

B-SPLINE CURVES

Rodrigo Silveira

Curve and Surface Design
Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

INTRODUCTION TO B-SPLINES

Improving over Bézier curves

INTRODUCTION TO B-SPLINES

Improving over Bézier curves

Bézier curves have some drawbacks:

- Degree is proportional to number of control points
- Does not offer true global control (at most “pseudo-local”)
- C^2 continuity is not so easy to obtain for composite curves

INTRODUCTION TO B-SPLINES

Improving over Bézier curves

Bézier curves have some drawbacks:

- Degree is proportional to number of control points
- Does not offer true global control (at most “pseudo-local”)
- C^2 continuity is not so easy to obtain for composite curves

To overcome this: **B-splines**

INTRODUCTION TO B-SPLINES

Improving over Bézier curves

Bézier curves have some drawbacks:

- Degree is proportional to number of control points
- Does not offer true global control (at most “pseudo-local”)
- C^2 continuity is not so easy to obtain for composite curves

To overcome this: **B-splines**

- “Discovered” by Schoenberg (1940s) (actually, even earlier), developed as we will see them by Riesenfeld and others in 1970s
- B-splines = Basis splines
- Several flavors: uniform, non-uniform, rational non-uniform (NURBs)...

QUADRATIC UNIFORM B-SPLINES

Deriving the formula for the quadratic B-splines

Setting:

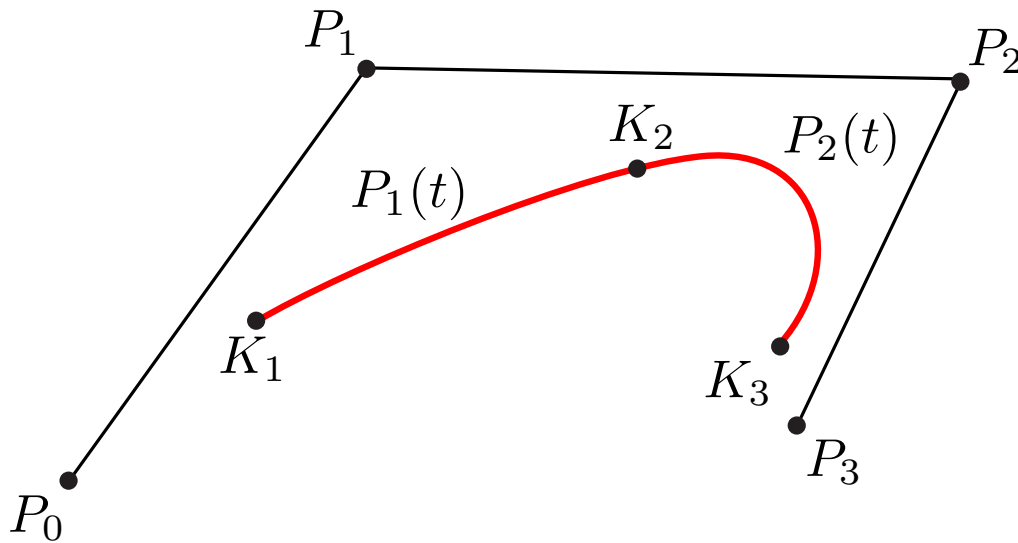
- Input: $n + 1$ control points P_0, \dots, P_n
- Output: **spline curve** where each segment $P_i(t)$ is a **quadratic** parametric polynomial based on P_{i-1}, P_i and P_{i+1} . Segments start and end at points K_i , called *knots*.

QUADRATIC UNIFORM B-SPLINES

Deriving the formula for the quadratic B-splines

Setting:

- Input: $n + 1$ control points P_0, \dots, P_n
- Output: **spline curve** where each segment $P_i(t)$ is a **quadratic** parametric polynomial based on P_{i-1}, P_i and P_{i+1} . Segments start and end at points K_i , called *knots*.



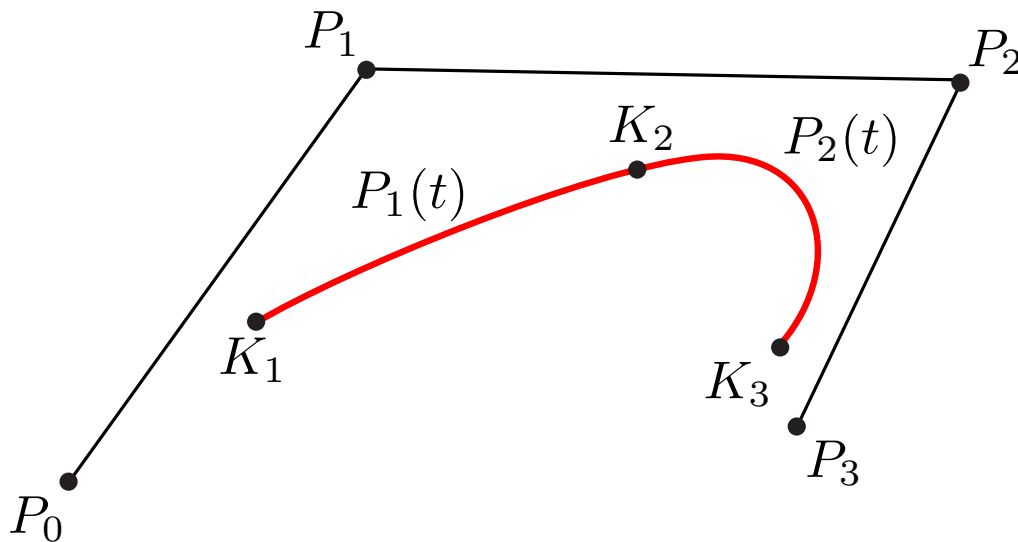
sketch of setting for $n = 3$ (not accurate!)

