Naval Architecture & Ocean Engineering

Computer Aided Ship design

-Part II. Hull Form Modeling-

September, 2009

Prof. Kyu-Yeul Lee

Department of Naval Architecture and Ocean Engineering,
Seoul National University of College of Engineering

Seoul National Univ.

@SDAL
Advanced Ship Design Automation Lab.
http://asdal.snu.ac.kr

2.3 B[asis]-spline curves

2.3.1 Definition of B-spline curves

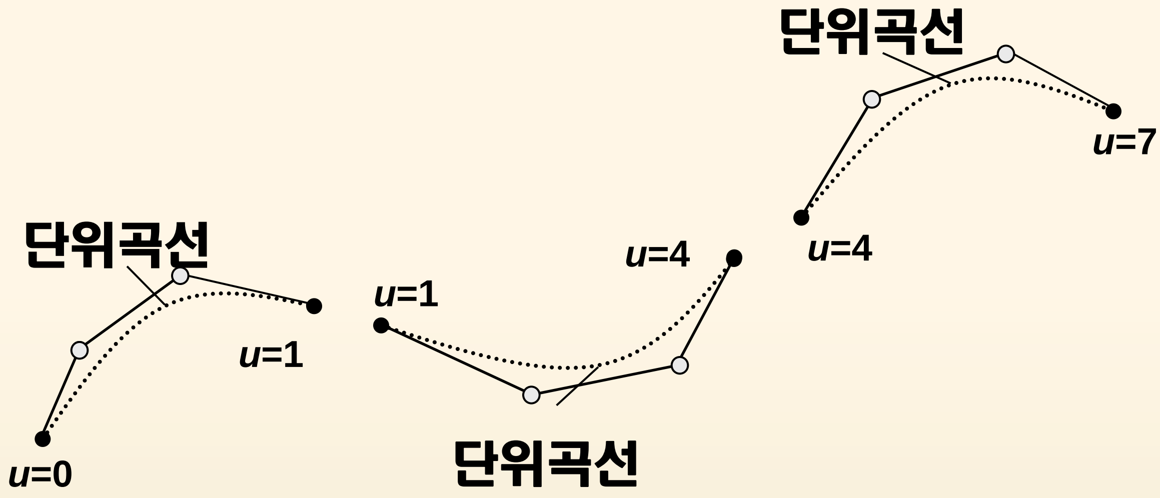
2.3.1.1 Knots, spline curves

2.3.1.2 Definition of B-spline curves

2.3.1.3 Geometric meanings of cubic B-spline curve

Seoul National Univ.
@SDAL Advanced Ship Design Automation Lab.
http://asdal.snu.ac.kr

2.3.1.1 Knot & spline curves

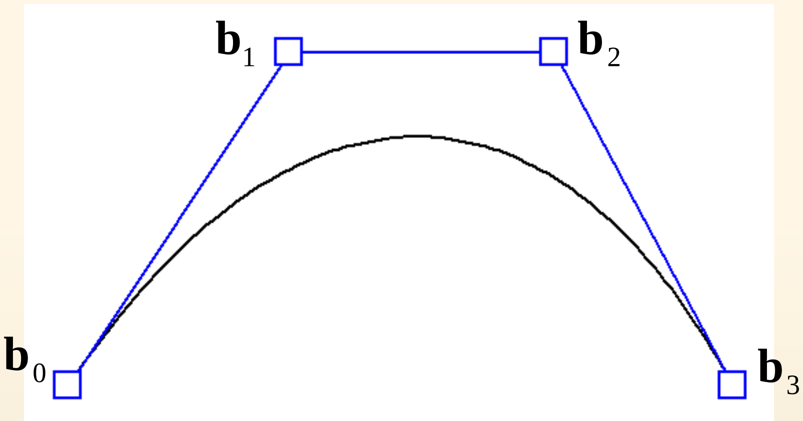


노트 = {…, 0, 1, 4, 7, …}

- 단위 곡선들을 "부드럽게" 연결한 곡선: spline curve
- 단위 곡선을 묶는 매듭 : 놋트(knot)

2.3.1.2 Definition of B-spline curves

Cubic Bezier Curve



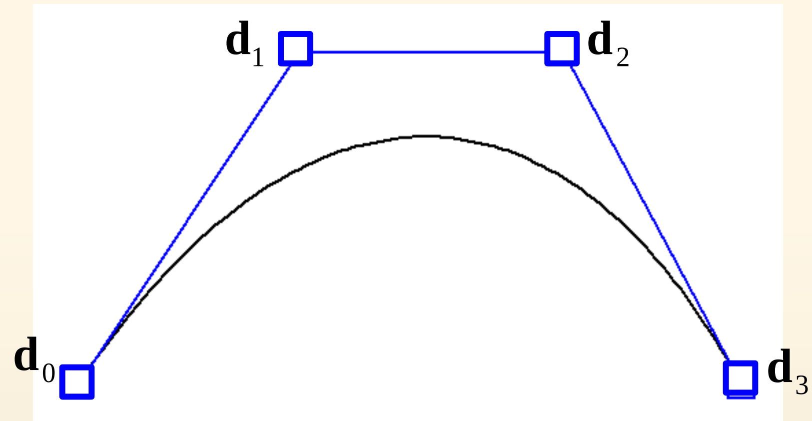
Given: $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, t$

Find: points on curve at parameter t

$$\mathbf{r}(t) = \mathbf{b}_0 B_0^3(t) + \mathbf{b}_1 B_1^3(t) + \mathbf{b}_2 B_2^3(t) + \mathbf{b}_3 B_3^3(t)$$

Bernstein Polynomial Function

Cubic B-spline Curve



Given: $\mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, u$

Find: points on curve at parameter u

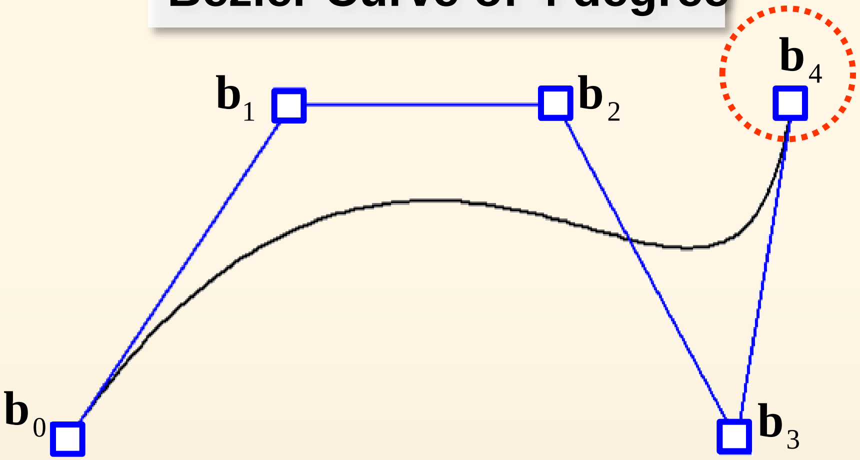
$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u)$$

B-spline Basis Function
(Cox-de Boor Recursive Formula)

Control Point를 하나 더 추가하면 어떻게 될까?

2.3.1.2 Definition of B-spline curves

Bezier Curve of 4 degree



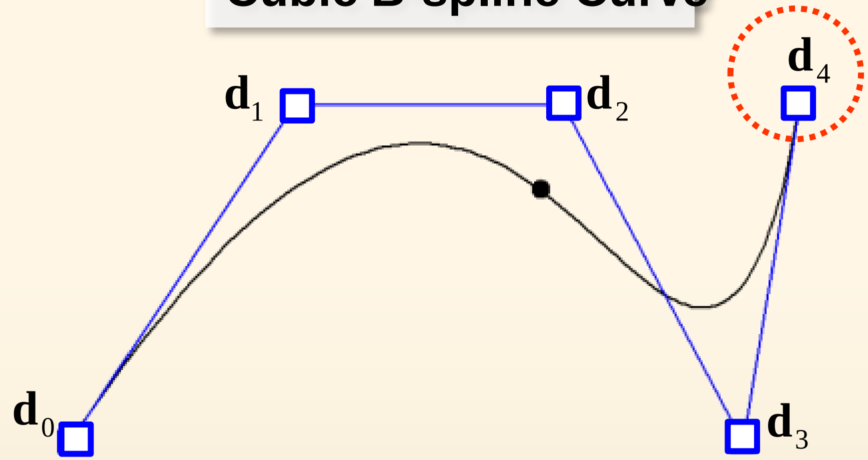
Given: $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4, t$

Find: points on curve at parameter ***t***

$$\mathbf{r}(t) = \mathbf{b}_0 B_0^4(t) + \mathbf{b}_1 B_1^4(t) + \mathbf{b}_2 B_2^4(t) + \mathbf{b}_3 B_3^4(t) + \mathbf{b}_4 B_4^4(t)$$

Bezier Curve를 사용할 경우
Control Point의 개수가 늘어나면
Curve의 차수도 늘어남

Cubic B-spline Curve



Given: $\mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4, u$

Find: points on curve at parameter ***u***

$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u) + \mathbf{d}_4 N_4^3(u)$$

B-spline Curve를 사용할 경우
차수는 변경하지 않은 채
곡선 2개가 생성됨

2.3.1.2 Definition of B-spline curves

Cubic Bezier Curve

Given: $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, t$

Find

$$\mathbf{r}(t) = \mathbf{b}_0 B_0^3(t) + \mathbf{b}_1 B_1^3(t) + \mathbf{b}_2 B_2^3(t) + \mathbf{b}_3 B_3^3(t)$$

Bernstein polynomial function

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i},$$

$$\binom{n}{i} = {}_nC_i = \begin{cases} \dfrac{n!}{i!(n-i)!} & \textbf{if } 0 \le i \le n \\ \mathbf{0} & \textbf{else} \end{cases}$$

☑ Ex): Cubic B-spline curves

- **Given:** d_i, u_j
- **Find:** r(u) (Points on curve at parameter u)

$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \cdots + \mathbf{d}_{D-1} N_{D-1}^3(u)$$

$\mathbf{d}_i$: de Boor points (control points), i = 0, 1, … , D-1

$N_i^n(u)$: B-spline basis function of degree n(=3)

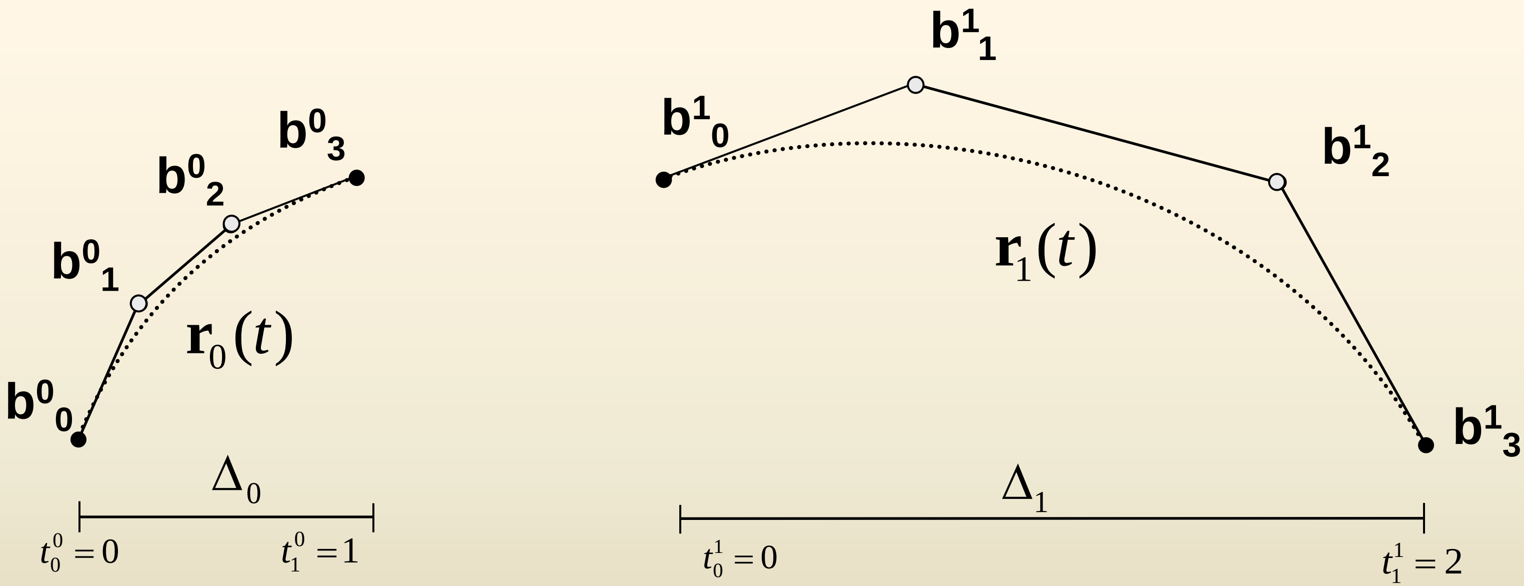
u_j : knots, j = 0, 1, …, K-1

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \textbf{if } u_{i-1} \le u < u_i \\ 0 & \textbf{else} \end{cases}, \quad \sum_{i=0}^{D-1} N_i^n(u) = 1$$

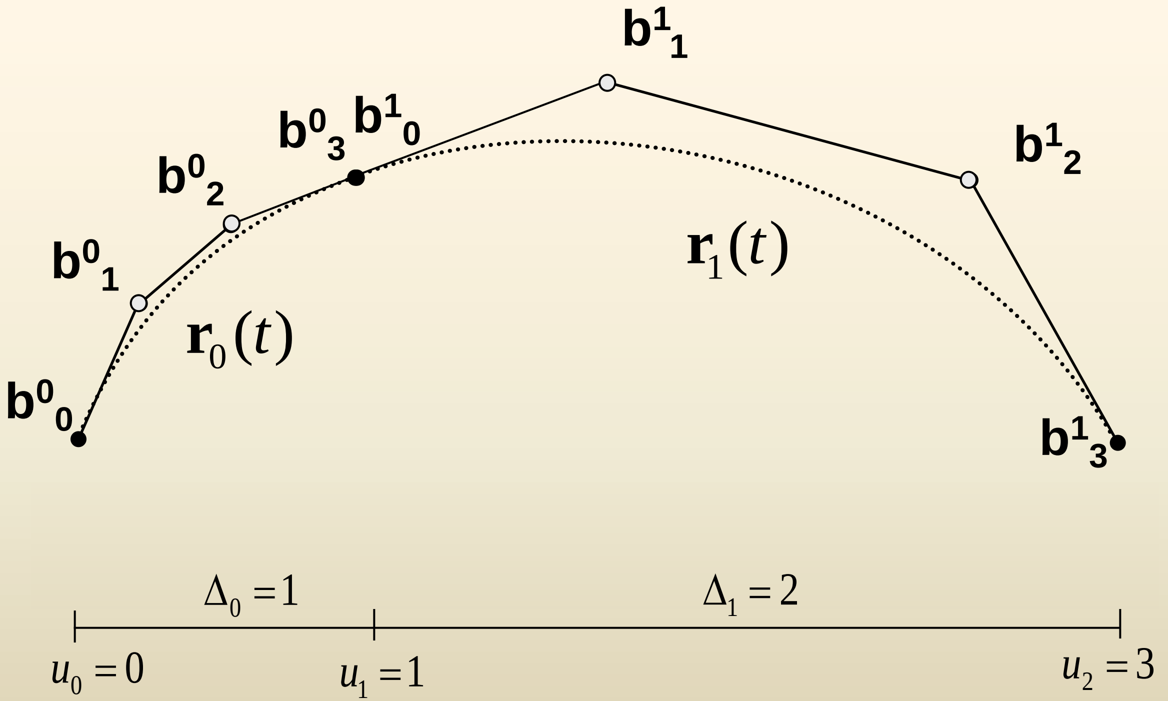
2.3.1.3 Geometric meanings of cubic B-spline curve (1)

☑ 'Cubic' B-spline curve consist of 'cubic' Bezier curves, which are connected with the C^2 continuity condition



2.3.1.3 Geometric meanings of cubic B-spline curve (1)

☑ 'Cubic' B-spline curve consist of 'cubic' Bezier curves, which are connected with the C^2 continuity condition



Assign new global parameter *u* to jointed curve

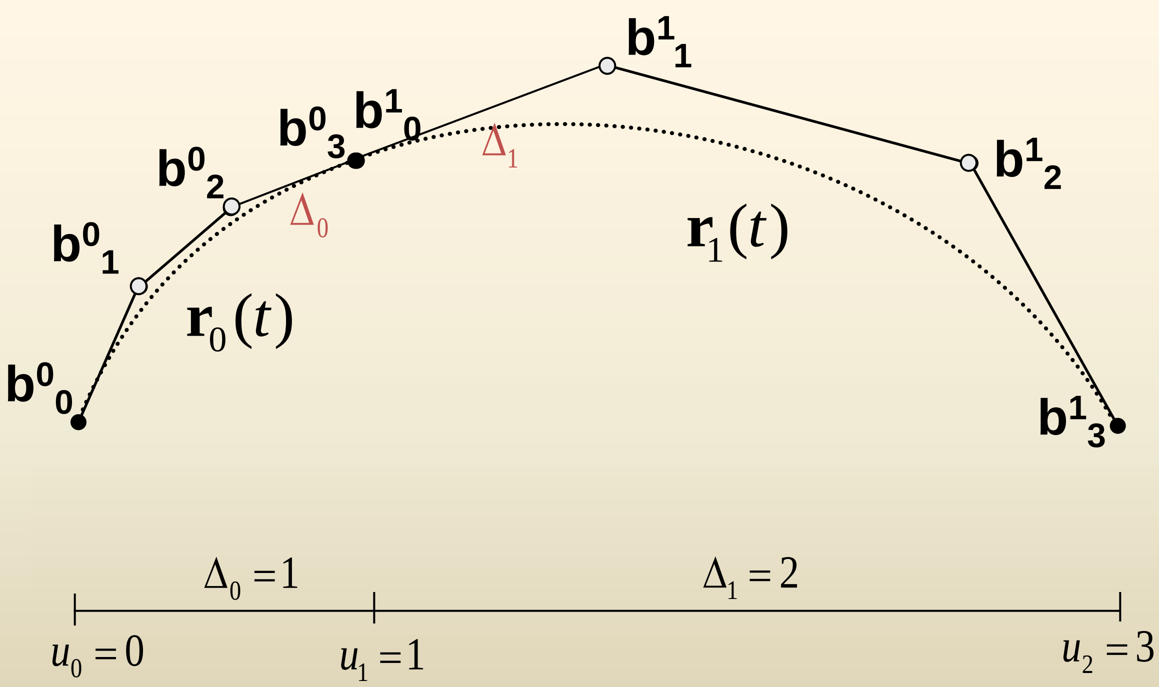
C^0 continuity condition

$$\mathbf{b}^0_3 = \mathbf{b}^1_0$$

2.3.1.3 Geometric meanings of cubic B-spline curve (1)

Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$

☑ 'Cubic' B-spline curve consist of 'cubic' Bezier curves, which are connected with the C^2 continuity condition



Assign new global parameter *u* to jointed curve

C¹ continuity condition

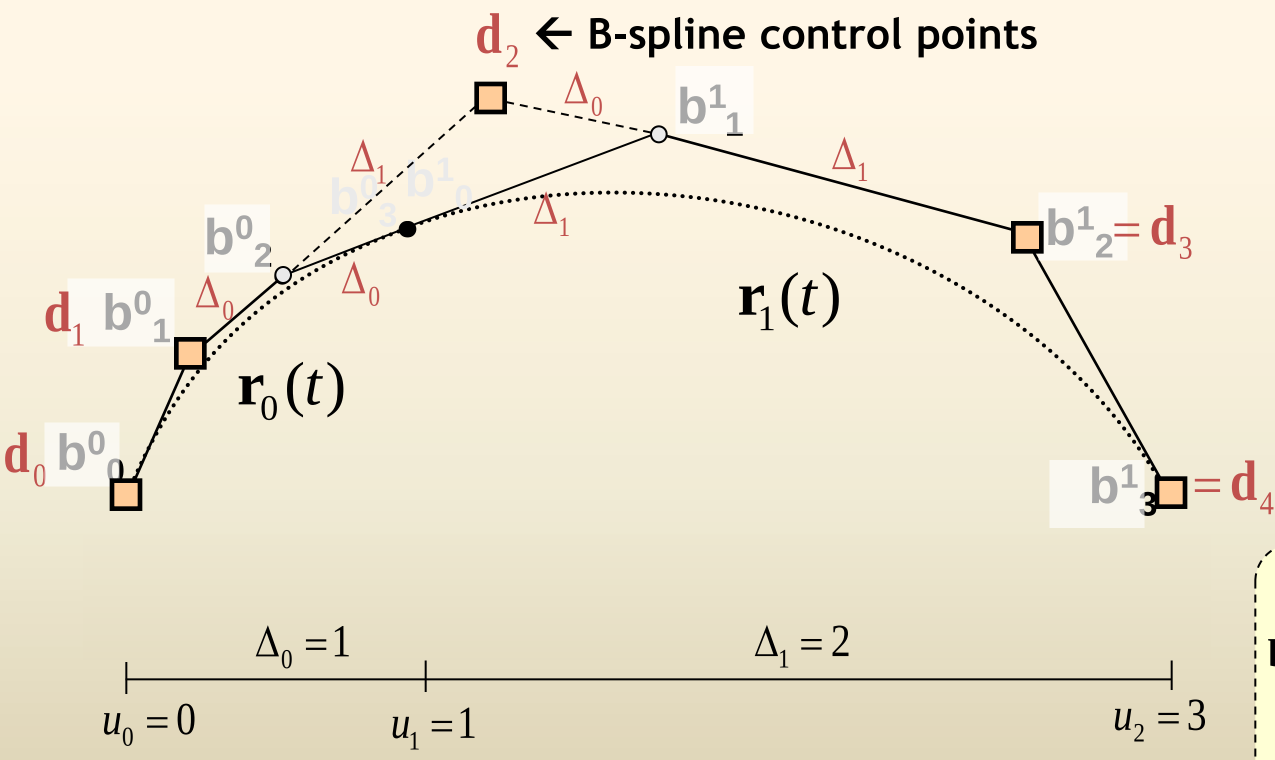
$$\mathbf{b}_3^0 = \mathbf{b}_0^1$$

$$\mathbf{b}_3^0 = \mathbf{b}_0^1 = \frac{\Delta_1}{\Delta_0 + \Delta_1}\mathbf{b}_2^0 + \frac{\Delta_0}{\Delta_0 + \Delta_1}\mathbf{b}_1^1$$

2.3.1.3 Geometric meanings of cubic B-spline curve (1)

Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$

☑ 'Cubic' B-spline curve consist of 'cubic' Bezier curves, which are connected with the C^2 continuity condition



Assign new global parameter u to jointed curve

C^2 continuity condition

$$\mathbf{b}_3^0 = \mathbf{b}_0^1$$

$$\mathbf{b}_3^0 = \mathbf{b}_0^1 = \frac{\Delta_1}{\Delta_0 + \Delta_1}\mathbf{b}_2^0 + \frac{\Delta_0}{\Delta_0 + \Delta_1}\mathbf{b}_1^1$$

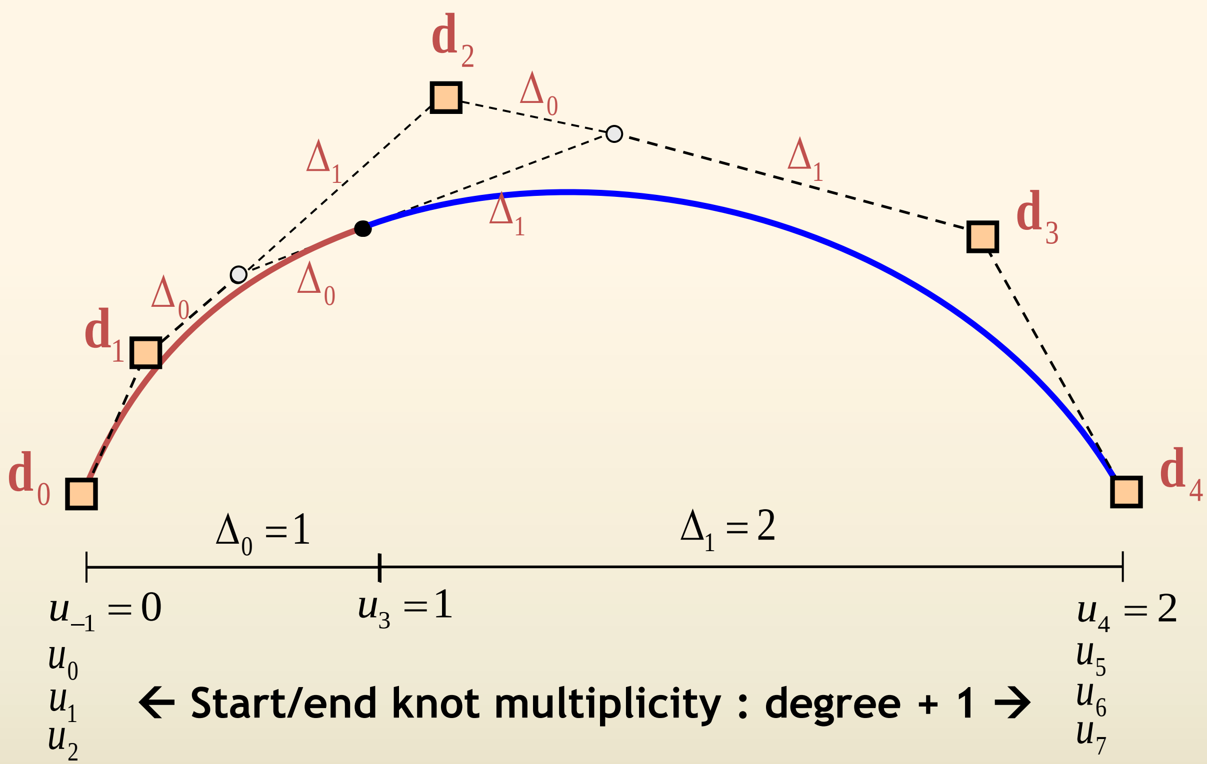
$$\mathbf{b}_2^0 = \{\frac{\Delta_1}{\Delta_0 + \Delta_1}\}\mathbf{d}_1 + \{\frac{\Delta_0}{\Delta_0 + \Delta_1}\}\mathbf{d}_2$$

$$\mathbf{b}_1^1 = \{\frac{\Delta_1}{\Delta_0 + \Delta_1}\}\mathbf{d}_2 + \{\frac{\Delta_0}{\Delta_0 + \Delta_1}\}\mathbf{d}_3$$

2.3.1.3 Geometric meanings of cubic B-spline curve (2)

Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$

- ☑ **n: degree**
- ☑ **S: number of Bezier curve segments**
- ☑ **number of knot = (S-1) + 2(n+1)**
- ☑ **number of control points = 4 + (S-1) = (n+1) + (S-1)**

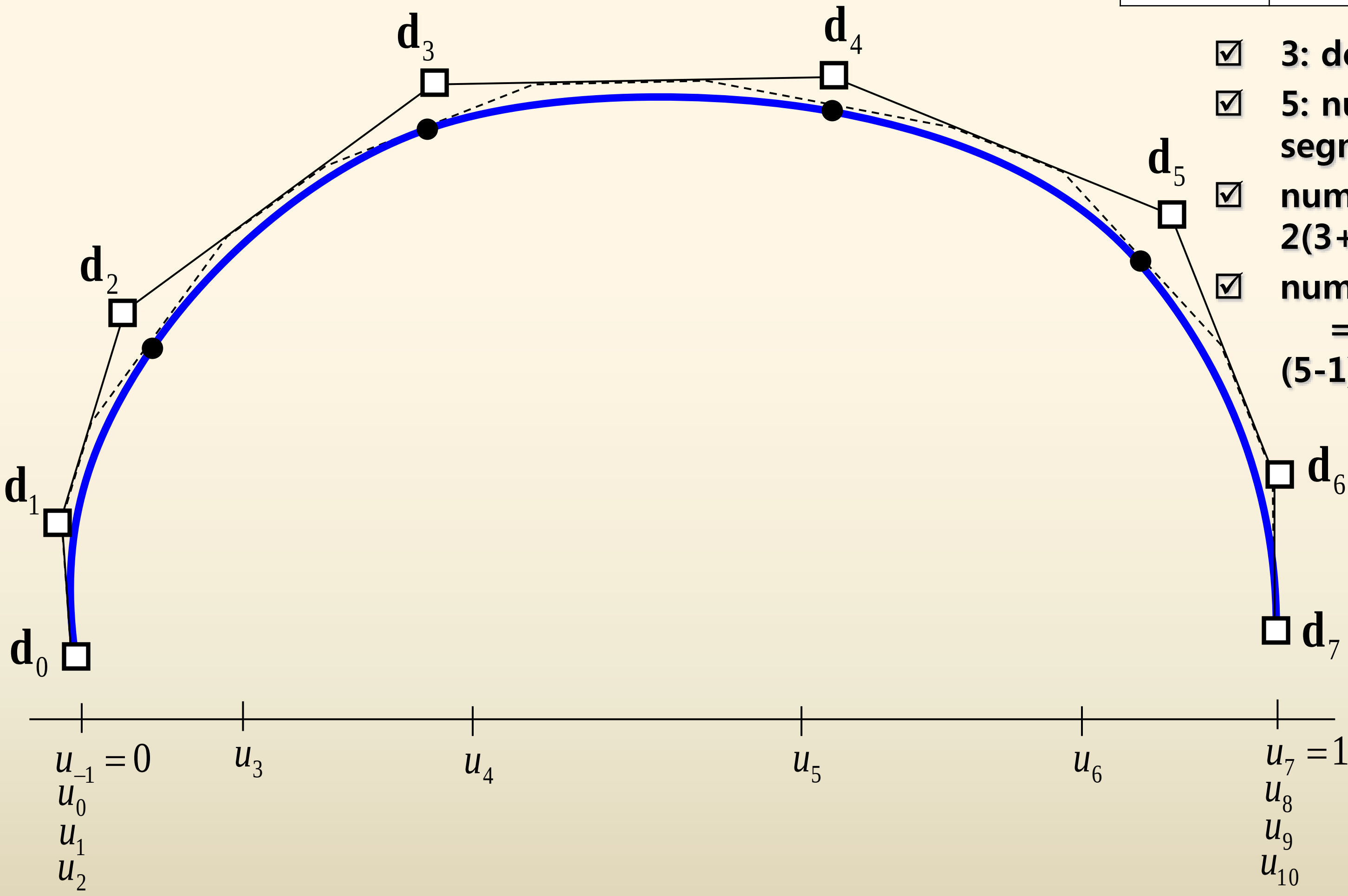


$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u) + \mathbf{d}_4 N_4^3(u)$$

