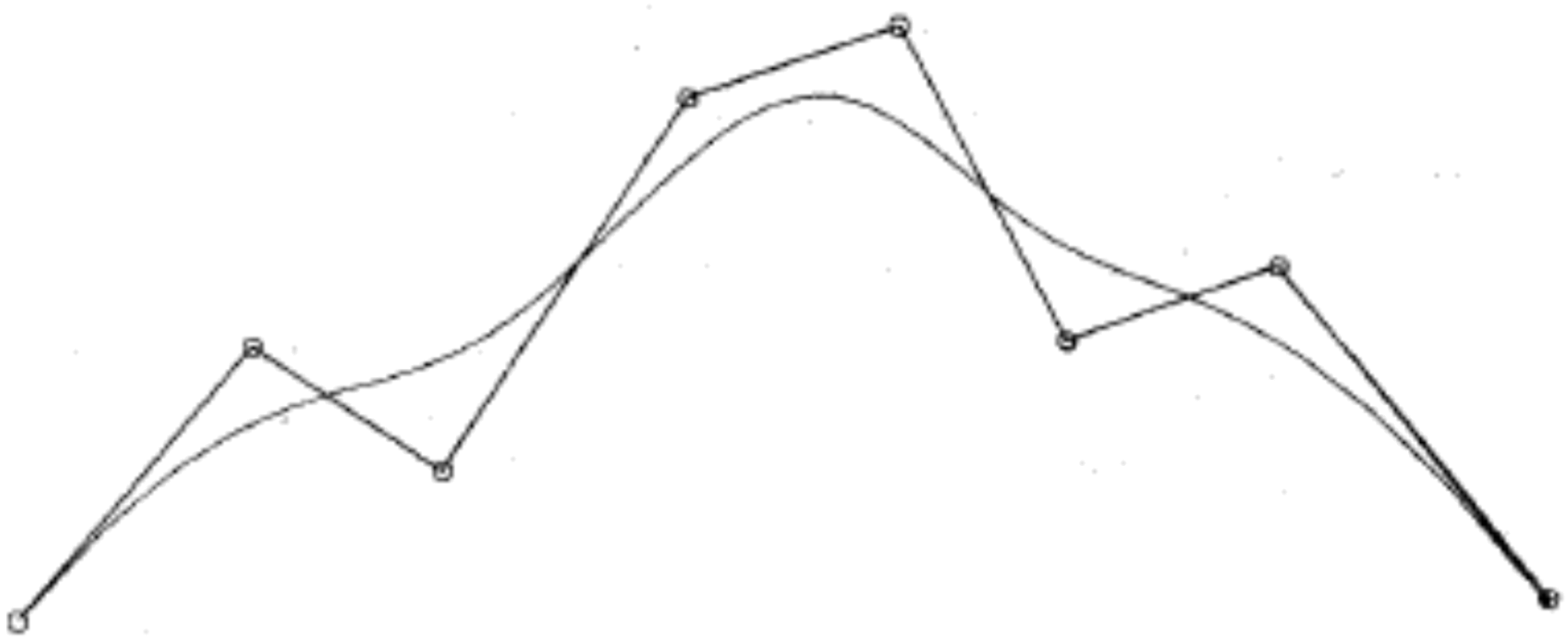
- Example: $k = 4, n = 5$





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

B-spline curves properties

- $\boldsymbol{x}(t_0) = \boldsymbol{d}_0, \boldsymbol{x}(t_{n+1}) = \boldsymbol{d}_n$ (end point interpolation)

- $\boldsymbol{x}'(t_0) = \frac{k-1}{t_k - t_0}(\boldsymbol{d}_1 - \boldsymbol{d}_0)$ (tangent direction at $\boldsymbol{d}_0$, similar in $\boldsymbol{d}_n$)

- $\boldsymbol{x}(t)$ consists of $n - k + 2$ polynomial curve segments of degree $k - 1$ (assuming no multiple inner knots)