QUADRATIC UNIFORM B-SPLINES

Deriving the formula for the quadratic B-splines

Setting:

- Input: $n + 1$ control points P_0, \dots, P_n
- Output: **spline curve** where each segment $P_i(t)$ is a **quadratic** parametric polynomial based on P_{i-1}, P_i and P_{i+1} . Segments start and end at points K_i , called *knots*.



$$P_i(t) = (t^2, t, 1) \begin{pmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \end{pmatrix}$$

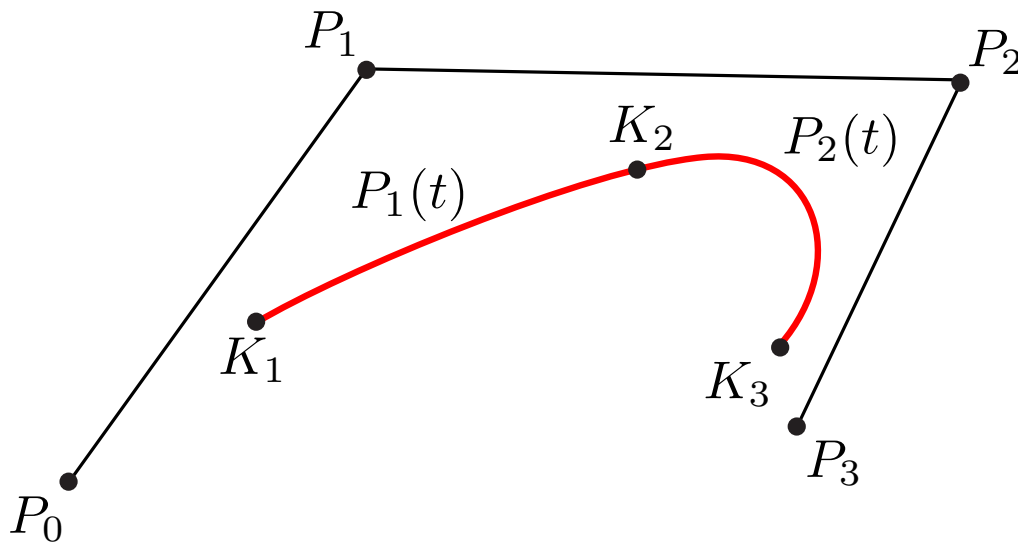
sketch of setting for $n = 3$ (not accurate!)

QUADRATIC UNIFORM B-SPLINES

Deriving the formula for the quadratic B-splines

Setting:

- Input: $n + 1$ control points P_0, \dots, P_n
- Output: **spline curve** where each segment $P_i(t)$ is a **quadratic** parametric polynomial based on P_{i-1}, P_i and P_{i+1} . Segments start and end at points K_i , called *knots*.



$$P_i(t) = (t^2, t, 1) \begin{pmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \end{pmatrix}$$
$$= (t^2, t, 1) \begin{pmatrix} a_2 & b_2 & c_2 \\ a_1 & b_1 & c_1 \\ a_0 & b_0 & c_0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \end{pmatrix}$$

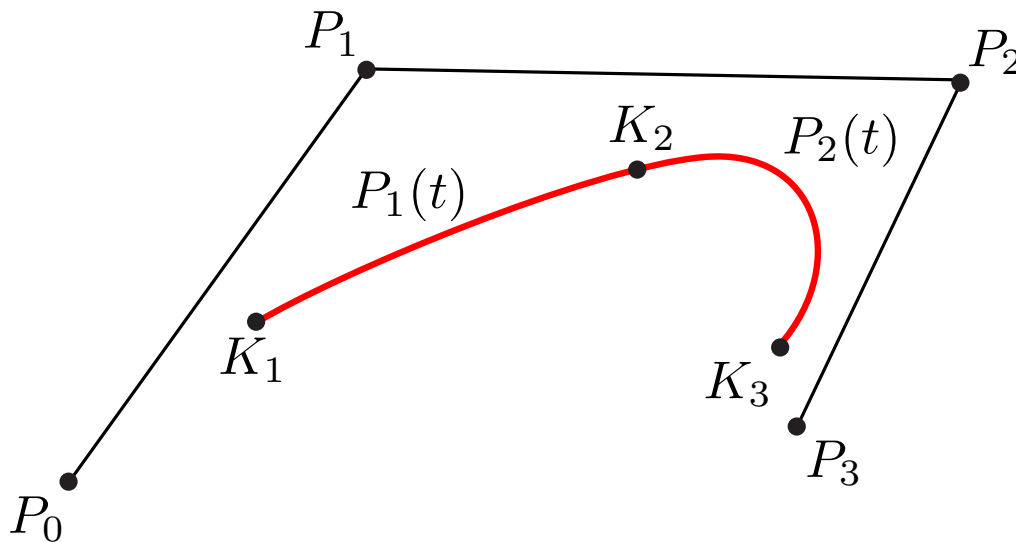
sketch of setting for $n = 3$ (not accurate!)

QUADRATIC UNIFORM B-SPLINES

Deriving the formula for the quadratic B-splines

Setting:

- Input: $n + 1$ control points P_0, \dots, P_n
- Output: **spline curve** where each segment $P_i(t)$ is a **quadratic** parametric polynomial based on P_{i-1}, P_i and P_{i+1} . Segments start and end at points K_i , called *knots*.



sketch of setting for $n = 3$ (not accurate!)

$$P_i(t) = (t^2, t, 1) \begin{pmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \end{pmatrix}$$
$$= (t^2, t, 1) \begin{pmatrix} a_2 & b_2 & c_2 \\ a_1 & b_1 & c_1 \\ a_0 & b_0 & c_0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \end{pmatrix}$$

Requirements:

1. $P_1(t)$ and $P_2(t)$ meet smoothly at common point
2. Affine combination of control points

Question: what is the matrix?

QUADRATIC UNIFORM B-SPLINES

Deriving the formula for the quadratic B-splines

$$\begin{aligned} P_i(t) &= \frac{1}{2}(t^2, t, 1) \begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \end{pmatrix}, i = 1, 2, \dots \\ &= \frac{1}{2}(t^2 - 2t + 1)P_{i-1} + \frac{1}{2}(-2t^2 + 2t + 1)P_i + \frac{t^2}{2}P_{i+1} \end{aligned}$$

QUADRATIC UNIFORM B-SPLINES

Deriving the formula for the quadratic B-splines

$$\begin{aligned} P_i(t) &= \frac{1}{2}(t^2, t, 1) \begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \end{pmatrix}, i = 1, 2, \dots \\ &= \frac{1}{2}(t^2 - 2t + 1)P_{i-1} + \frac{1}{2}(-2t^2 + 2t + 1)P_i + \frac{t^2}{2}P_{i+1} \end{aligned}$$

- Start and endpoints: K_i and K_{i+1}

QUADRATIC UNIFORM B-SPLINES

Deriving the formula for the quadratic B-splines

$$\begin{aligned} P_i(t) &= \frac{1}{2}(t^2, t, 1) \begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \end{pmatrix}, i = 1, 2, \dots \\ &= \frac{1}{2}(t^2 - 2t + 1)P_{i-1} + \frac{1}{2}(-2t^2 + 2t + 1)P_i + \frac{t^2}{2}P_{i+1} \end{aligned}$$