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u) \quad N_i^0(u) = \begin{cases} 1 & \textbf{if } u_{i-1} \le u < u_i \\ 0 & \textbf{else} \end{cases}, \sum_{i=0}^{D-1} N_i^n(u) = 1$$

2.3.1.3 Geometric meanings of cubic B-spline curve (3)

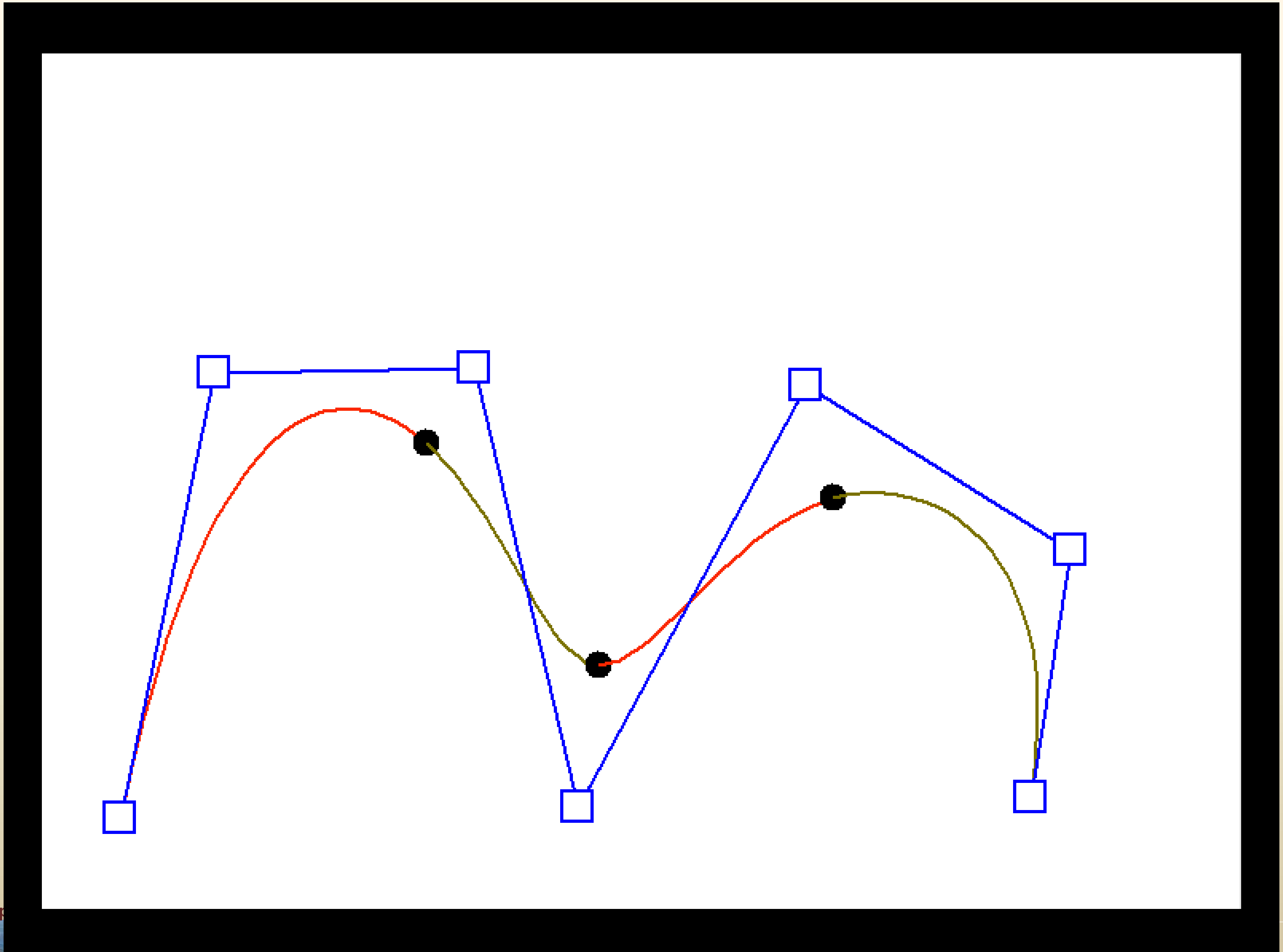
Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$



- ☑ **3: degree**
- ☑ **5: number of Bezier curve segments**
- ☑ **number of knot = (5-1) + 2(3+1)**
- ☑ **number of control points = 4 + (5-1) = (3+1) + (5-1)**

$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u) + \mathbf{d}_4 N_4^3(u) + \mathbf{d}_5 N_5^3(u) + \mathbf{d}_6 N_6^3(u) + \mathbf{d}_7 N_7^3(u)$$

Example of B-spline Curve



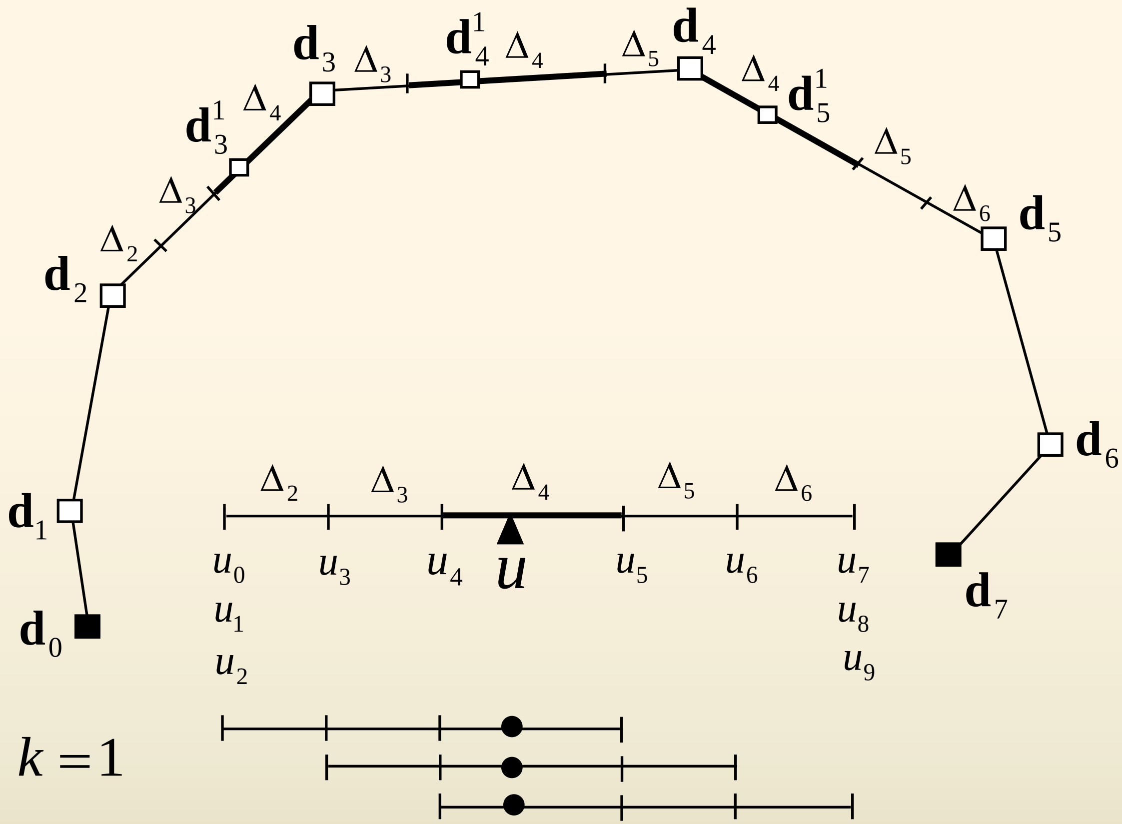
2.3.2 de Boor algorithm

2.3.2.1 de Boor algorithm

2.3.2.2 Relationship between de Boor algorithm & B-spline curves

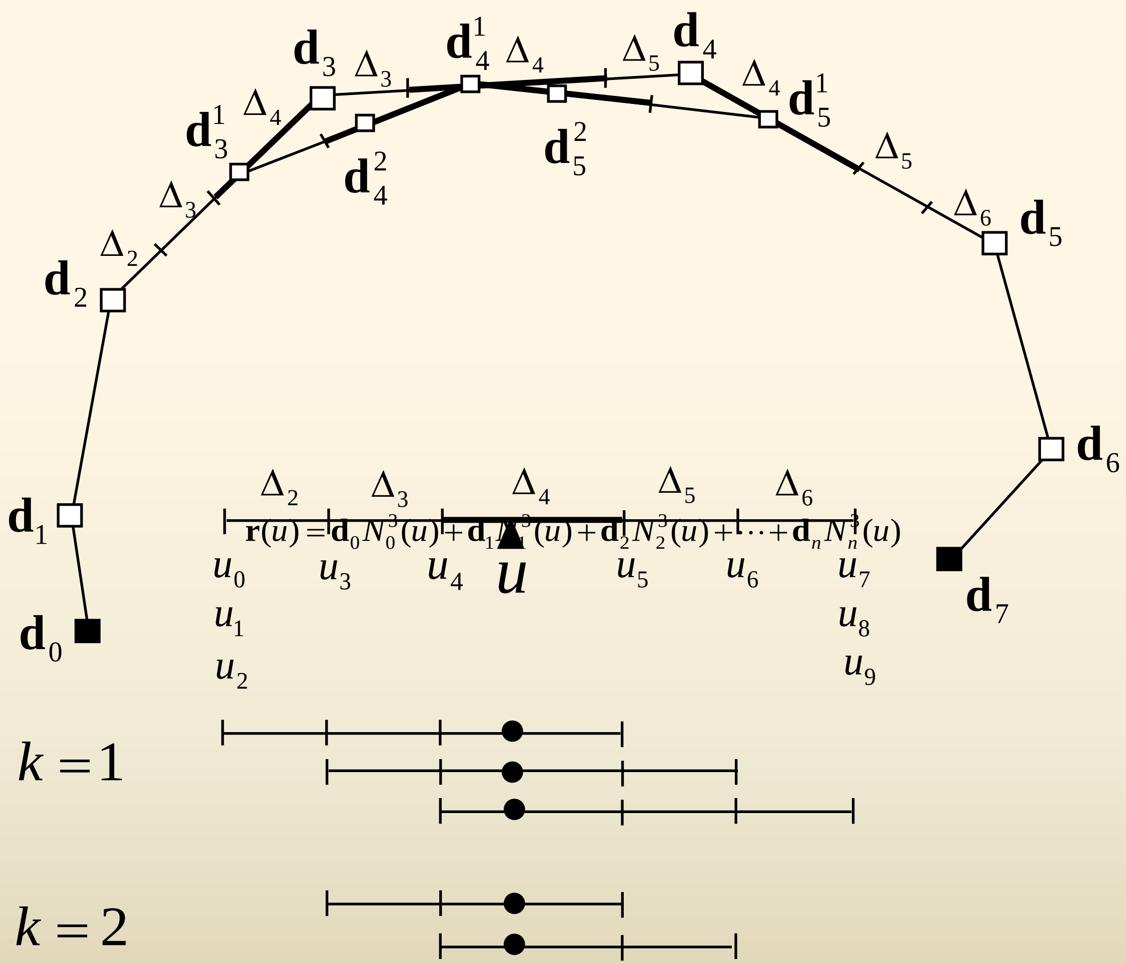
Seoul National Univ.
@SDAL
Advanced Ship Design Automation Lab.
http://asdal.snu.ac.kr

2.3.2.1 de Boor Algorithm (1)

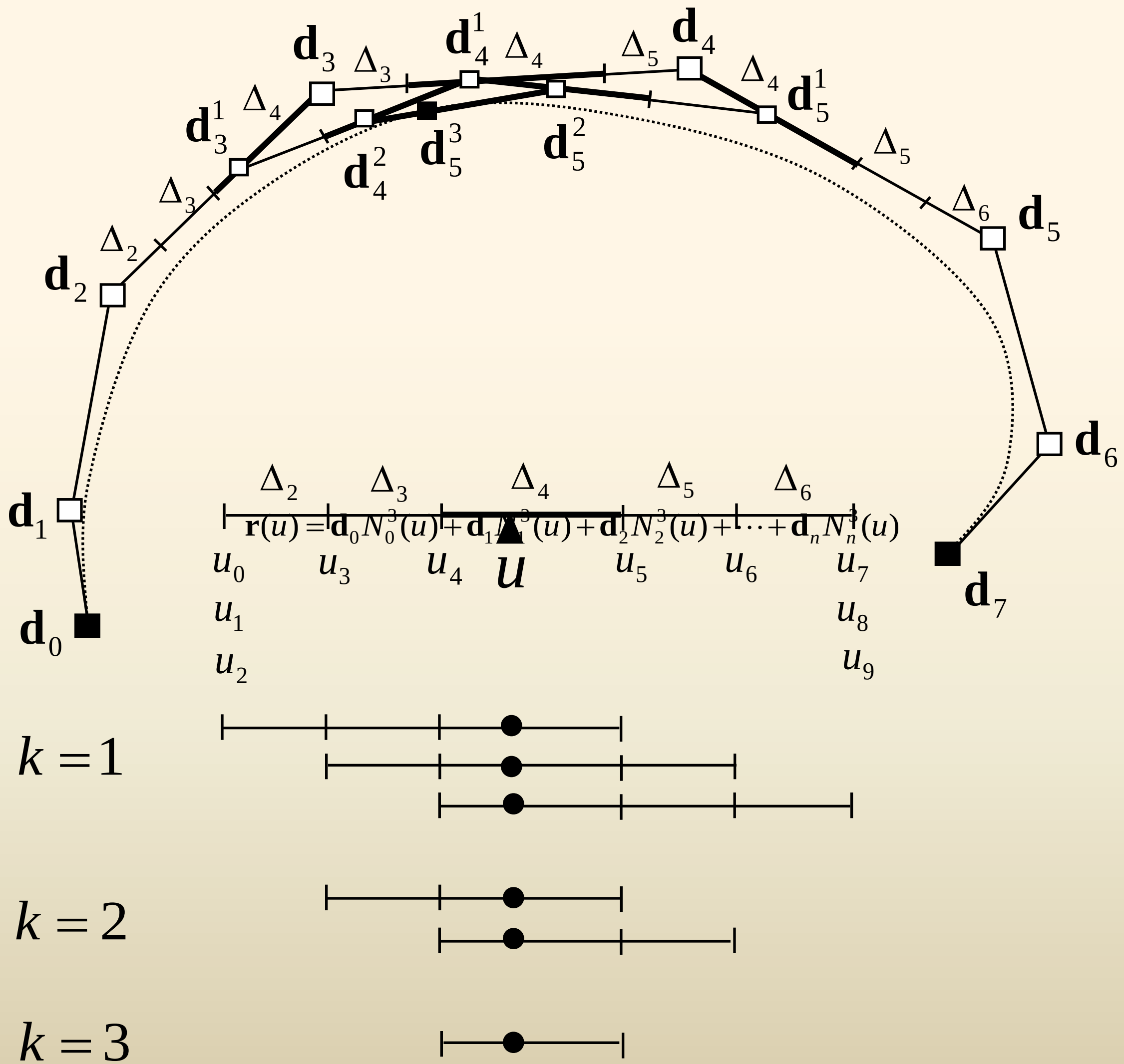


- Linear Interpolation 비율이 t:(1-t)로 일정했던 de Casteljau algorithm에 비하여 de Boor algorithm에서는 Linear Interpolation 비율이 변한다
- 이는 B-spline curve 를 구성하는 Bezier curve segment의 매개변수 간격이 서로 다르기 때문이다

2.3.2.1 de Boor Algorithm (2)



2.3.2.1 de Boor Algorithm (3)



2.3.2.2. Relationship between de Boor algorithm & B-spline curves

☑ **de Boor 알고리즘 : "Constructive Approach"**

Input: $\mathbf{d}_i$ **(de Boor Points)**

Processor: 구간별로 $\mathbf{d}_i$**를** n**번 순차적 'linear interpolation'**

Output : n**차 곡선상의 점**

→ 'B-spline function'(Cox-de Boor recurrence formula) 형태로 표현 됨

$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \cdots + \mathbf{d}_n N_n^3(u)$$

2.3.3 B-spline curve Interpolation

2.3.3.1 Determine number of curve segments & Knots values

2.3.3.2 Problem definition of B-spline curve interpolation

2.3.3.3 Determine Bezier end control points by end tangent vectors

2.3.3.4 Determine Bezier control points by C^1 continuity condition

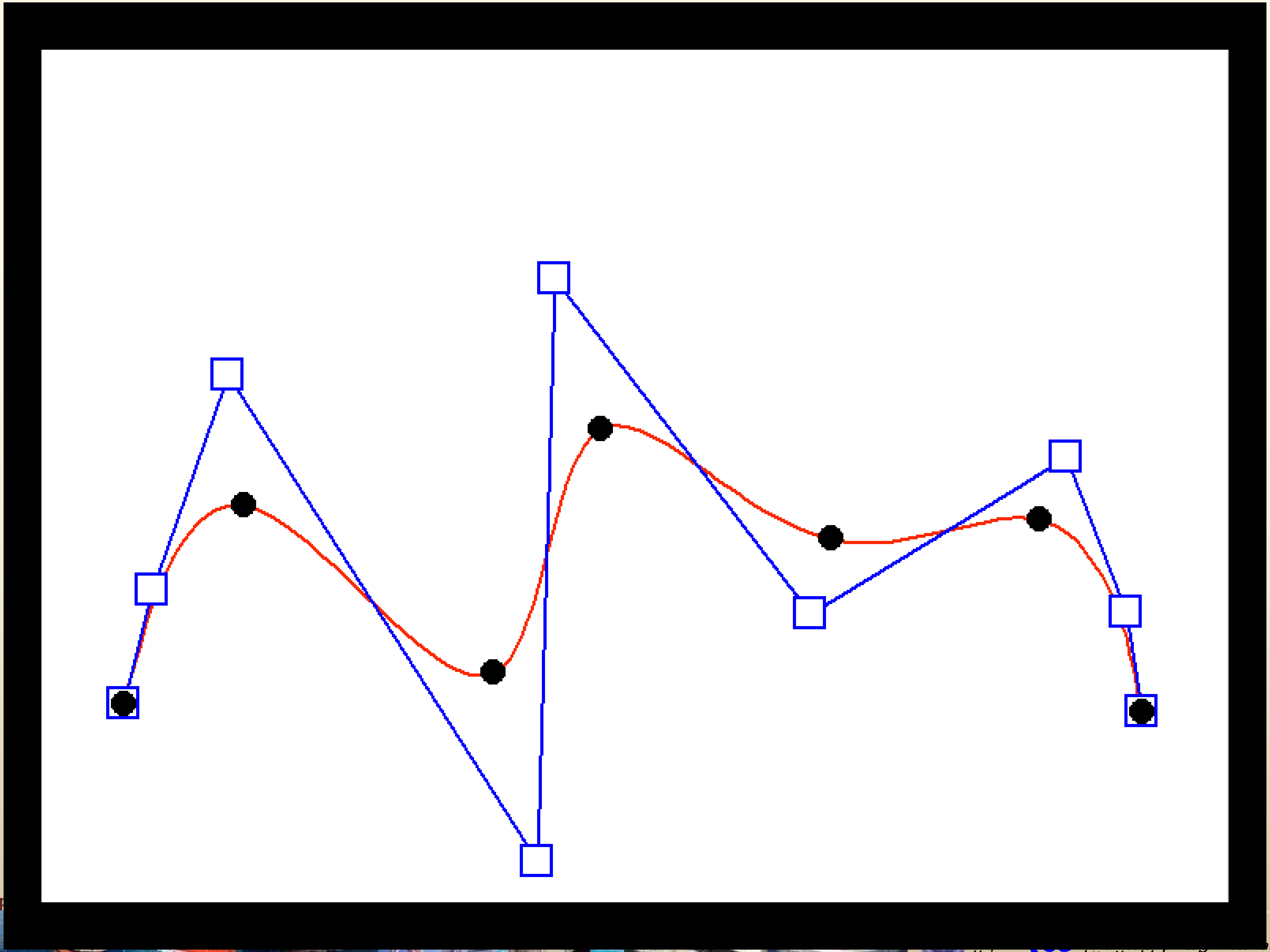
2.3.3.5 Determine B-spline control points by C^2 continuity condition

2.3.3.6 Tridiagonal matrix 해법을 이용한 B-spline 곡선 조정점 결정

2.3.3.7 Bessel end condition

2.3.3.8 Sample code of cubic B-spline curve interpolation

Example of B-spline Interpolation



2.3.3.1 Determine number of Bezier curve segment & Knot value (1)

☑ **Given: fitting points $\mathbf{P}_i$ and corresponding parameter t_i**

where, $i = 0, 1, \ldots, m$ **and** $t_0 = 0, \; t_m = 1,$

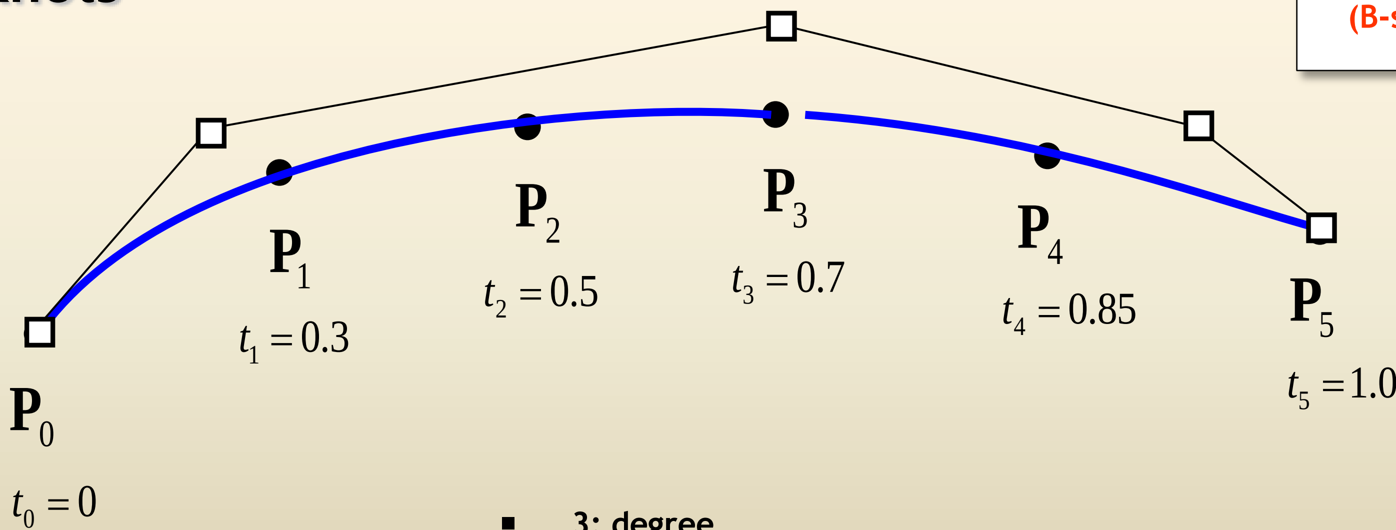
☑ **First, determine number of Bezier curve segment and its knots**

Given:

곡선 상의 점 p_i, t_i
곡선의 놋트 u_j
양끝단의 접선 벡터 t_0, t_1

Find:

곡선 상의 점 p_i을 지나고
C^2 연속 조건을 만족하는
3차 B-spline 곡선 r(u)
(B-spline 조정점: d_i)



- 3: degree
- 2: number of Bezier curve segments
- number of control points
  = 4 + (2-1) = 5

2.3.3.1 Determine number of Bezier curve segment & Knot value (2)

☑ Given: fitting points $\mathbf{P}_i$ and corresponding parameter t_i

where, $i = 0,1,\ldots,m$ and $t_0 = 0, \quad t_m = 1,$

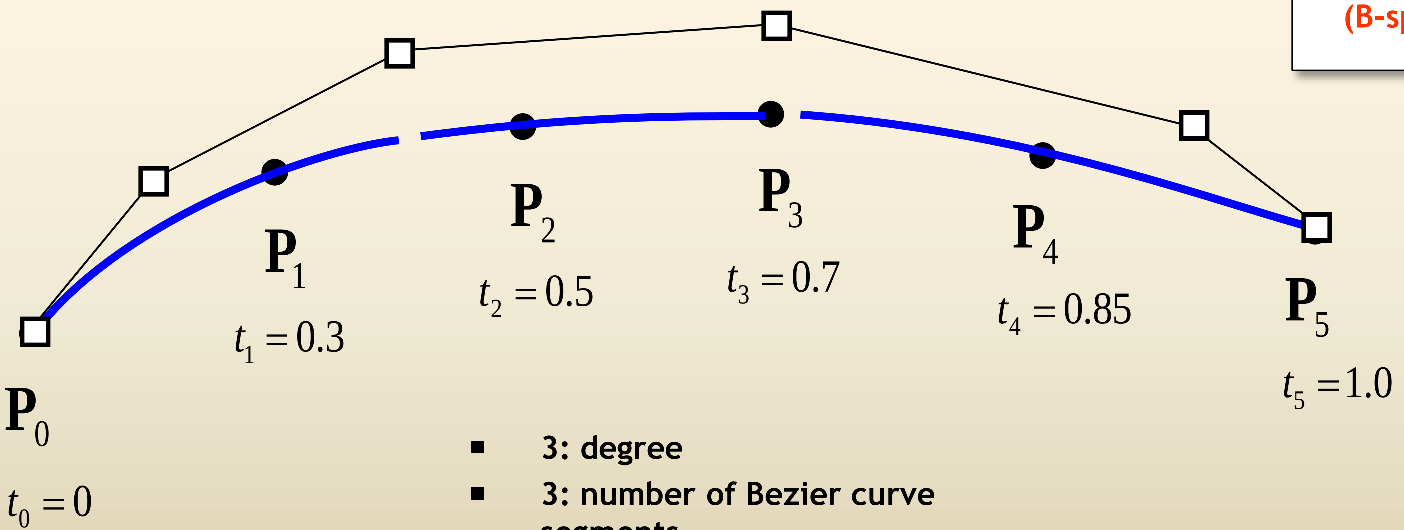
☑ First, determine number of Bezier curve segment and i

Given:

곡선 상의 점 p_i, t_i
곡선의 놋트 u_j
양끝단의 접선 벡터 t_0, t_1

Find:

곡선 상의 점 p_i을 지나고
C^2 연속 조건을 만족하는
3차 B-spline 곡선 $r(u)$
(B-spline 조정점: d_i)