B-spline curves properties

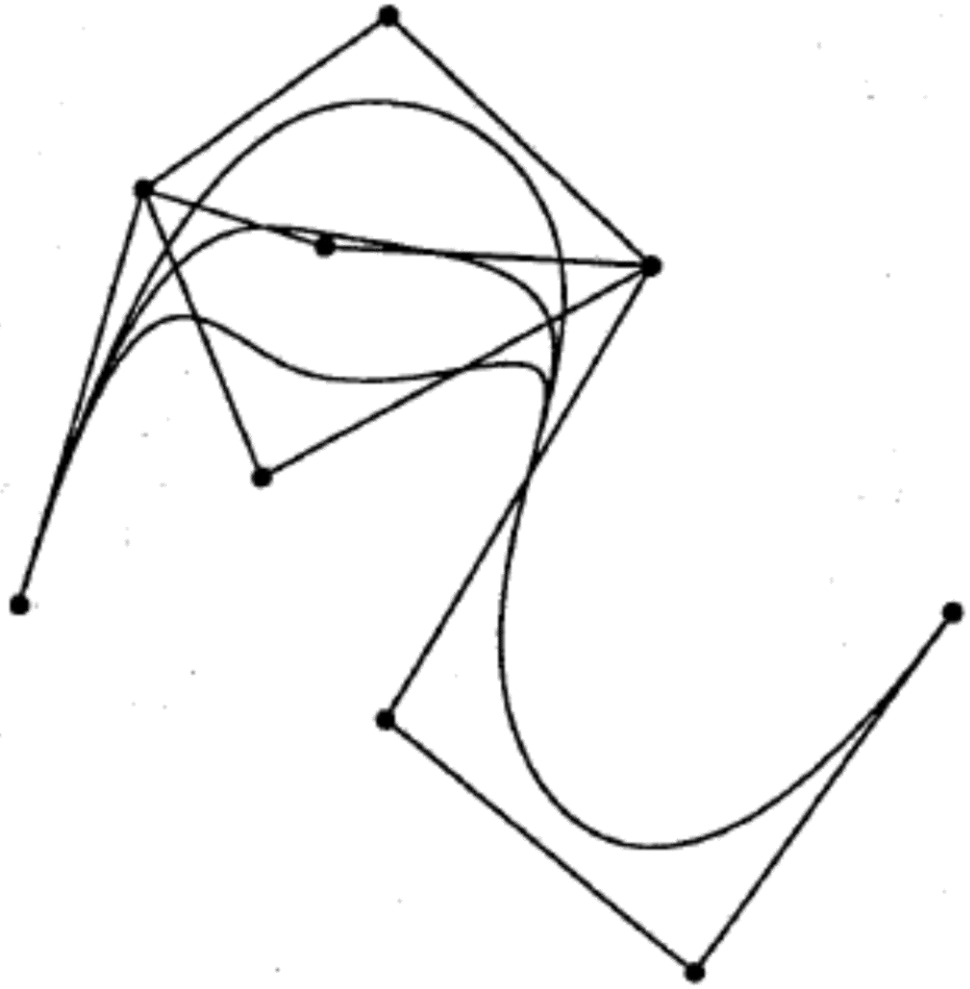
- Multiple inner knots ⇒ reduction of continuity of $x(t)$.

  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 Bézier curves;
  consists of a number of linear interpolations on the de Boor polygon

The de Boor algorithm

- Given:

  $\boldsymbol{d}_0, \ldots, \boldsymbol{d}_n$: de Boor points

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

  Knot vector

- wanted:

  Curve point $\boldsymbol{x}(t)$ 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 $i = r - k + 1, \ldots, r$

   $d_i^0 = d_i$

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

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

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

  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:

$$
\begin{array}{lllll}
d_{r-k+1} = d_{r-k+1}^0 & & & & \\
d_{r-k+2} = d_{r-k+2}^0 & d_{r-k+2}^1 & & & \\
\quad\cdots & & & & \\
d_{r-1} = d_{r-1}^0 & d_{r-1}^1 & \cdots & d_{r-1}^{k-2} & \\
d_r = d_r^0 & d_r^1 & \cdots & d_r^{k-2} & d_r^{k-1} = x(t)
\end{array}
$$