- Start and endpoints: K_i and K_{i+1}

Since $K_i = P_i(0)$ and $K_{i+1} = P_i(1)$, we have:

$$K_i = P_i(0) = \frac{1}{2}(P_{i-1} + P_i)$$

$$K_{i+1} = P_i(1) = \frac{1}{2}(P_i + P_{i+1})$$

QUADRATIC UNIFORM B-SPLINES

Deriving the formula for the quadratic B-splines

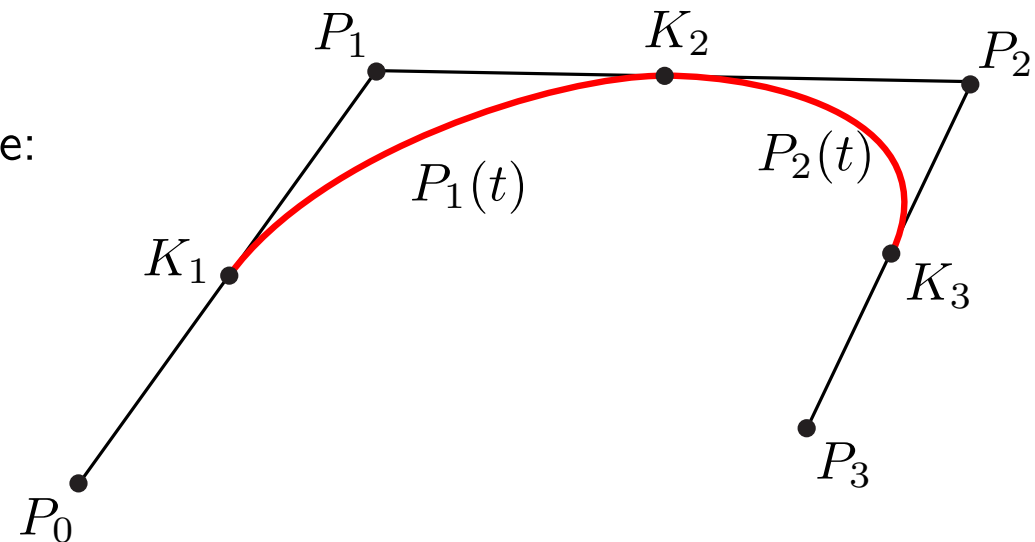
$$\begin{aligned} P_i(t) &= \frac{1}{2}(t^2, t, 1) \begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \end{pmatrix}, i = 1, 2 \\ &= \frac{1}{2}(t^2 - 2t + 1)P_{i-1} + \frac{1}{2}(-2t^2 + 2t + 1)P_i + \frac{t^2}{2}P_{i+1} \end{aligned}$$

- Start and endpoints: K_i and K_{i+1}

Since $K_i = P_i(0)$ and $K_{i+1} = P_i(1)$, we have:

$$K_i = P_i(0) = \frac{1}{2}(P_{i-1} + P_i)$$

$$K_{i+1} = P_i(1) = \frac{1}{2}(P_i + P_{i+1})$$



More accurate picture

QUADRATIC UNIFORM B-SPLINES

Deriving the formula for the quadratic B-splines

$$\begin{aligned} P_i(t) &= \frac{1}{2}(t^2, t, 1) \begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \end{pmatrix}, i = 1, 2 \\ &= \frac{1}{2}(t^2 - 2t + 1)P_{i-1} + \frac{1}{2}(-2t^2 + 2t + 1)P_i + \frac{t^2}{2}P_{i+1} \end{aligned}$$

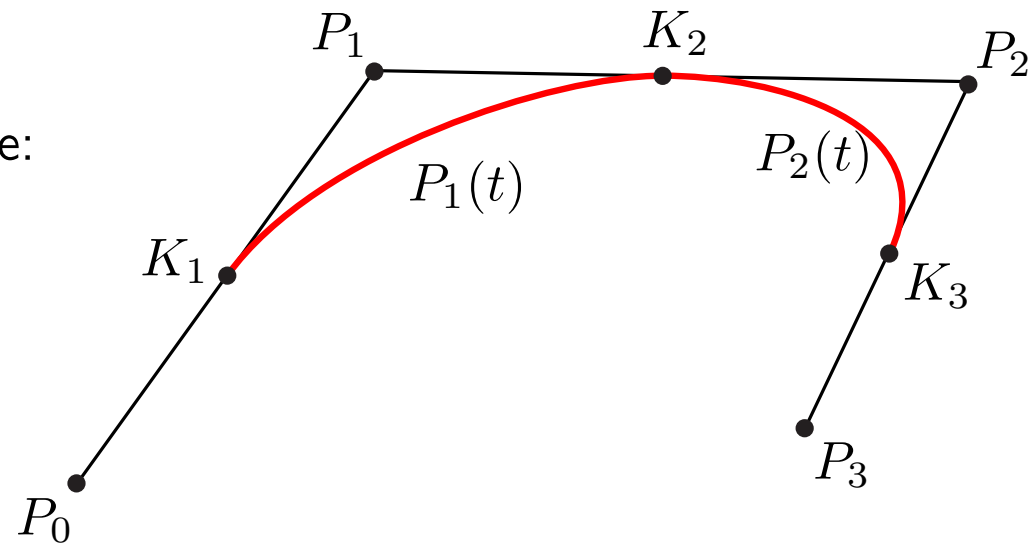
- Start and endpoints: K_i and K_{i+1}

Since $K_i = P_i(0)$ and $K_{i+1} = P_i(1)$, we have:

$$K_i = P_i(0) = \frac{1}{2}(P_{i-1} + P_i)$$

$$K_{i+1} = P_i(1) = \frac{1}{2}(P_i + P_{i+1})$$

Question: what are the tangent vectors at the ends?



More accurate picture

QUADRATIC UNIFORM B-SPLINES

Deriving the formula for the quadratic B-splines

$$\begin{aligned} P_i(t) &= \frac{1}{2}(t^2, t, 1) \begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \end{pmatrix}, i = 1, 2 \\ &= \frac{1}{2}(t^2 - 2t + 1)P_{i-1} + \frac{1}{2}(-2t^2 + 2t + 1)P_i + \frac{t^2}{2}P_{i+1} \end{aligned}$$

- Start and endpoints: K_i and K_{i+1}

Since $K_i = P_i(0)$ and $K_{i+1} = P_i(1)$, we have:

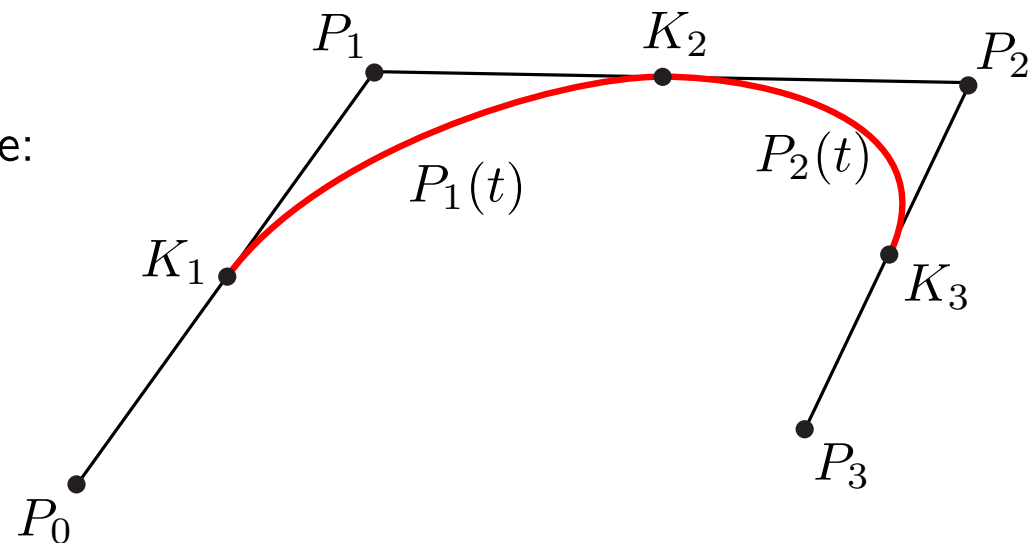
$$K_i = P_i(0) = \frac{1}{2}(P_{i-1} + P_i)$$

$$K_{i+1} = P_i(1) = \frac{1}{2}(P_i + P_{i+1})$$

Question: what are the tangent vectors at the ends?

$$P'_i(0) = P_i - P_{i-1}$$

$$P'_i(1) = P_{i+1} - P_i$$



More accurate picture

QUADRATIC UNIFORM B-SPLINES

Deriving the formula for the quadratic B-splines

$$\begin{aligned} P_i(t) &= \frac{1}{2}(t^2, t, 1) \begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \end{pmatrix}, i = 1, 2 \\ &= \frac{1}{2}(t^2 - 2t + 1)P_{i-1} + \frac{1}{2}(-2t^2 + 2t + 1)P_i + \frac{t^2}{2}P_{i+1} \end{aligned}$$

- Start and endpoints: K_i and K_{i+1}

Since $K_i = P_i(0)$ and $K_{i+1} = P_i(1)$, we have:

$$K_i = P_i(0) = \frac{1}{2}(P_{i-1} + P_i)$$

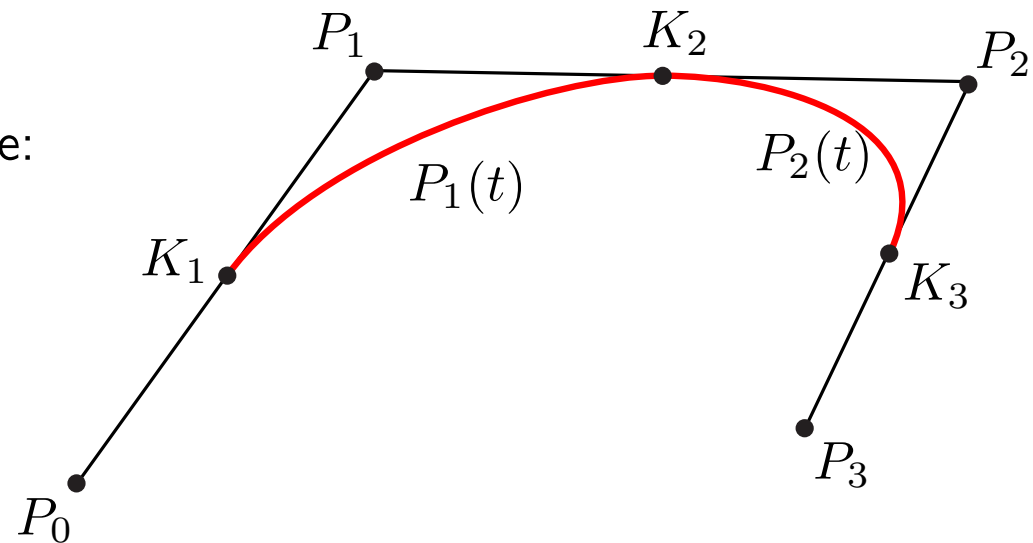
$$K_{i+1} = P_i(1) = \frac{1}{2}(P_i + P_{i+1})$$

Question: what are the tangent vectors at the ends?

$$P'_i(0) = P_i - P_{i-1}$$

$$P'_i(1) = P_{i+1} - P_i$$

Example: use control points
 $\{(1, 0), (1, 1), (2, 1), (2, 0)\}$



More accurate picture

CUBIC UNIFORM B-SPLINES

Deriving the formula for the cubic B-splines

CUBIC UNIFORM B-SPLINES

Deriving the formula for the cubic B-splines

Setting:

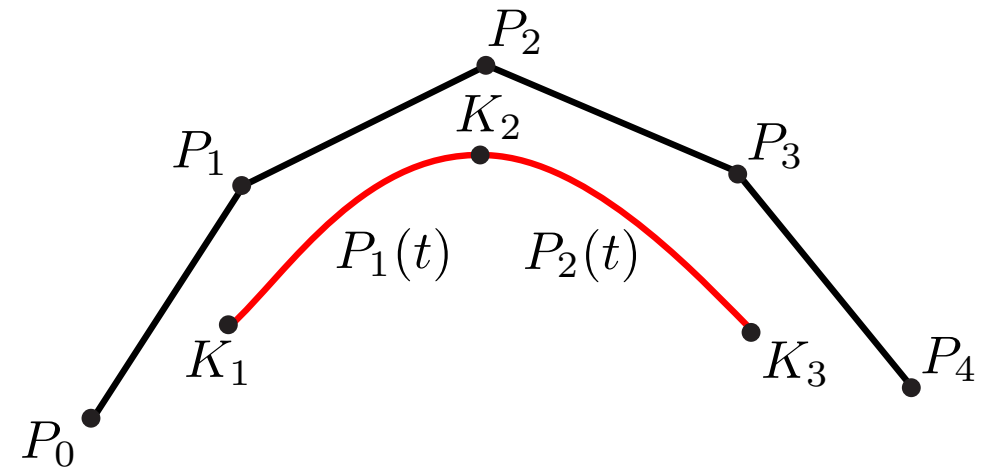
- Input: $n + 1$ control points P_0, \dots, P_n
- Output: spline curve where each segment $P_i(t)$ is a **cubic** parametric polynomial based on $P_{i-1}, P_i, P_{i+1}, P_{i+2}$

CUBIC UNIFORM B-SPLINES

Deriving the formula for the cubic B-splines

Setting:

- Input: $n + 1$ control points P_0, \dots, P_n
- Output: spline curve where each segment $P_i(t)$ is a **cubic** parametric polynomial based on $P_{i-1}, P_i, P_{i+1}, P_{i+2}$



sketch of setting for $n = 4$ (not accurate!)