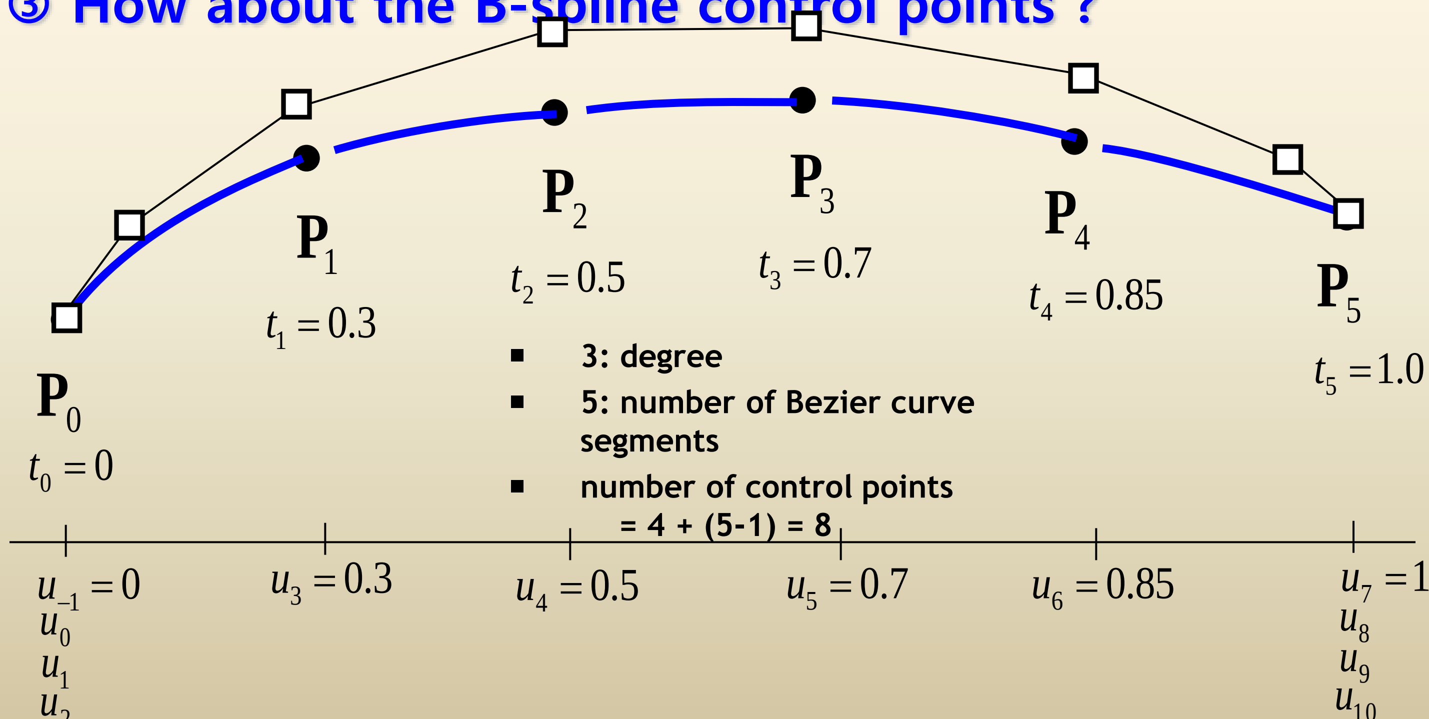
- 3: degree
- 3: number of Bezier curve segments
- number of control points = 4 + (3-1) = 6
- How do we determine Knots? (= start / end points of each cubic Bezier curve)

2.3.3.1 Determine number of Bezier curve segment & Knot value (3)

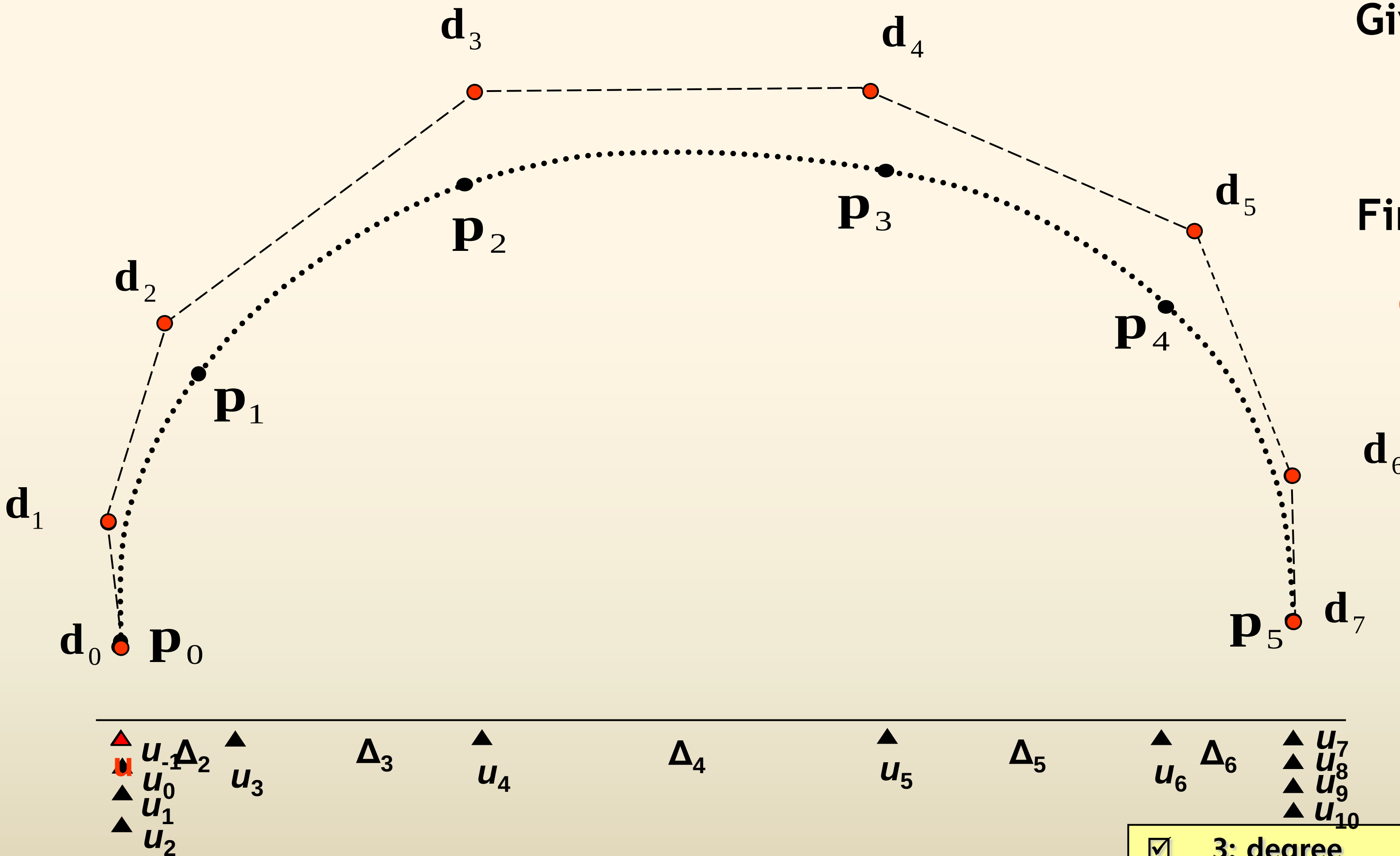
- ☑ Given: fitting points $\mathbf{P}_i$ and corresponding parameter t_i

  where, $i = 0, 1, \ldots, m$ and $t_0 = 0, \; t_m = 1,$

- ☑ ① determine number of Bezier curve segment to be (number of fitting point -1)
- ☑ ② We can determine knots to be the same as the parameters t_i
- ☑ ③ How about the B-spline control points ?



2.3.3.2 Problem definition of cubic B-spline curve interpolation



Given:

곡선 상의 점 p_i
곡선의 놋트 u_j
양끝단의 접선 벡터 t_0, t_1

Find:

곡선 상의 점 p_i을 지나고
C^2 연속 조건을 만족하는
3차 B-spline 곡선 r(u)
(B-spline 조정점: d_i)

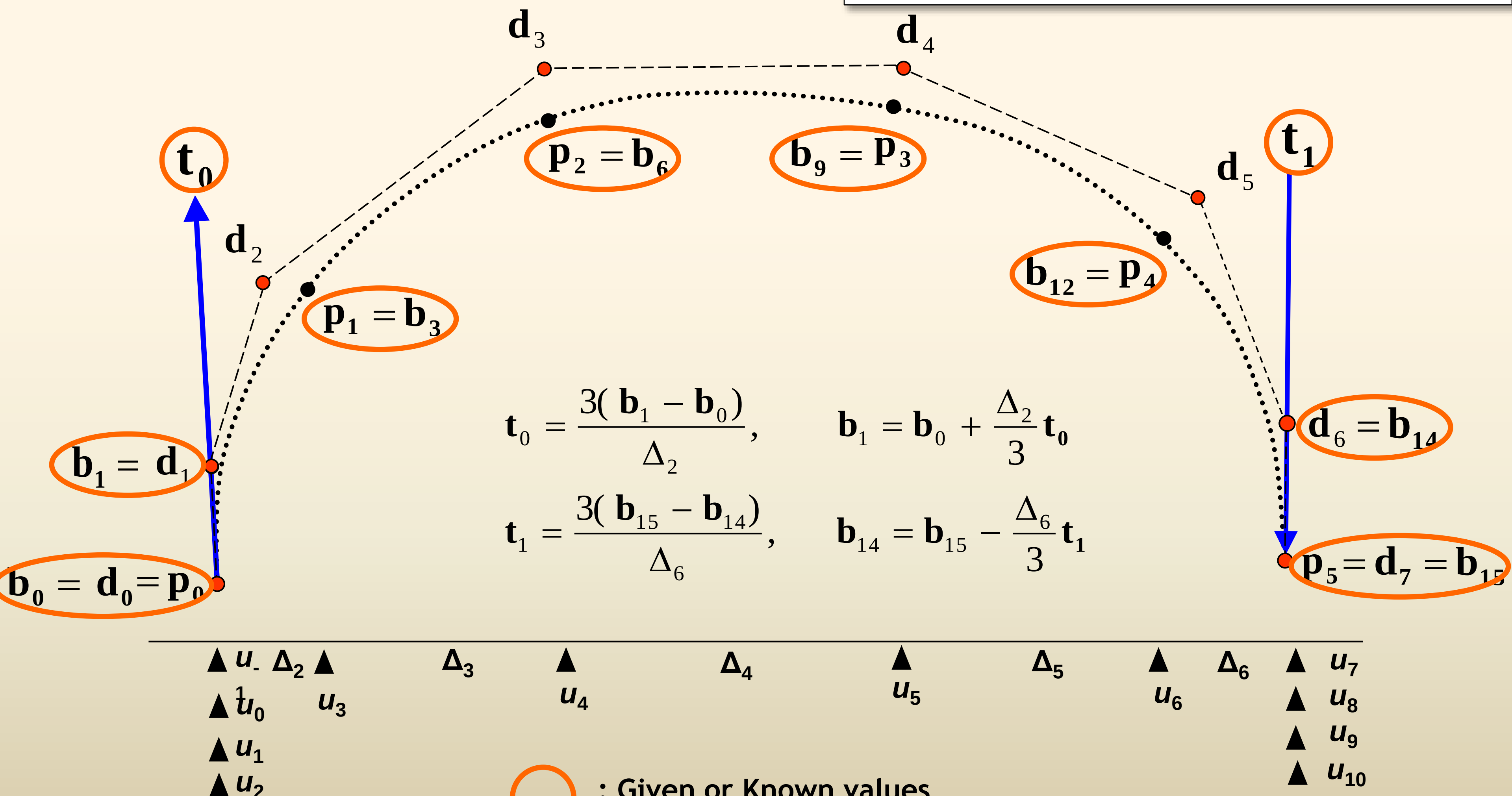
- ☑ 3: degree
- ☑ 5: number of Bezier curve segments
- ☑ number of knot = (5-1) + 2(3+1)
- ☑ number of control points = 4 + (5-1) = (3+1) + (5-1)

가정 : 각 곡선 세그먼트는 3차 Bezier Curve 이다.
연결점에서는 C^1, C^2 연속조건을 만족한다.

2.3.3.3 Determine Bezier end control points by end tangent vectors

Given:
- 곡선 상의 점 p_i
- 곡선의 놋트 u_j
- 양끝단의 접선 벡터 t_0, t_1

Find:
- 곡선 상의 점 p_i을 지나고
- C^2 연속 조건을 만족하는
- 3차 B-spline 곡선 $r(u)$
- (B-spline 조정점: d_i)



$$\mathbf{t}_0 = \frac{3(\mathbf{b}_1 - \mathbf{b}_0)}{\Delta_2}, \qquad \mathbf{b}_1 = \mathbf{b}_0 + \frac{\Delta_2}{3}\mathbf{t}_0$$

$$\mathbf{t}_1 = \frac{3(\mathbf{b}_{15} - \mathbf{b}_{14})}{\Delta_6}, \qquad \mathbf{b}_{14} = \mathbf{b}_{15} - \frac{\Delta_6}{3}\mathbf{t}_1$$

○ : Given or Known values

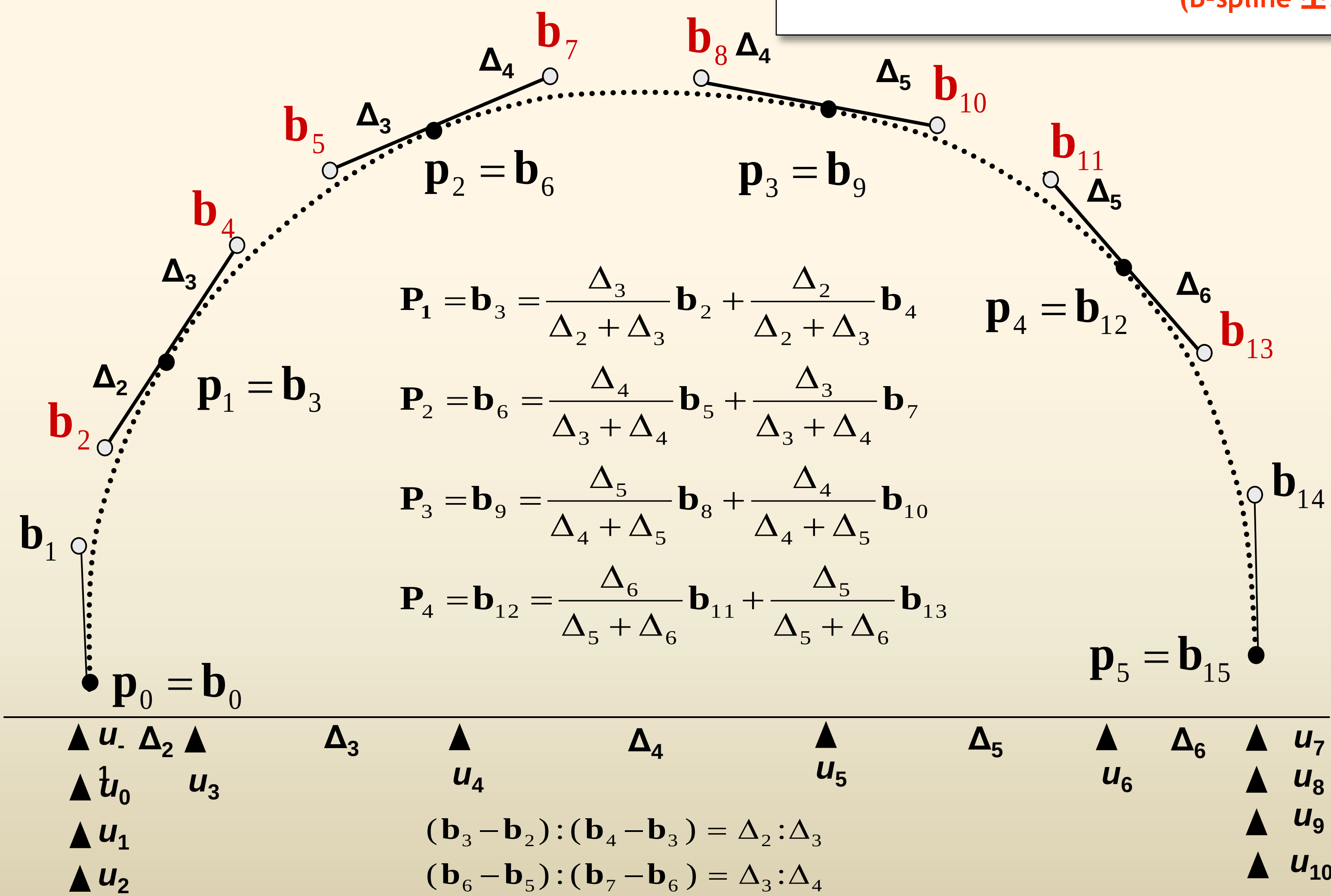
2.3.3.4 Determine Bezier control points by C^1 continuity condition

Given:

곡선 상의 점 p_i
곡선의 놋트 u_j
양끝단의 접선 벡터 t_0, t_1

Find:

곡선 상의 점 p_i을 지나고
C^2 연속 조건을 만족하는
3차 B-spline 곡선 $r(u)$
(B-spline 조정점: d_i)



$$\mathbf{P}_1 = \mathbf{b}_3 = \frac{\Delta_3}{\Delta_2+\Delta_3}\mathbf{b}_2 + \frac{\Delta_2}{\Delta_2+\Delta_3}\mathbf{b}_4$$

$$\mathbf{P}_2 = \mathbf{b}_6 = \frac{\Delta_4}{\Delta_3+\Delta_4}\mathbf{b}_5 + \frac{\Delta_3}{\Delta_3+\Delta_4}\mathbf{b}_7$$

$$\mathbf{P}_3 = \mathbf{b}_9 = \frac{\Delta_5}{\Delta_4+\Delta_5}\mathbf{b}_8 + \frac{\Delta_4}{\Delta_4+\Delta_5}\mathbf{b}_{10}$$

$$\mathbf{P}_4 = \mathbf{b}_{12} = \frac{\Delta_6}{\Delta_5+\Delta_6}\mathbf{b}_{11} + \frac{\Delta_5}{\Delta_5+\Delta_6}\mathbf{b}_{13}$$

$$(\mathbf{b}_3-\mathbf{b}_2):(\mathbf{b}_4-\mathbf{b}_3) = \Delta_2:\Delta_3$$

$$(\mathbf{b}_6-\mathbf{b}_5):(\mathbf{b}_7-\mathbf{b}_6) = \Delta_3:\Delta_4$$

$$(\mathbf{b}_9-\mathbf{b}_8):(\mathbf{b}_{10}-\mathbf{b}_9) = \Delta_4:\Delta_5$$

$$(\mathbf{b}_{12}-\mathbf{b}_{11}):(\mathbf{b}_{13}-\mathbf{b}_{12}) = \Delta_5:\Delta_6$$

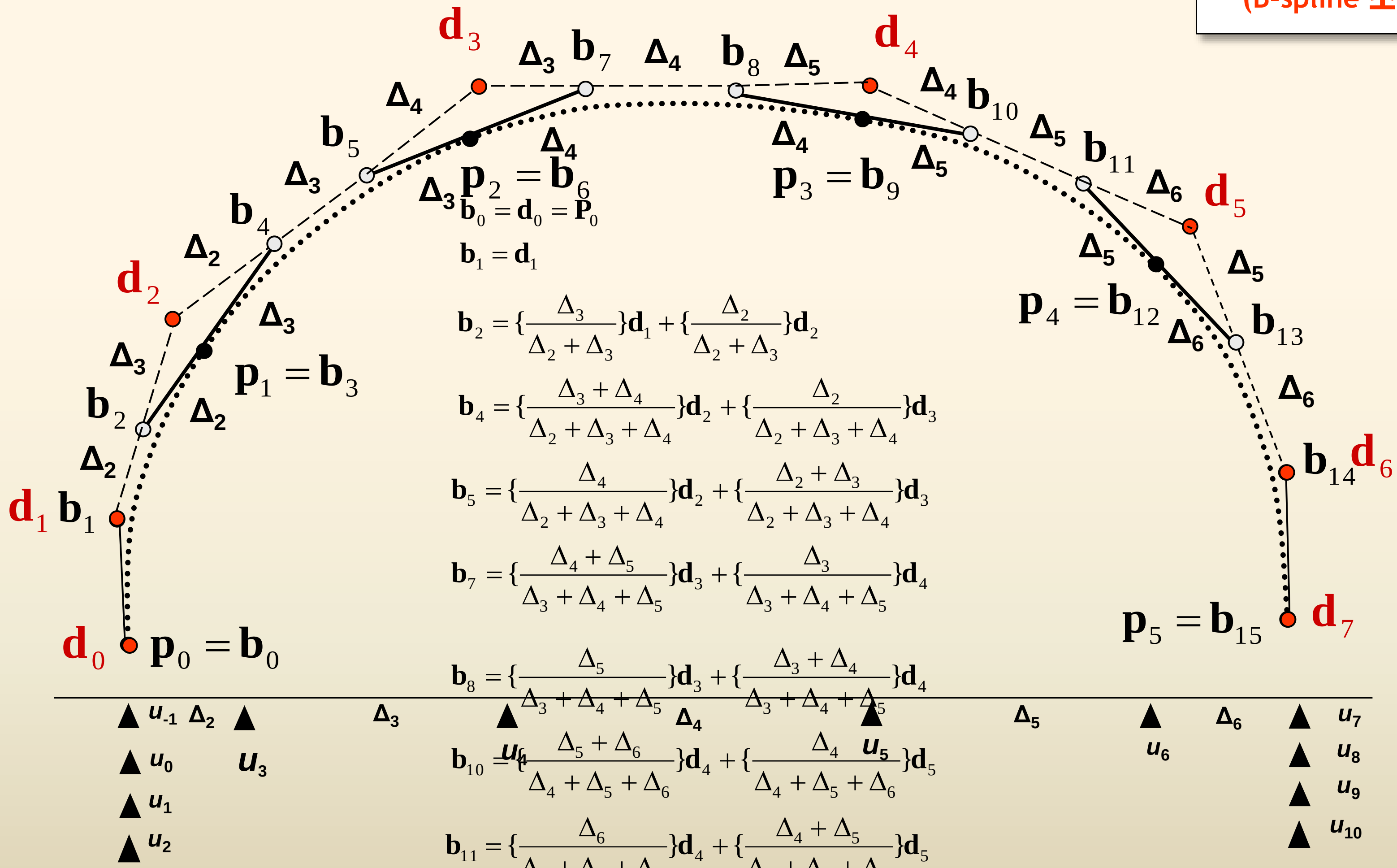
2.3.3.5 Determine B-spline control points by C^2 continuity condition (1)

Given:

곡선 상의 점 p_i
곡선의 놋트 u_j
양끝단의 접선 벡터 t_0, t_1

Find:

곡선 상의 점 p_i을 지나고
C^2 연속 조건을 만족하는
3차 B-spline 곡선 $r(u)$
(B-spline 조정점: d_i)



$$\mathbf{b}_0 = \mathbf{d}_0 = \mathbf{P}_0$$

$$\mathbf{b}_1 = \mathbf{d}_1$$

$$\mathbf{b}_2 = \left\{\frac{\Delta_3}{\Delta_2+\Delta_3}\right\}\mathbf{d}_1 + \left\{\frac{\Delta_2}{\Delta_2+\Delta_3}\right\}\mathbf{d}_2$$

$$\mathbf{b}_4 = \left\{\frac{\Delta_3+\Delta_4}{\Delta_2+\Delta_3+\Delta_4}\right\}\mathbf{d}_2 + \left\{\frac{\Delta_2}{\Delta_2+\Delta_3+\Delta_4}\right\}\mathbf{d}_3$$

$$\mathbf{b}_5 = \left\{\frac{\Delta_4}{\Delta_2+\Delta_3+\Delta_4}\right\}\mathbf{d}_2 + \left\{\frac{\Delta_2+\Delta_3}{\Delta_2+\Delta_3+\Delta_4}\right\}\mathbf{d}_3$$

$$\mathbf{b}_7 = \left\{\frac{\Delta_4+\Delta_5}{\Delta_3+\Delta_4+\Delta_5}\right\}\mathbf{d}_3 + \left\{\frac{\Delta_3}{\Delta_3+\Delta_4+\Delta_5}\right\}\mathbf{d}_4$$

$$\mathbf{b}_8 = \left\{\frac{\Delta_5}{\Delta_3+\Delta_4+\Delta_5}\right\}\mathbf{d}_3 + \left\{\frac{\Delta_3+\Delta_4}{\Delta_3+\Delta_4+\Delta_5}\right\}\mathbf{d}_4$$

$$\mathbf{b}_{10} = \left\{\frac{\Delta_5+\Delta_6}{\Delta_4+\Delta_5+\Delta_6}\right\}\mathbf{d}_4 + \left\{\frac{\Delta_4}{\Delta_4+\Delta_5+\Delta_6}\right\}\mathbf{d}_5$$

$$\mathbf{b}_{11} = \left\{\frac{\Delta_6}{\Delta_4+\Delta_5+\Delta_6}\right\}\mathbf{d}_4 + \left\{\frac{\Delta_4+\Delta_5}{\Delta_4+\Delta_5+\Delta_6}\right\}\mathbf{d}_5$$

$$\mathbf{b}_{13} = \left\{\frac{\Delta_6}{\Delta_5+\Delta_6}\right\}\mathbf{d}_5 + \left\{\frac{\Delta_5}{\Delta_5+\Delta_6}\right\}\mathbf{d}_6$$

$$\mathbf{b}_{14} = \mathbf{d}_6$$

$$\mathbf{b}_{15} = \mathbf{d}_7 = \mathbf{p}_5$$

2.3.3.5 Determine B-spline control points by C^2 continuity condition (2)

Given:
곡선 상의 점 p_i
곡선의 놋트 u_j
양끝단의 접선 벡터 t_0, t_1

Find:
곡선 상의 점 p_i을 지나고
C^2 연속 조건을 만족하는
3차 B-spline 곡선 $r(u)$
(B-spline 조정점: d_i)

C^1, C^2 조건을 이용하여 P_i에 관한 식 유도

C^1 조건

$$\mathbf{P}_1 = \mathbf{b}_3 = \frac{\Delta_3}{\Delta_2+\Delta_3}\mathbf{b}_2 + \frac{\Delta_2}{\Delta_2+\Delta_3}\mathbf{b}_4$$

$$\mathbf{P}_2 = \mathbf{b}_6 = \frac{\Delta_4}{\Delta_3+\Delta_4}\mathbf{b}_5 + \frac{\Delta_3}{\Delta_3+\Delta_4}\mathbf{b}_7$$

$$\mathbf{P}_3 = \mathbf{b}_9 = \frac{\Delta_5}{\Delta_4+\Delta_5}\mathbf{b}_8 + \frac{\Delta_4}{\Delta_4+\Delta_5}\mathbf{b}_{10}$$

$$\mathbf{P}_4 = \mathbf{b}_{12} = \frac{\Delta_6}{\Delta_5+\Delta_6}\mathbf{b}_{11} + \frac{\Delta_5}{\Delta_5+\Delta_6}\mathbf{b}_{13}$$

C^2 조건

$$\mathbf{b}_0 = \mathbf{d}_0 = \mathbf{P}_0$$

$$\mathbf{b}_1 = \mathbf{d}_1$$

$$\mathbf{b}_2 = \{\frac{\Delta_3}{\Delta_2+\Delta_3}\}\mathbf{d}_1 + \{\frac{\Delta_2}{\Delta_2+\Delta_3}\}\mathbf{d}_2$$

$$\mathbf{b}_4 = \{\frac{\Delta_3+\Delta_4}{\Delta_2+\Delta_3+\Delta_4}\}\mathbf{d}_2 + \{\frac{\Delta_2}{\Delta_2+\Delta_3+\Delta_4}\}\mathbf{d}_3$$

$$\mathbf{b}_5 = \{\frac{\Delta_4}{\Delta_2+\Delta_3+\Delta_4}\}\mathbf{d}_2 + \{\frac{\Delta_2+\Delta_3}{\Delta_2+\Delta_3+\Delta_4}\}\mathbf{d}_3$$

$$\mathbf{b}_7 = \{\frac{\Delta_4+\Delta_5}{\Delta_3+\Delta_4+\Delta_5}\}\mathbf{d}_3 + \{\frac{\Delta_3}{\Delta_3+\Delta_4+\Delta_5}\}\mathbf{d}_4$$

$$\mathbf{b}_8 = \{\frac{\Delta_5}{\Delta_3+\Delta_4+\Delta_5}\}\mathbf{d}_3 + \{\frac{\Delta_3+\Delta_4}{\Delta_3+\Delta_4+\Delta_5}\}\mathbf{d}_4$$

$$\mathbf{b}_{10} = \{\frac{\Delta_5+\Delta_6}{\Delta_4+\Delta_5+\Delta_6}\}\mathbf{d}_4 + \{\frac{\Delta_4}{\Delta_4+\Delta_5+\Delta_6}\}\mathbf{d}_5$$

$$\mathbf{b}_{11} = \{\frac{\Delta_6}{\Delta_4+\Delta_5+\Delta_6}\}\mathbf{d}_4 + \{\frac{\Delta_4+\Delta_5}{\Delta_4+\Delta_5+\Delta_6}\}\mathbf{d}_5$$

$$\mathbf{b}_{13} = \{\frac{\Delta_6}{\Delta_5+\Delta_6}\}\mathbf{d}_5 + \{\frac{\Delta_5}{\Delta_5+\Delta_6}\}\mathbf{d}_6$$

$$\mathbf{b}_{14} = \mathbf{d}_6$$

$$\mathbf{b}_{15} = \mathbf{d}_7 = \mathbf{p}_5$$

2.3.3.5 Determine B-spline control points by C^2 continuity condition (3)

Given:

곡선 상의 점 p_i
곡선의 놋트 u_j
양끝단의 접선 벡터 t_0, t_1

Find:

곡선 상의 점 p_i을 지나고
C^2 연속 조건을 만족하는
3차 B-spline 곡선 $r(u)$
(B-spline 조정점: d_i)

$$\mathbf{P}_1 = \frac{1}{(\Delta_2+\Delta_3)(\Delta_2+\Delta_3+\Delta_4)}[(\Delta_3)^2(\Delta_2+\Delta_3+\Delta_4)/(\Delta_2+\Delta_3)\mathbf{d}_1$$
$$+\{\Delta_2\Delta_3(\Delta_2+\Delta_3+\Delta_4)+\Delta_2(\Delta_2+\Delta_3)(\Delta_3+\Delta_4)\}/(\Delta_2+\Delta_3)\mathbf{d}_2+(\Delta_2)^2\mathbf{d}_3]$$
$$=\alpha_1\mathbf{d}_1+\beta_1\mathbf{d}_2+\gamma_1\mathbf{d}_3$$

$$\alpha_i = \frac{(\Delta_{i+2})^2}{(\Delta_i+\Delta_{i+1}+\Delta_{i+2})(\Delta_{i+1}+\Delta_{i+2})}$$
$$\beta_i = \left\{\frac{\Delta_{i+2}(\Delta_i+\Delta_{i+1})}{(\Delta_i+\Delta_{i+1}+\Delta_{i+2})}+\frac{\Delta_{i+1}(\Delta_{i+2}+\Delta_{i+3})}{(\Delta_{i+1}+\Delta_{i+2}+\Delta_{i+3})}\right\}/(\Delta_{i+1}+\Delta_{i+2})$$
$$\gamma_i = \frac{(\Delta_{i+1})^2}{(\Delta_{i+1}+\Delta_{i+2}+\Delta_{i+3})(\Delta_{i+1}+\Delta_{i+2})}$$

$$\mathbf{P}_2 = \frac{1}{(\Delta_3+\Delta_4)(\Delta_3+\Delta_4+\Delta_5)}[(\Delta_4)^2\mathbf{d}_2+\{\Delta_4(\Delta_2+\Delta_3)+$$
$$\Delta_3(\Delta_4+\Delta_5)\}\mathbf{d}_3+(\Delta_3)^2\mathbf{d}_4]=\alpha_2\mathbf{d}_2+\beta_2\mathbf{d}_3+\gamma_2\mathbf{d}_4$$

$$\mathbf{P}_3 = \frac{1}{(\Delta_4+\Delta_5)(\Delta_3+\Delta_4+\Delta_5)}[(\Delta_5)^2\mathbf{d}_3+\{\Delta_5(\Delta_3+\Delta_4)(\Delta_4+\Delta_5+\Delta_6)$$
$$+\Delta_4(\Delta_5+\Delta_6)(\Delta_3+\Delta_4+\Delta_5)\}/(\Delta_4+\Delta_5+\Delta_6)\mathbf{d}_4+(\Delta_4)^2(\Delta_3+\Delta_4+\Delta_5)$$
$$/(\Delta_4+\Delta_5+\Delta_6)\mathbf{d}_5]=\alpha_3\mathbf{d}_3+\beta_3\mathbf{d}_4+\gamma_3\mathbf{d}_5$$

$$\mathbf{P}_4 = \frac{1}{(\Delta_5+\Delta_6)(\Delta_4+\Delta_5+\Delta_6)}[(\Delta_6)^2\mathbf{d}_4+$$
$$\{\Delta_6(\Delta_4+\Delta_5)+\Delta_5\Delta_6(\Delta_4+\Delta_5+\Delta_6)\}\mathbf{d}_5$$
$$+(\Delta_5)^2(\Delta_4+\Delta_5+\Delta_6)\mathbf{d}_6]=\alpha_4\mathbf{d}_4+\beta_4\mathbf{d}_5+\gamma_4\mathbf{d}_6$$

주어진 것 　　 구해야 하는 것

$$\begin{vmatrix}\mathbf{p}_0\\ \mathbf{t}_0\\ \mathbf{p}_1\\ \mathbf{p}_2\\ \mathbf{p}_3\\ \mathbf{p}_4\\ \mathbf{t}_1\\ \mathbf{p}_5\end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ \frac{-3}{\Delta_2} & \frac{3}{\Delta_2} & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & \alpha_1 & \beta_1 & \gamma_1 & 0 & 0 & 0 & 0\\ 0 & 0 & \alpha_2 & \beta_2 & \gamma_2 & 0 & 0 & 0\\ 0 & 0 & 0 & \alpha_3 & \beta_3 & \gamma_3 & 0 & 0\\ 0 & 0 & 0 & 0 & \alpha_4 & \beta_4 & \gamma_4 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-3}{\Delta_6} & \frac{3}{\Delta_6}\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\end{vmatrix}\begin{vmatrix}\mathbf{d}_0\\ \mathbf{d}_1\\ \mathbf{d}_2\\ \mathbf{d}_3\\ \mathbf{d}_4\\ \mathbf{d}_5\\ \mathbf{d}_6\\ \mathbf{d}_7\end{vmatrix}$$

2.3.3.6 Tridiagonal matrix 해법을 이용한 B-spline 곡선 조정점($\mathbf{d}_i$) 결정(1)

$$\begin{bmatrix} \mathbf{p}_0 \\ \mathbf{t}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \mathbf{p}_4 \\ \mathbf{t}_1 \\ \mathbf{p}_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-3}{\Delta_2} & \frac{3}{\Delta_2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_1 & \beta_1 & \gamma_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_2 & \beta_2 & \gamma_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_3 & \beta_3 & \gamma_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_4 & \beta_4 & \gamma_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-3}{\Delta_6} & \frac{3}{\Delta_6} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \\ \mathbf{d}_4 \\ \mathbf{d}_5 \\ \mathbf{d}_6 \\ \mathbf{d}_7 \end{bmatrix}$$

$= \mathbf{D}$ 주어진 것

$= \mathbf{A}$ 계산할 수 있는 것

$= \mathbf{X}$ 구해야 하는 것

$$\mathbf{D} = \mathbf{A}\mathbf{X}$$

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{D}$$

그런데 행렬 A가 Tri-diagonal matrix이므로 간단하게 A^{-1}를 계산할 수 있음

2.3.3.6 Tridiagonal matrix 해법을 이용한 B-spline 곡선 조정점($\mathbf{d}_i$) 결정(2)

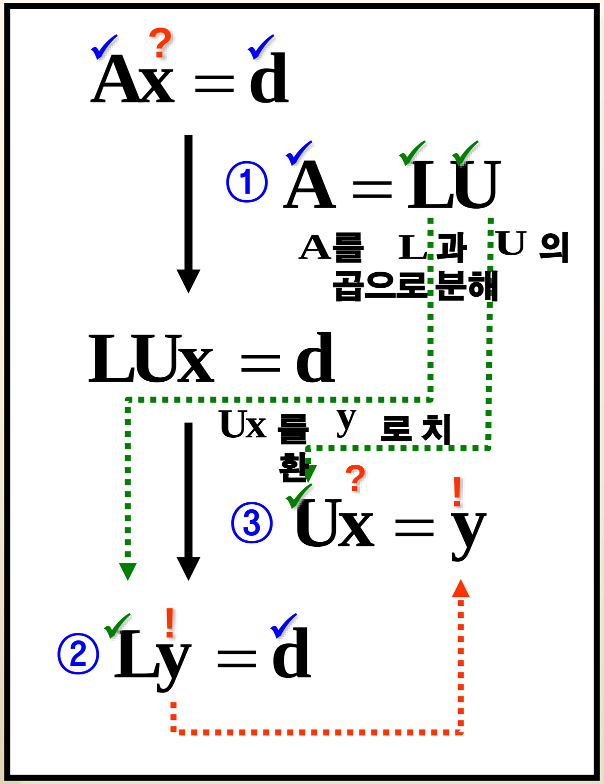
Tridiagonal matrix

대각 성분과 그 위/아래, 좌/우 성분만 0이 아닌 값이고, 나머지 성분은 0 값인 행렬
즉, 대각 성분을 중심으로 3개의 성분만 0이 아닌 값 ➡ Tri + Diagonal

$$\begin{bmatrix} b_0 & c_0 & 0 & & & & & \\ a_1 & b_1 & c_1 & 0 & & & & \\ 0 & a_2 & b_2 & c_2 & 0 & & & \\ & & & \ddots & & & & \\ & & & & \ddots & & & \\ & & & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}$$

$$\mathbf{A} \qquad \mathbf{x} = \mathbf{d}$$

$\mathbf{A}$와 $\mathbf{d}$를 알고 있을 때, $\mathbf{x}$ 구하기



① $\mathbf{A}$를 $\mathbf{L}$과 $\mathbf{U}$의 곱으로 분해

② $\mathbf{Ly} = \mathbf{d}$를 만족하는 $\mathbf{y}$ 구하기

③ $\mathbf{Ux} = \mathbf{y}$를 만족하는 $\mathbf{x}$를 구하면, 곧 $\mathbf{Ax} = \mathbf{d}$를 만족하는 $\mathbf{x}$를 구하는 것임

2.3.3.6 Tridiagonal matrix 해법을 이용한 B-spline 곡선 조정점($\mathbf{d}_i$) 결정(3)

① $\mathbf{A} = \mathbf{L}\mathbf{U}$

$$\begin{bmatrix} b_0 & c_0 & 0 & & & & \\ a_1 & b_1 & c_1 & 0 & & & \\ 0 & a_2 & b_2 & c_2 & 0 & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & 0 & a_n & b_n \end{bmatrix} = \begin{bmatrix} \beta_0 & 0 & & & & & \\ \alpha_1 & \beta_1 & 0 & & & & \\ 0 & \alpha_2 & \beta_2 & 0 & & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & 0 & \alpha_{n-1} & \beta_{n-1} & 0 \\ & & & & 0 & \alpha_n & \beta_n \end{bmatrix} \begin{bmatrix} 1 & \gamma_1 & 0 & & & & \\ 0 & 1 & \gamma_2 & 0 & & & \\ & 0 & 1 & \gamma_3 & 0 & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & 0 & 1 & \gamma_n \\ & & & & & 0 & 1 \end{bmatrix}$$

$$\mathbf{A} = \mathbf{L} \quad \mathbf{U}$$

$$\begin{array}{lll} & b_0 = \beta_0 & c_0 = \beta_0\gamma_1 \\ a_1 = \alpha_1 & b_1 = \alpha_1\gamma_1 + \beta_1 & c_1 = \beta_1\gamma_2 \\ a_2 = \alpha_2 & b_2 = \alpha_2\gamma_2 + \beta_2 & c_2 = \beta_2\gamma_3 \\ \vdots & \vdots & \vdots \\ a_{n-1} = \alpha_{n-1} & b_{n-1} = \alpha_{n-1}\gamma_{n-1} + \beta_{n-1} & c_{n-1} = \beta_{n-1}\gamma_n \\ a_n = \alpha_n & b_n = \alpha_n\gamma_n + \beta_n & \end{array}$$

$$\alpha_i = a_i \qquad i = 1, \cdots, n$$

$$\gamma_{i+1} = \frac{c_i}{\beta_i} \qquad i = 0, \cdots, n-1$$

$$\beta_{i+1} = b_{i+1} - \alpha_{i+1}\gamma_{i+1} \qquad i = 0, \cdots, n-1$$

with $\beta_0 = b_0$

$\mathbf{A}\mathbf{x} = \mathbf{d}$

① $\mathbf{A} = \mathbf{L}\mathbf{U}$

$\mathbf{L}\mathbf{U}\mathbf{x} = \mathbf{d}$

③ $\mathbf{U}\mathbf{x} = \mathbf{y}$

② $\mathbf{L}\mathbf{y} = \mathbf{d}$

2.3.3.6 Tridiagonal matrix 해법을 이용한 B-spline 곡선 조정점($\mathbf{d}_i$) 결정(4)

② $\mathbf{L}\mathbf{y} = \mathbf{d}$

$$\begin{bmatrix} \beta_0 & 0 & & & & & \\ \alpha_1 & \beta_1 & 0 & & & & \\ 0 & \alpha_2 & \beta_2 & 0 & & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & 0 & \alpha_{n-1} & \beta_{n-1} & 0 \\ & & & & 0 & \alpha_n & \beta_n \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}$$

$$\mathbf{L} \quad \mathbf{y} = \mathbf{d}$$

$$\beta_0 y_0 = d_0$$
$$\alpha_1 y_0 + \beta_1 y_1 = d_1$$
$$\alpha_2 y_1 + \beta_2 y_2 = d_2$$
$$\vdots$$
$$\alpha_{n-1} y_{n-2} + \beta_{n-1} y_{n-1} = d_{n-1}$$
$$\alpha_n y_{n-1} + \beta_n y_n = d_n$$

Forward substitution

$$y_i = \frac{d_i - \alpha_i y_{i-1}}{\beta_i} \qquad i = 1, \cdots, n$$

$$\text{with } y_0 = \frac{d_0}{\beta_0}$$

$\mathbf{A}\mathbf{x} = \mathbf{d}$

① $\mathbf{A} = \mathbf{L}\mathbf{U}$

$\mathbf{L}\mathbf{U}\mathbf{x} = \mathbf{d}$

③ $\mathbf{U}\mathbf{x} = \mathbf{y}$

② $\mathbf{L}\mathbf{y} = \mathbf{d}$

2.3.3.6 Tridiagonal matrix 해법을 이용한 B-spline 곡선 조정점($\mathbf{d}_i$) 결정(5)

③ $\mathbf{U}\mathbf{x} = \mathbf{y}$

$$\underbrace{\begin{bmatrix} 1 & \gamma_1 & 0 & & & & \\ 0 & 1 & \gamma_2 & 0 & & & \\ & 0 & 1 & \gamma_3 & 0 & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & 0 & 1 & \gamma_n \\ & & & & & 0 & 1 \end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix}}_{\mathbf{y}}$$

$$x_0 + \gamma_0 x_1 = y_0$$
$$x_1 + \gamma_1 x_2 = y_1$$
$$x_2 + \gamma_2 x_3 = y_2$$
$$\vdots$$
$$x_{n-1} + \gamma_{n-1} x_n = y_{n-1}$$
$$x_n = y_n$$

Backward substitution

$$x_i = y_i - \gamma_{i+1} x_{i+1}$$
$$i = n-1, \cdots, 0$$
$$\text{with } x_n = y_n$$

$\mathbf{A}\mathbf{x} = \mathbf{d}$

① $\mathbf{A} = \mathbf{L}\mathbf{U}$

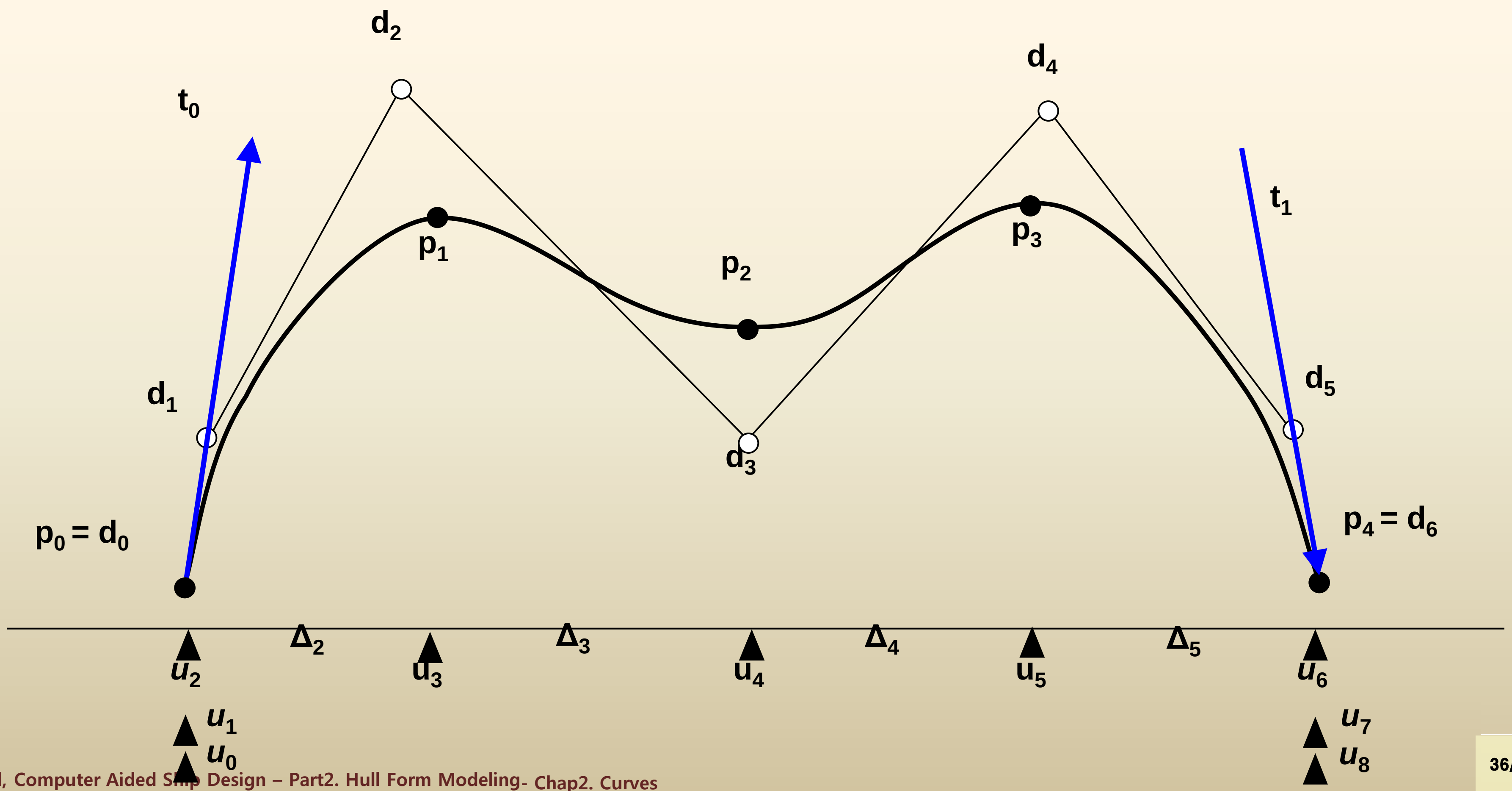
$\mathbf{L}\mathbf{U}\mathbf{x} = \mathbf{d}$

③ $\mathbf{U}\mathbf{x} = \mathbf{y}$

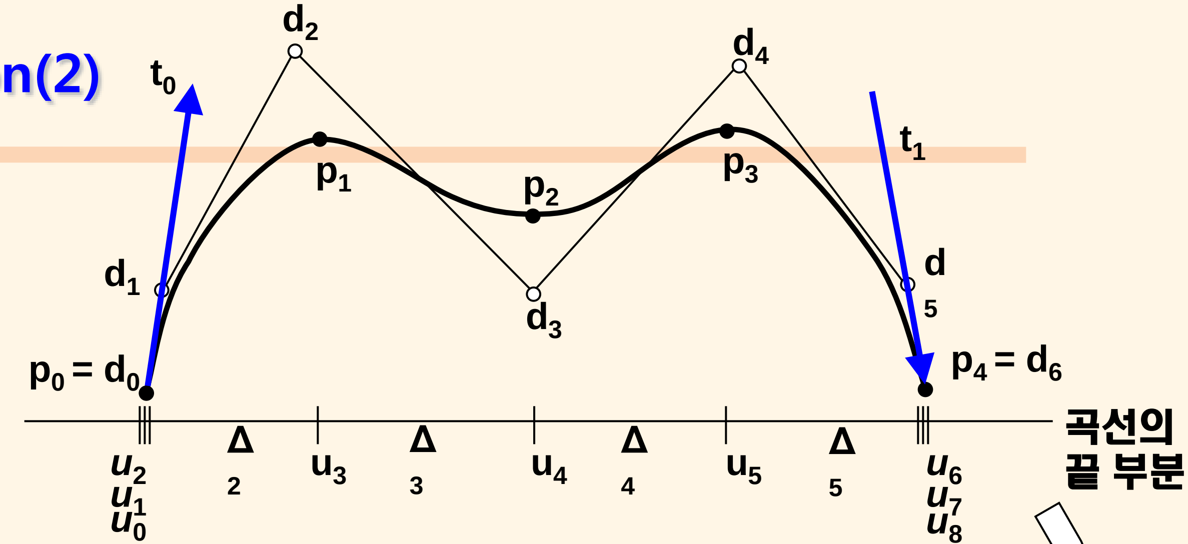
② $\mathbf{L}\mathbf{y} = \mathbf{d}$

2.3.3.7 Bessel End Condition (1)

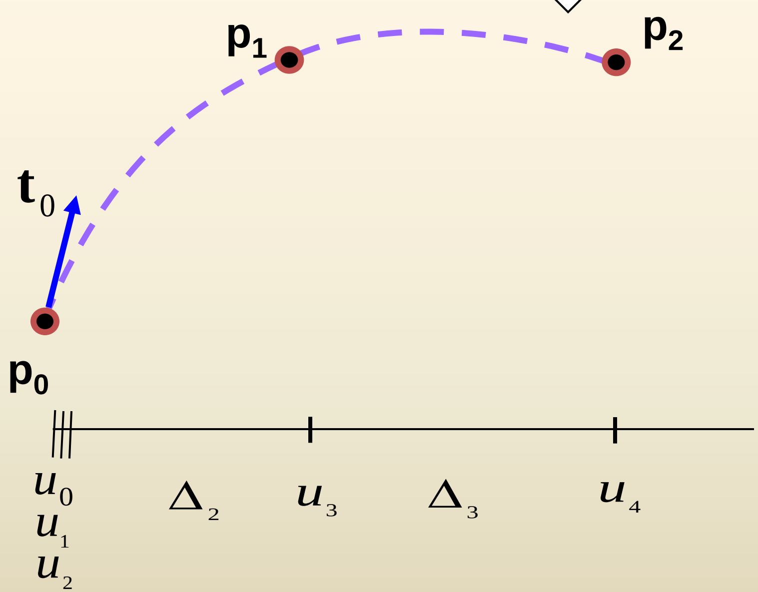
☑ B-spline curve interpolation에서 양끝점에서의 접선벡터 t_0, t_1이 주어지지 않았을 때,
(1) 곡선의 양끝의 연속된 세 점으로부터 2차 곡선(quadratic curve)을 생성하고,
(2) 생성된 2차 곡선의 양 끝점에서의 1차 미분값을 우리가 생성하고자 하는 B-spline curve의 양 끝점에서의 접선 벡터로 가정하는 방법

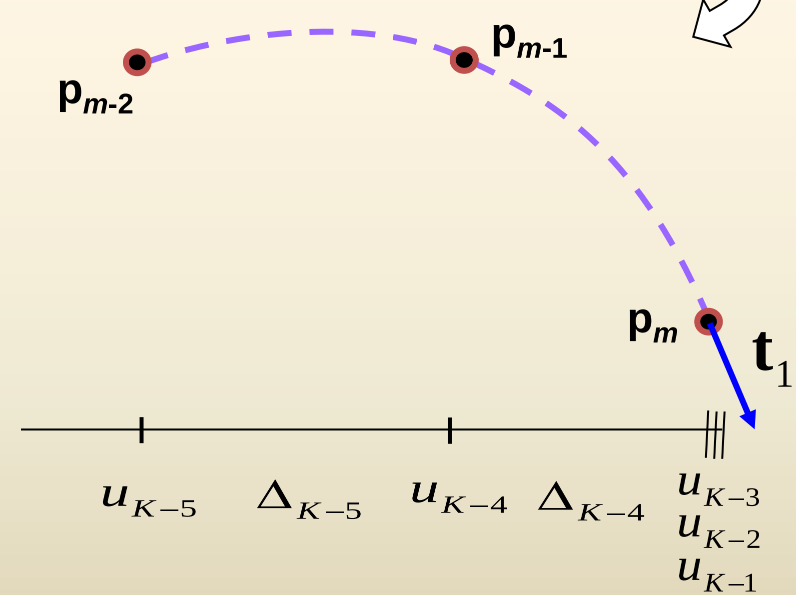


2.3.3.7 Bessel End Condition(2)



곡선의 시작 부분





$$\mathbf{t}_s = \left(-\frac{2\Delta_2+\Delta_3}{\Delta_2(\Delta_2+\Delta_3)}\mathbf{p}_0 + \frac{(\Delta_2+\Delta_3)}{\Delta_2\Delta_3}\mathbf{p}_1 - \frac{\Delta_2}{\Delta_3(\Delta_2+\Delta_3)}\mathbf{p}_2\right)$$

$$\mathbf{t}_e = \left(\frac{\Delta_{K-4}}{\Delta_{K-5}(\Delta_{K-5}+\Delta_{K-4})}\mathbf{p}_{m-2} - \frac{(\Delta_{K-5}+\Delta_{K-4})}{\Delta_{K-5}\Delta_{K-4}}\mathbf{p}_{m-1} + \frac{(2\Delta_{K-4}+\Delta_{K-5})}{(\Delta_{K-5}+\Delta_{K-4})\Delta_{K-4}}\mathbf{p}_m\right)$$

2.3.3.8 Sample code of Cubic B-spline Curve (1)

```
numberifndef __CubicBspline_h__
numberdefine __CubicBspline_h__

numberinclude "vector.h“

class CubicBsplineCurve {
public:
    Vector* m_ControlPoint;   int m_nControlPoint;
    double* m_Knot; int m_nKnot;  int m_nDegree;

    .........

    void SetControlPoint(Vector* pControlPoint, int nControlPoint);
    void SetKnot(double* pKnot, int nKnot);
    Vector CalcPoint(double u);
    double N(int d, int i, double u);
    void Interpolate(Vector *pFittingPoint, int nFittingPoint);
    void Parameterization(int nType, Vector* FittingPoint, int nPoint, double* t);
};
numberendif
```

2.3.3.8 Sample code of Cubic B-spline Curve (2)

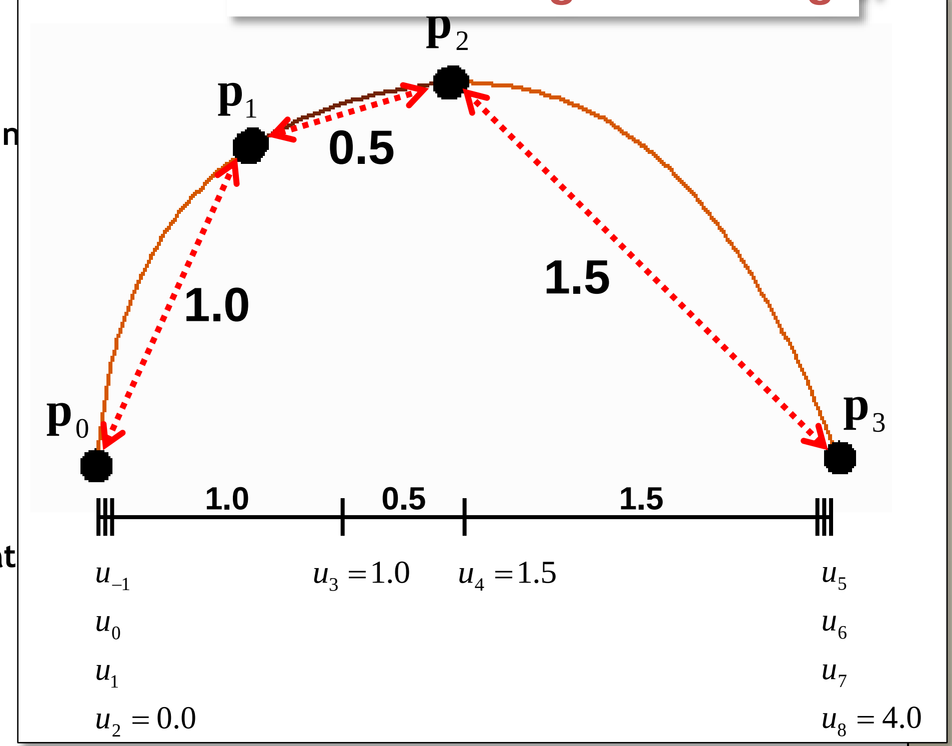
```
void CubicBsplineCurve::Interpolate(Vector *pFittingPoint, in
{
  // Generate Knot
    if(m_Knot) delete[] m_Knot;
    m_nKnot = (m_nFittingPoint - 2) + 2*(3+1);
    m_Knot = new double [m_nKnot];
    // Use Chord length or Centripetal method
    ......

//-----------------------------------------
// Generate Matrix : (L+1) * (L+1)
   int L = m_nFittingPoint + 1;          // (L+1)*(L+1) size Mat

  // Fill rhs
    Vector* rhs = new Vector[L+1];
    for(i = 1; i <= L-1 ; i++) rhs[i] = pFittingPoint[i-1];

  // Bessel End condition
    rhs[0] = rhs[1]; rhs[L] = rhs[L-1];
    rhs[1] = StartTangentByBesselEndCondition;   rhs[L-1] = EndTangentByBesselEndCondition;
```

Set knot using chord length



2.3.3.8 Sample code of Cubic B-spline Curve (2)

```
void CubicBsplineCurve::Interpolate(Vector *pFittingPoint, int nFittingPoint)
{
  // Generate Knot
     if(m_Knot) delete[] m_Knot;
     m_nKnot = (m_nFittingPoint - 2) + 2*(3+1);
     m_Knot = new double [m_nKnot];
     // Use Chord length or Centripetal method
     ......

//-----------------------------------------
// Generate Matrix : (L+1) * (L+1)
   int L = m_nFittingPoint + 1;           // (L+1)*(L+1) size Matrix

  // Fill rhs
     Vector* rhs = new Vector[L+1];
     for(i = 1; i <= L-1 ; i++) rhs[i] = pFittingPoint[i-1];

  // Bessel End condition
     rhs[0] = rhs[1]; rhs[L] = rhs[L-1];
     rhs[1] = StartTangentByBesselEndCondition;   rhs[L-1] = EndTangentByBe
```