B-spline curves: interpolation

Interpolating B-spline curves

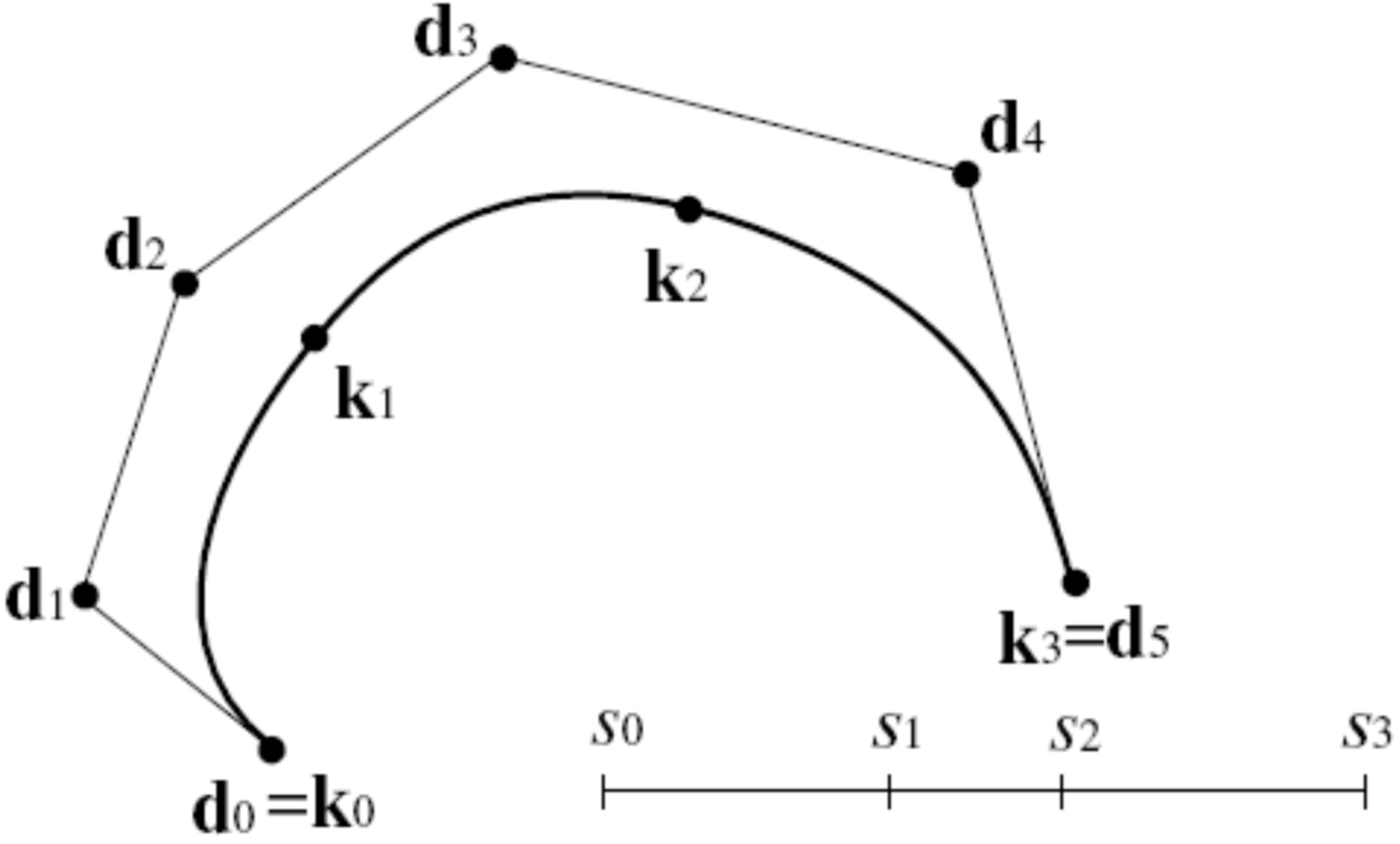
- Given: $n+1$ control points $\boldsymbol{k}_0,\cdots,\boldsymbol{k}_n$

  knot sequence $s_0,\cdots,s_n$
- Wanted: piecewise cubic interpolating B-spline curve $\boldsymbol{x}$

  i.e., $\boldsymbol{x}(s_i) = \boldsymbol{k}_i$ for $i = 0, \dots, n$
- Approach: piecewise cubic ⇒ $k = 4$
  - $\boldsymbol{x}(t)$ consists of n segments ⇒ $n+3$ de Boor points

B-spline curves: interpolation

- Example: $n = 3$



B-spline curves: interpolation

- We choose the knot vector
  - $T = (t_0, t_1, t_2, t_3, t_4, \dots, t_{n+2}, t_{n+3}, t_{n+4}, t_{n+5}, t_{n+6})$
    $= (s_0, s_0, s_0, s_0, s_1, \dots, s_{n-1}, s_n, \quad s_n, \quad s_n, \quad s_n)$

- Then, the following conditions arise:

$$\boldsymbol{x}(s_0) = \boldsymbol{k}_0 = \boldsymbol{d}_0$$
$$\boldsymbol{x}(s_i) = \boldsymbol{k}_i = N_{i,4}(s_i)\boldsymbol{d}_i + N_{i+1,4}(s_i)\boldsymbol{d}_{i+1} + N_{i+2,4}(s_i)\boldsymbol{d}_{i+2}$$
$$\text{for } i = 1, \dots, n-1$$
$$\boldsymbol{x}(s_n) = \boldsymbol{k}_n = \boldsymbol{d}_{n+2}$$

- Total: $n + 1$ conditions for $n + 3$ unknown de Boor points

  $\rightarrow$ 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:

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

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

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