CUBIC UNIFORM B-SPLINES

Deriving the formula for the cubic B-splines

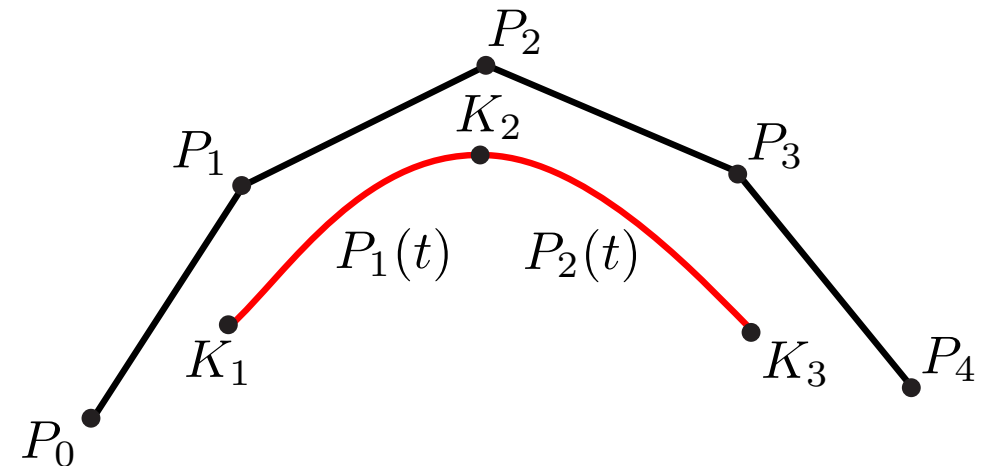
Setting:

- Input: $n + 1$ control points P_0, \dots, P_n
- Output: spline curve where each segment $P_i(t)$ is a **cubic** parametric polynomial based on $P_{i-1}, P_i, P_{i+1}, P_{i+2}$

$$P_i(t) = (t^3, t^2, t, 1)M \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$

Requirements:

1. Consecutive segments meet with C^2 -continuity
2. Entire curve is affine combination of control points



sketch of setting for $n = 4$ (not accurate!)

CUBIC UNIFORM B-SPLINES

Deriving the formula for the cubic B-splines

$$P_i(t) = (t^3, t^2, t, 1) \begin{pmatrix} a_3 & b_3 & c_3 & d_3 \\ a_2 & b_2 & c_2 & d_2 \\ a_1 & b_1 & c_1 & d_1 \\ a_0 & b_0 & c_0 & d_0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$

Requirements:

1. Consecutive segments meet with C^2 continuity
2. Entire curve is affine combination of control points

CUBIC UNIFORM B-SPLINES

Deriving the formula for the cubic B-splines

$$P_i(t) = (t^3, t^2, t, 1) \begin{pmatrix} a_3 & b_3 & c_3 & d_3 \\ a_2 & b_2 & c_2 & d_2 \\ a_1 & b_1 & c_1 & d_1 \\ a_0 & b_0 & c_0 & d_0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$

Requirements:

1. Consecutive segments meet with C^2 continuity
2. Entire curve is affine combination of control points

Matrix derived in similar way:

CUBIC UNIFORM B-SPLINES

Deriving the formula for the cubic B-splines

$$P_i(t) = (t^3, t^2, t, 1) \begin{pmatrix} a_3 & b_3 & c_3 & d_3 \\ a_2 & b_2 & c_2 & d_2 \\ a_1 & b_1 & c_1 & d_1 \\ a_0 & b_0 & c_0 & d_0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$

Requirements:

1. Consecutive segments meet with C^2 continuity
2. Entire curve is affine combination of control points

Matrix derived in similar way:

- Equations for equal endpoints, equal first and second derivatives at $t = 0$ and $t = 1$?

CUBIC UNIFORM B-SPLINES

Deriving the formula for the cubic B-splines

$$P_i(t) = (t^3, t^2, t, 1) \begin{pmatrix} a_3 & b_3 & c_3 & d_3 \\ a_2 & b_2 & c_2 & d_2 \\ a_1 & b_1 & c_1 & d_1 \\ a_0 & b_0 & c_0 & d_0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$

Requirements:

1. Consecutive segments meet with C^2 continuity
2. Entire curve is affine combination of control points

Matrix derived in similar way:

- Equations for equal endpoints, equal first and second derivatives at $t = 0$ and $t = 1$

15 equations

CUBIC UNIFORM B-SPLINES

Deriving the formula for the cubic B-splines

$$P_i(t) = (t^3, t^2, t, 1) \begin{pmatrix} a_3 & b_3 & c_3 & d_3 \\ a_2 & b_2 & c_2 & d_2 \\ a_1 & b_1 & c_1 & d_1 \\ a_0 & b_0 & c_0 & d_0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$

Requirements:

1. Consecutive segments meet with C^2 continuity
2. Entire curve is affine combination of control points

Matrix derived in similar way:

- Equations for equal endpoints, equal first and second derivatives at $t = 0$ and $t = 1$?
- Equations for affine combination

15 equations

CUBIC UNIFORM B-SPLINES

Deriving the formula for the cubic B-splines

$$P_i(t) = (t^3, t^2, t, 1) \begin{pmatrix} a_3 & b_3 & c_3 & d_3 \\ a_2 & b_2 & c_2 & d_2 \\ a_1 & b_1 & c_1 & d_1 \\ a_0 & b_0 & c_0 & d_0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$

Requirements:

1. Consecutive segments meet with C^2 continuity
2. Entire curve is affine combination of control points

Matrix derived in similar way:

- Equations for equal endpoints, equal first and second derivatives at $t = 0$ and $t = 1$
- Equations for affine combination

15 equations

4 equations

CUBIC UNIFORM B-SPLINES

Deriving the formula for the cubic B-splines

$$P_i(t) = (t^3, t^2, t, 1) \begin{pmatrix} a_3 & b_3 & c_3 & d_3 \\ a_2 & b_2 & c_2 & d_2 \\ a_1 & b_1 & c_1 & d_1 \\ a_0 & b_0 & c_0 & d_0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$

Requirements:

1. Consecutive segments meet with C^2 continuity
2. Entire curve is affine combination of control points

Matrix derived in similar way:

- Equations for equal endpoints, equal first and second derivatives at $t = 0$ and $t = 1$?
- Equations for affine combination