Bessel End Condition

$$\mathbf{t}_s = -\frac{2\Delta_2 + \Delta_3}{\Delta_2(\Delta_2 + \Delta_3)}\mathbf{p}_0 + \frac{(\Delta_2 + \Delta_3)}{\Delta_2\Delta_3}\mathbf{p}_1 - \frac{\Delta_2}{\Delta_3(\Delta_2 + \Delta_3)}\mathbf{p}_2$$

$$\mathbf{t}_e = \frac{\Delta_{K-4}}{\Delta_{K-5}(\Delta_{K-5} + \Delta_{K-4})}\mathbf{p}_{m-2} - \frac{(\Delta_{K-5} + \Delta_{K-4})}{\Delta_{K-5}\Delta_{K-4}}\mathbf{p}_{m-1} + \frac{(2\Delta_{K-4} + \Delta_{K-5})}{(\Delta_{K-5} + \Delta_{K-4})\Delta_{K-4}}\mathbf{p}_m$$

2.3.3.8 Sample code of Cubic B-spline Curve (3)

```
    double* alpha = new double[L+1];
    double* beta = new double[L+1];
    double* gamma = new double[L+1];
    double* up = new double[L+1];
    double* low = new double[L+1];
    if(m_ControlPoint) delete[] m_ControlPoint;
    m_nControlPoint = L+1;
    m_ControlPoint = new Vector[m_nControlPoint];
  // Fill alpha, beta, gamma
    ......
  // Solve LU system
    l_u_system(alpha, beta, gamma, L, up, low);
    solve_system(up, low, gamma, L, rhs, m_ControlPoint);

//------------------------------
// Release memory
  delete[] rhs;   delete[] alpha;   delete[] beta;   delete[] gamma;   delete[] up;   delete[] low;
}
```

LU 분해법을 이용하여 역행렬을 계산

2.3.4 C^1 and C^2 continuity condition

2.3.4.1 1st Derivatives of Cubic Bezier Curves at Junction point

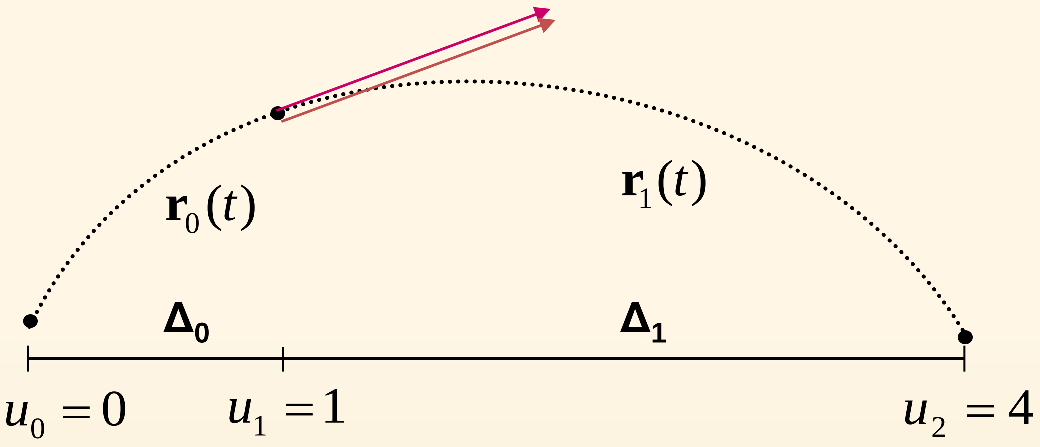
2.3.4.2 C^1 continuity condition of composite curves

2.3.4.3 2nd Derivatives of Cubic Bezier Curves

2.3.4.4 C^2 continuity condition of composite curves

Seoul National Univ.
@SDAL Advanced Ship Design Automation Lab. http://asdal.snu.ac.kr

2.3.4.1 1st Derivatives of Cubic Bezier Curves at Junction point



$$t = \frac{u - u_i}{u_{i+1} - u_i} = \frac{u - u_i}{\Delta_i}$$ t는 [0,1] 구간의 국부매개변수('local parameter')

$$\frac{d\mathbf{r}(u(t))}{du} = \frac{d\mathbf{r}_i(t)}{dt}\frac{dt}{du} = \frac{1}{\Delta_i}\frac{d\mathbf{r}_i(t)}{dt}$$

$\frac{d\mathbf{r}(u)}{du}$ 의 $u_0 \le u \le u_1$ 에서의 미분 값

$$t = \frac{u - u_0}{u_1 - u_0} = \frac{u - u_0}{\Delta_0}$$ t는 [0,1] 구간의 국부 매개 변수

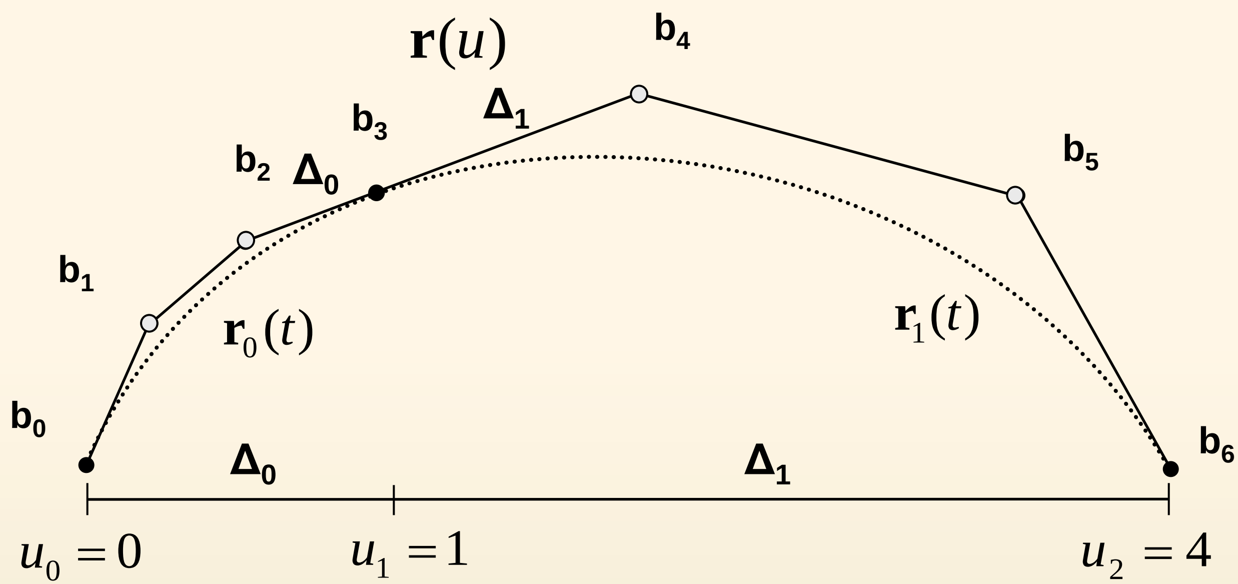
$$\frac{d\mathbf{r}(u)}{du} = \frac{d\,\mathbf{r}_0(u(t))}{dt}\frac{dt}{du} = \frac{1}{\Delta_0}\frac{d\,\mathbf{r}_0(t)}{dt}$$

$\frac{d\mathbf{r}(u)}{du}$ 의 $u_1 \le u \le u_2$ 에서의 미분 값

$$t = \frac{u - u_1}{u_2 - u_1} = \frac{u - u_1}{\Delta_1}$$ t는 [0,1] 구간의 국부 매개 변수

$$\frac{d\mathbf{r}(u)}{du} = \frac{d\,\mathbf{r}_1(t)}{dt}\frac{dt}{du} = \frac{1}{\Delta_1}\frac{d\,\mathbf{r}_1(t)}{dt}$$

2.3.4.2 C^1 continuity condition of composite curves



$\mathbf{r}(u = u_1) = \mathbf{r}_0(t = 1) = \mathbf{r}_1(t = 0)$ 연결 점에서 C^1 조건을 만족 해야 하므로

$$\left.\frac{d\mathbf{r}(u)}{du}\right|_{u_1=1} = \frac{1}{\Delta_0}\left.\frac{d\mathbf{r}_0(t)}{dt}\right|_{t=1} = \frac{1}{\Delta_0}3(\mathbf{b}_3-\mathbf{b}_2)$$

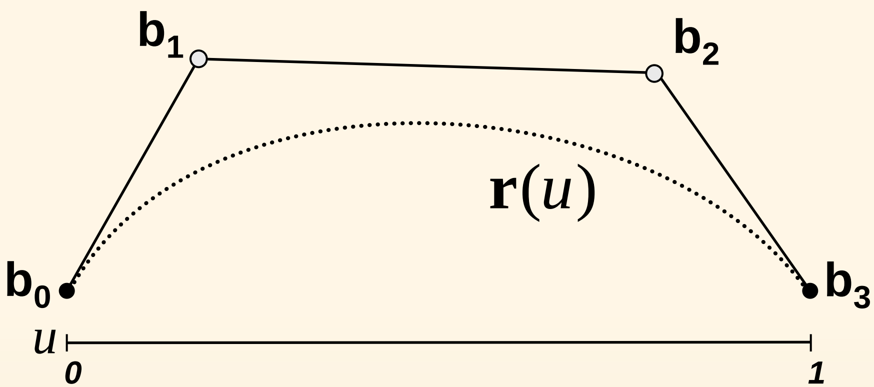
$$= \frac{1}{\Delta_1}\left.\frac{d\mathbf{r}_1(t)}{dt}\right|_{t=0} = \frac{1}{\Delta_1}\cdot 3(\mathbf{b}_4-\mathbf{b}_3)$$

$$(\mathbf{b}_3-\mathbf{b}_2):(\mathbf{b}_4-\mathbf{b}_3) = \Delta_0 : \Delta_1$$

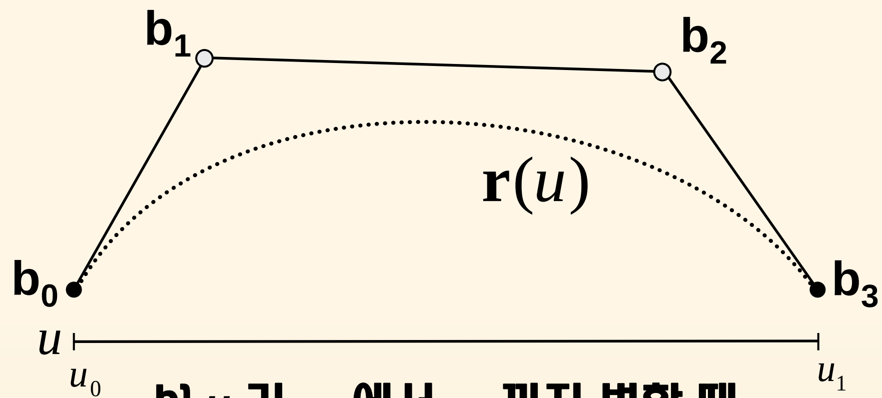
$$\mathbf{b}_3 = \frac{\Delta_1}{\Delta}\mathbf{b}_2 + \frac{\Delta_0}{\Delta}\mathbf{b}_4$$

parameter u를 시간이라고 생각하면, 1차 미분 계수는 곡선상을 지나는 점의 속도라고 생각할 수 있다.
연결점 b_3에서 1차 미분 계수가 연속이라면 그 점에서 속도가 연속이어야 한다는 의미이다.
그러므로 시간 간격이 Δ_0 에서 Δ_1 으로 변하면 즉, 시간 간격이 변하면,
그 거리도 비례하여 변하여야 연결점에서 속도가 연속이다.!!!

2.3.4.3 2nd Derivatives of Cubic Bezier Curves



a) u 가 0 에서 1까지 변할 때



b) u 가 u_0 에서 u_1 까지 변할 때

n차 Bezier곡선 2차 미분

$$\frac{d^2\mathbf{r}(u)}{du^2} = n(n-1)\sum_{i=0}^{n-2}(\mathbf{b}_{i+2} - 2\mathbf{b}_{i+1} + \mathbf{b}_i)B_i^{n-2}$$

3차 Bezier곡선 2차 미분

$$\frac{d^2\mathbf{r}(u)}{du^2} = 3(3-1)\sum_{i=0}^{1}(\mathbf{b}_{i+2} - 2\mathbf{b}_{i+1} + \mathbf{b}_i)B_i^1(u)$$

$u = 1$일 때

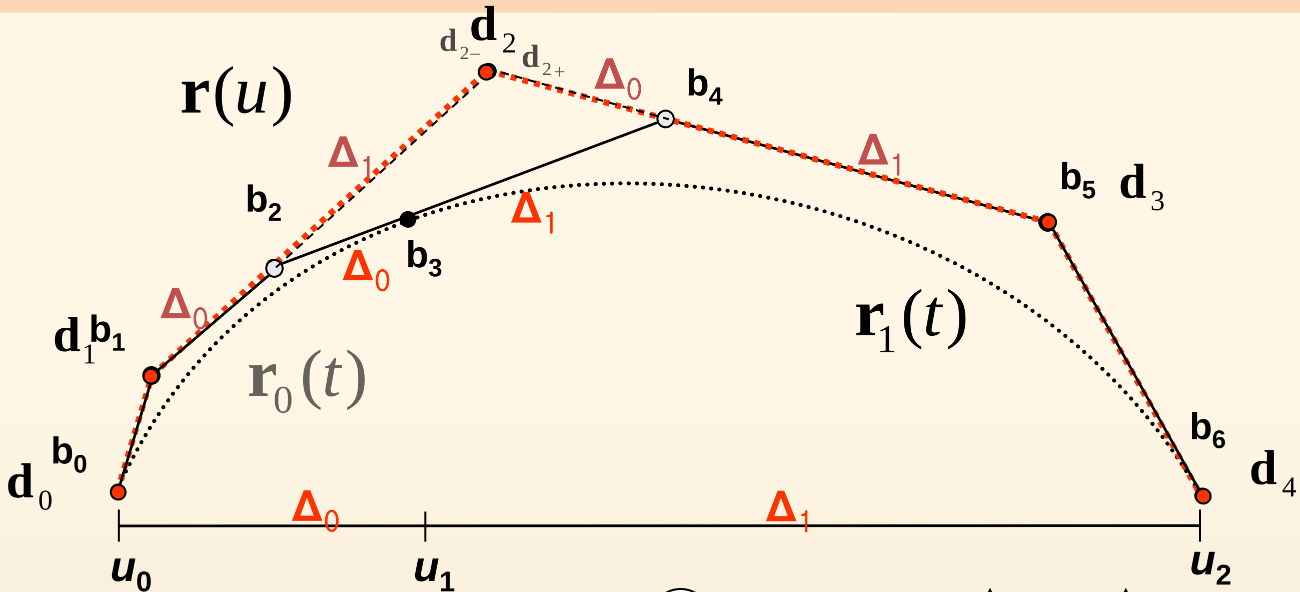
$$\frac{d^2\mathbf{r}(1)}{du^2} = 3(3-1)(\mathbf{b}_3 - 2\mathbf{b}_2 + \mathbf{b}_1)$$

$$\frac{d^2\mathbf{r}(u(t))}{du^2} = \frac{1}{(\Delta)^2}\frac{d^2\mathbf{r}(t)}{dt^2} \qquad (\Delta = u_1 - u_0)$$

$u = u_1$ 일 때

$$\frac{d^2\mathbf{r}(u_1)}{du^2} = \frac{1}{(\Delta)^2}\frac{d^2\mathbf{r}(1)}{dt^2} = \frac{1}{(\Delta)^2}3(3-1)(\mathbf{b}_3 - 2\mathbf{b}_2 + \mathbf{b}_1)$$

2.3.4.4. C² continuity condition of composite curves



1 연결점 $\mathbf{b}_3$에서 C^2조건

$$\frac{d^2\mathbf{r}(u_{1-})}{du^2} = \frac{1}{(\Delta_0)^2}\frac{d^2\mathbf{r}_0(1)}{dt^2} = \frac{1}{(\Delta_0)^2}3(3-1)(\mathbf{b}_3 - 2\mathbf{b}_2 + \mathbf{b}_1)$$

$$\frac{d^2\mathbf{r}(u_{1+})}{du^2} = \frac{1}{(\Delta_1)^2}\frac{d^2\mathbf{r}_1(0)}{dt^2} = \frac{1}{(\Delta_1)^2}3(3-1)(\mathbf{b}_5 - 2\mathbf{b}_4 + \mathbf{b}_3)$$

2 $\frac{d^2\mathbf{r}(u_{1-})}{du^2} = \frac{d^2\mathbf{r}(u_{1+})}{du^2}$ 이어야 하므로

$\frac{6}{(\Delta_0)^2}(\mathbf{b}_3 - 2\mathbf{b}_2 + \mathbf{b}_1) = \frac{6}{(\Delta_0)^2}(\mathbf{b}_5 - 2\mathbf{b}_4 + \mathbf{b}_3)$이다.

3 그리고 C^1조건($\mathbf{b}_3 = \frac{\Delta_1}{\Delta}\mathbf{b}_2 + \frac{\Delta_0}{\Delta}\mathbf{b}_4$)을 대입하여

정리하면

$$\Rightarrow -\frac{\Delta_1}{\Delta_0}\mathbf{b}_1 + \frac{\Delta}{\Delta_0}\mathbf{b}_2 = \frac{\Delta}{\Delta_1}\mathbf{b}_4 - \frac{\Delta_0}{\Delta_1}\mathbf{b}_5$$

4 좌변을 $\mathbf{d}_{2-} = -\frac{\Delta_1}{\Delta_0}\mathbf{b}_1 + \frac{\Delta}{\Delta_0}\mathbf{b}_2$ 라 하면

$$\mathbf{b}_2 = \frac{\Delta_1}{\Delta}\mathbf{b}_1 + \frac{\Delta_0}{\Delta}\mathbf{d}_{2-}$$

5 우변을 $\mathbf{d}_{2+} = \frac{\Delta}{\Delta_1}\mathbf{b}_4 - \frac{\Delta_0}{\Delta_1}\mathbf{b}_5$ 라 하면

$$\mathbf{b}_4 = \frac{\Delta_1}{\Delta}\mathbf{d}_{2+} + \frac{\Delta_1}{\Delta}\mathbf{b}_5$$

6 **즉, $(\mathbf{d}_{2-} = \mathbf{d}_{2+} = \mathbf{d}_2)$ 인 점이 존재하면 C² 조건을 만족한다**

7 연결점에서 C^2조건

$$-\frac{\Delta_1}{\Delta_0}\mathbf{b}_1 + \frac{\Delta}{\Delta_0}\mathbf{b}_2 = \frac{\Delta}{\Delta_1}\mathbf{b}_4 - \frac{\Delta_0}{\Delta_1}\mathbf{b}_5$$

$$ratio(\mathbf{b}_1, \mathbf{b}_2, d_2) = ratio(d_2, \mathbf{b}_4, \mathbf{b}_5) = \frac{\Delta_0}{\Delta_1}$$

2.3.5 B-spline basis function

(Cox-de Boor recurrence formula)

2.3.5.1 Cox-de Boor recurrence formula

2.3.5.2 B-spline curves

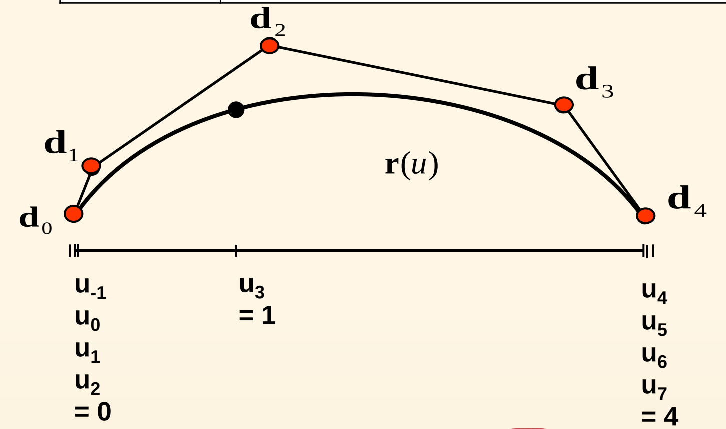
2.3.5.3 Relationship between de Boor algorithm & B-spline curves

2.3.5.4 Sample code of cubic B-spline curves

Seoul National Univ.
@SDAL Advanced Ship Design Automation Lab.
http://asdal.snu.ac.kr

2.3.5.1 Cox-de Boor Recurrence Formula (B-spline function) (1)

Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$



☑ 예: Cubic B-spline 곡선

$$\mathbf{r}(u)=\begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix}=\sum_{i=0}^{D-1}\mathbf{d}_i N_i^n(u)$$

$$=\mathbf{d}_0 N_0^3(u)+\mathbf{d}_1 N_1^3(u)+\mathbf{d}_2 N_2^3(u)+\mathbf{d}_3 N_3^3(u)+\mathbf{d}_4 N_4^3(u)$$

- Cox-de Boor Recurrence Formula (B-spline function)

$$N_i^n(u)=\frac{u-u_{i-1}}{u_{i+n-1}-u_{i-1}}N_i^{n-1}(u)+\frac{u_{i+n}-u}{u_{i+n}-u_i}N_{i+1}^{n-1}(u)$$

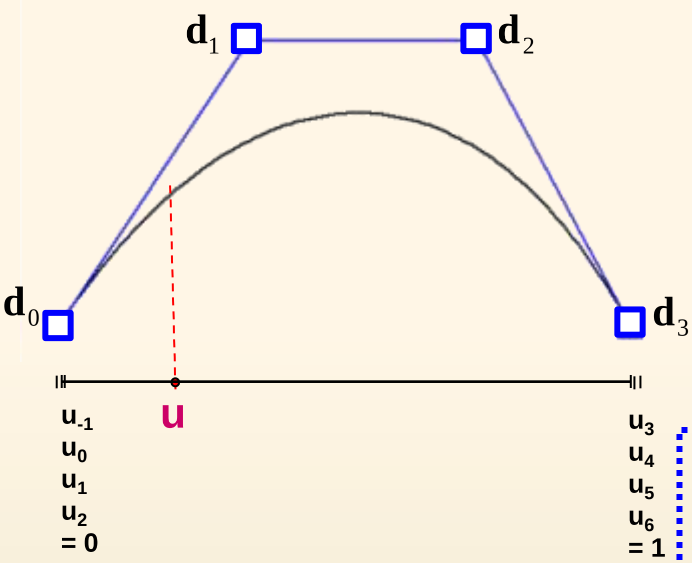
$$N_i^0(u)=\begin{cases}1 & \text{if } u_{i-1}\le u<u_i \\ 0 & \text{else}\end{cases}$$

2.3.5.1 Cox-de Boor Recurrence Formula (B-spline function) (2)

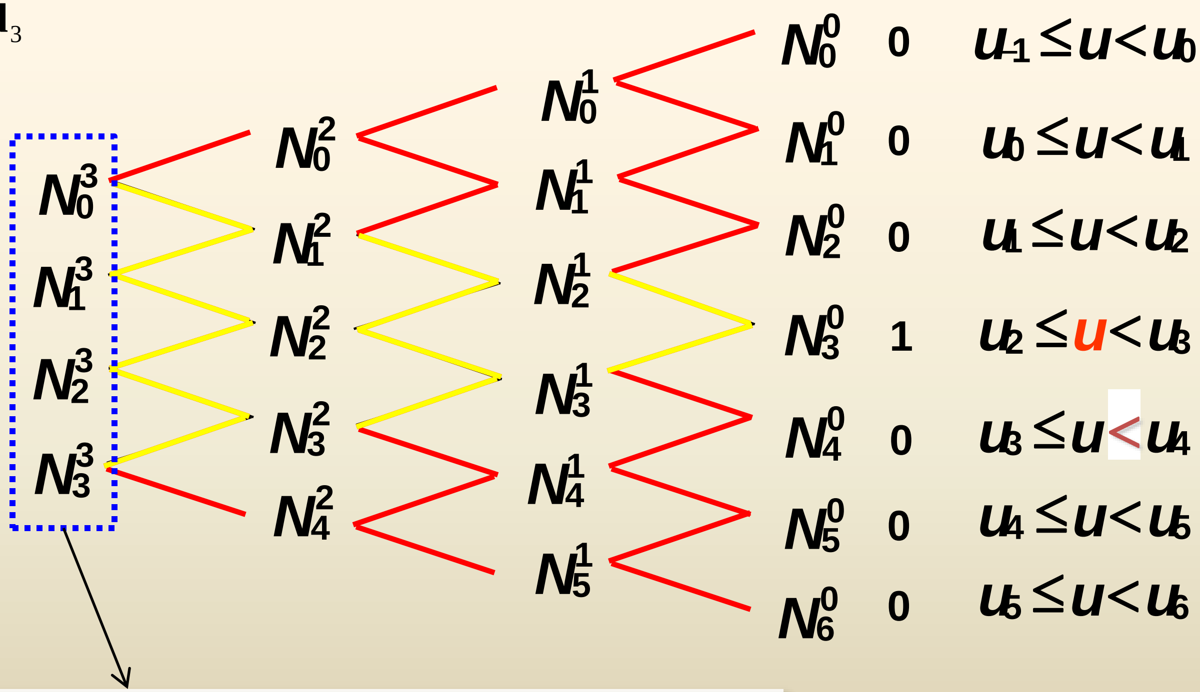
Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$



$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u)$$



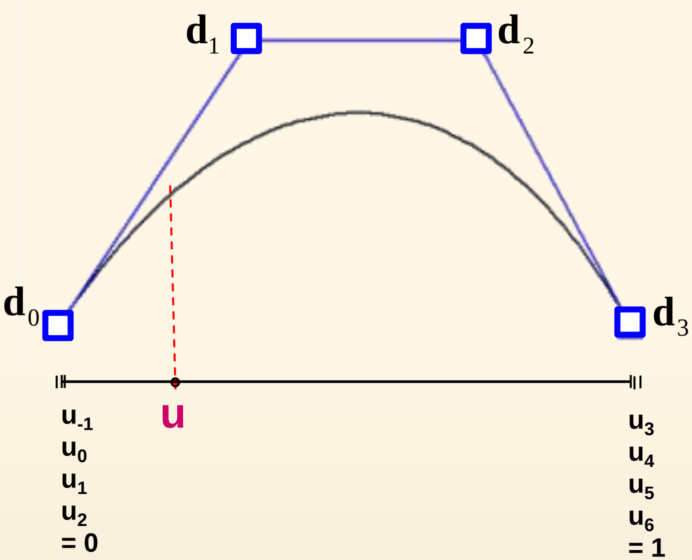
Parameter *u*에 대한 B-spline 곡선 상의 점을 구하기 위해서 위의 B-spline basis function을 계산해야 한다.

2.3.5.1 Cox-de Boor Recurrence Formula (B-spline function) (3)

Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$



$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u)$$

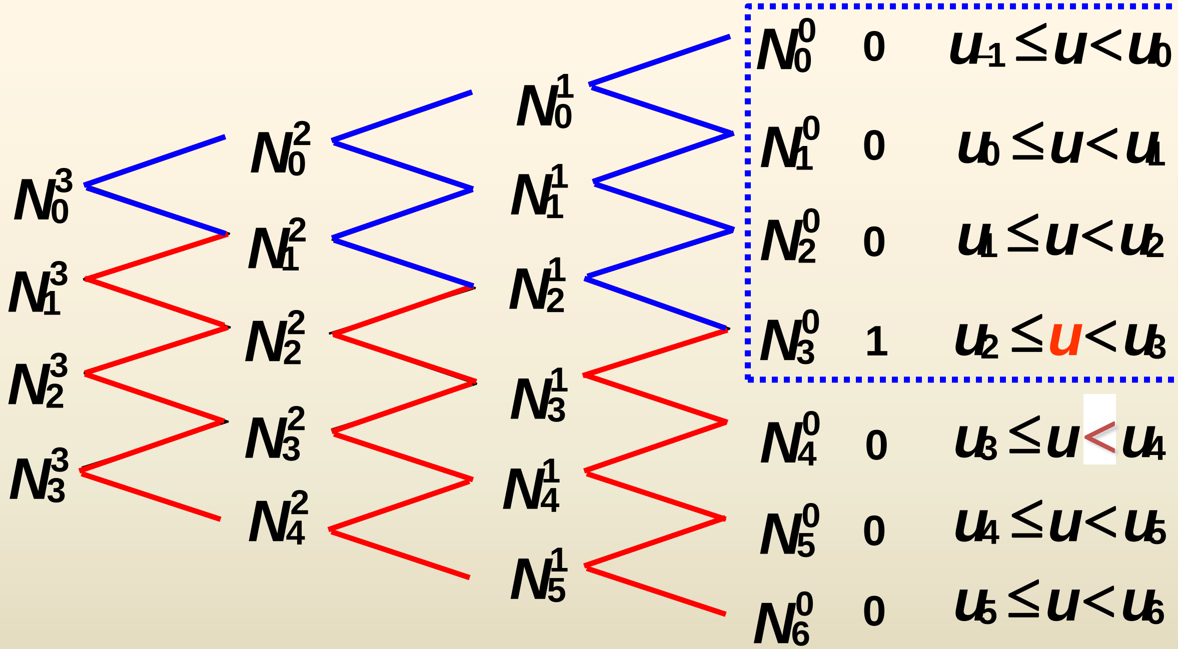
From $u_2 \le u < u_3$,

we can get $N_0^0(u) = 0$

$N_1^0(u) = 0$

$N_2^0(u) = 0$

$N_3^0(u) = 1$

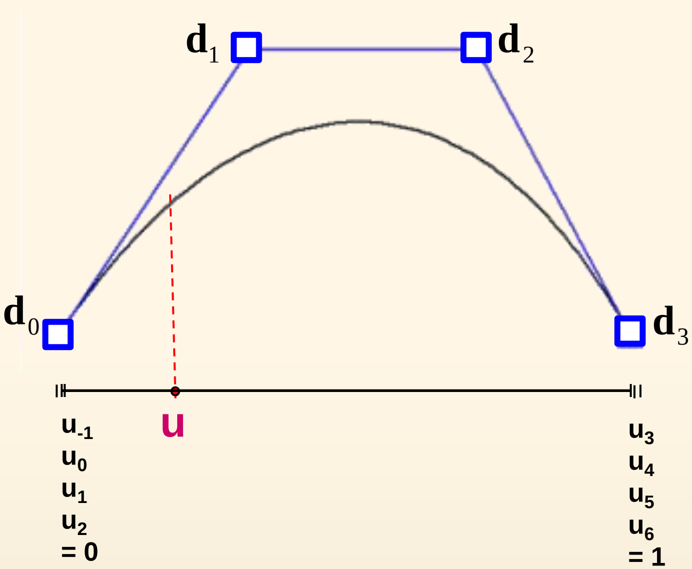


2.3.5.1 Cox-de Boor Recurrence Formula (B-spline function) (4)

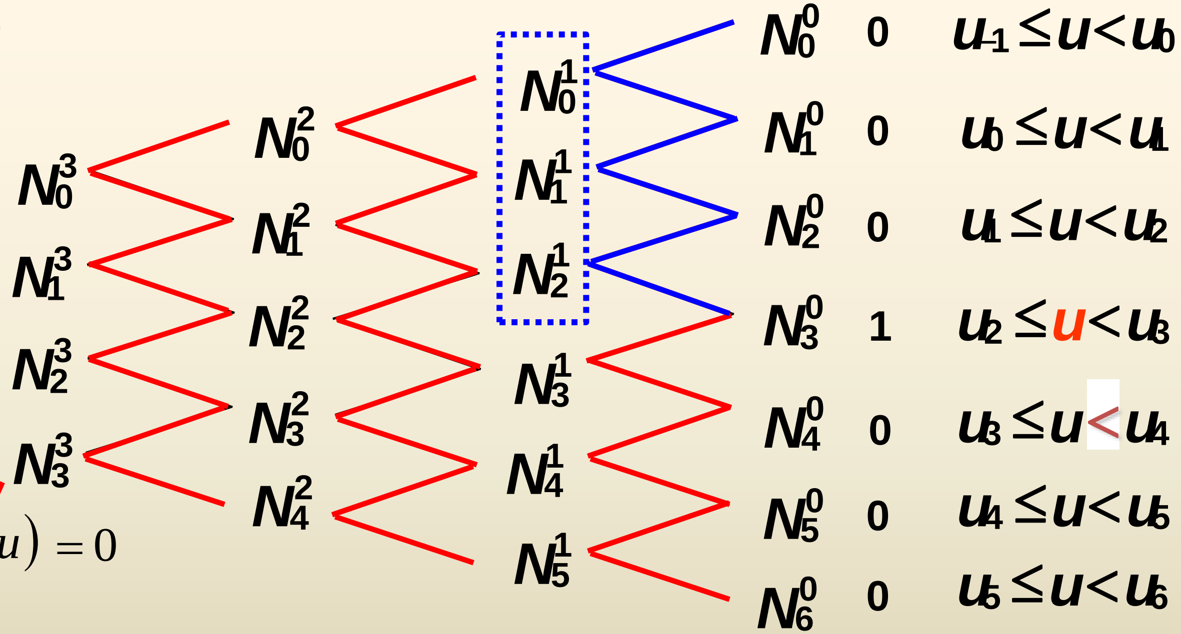
Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$



$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u)$$



$$N_0^0(u) = 0, \quad N_1^0(u) = 0$$

$$N_2^0(u) = 0, \quad N_3^0(u) = 1$$

$$N_0^1(u) = \frac{u - u_{-1}}{u_0 - u_{-1}} N_0^0(u) + \frac{u_1 - u}{u_1 - u_0} N_1^0(u) = 0$$

$$N_1^1(u) = \frac{u - u_0}{u_1 - u_0} N_1^0(u) + \frac{u_2 - u}{u_2 - u_1} N_2^0(u) = 0$$

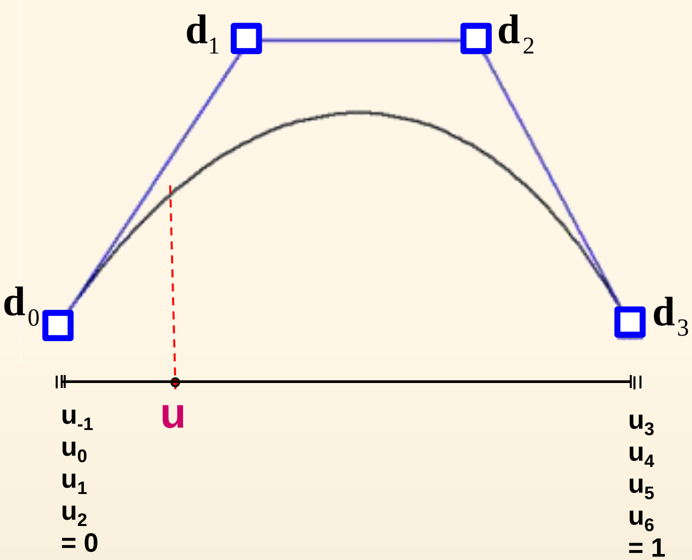
$$N_2^1(u) = \frac{u - u_1}{u_2 - u_1} N_2^0(u) + \frac{u_3 - u}{u_3 - u_2} N_3^0(u) = \frac{u_3 - u}{u_3 - u_2} = 1 - u$$

2.3.5.1 Cox-de Boor Recurrence Formula (B-spline function) (5)

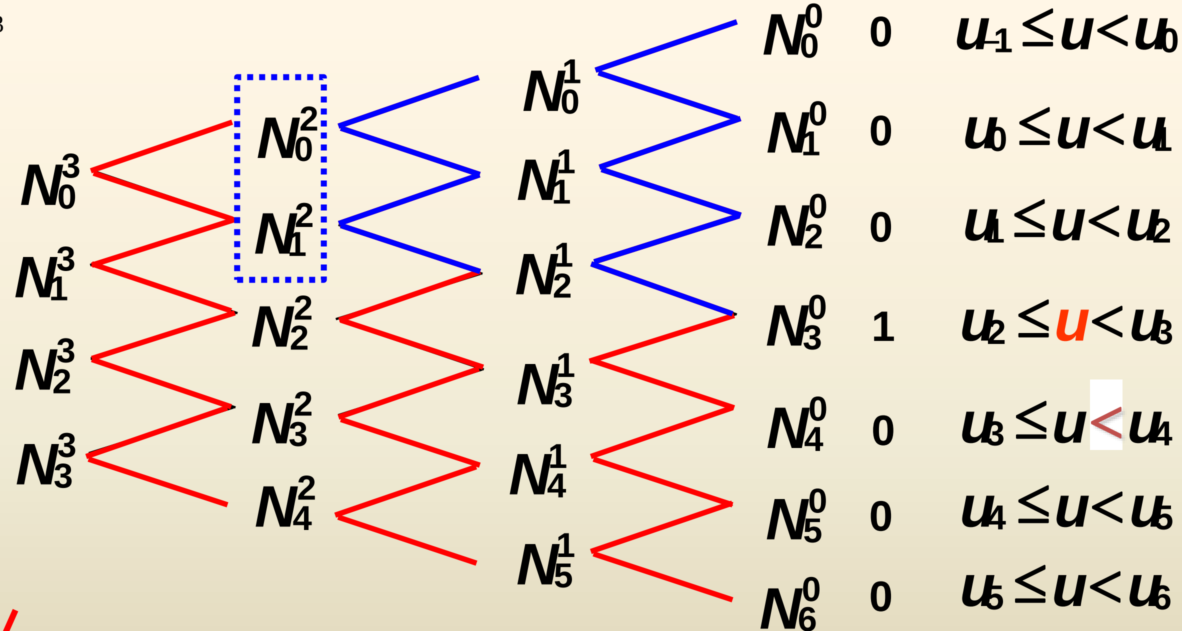
Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$



$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u)$$



$N_0^0(u) = 0, \quad N_1^0(u) = 0$

$N_2^0(u) = 0, \quad N_3^0(u) = 1$

$N_0^1(u) = 0, \quad N_1^1(u) = 0, \quad N_2^1(u) = 1 - u$

$$N_0^2(u) = \frac{u - u_{-1}}{u_1 - u_{-1}} N_0^1(u) + \frac{u_2 - u}{u_2 - u_0} N_1^1(u) = 0$$

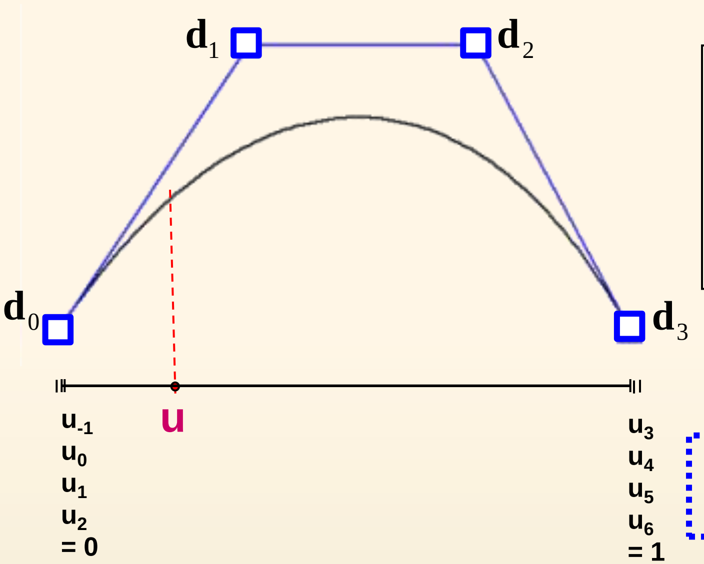
$$N_1^2(u) = \frac{u - u_0}{u_2 - u_0} N_1^1(u) + \frac{u_3 - u}{u_3 - u_1} N_2^1(u) = \frac{u_3 - u}{u_3 - u_1}(1 - u) = (1 - u)^2$$

2.3.5.1 Cox-de Boor Recurrence Formula (B-spline function) (6)

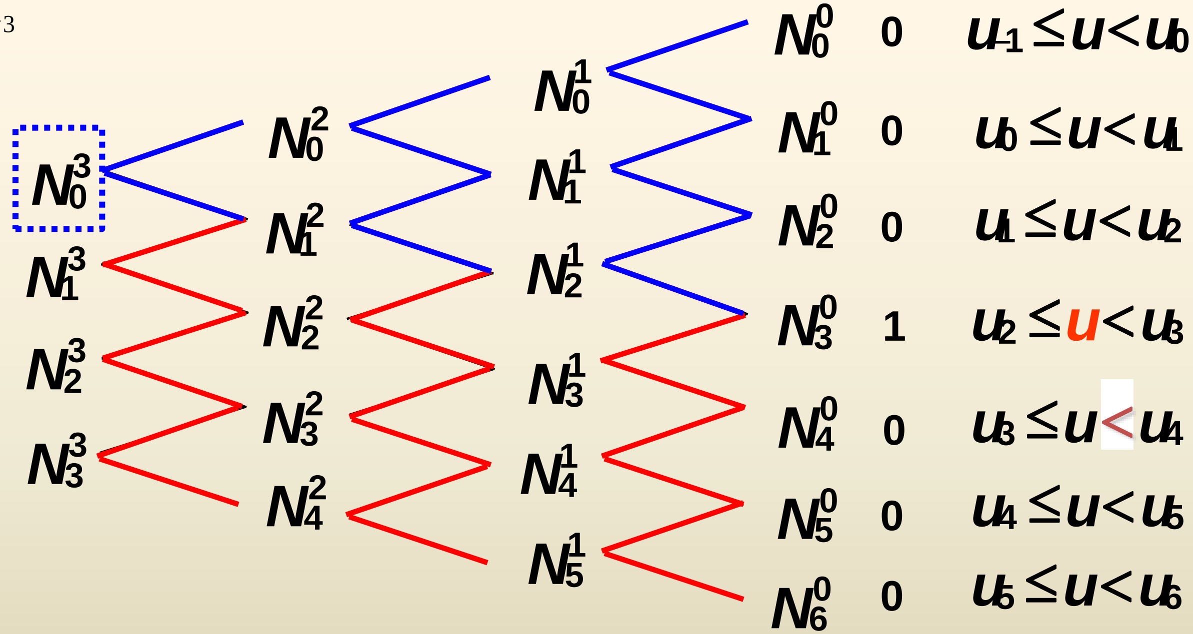
Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$



$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u)$$



$N_0^0(u) = 0,\ N_1^0(u) = 0$

$N_2^0(u) = 0,\ N_3^0(u) = 1$

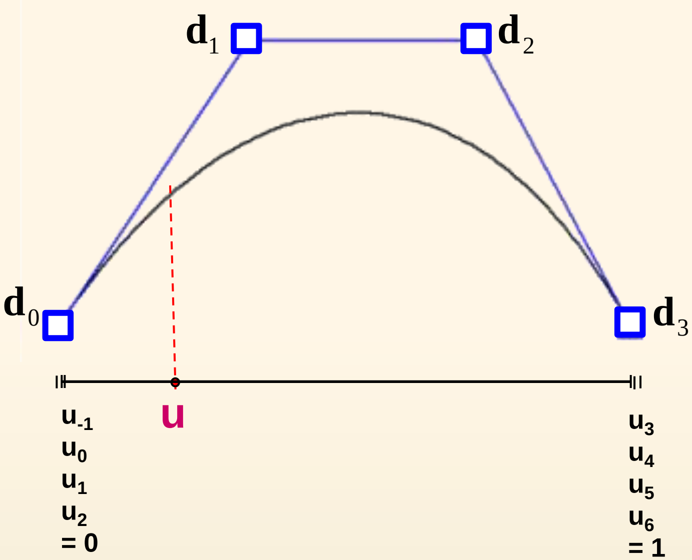
$N_0^1(u) = 0,\ N_1^1(u) = 0,\ N_2^1(u) = 1 - u$

$N_0^2(u) = 0,\ N_1^2(u) = (1-u)^2$

$$N_0^3(u) = \frac{u - u_{-1}}{u_2 - u_{-1}} N_0^2(u) + \frac{u_3 - u}{u_3 - u_0} N_1^2(u) = \frac{u_3 - u}{u_3 - u_0}(1-u)^2 = (1-u)^3$$

2.3.5.1 Cox-de Boor Recurrence Formula (B-spline function) (7)

Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$



$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$

$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u)$$

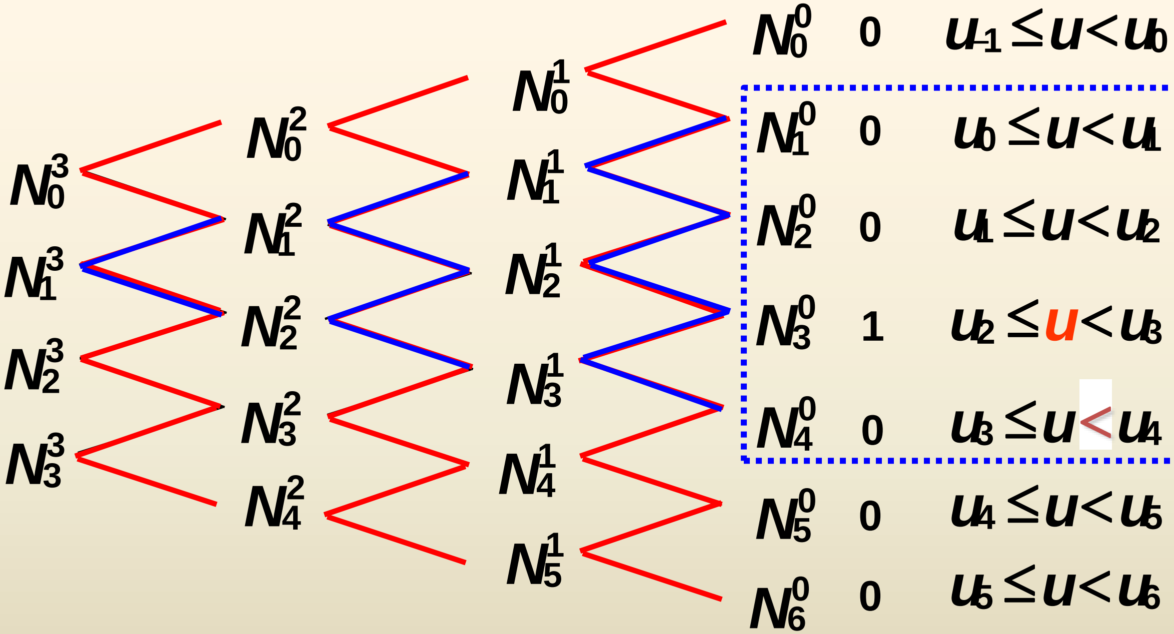
From $u_2 \le u < u_3$,

we can get $N_1^0(u) = 0$

$N_2^0(u) = 0$

$N_3^0(u) = 1$

$N_4^0(u) = 0$

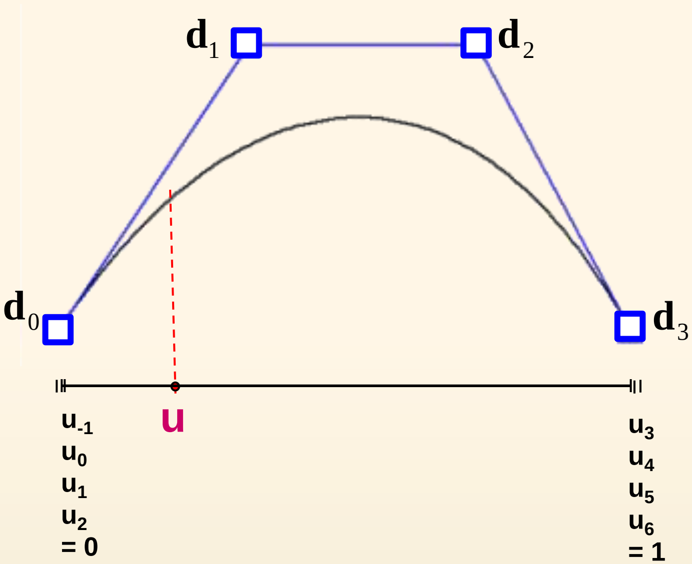


2.3.5.1 Cox-de Boor Recurrence Formula (B-spline function) (8)

Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

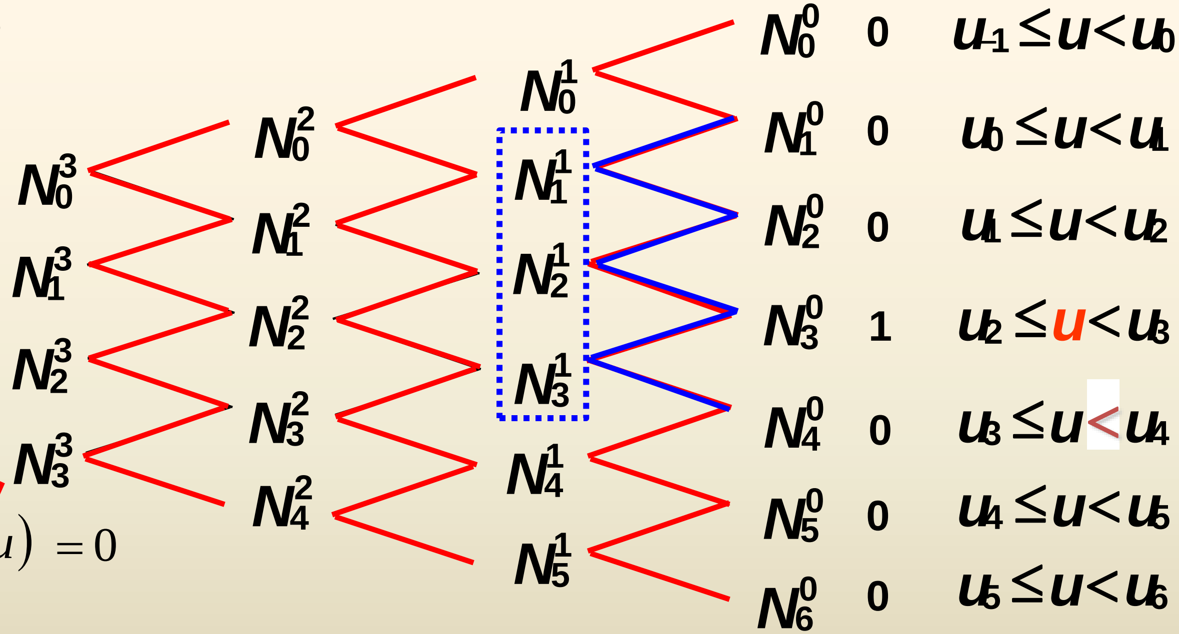
$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$



$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u)$$

$$N_1^0(u) = 0, \quad N_2^0(u) = 0$$

$$N_3^0(u) = 1, \quad N_4^0(u) = 0$$



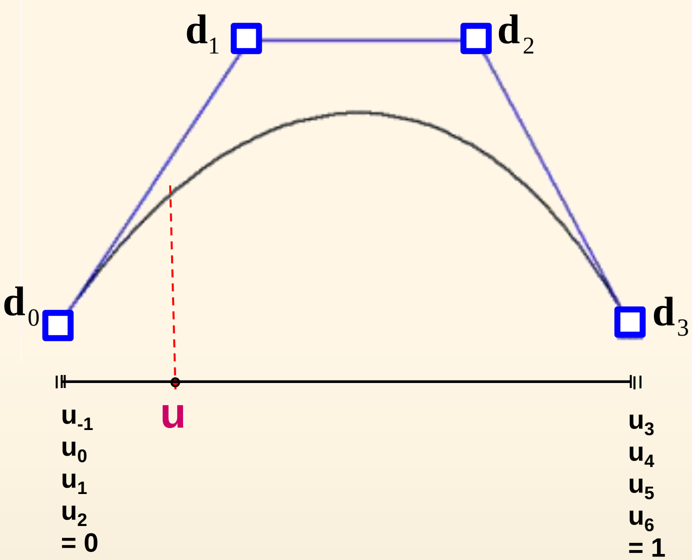
$$N_1^1(u) = \frac{u - u_0}{u_1 - u_0} N_1^0(u) + \frac{u_2 - u}{u_2 - u_0} N_2^0(u) = 0$$

$$N_2^1(u) = \frac{u - u_1}{u_2 - u_1} N_2^0(u) + \frac{u_3 - u}{u_3 - u_2} N_3^0(u) = \frac{u_3 - u}{u_3 - u_2} = 1 - u$$

$$N_3^1(u) = \frac{u - u_1}{u_3 - u_2} N_3^0(u) + \frac{u_4 - u}{u_4 - u_3} N_4^0(u) = \frac{u - u_1}{u_3 - u_2} = u$$

2.3.5.1 Cox-de Boor Recurrence Formula (B-spline function) (9)

Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$



$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

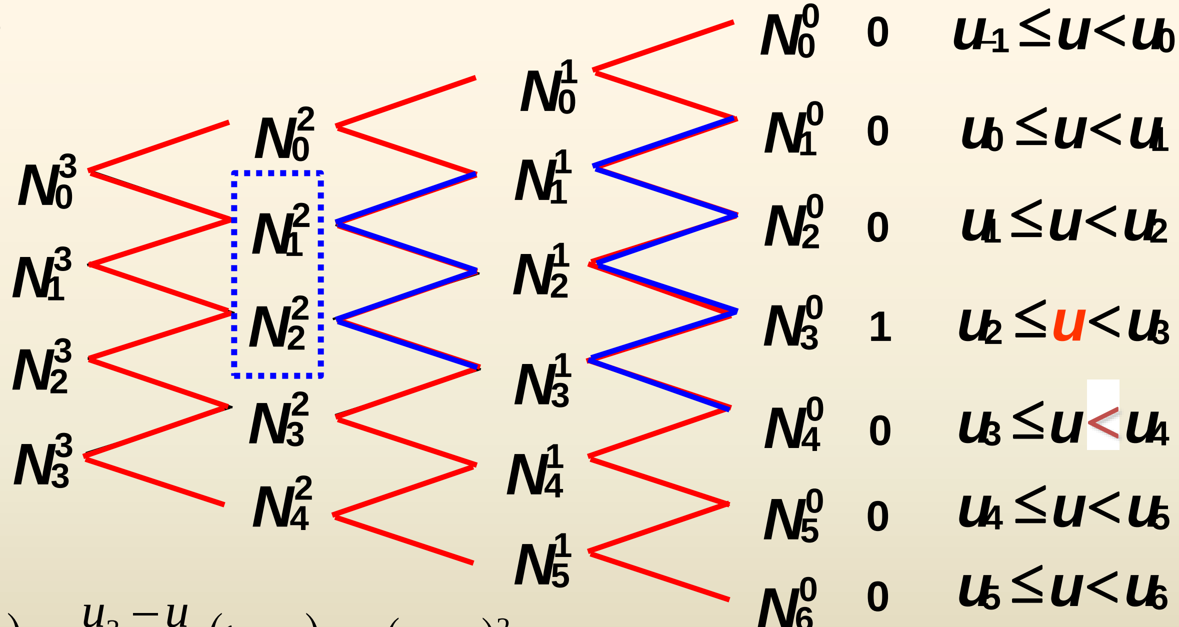
$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$

$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u)$$

$N_1^0(u) = 0, \quad N_2^0(u) = 0$

$N_3^0(u) = 1, \quad N_4^0(u) = 0$

$N_1^1(u) = 0, \; N_2^1(u) = 1 - u, \; N_2^1(u) = u$

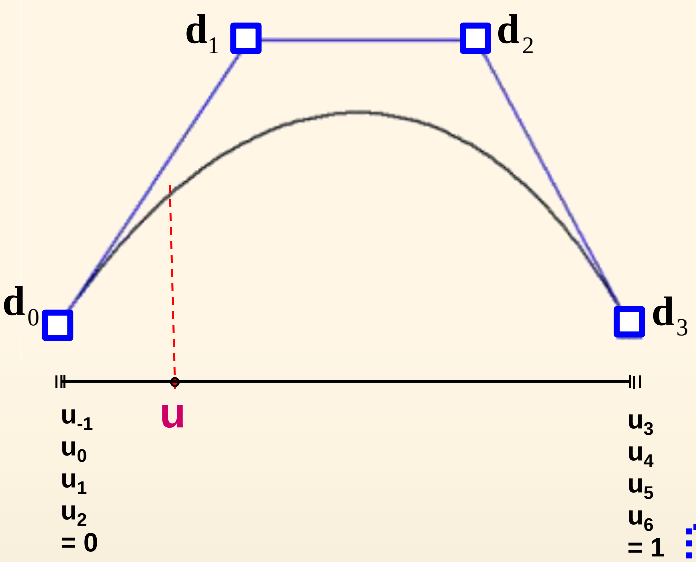


$$N_1^2(u) = \frac{u - u_0}{u_2 - u_0} N_1^1(u) + \frac{u_3 - u}{u_3 - u_1} N_2^1(u) = \frac{u_3 - u}{u_3 - u_1}(1 - u) = (1 - u)^2$$

$$N_2^2(u) = \frac{u - u_1}{u_3 - u_1} N_2^1(u) + \frac{u_4 - u}{u_4 - u_2} N_3^1(u) = \frac{u - u_1}{u_3 - u_1}(1 - u) + \frac{u_4 - u}{u_4 - u_2} u = 2u(1 - u)$$