15 equations

4 equations

16 of them are
independent \implies
unique solution

CUBIC UNIFORM B-SPLINES

Formula for the cubic B-splines

CUBIC UNIFORM B-SPLINES

Formula for the cubic B-splines

$$P_i(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$

CUBIC UNIFORM B-SPLINES

Formula for the cubic B-splines

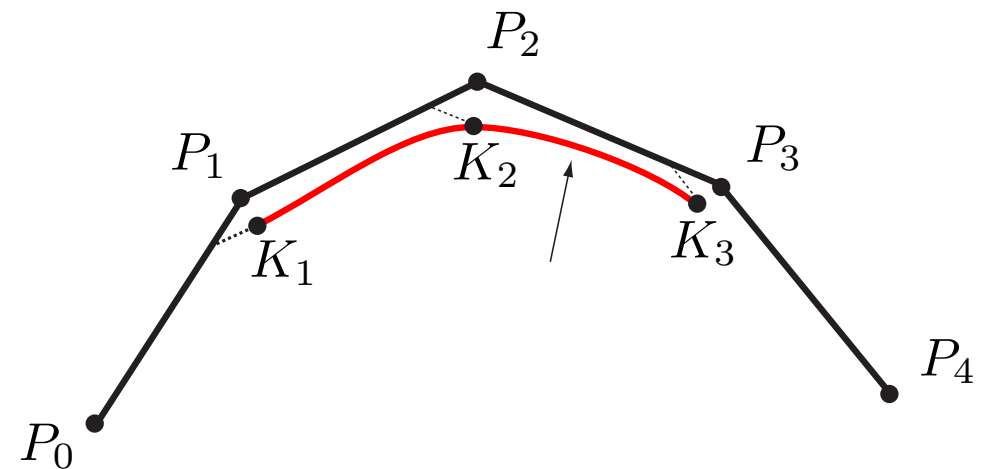
$$P_i(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$
$$= \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)P_{i-1} + \frac{1}{6}(3t^3 - 6t^2 + 4)P_i + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)P_{i+1} + \frac{t^3}{6}P_{i+2}$$

CUBIC UNIFORM B-SPLINES

Formula for the cubic B-splines

$$P_i(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$

$$= \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)P_{i-1} + \frac{1}{6}(3t^3 - 6t^2 + 4)P_i + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)P_{i+1} + \frac{t^3}{6}P_{i+2}$$

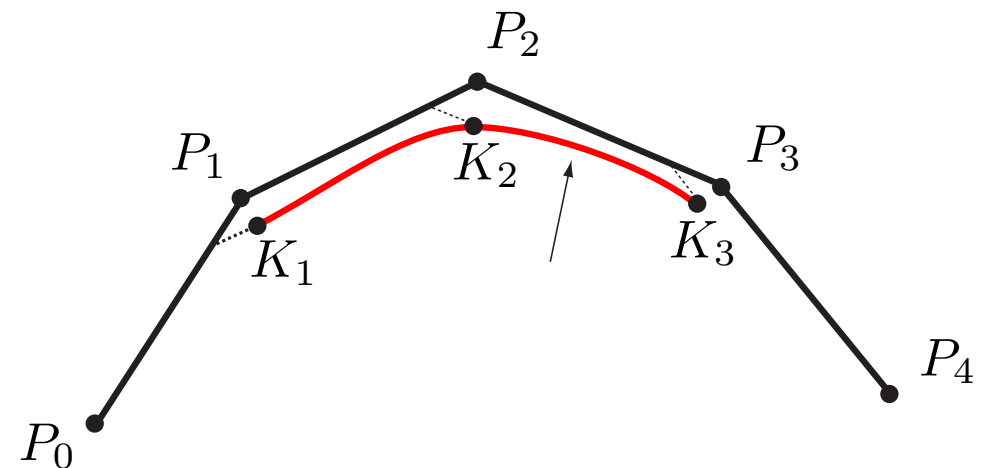


CUBIC UNIFORM B-SPLINES

Formula for the cubic B-splines

$$P_i(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$
$$= \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)P_{i-1} + \frac{1}{6}(3t^3 - 6t^2 + 4)P_i + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)P_{i+1} + \frac{t^3}{6}P_{i+2}$$

The two endpoints of each curve segment are



CUBIC UNIFORM B-SPLINES

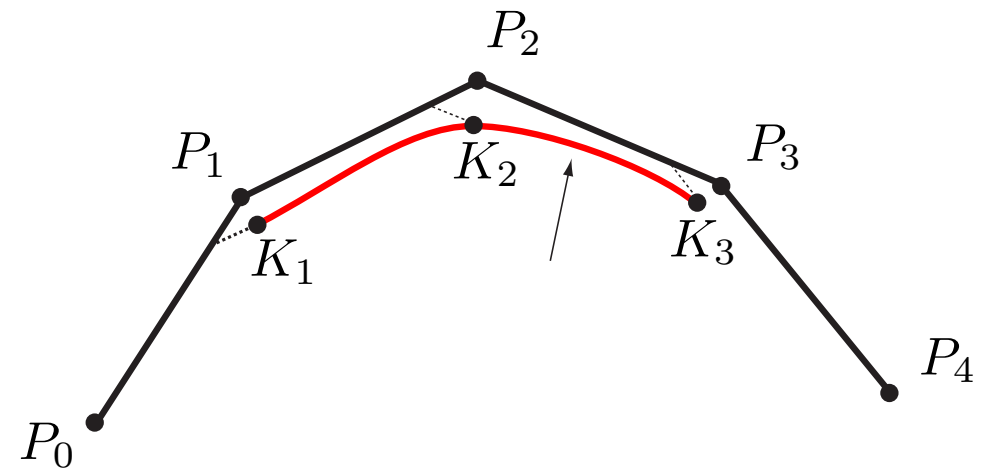
Formula for the cubic B-splines

$$P_i(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$
$$= \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)P_{i-1} + \frac{1}{6}(3t^3 - 6t^2 + 4)P_i + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)P_{i+1} + \frac{t^3}{6}P_{i+2}$$

The two endpoints of each curve segment are

$$K_i = P_i(0) = \frac{1}{6}(P_{i-1} + 4P_i + P_{i+1})$$

$$K_{i+1} = P_i(1) = \frac{1}{6}(P_i + 4P_{i+1} + P_{i+2})$$



CUBIC UNIFORM B-SPLINES

Formula for the cubic B-splines

$$P_i(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$
$$= \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)P_{i-1} + \frac{1}{6}(3t^3 - 6t^2 + 4)P_i + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)P_{i+1} + \frac{t^3}{6}P_{i+2}$$