2.3.5.1 Cox-de Boor Recurrence Formula (B-spline function) (10)

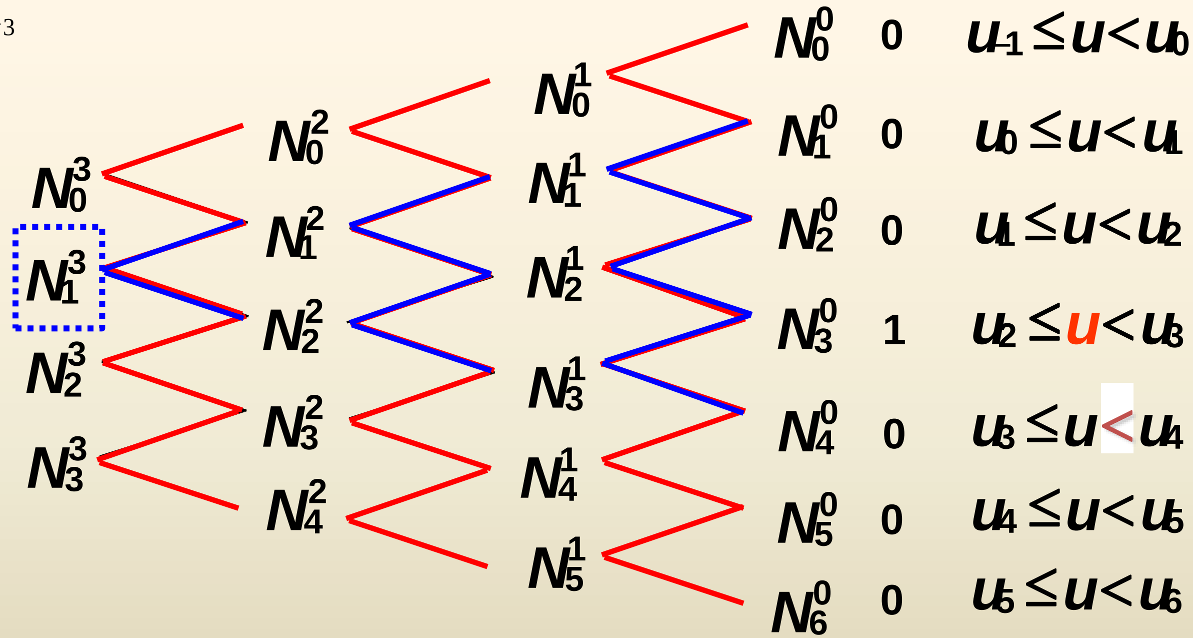
Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$



$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$

$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u)$$



$N_1^0(u) = 0, \quad N_2^0(u) = 0$

$N_3^0(u) = 1, \quad N_4^0(u) = 0$

$N_0^1(u) = 0, \quad N_1^1(u) = 0, \quad N_2^1(u) = 1 - u$

$N_1^2(u) = (1-u)^2, \quad N_2^2(u) = 2u(1-u)$

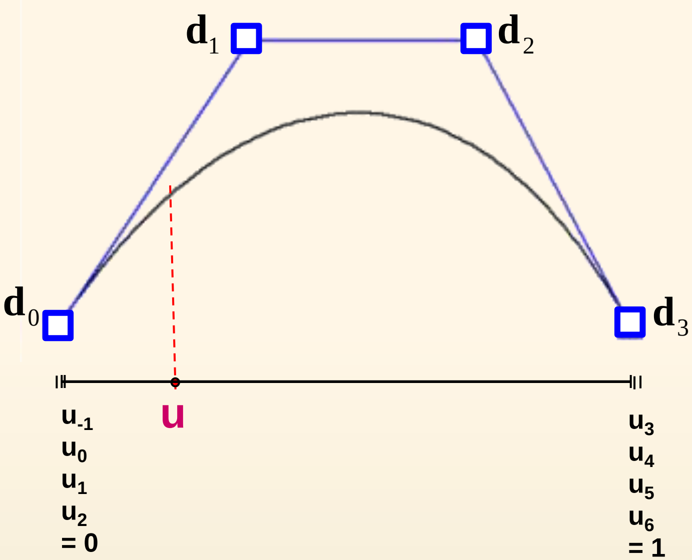
$$N_1^3(u) = \frac{u - u_0}{u_3 - u_0} N_1^2(u) + \frac{u_4 - u}{u_4 - u_1} N_2^2(u) = \frac{u - u_0}{u_3 - u_0}(1-u)^2 + \frac{u_4 - u}{u_4 - u_1} \cdot 2u(1-u) = 3u(1-u)^2$$

2.3.5.1 Cox-de Boor Recurrence Formula (B-spline function) (11)

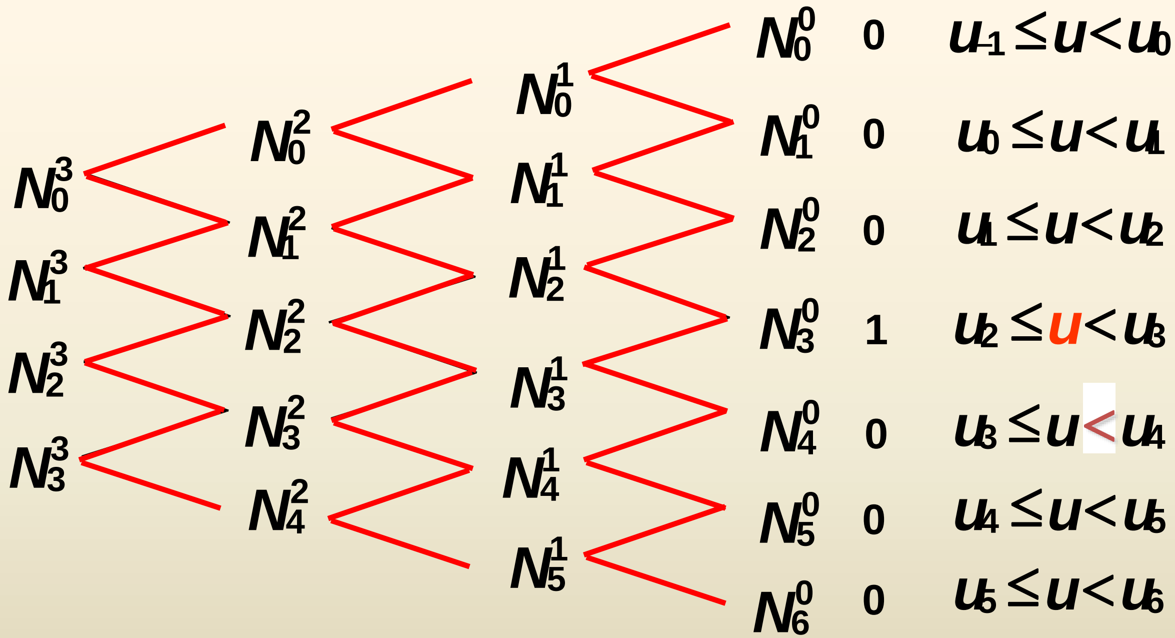
Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$



$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u)$$



$N_2^3(u)$, $N_2^3(u)$ 도 동일하게 계산

$$N_0^3(u) = (1-u)^3$$

$$N_1^3(u) = 3u(1-u)^2$$

$$N_2^3(u) = 3u^2(1-u)$$

$$N_3^3(u) = 3u^3$$

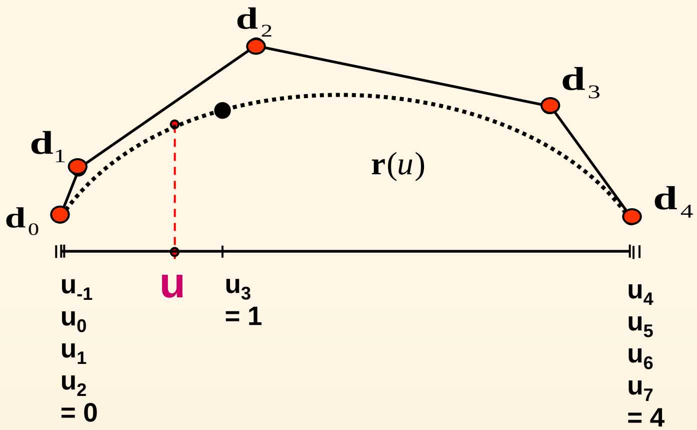
$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u)$$

$$\mathbf{r}(u) = (1-u)^3 \mathbf{d}_0 + 3u(1-u)^2 \mathbf{d}_1 + 3u^2(1-u)\mathbf{d}_2 + u^3 \mathbf{d}_3$$

➔3차 Bezier Curve 곡선식과 동일

2.3.5.1 Cox-de Boor Recurrence Formula (B-spline function) (12)

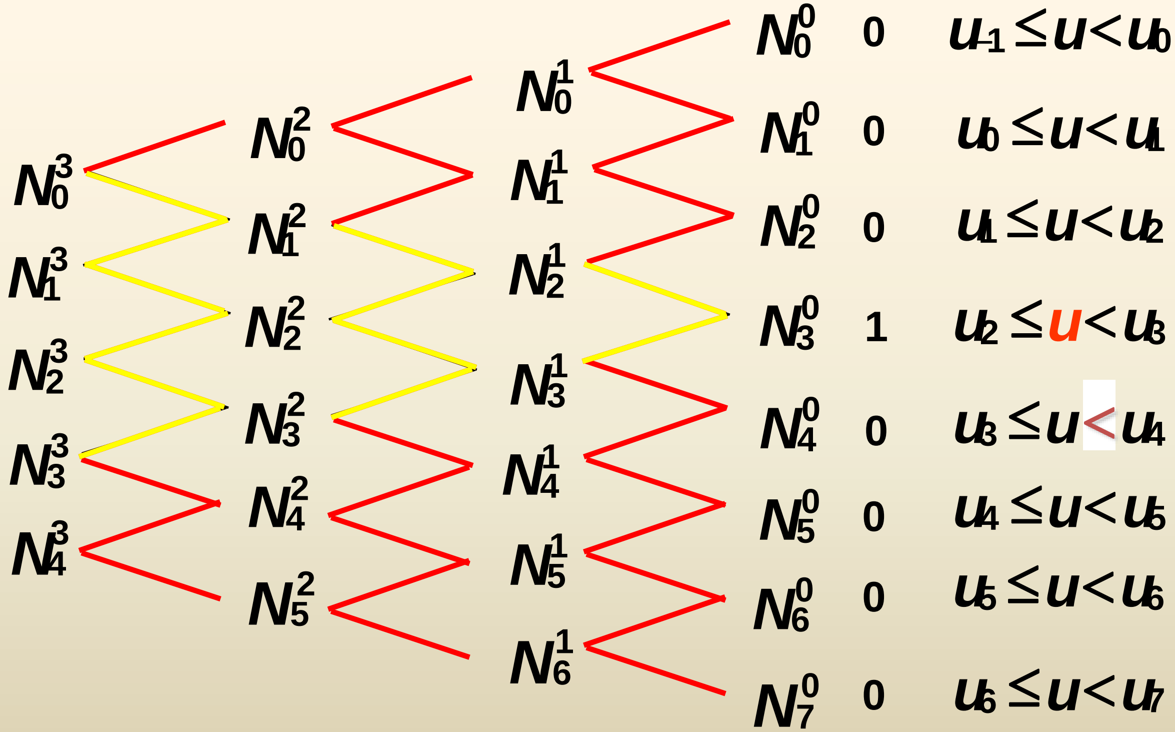
Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$



$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

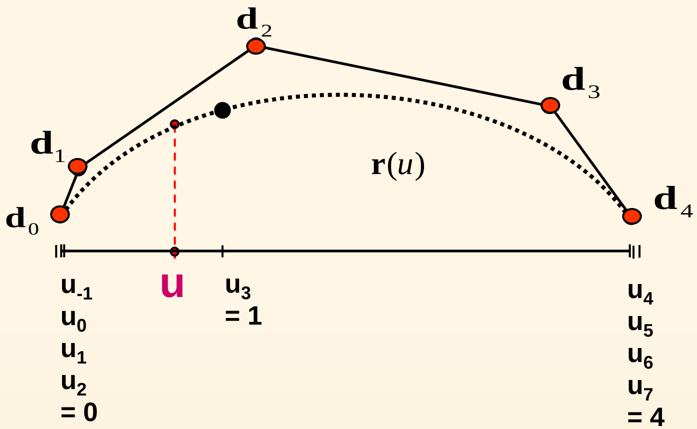
$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$

$$\mathbf{r}(u) = \sum_{i=0}^{D-1} \mathbf{d}_i N_i^n(u)$$



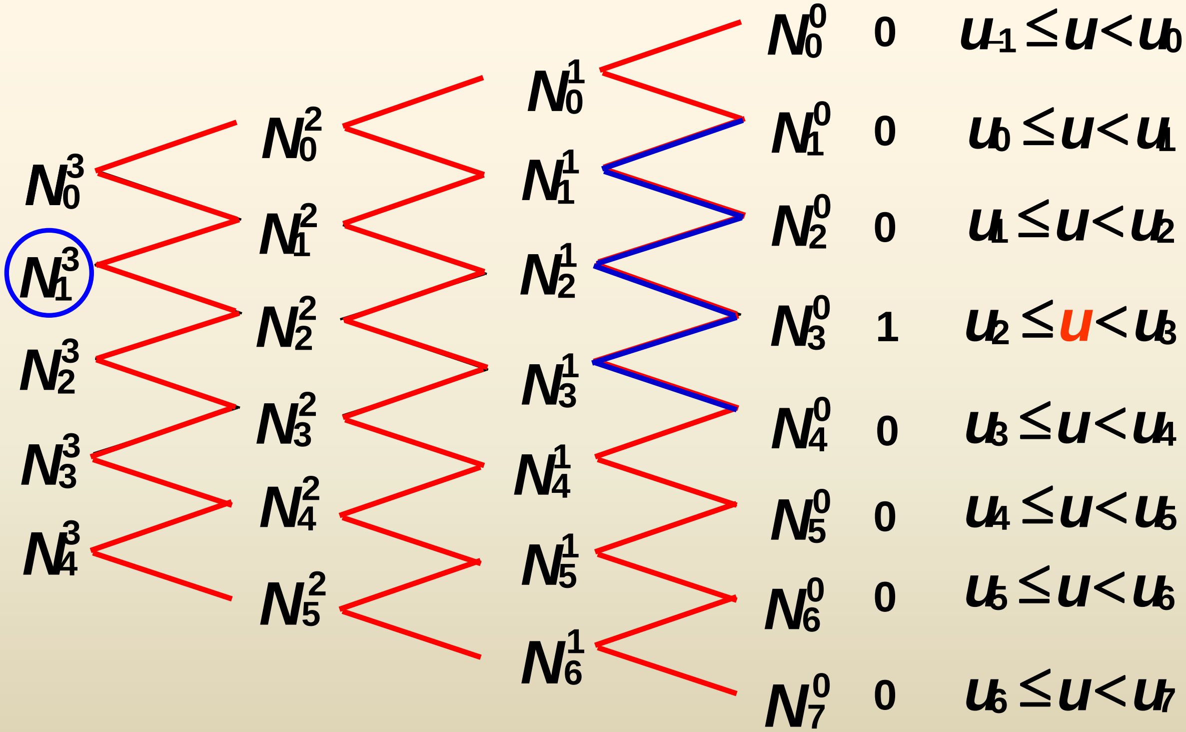
2.3.5.1 Cox-de Boor Recurrence Formula (B-spline function) (13)

Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$



$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$

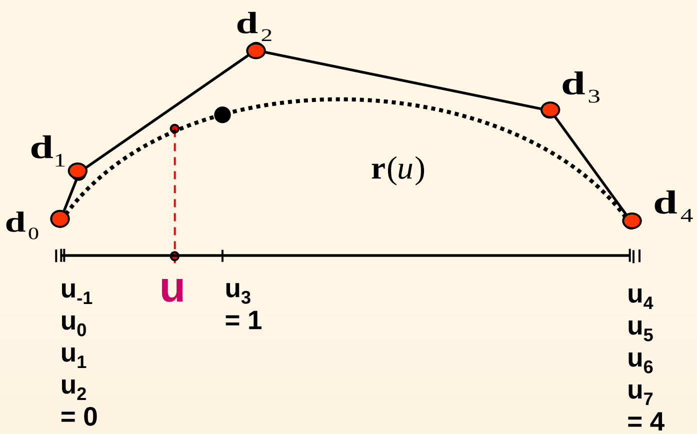


$$N_1^1 = 0 \quad N_2^1 = \frac{u_2 - u}{u_3 - u_2} \quad N_3^1 = \frac{u - u_2}{u_3 - u_2}$$

$$\mathbf{r}(u) = \sum_{i=0}^{D-1} \mathbf{d}_i N_i^n(u)$$

2.3.5.1 Cox-de Boor Recurrence Formula (B-spline function) (14)

Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$



$$N_i^n(u) = \frac{u-u_{i-1}}{u_{i+n-1}-u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n}-u}{u_{i+n}-u_i} N_{i+1}^{n-1}(u)$$

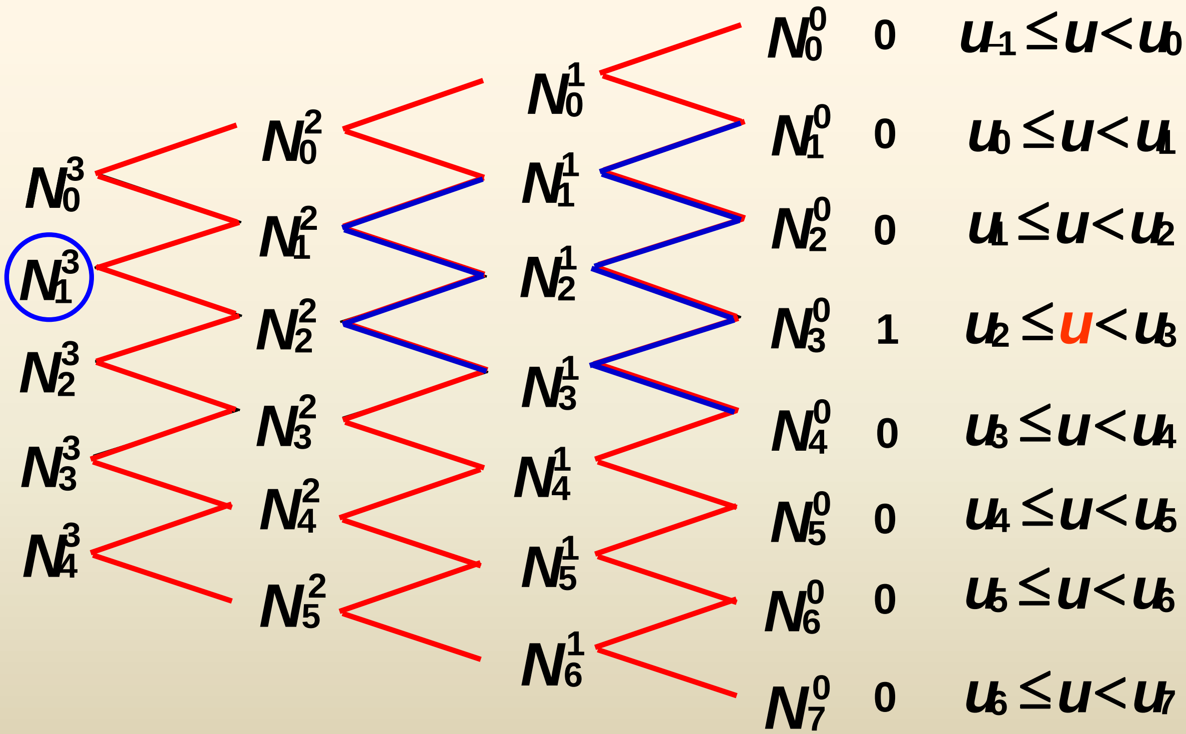
$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$

$$N_1^1 = 0 \quad N_2^1 = \frac{u_2-u}{u_3-u_2} \quad N_3^1 = \frac{u-u_2}{u_3-u_2}$$

$$N_1^2 = \frac{u_3-u}{u_3-u_1} N_2^1 = \frac{u_3-u}{u_3-u_1} \cdot \frac{u_2-u}{u_3-u_2}$$

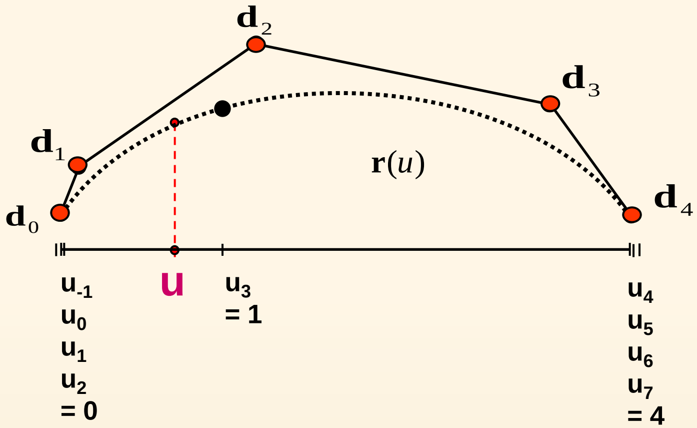
$$N_2^2 = \frac{u-u_1}{u_3-u_1} N_2^1 + \frac{u_4-u}{u_4-u_2} N_3^1$$

$$= \frac{u-u_1}{u_3-u_1} \cdot \frac{u_2-u}{u_3-u_2} + \frac{u_4-u}{u_4-u_2} \cdot \frac{u-u_2}{u_3-u_2}$$



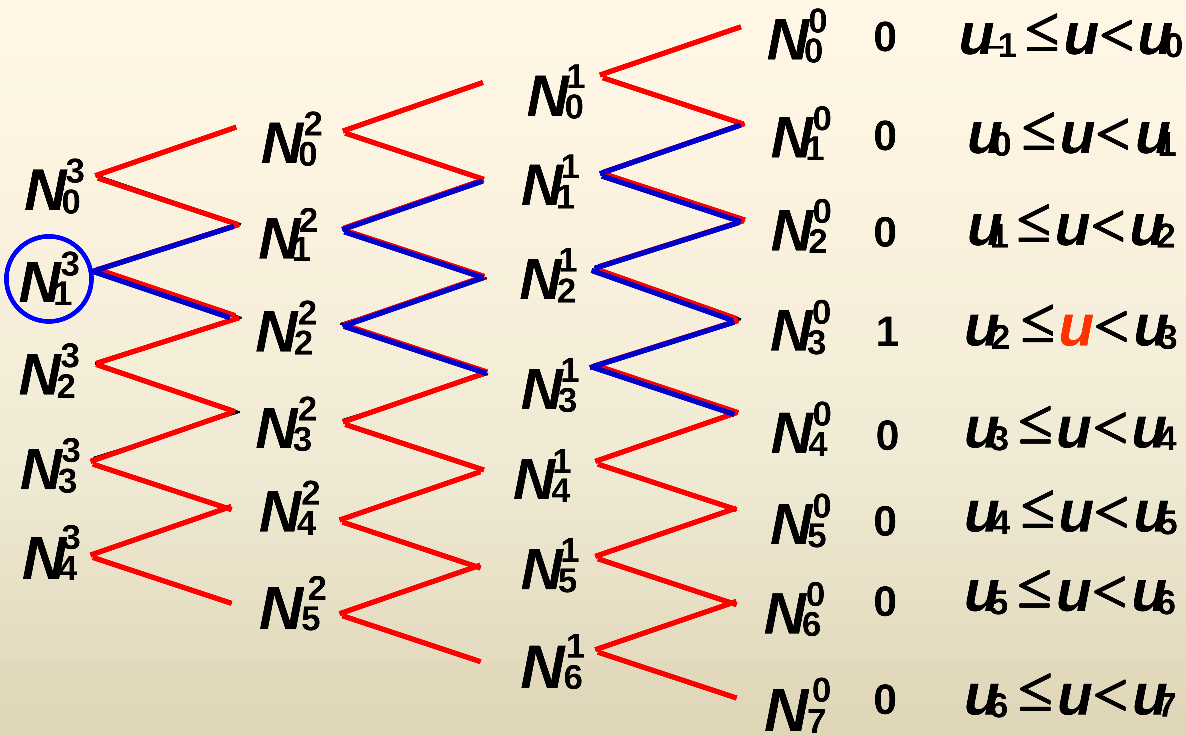
2.3.5.1 Cox-de Boor Recurrence Formula (B-spline function) (15)

Given	B-spline Control Point $\mathbf{d}_i$ Parameter u B-spline Basis Func. $N_i^n(u)$
Find	B-spline Curve $\mathbf{r}(u)$



$$N_i^n(u) = \frac{u-u_{i-1}}{u_{i+n-1}-u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n}-u}{u_{i+n}-u_i} N_{i+1}^{n-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \text{if } \ u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$



$$N_1^1 = 0 \qquad N_2^1 = \frac{u_2-u}{u_3-u_2} \qquad N_3^1 = \frac{u-u_2}{u_3-u_2}$$

$$N_1^2 = \frac{u_3-u}{u_3-u_1} N_2^1 = \frac{u_3-u}{u_3-u_1} \cdot \frac{u_2-u}{u_3-u_2}$$

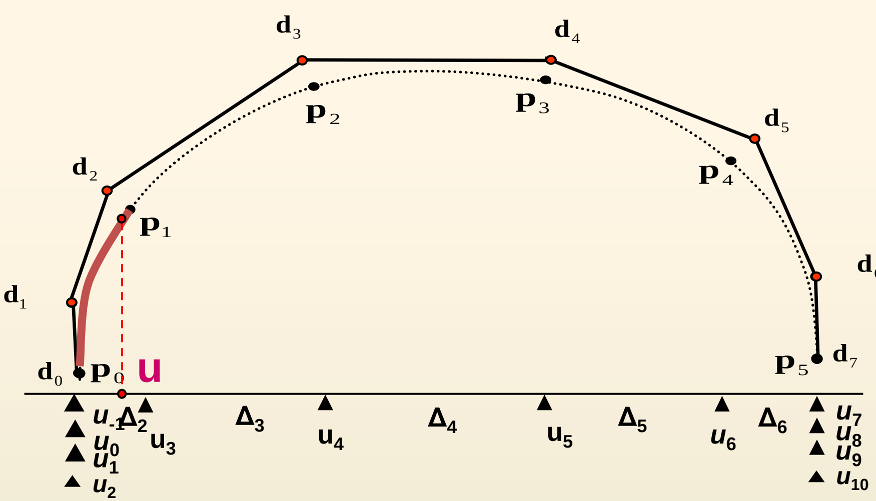
$$N_2^2 = \frac{u-u_1}{u_3-u_1} N_2^1 + \frac{u_4-u}{u_4-u_2} N_3^1$$

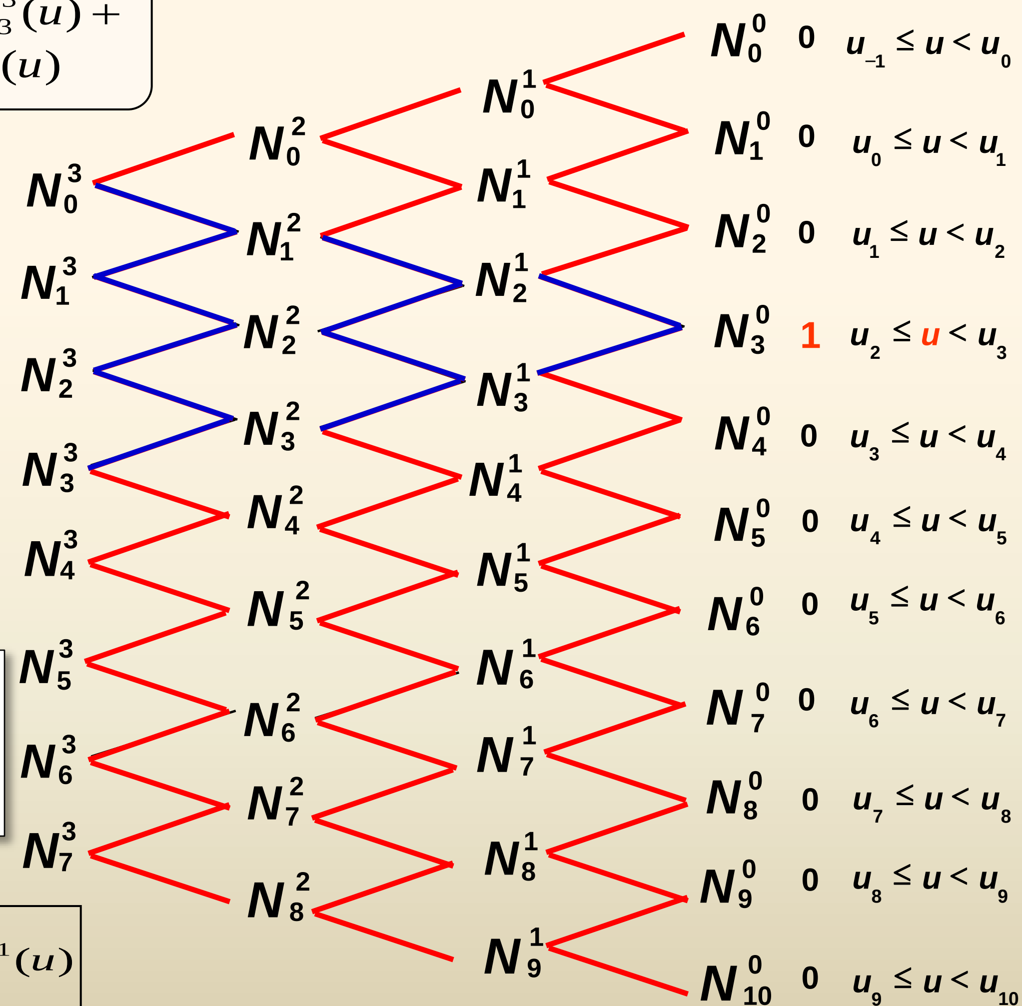
$$= \frac{u-u_1}{u_3-u_1} \cdot \frac{u_2-u}{u_3-u_2} + \frac{u_4-u}{u_4-u_2} \cdot \frac{u-u_2}{u_3-u_2}$$

$$N_1^3 = \frac{u-u_0}{u_3-u_2} N_1^2 + \frac{u_4-u}{u_4-u_1} N_2^2 = \frac{u-u_0}{u_3-u_2} \cdot \frac{u_3-u}{u_3-u_1} \cdot \frac{u_2-u}{u_3-u_2} + \frac{u_4-u}{u_4-u_1} \cdot \frac{u-u_1}{u_3-u_1} \cdot \frac{u_2-u}{u_3-u_2} + \frac{u_4-u}{u_4-u_1} \cdot \frac{u_4-u}{u_4-u_2} \cdot \frac{u-u_2}{u_3-u_2}$$

2.3.5.2 B-spline curves (1)

$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u) + \mathbf{d}_4 N_4^3(u) + \mathbf{d}_5 N_5^3(u) + \mathbf{d}_6 N_6^3(u) + \mathbf{d}_7 N_7^3(u)$$





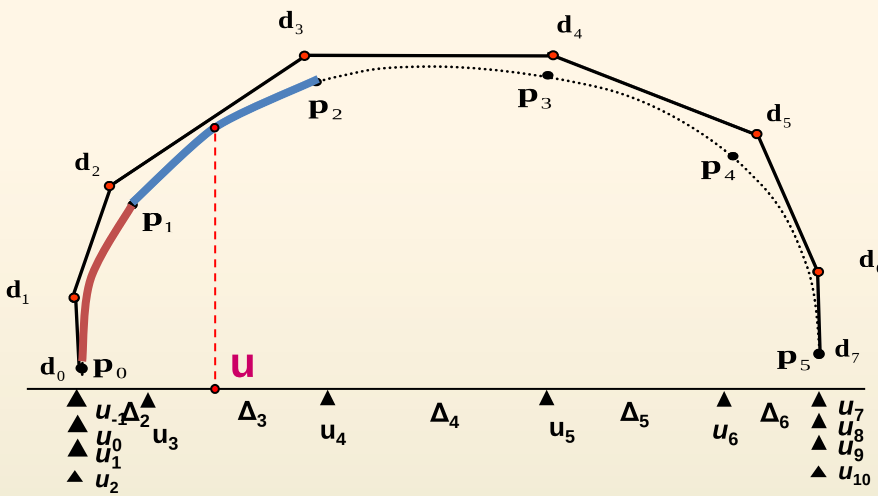
$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_4 N_4^3(u) + \mathbf{d}_4 N_4^3(u) + \mathbf{d}_5 N_5^3(u) + \mathbf{d}_6 N_6^3(u) + \mathbf{d}_7 N_7^3(u)$$

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

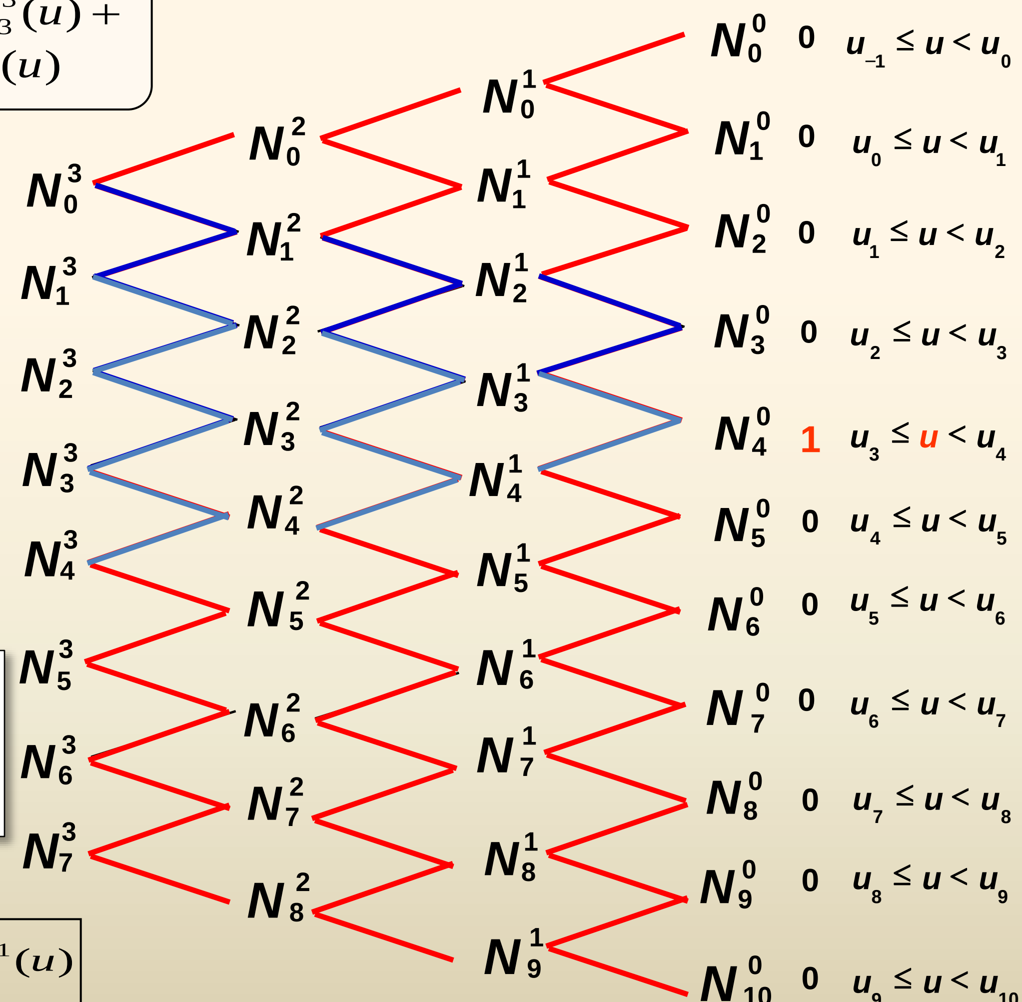
$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$

2.3.5.2 B-spline curves (2)

$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u) + \mathbf{d}_4 N_4^3(u) + \mathbf{d}_5 N_5^3(u) + \mathbf{d}_6 N_6^3(u) + \mathbf{d}_7 N_7^3(u)$$



$$\mathbf{r}(u) = \mathbf{d}_0 \cancel{N_0^3}(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_4 N_4^3(u) + \mathbf{d}_4 N_4^3(u) + \mathbf{d}_5 \cancel{N_5^3}(u) + \mathbf{d}_6 \cancel{N_6^3}(u) + \mathbf{d}_7 \cancel{N_7^3}(u)$$

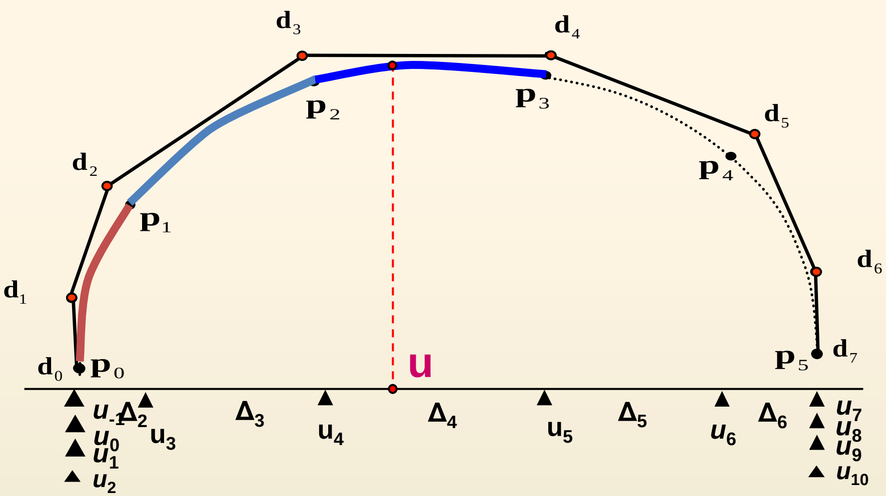


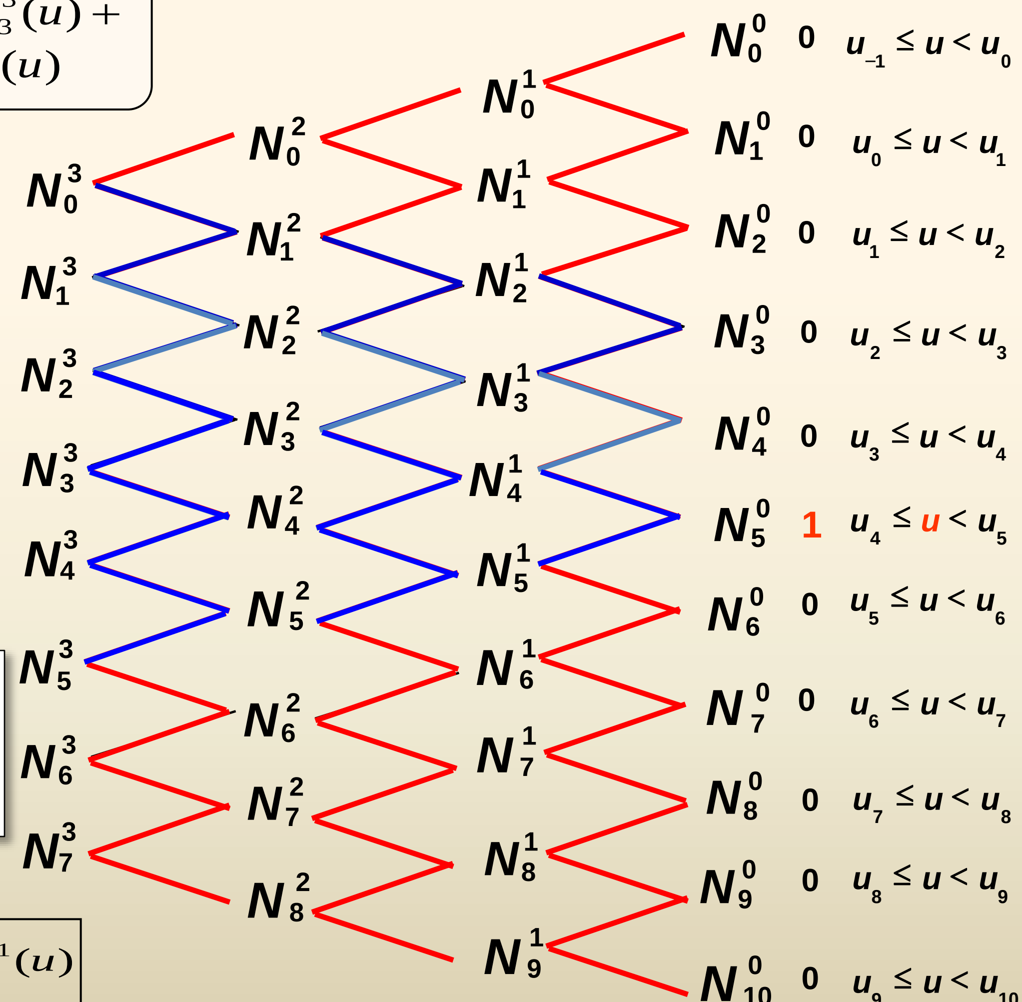
$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \text{if} \quad u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$

2.3.5.2 B-spline curves (3)

$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u) + \mathbf{d}_4 N_4^3(u) + \mathbf{d}_5 N_5^3(u) + \mathbf{d}_6 N_6^3(u) + \mathbf{d}_7 N_7^3(u)$$





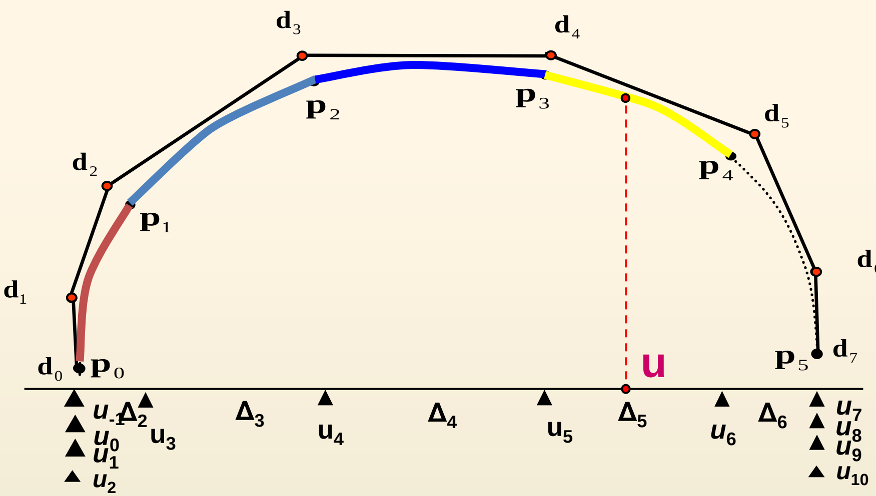
$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_4 N_4^3(u) + \mathbf{d}_4 N_4^3(u) + \mathbf{d}_5 N_5^3(u) + \mathbf{d}_6 N_6^3(u) + \mathbf{d}_7 N_7^3(u)$$

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

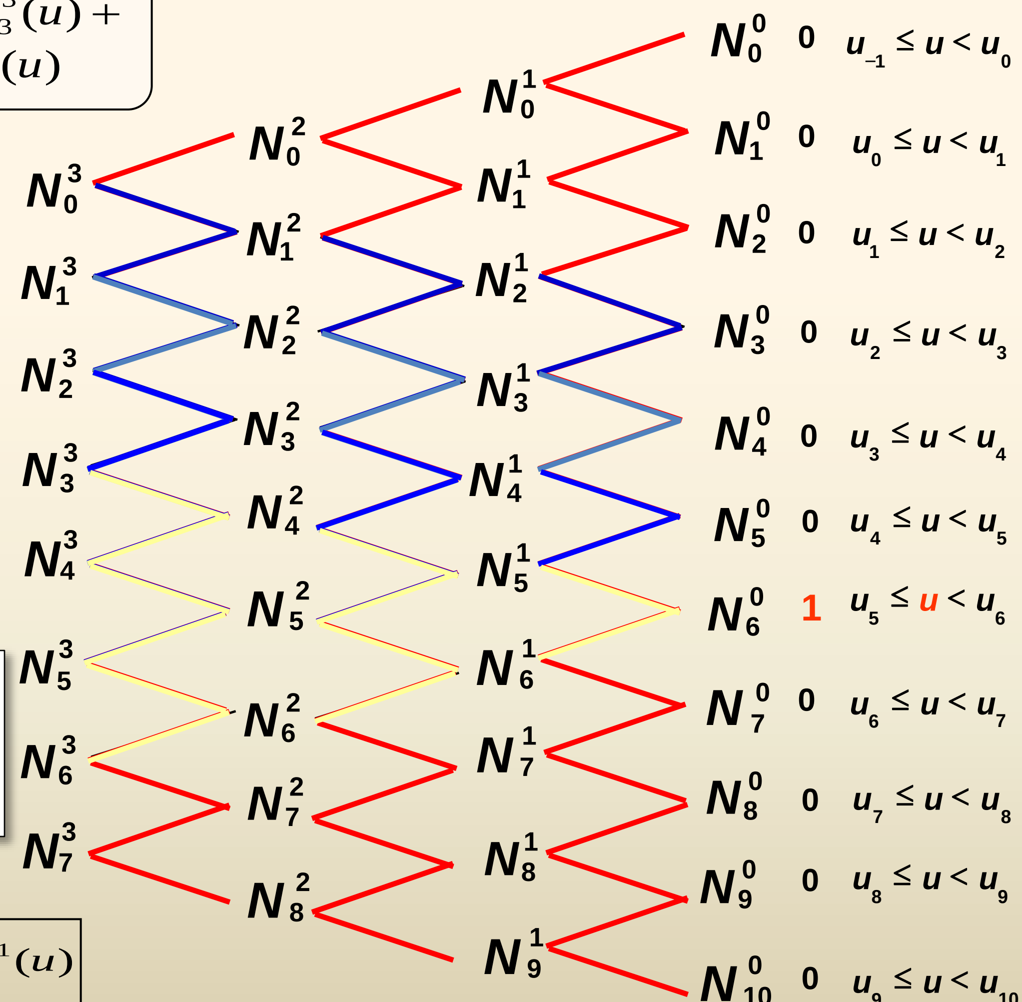
$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$

2.3.5.2 B-spline curves (4)

$$\mathbf{r}(u)=\mathbf{d}_0N_0^3(u)+\mathbf{d}_1N_1^3(u)+\mathbf{d}_2N_2^3(u)+\mathbf{d}_3N_3^3(u)+\mathbf{d}_4N_4^3(u)+\mathbf{d}_5N_5^3(u)+\mathbf{d}_6N_6^3(u)+\mathbf{d}_7N_7^3(u)$$



$$\mathbf{r}(u)=\mathbf{d}_0N_0^3(u)+\mathbf{d}_1N_1^3(u)+\mathbf{d}_2N_2^3(u)+\mathbf{d}_4N_4^3(u)$$
$$+\mathbf{d}_4N_4^3(u)+\mathbf{d}_5N_5^3(u)+\mathbf{d}_6N_6^3(u)+\mathbf{d}_7N_7^3(u)$$

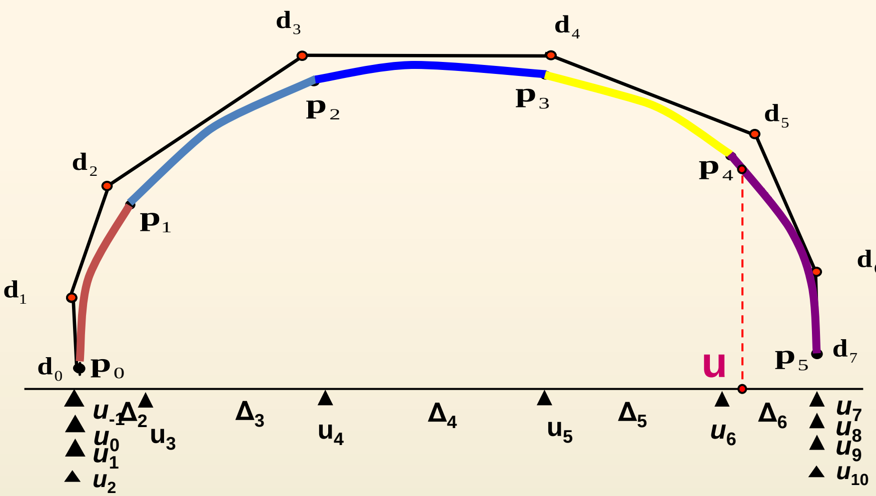


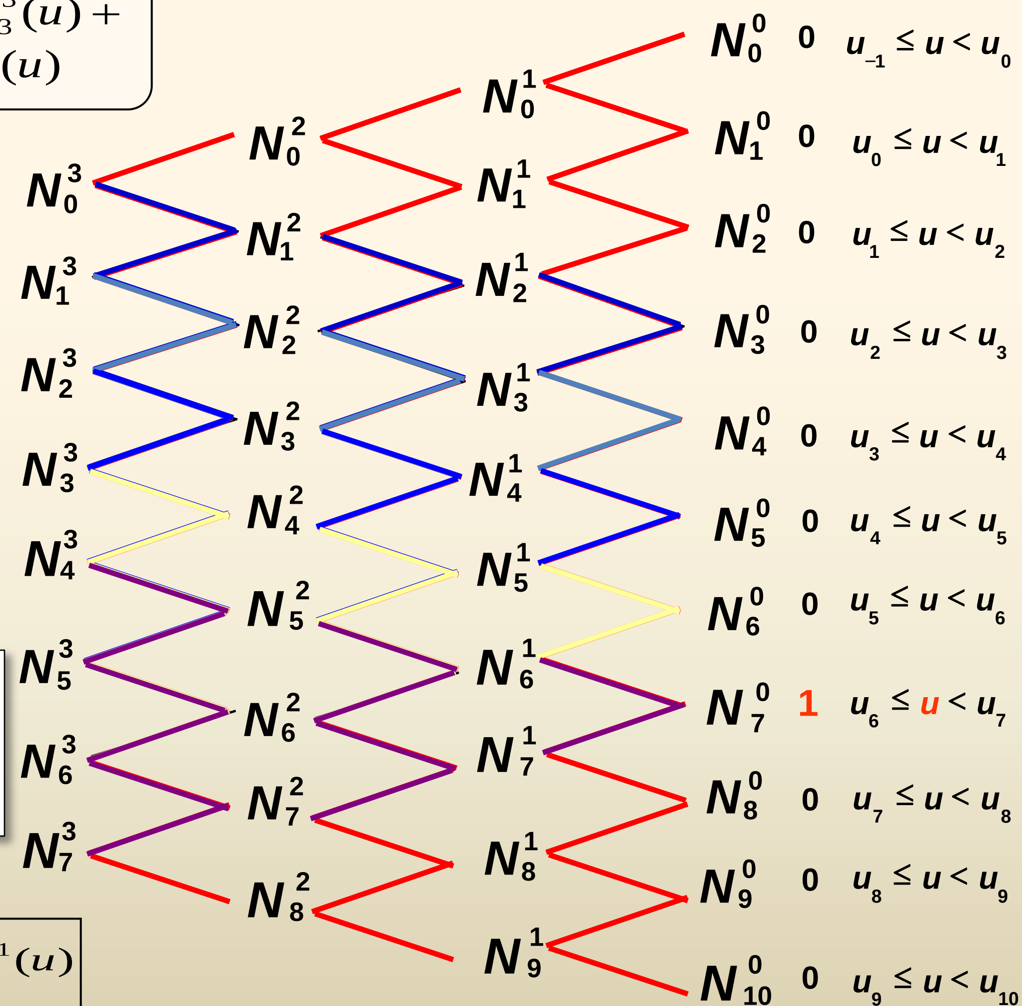
$$N_i^n(u)=\frac{u-u_{i-1}}{u_{i+n-1}-u_{i-1}}N_i^{n-1}(u)+\frac{u_{i+n}-u}{u_{i+n}-u_i}N_{i+1}^{n-1}(u)$$

$$N_i^0(u)=\begin{cases}1 & \text{if } u_{i-1}\le u<u_i\\ 0 & \text{else}\end{cases}$$

2.3.5.2 B-spline curves (5)

$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u) + \mathbf{d}_4 N_4^3(u) + \mathbf{d}_5 N_5^3(u) + \mathbf{d}_6 N_6^3(u) + \mathbf{d}_7 N_7^3(u)$$





$$\mathbf{r}(u) = \mathbf{d}_0 \cancel{N_0^3}(u) + \mathbf{d}_1 \cancel{N_1^3}(u) + \mathbf{d}_2 \cancel{N_2^3}(u) + \mathbf{d}_4 \cancel{N_4^3}(u) + \mathbf{d}_4 N_4^3(u) + \mathbf{d}_5 N_5^3(u) + \mathbf{d}_6 N_6^3(u) + \mathbf{d}_7 N_7^3(u)$$

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$

2.3.5.3 Relationship between de Boor algorithm & B-spline curves

☑ de Boor 알고리즘 : "Constructive Approach"

Input: $\mathbf{d}_i$ (de Boor Points)

Processor: 구간별로 $\mathbf{d}_i$를 n번 순차적 'linear interpolation'

Output : n차 곡선상의 점

→'B-spline function'(Cox-de Boor recurrence formula) 형태로 표현 됨

☑ B-spline 곡선식: "B-spline function evaluation Approach"

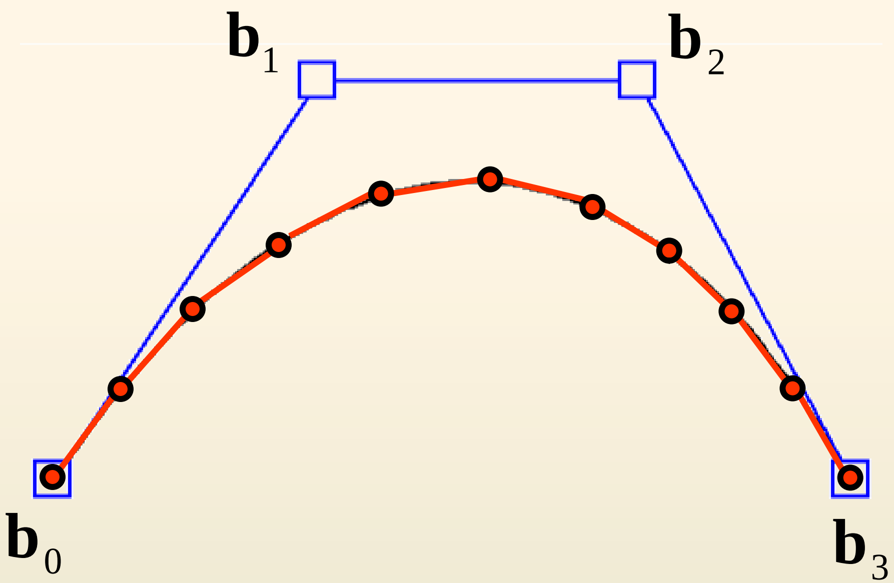
Input: $\mathbf{d}_i$ (de Boor Points)

Processor: 공간 상의 점 $\mathbf{d}_i$와 B-spline function을 "blending"하여 함수 값을 계산하면 곡선상의 점을 구할 수 있음

Output: B-spline function과 $\mathbf{d}_i$의 혼합 함수 형태로 표현

2.3.5.4 Programming B-spline Curve class

Cubic B-spline 예시



$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u)$$

1) B-spline Curve의 구성

- **Degree**
- **Control Point**

Member Variables of B-spline Curve Class
int n: degree of B-spline Curve
Vector* m_ControlPoint: Control Point
int m_nControlPoint: the number of Control Point

2) B-spline Basis Function 계산

(Cox-de Boor Recurrence Formula)

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \text{if} \quad u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$

3) B-spline Curve 작도

- **곡선을 Line Segment로 나누어 작도**
- **Parameter u를 u_{min}~u_{max}까지 n등분하여 각 u에 대한 곡선 상의 점을 구함**
- **위에서 구한 점을 직선으로 연결하여 곡선을 표현**

2.3.5.4 Sample code of Cubic B-spline Curve (1)

Member Variables of B-spline Curve Class

int m_nDegree : degree of B-spline Curve

Vector* m_ControlPoint: Control Point

int m_nControlPoint: the number of Control Point

```
#include "vector.h“

class CubicBsplineCurve {
public:
    Vector* m_ControlPoint;   int m_nControlPoint;
    double* m_Knot; int m_nKnot;
    int m_nDegree;

    CubicBsplineCurve();
    ~CubicBsplineCurve();

    void SetControlPoint(Vector* pControlPoint, int nControlPoint);
    void SetKnot(double* pKnot, int nKnot);
    Vector CalcPoint(double u);
    double N(int d, int i, double u);                    // B-spline basis function
};
```

2.3.5.4 Sample code of Cubic B-spline Curve (2)

```
CubicBsplineCurve::CubicBsplineCurve () {
   m_ControlPoint = 0;        m_Knot = 0;
   m_nControlPoint = 0;       m_nKnot = 0;       int m_nDegree =3;
}
CubicBsplineCurve::~CubicBsplineCurve () {
   if(m_ControlPoint)  delete[] m_ControlPoint;
   if(m_Knot)  delete[] m_Knot;
}
void CubicBsplineCurve::SetControlPoint(Vector* pControlPoint, int nControlPoint) {
   m_ControlPoint = new Vector[nControlPoint];
   for(int i=0; i < nControlPoint; i++) {
      m_ControlPoint[i] = pControlPoint[i];
   }
}
void CubicBsplineCurve::SetKnot(double* pKnot, int nKnot){
   m_Knot = new double[nKnot];
   for(int i=0; i < nKnot; i++) {
     m_Knot[i] = pKnot[i];
   }
}
```

2.3.5.4 Sample code of Cubic B-spline Curve (3)

```
Vector CubicBsplineCurve::CalcPoint(double u)
{
    Vector PointOnCurve(0,0,0);
    if ( t < m_Knot[0] || t > m_Knot[m_nKnot-1] ) {
        return PointOnCurve;
    }
    for(int i = 0; i < m_nControlPoint; i++){
        PointOnCurve  =  PointOnCurve  +  m_ControlPoint[i] * N(m_nDegree, i, u);
    }
    return PointOnCurve;
}
```

Get points on curve at parameter *u*

$$\mathbf{r}(u) = \mathbf{d}_0 N_0^3(u) + \mathbf{d}_1 N_1^3(u) + \mathbf{d}_2 N_2^3(u) + \mathbf{d}_3 N_3^3(u)$$

2.3.5.4 Sample code of Cubic B-spline Curve (4)

```
double CubicBsplineCurve:: N(int d, int i,
    // Find Span k
    // U i-1 <= U < U i  → k = i

    if( d == 0 ) {
        // return 0 or 1;
    } else {
        // return Cox de-Boor recurrence f
    }
}
```

B-spline Basis Function 계산

(Cox-de Boor Recurrence Formula)

$$N_i^n(u) = \frac{u - u_{i-1}}{u_{i+n-1} - u_{i-1}} N_i^{n-1}(u) + \frac{u_{i+n} - u}{u_{i+n} - u_i} N_{i+1}^{n-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \text{if } u_{i-1} \le u < u_i \\ 0 & \text{else} \end{cases}$$