The two endpoints of each curve segment are

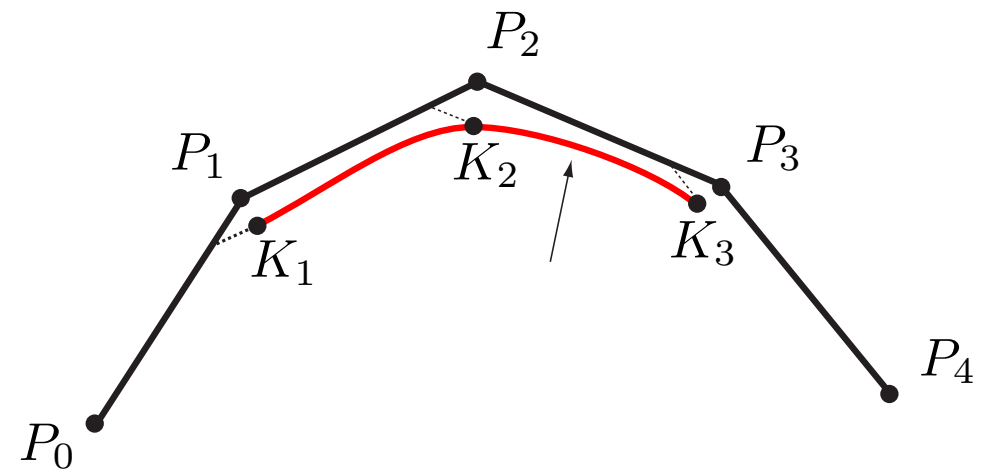
$$K_i = P_i(0) = \frac{1}{6}(P_{i-1} + 4P_i + P_{i+1})$$

$$K_{i+1} = P_i(1) = \frac{1}{6}(P_i + 4P_{i+1} + P_{i+2})$$

Geometrically, it makes more sense to rewrite as

$$K_i = \left(\frac{1}{6}P_{i-1} + \frac{5}{6}P_i\right) + \frac{1}{6}(P_{i+1} - P_i)$$

$$K_{i+1} = \left(\frac{1}{6}P_i + \frac{5}{6}P_{i+1}\right) + \frac{1}{6}(P_{i+2} - P_{i+1})$$



CUBIC UNIFORM B-SPLINES

Formula for the cubic B-splines

$$P_i(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$
$$= \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)P_{i-1} + \frac{1}{6}(3t^3 - 6t^2 + 4)P_i + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)P_{i+1} + \frac{t^3}{6}P_{i+2}$$

The two endpoints of each curve segment are

$$K_i = P_i(0) = \frac{1}{6}(P_{i-1} + 4P_i + P_{i+1})$$

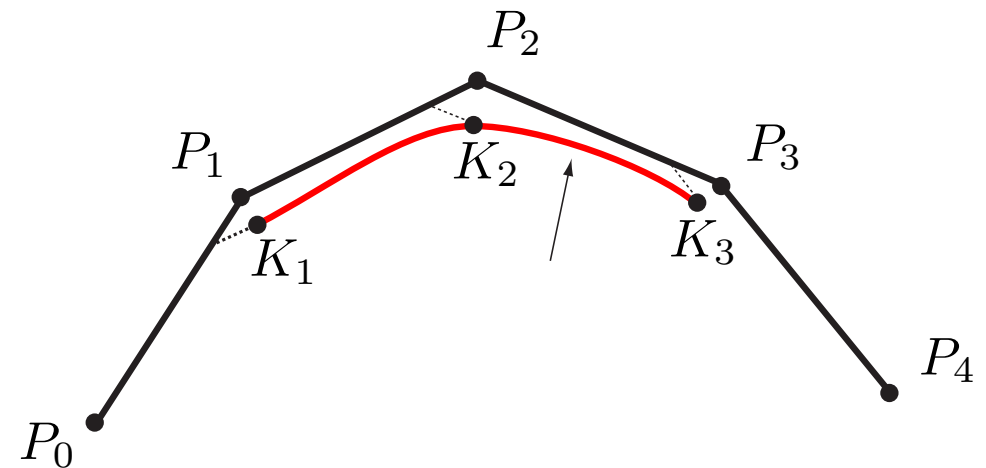
$$K_{i+1} = P_i(1) = \frac{1}{6}(P_i + 4P_{i+1} + P_{i+2})$$

Geometrically, it makes more sense to rewrite as

$$K_i = \left(\frac{1}{6}P_{i-1} + \frac{5}{6}P_i\right) + \frac{1}{6}(P_{i+1} - P_i)$$

$$K_{i+1} = \left(\frac{1}{6}P_i + \frac{5}{6}P_{i+1}\right) + \frac{1}{6}(P_{i+2} - P_{i+1})$$

Other geometric interpretations exist (e.g., $\frac{2}{3}$ rule)



CUBIC UNIFORM B-SPLINES

Making the curve go from P_0 to P_n

How can we force the curve go through P_0 and P_n ?

CUBIC UNIFORM B-SPLINES

Making the curve go from P_0 to P_n

How can we force the curve go through P_0 and P_n ?

Answer: add *dummy* control points P_{-1} and P_{n+1}

CUBIC UNIFORM B-SPLINES

Making the curve go from P_0 to P_n

How can we force the curve go through P_0 and P_n ?

Answer: add *dummy* control points P_{-1} and P_{n+1}

Question: what are P_{-1} and P_{n+1} points exactly?

CUBIC UNIFORM B-SPLINES

Making the curve go from P_0 to P_n

How can we force the curve go through P_0 and P_n ?

Answer: add *dummy* control points P_{-1} and P_{n+1}

Question: what are P_{-1} and P_{n+1} points exactly?

$$P_{-1} = 2P_0 - P_1 \text{ and } P_{n+1} = 2P_n - P_{n-1}$$

CUBIC UNIFORM B-SPLINES

Making the curve go from P_0 to P_n

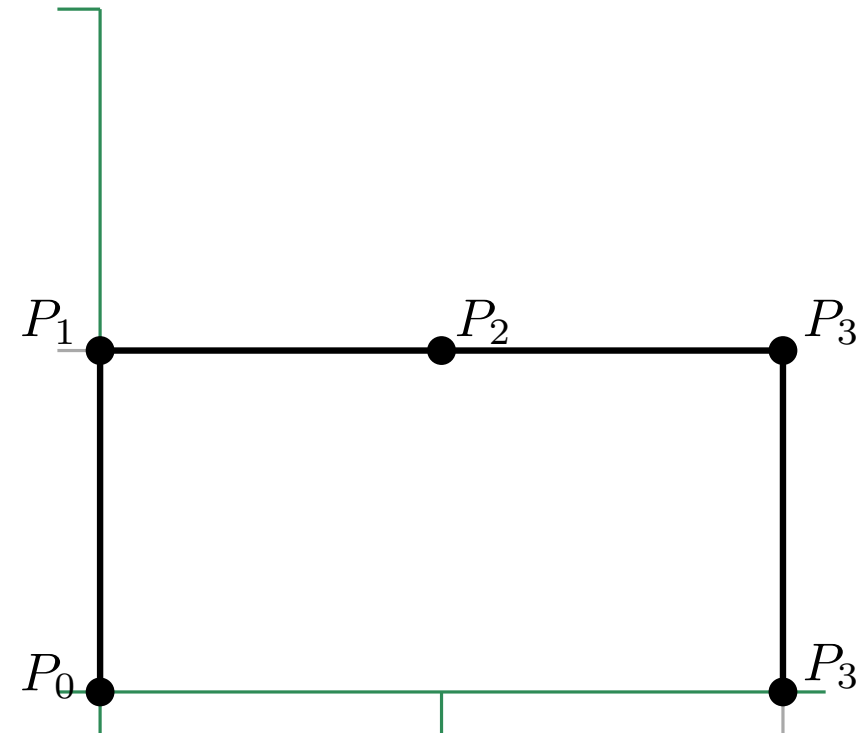
How can we force the curve go through P_0 and P_n ?

Answer: add *dummy* control points P_{-1} and P_{n+1}

Question: what are P_{-1} and P_{n+1} points exactly?

$$P_{-1} = 2P_0 - P_1 \text{ and } P_{n+1} = 2P_n - P_{n-1}$$

Example: use control points
 $\{(0, 0), (0, 1), (1, 1), (2, 1), (2, 0)\}$.
(i) Draw the first segment.
(ii) Add P_{-1} to fix beginning of curve at P_0 .



CUBIC UNIFORM B-SPLINES

Making the curve go from P_0 to P_n

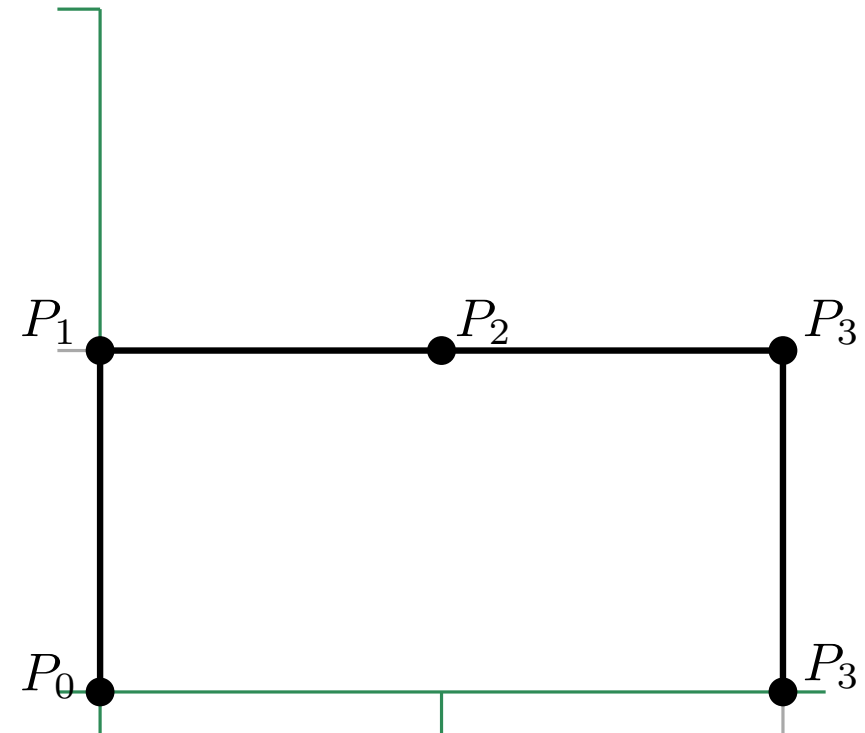
How can we force the curve go through P_0 and P_n ?

Answer: add *dummy* control points P_{-1} and P_{n+1}

Question: what are P_{-1} and P_{n+1} points exactly?

$$P_{-1} = 2P_0 - P_1 \text{ and } P_{n+1} = 2P_n - P_{n-1}$$

Example: use control points
 $\{(0, 0), (0, 1), (1, 1), (2, 1), (2, 0)\}$.
(i) Draw the first segment.
(ii) Add P_{-1} to fix beginning of curve at P_0 .



CUBIC UNIFORM B-SPLINES

Making the curve go from P_0 to P_n

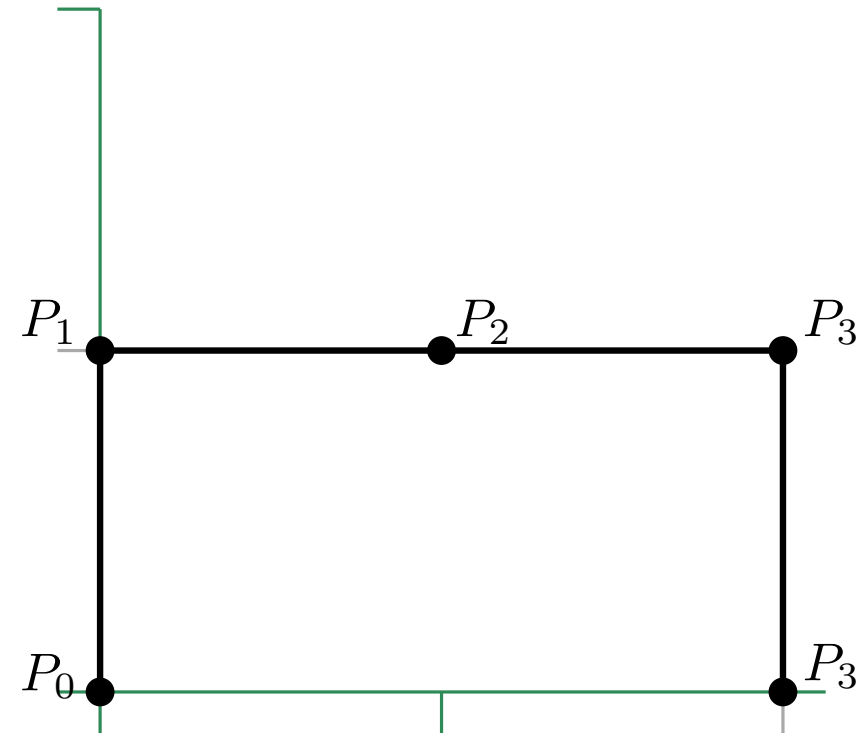
How can we force the curve go through P_0 and P_n ?

Answer: add *dummy* control points P_{-1} and P_{n+1}

Question: what are P_{-1} and P_{n+1} points exactly?

$$P_{-1} = 2P_0 - P_1 \text{ and } P_{n+1} = 2P_n - P_{n-1}$$

Example: use control points
 $\{(0, 0), (0, 1), (1, 1), (2, 1), (2, 0)\}$.
(i) Draw the first segment.
(ii) Add P_{-1} to fix beginning of curve at P_0 .



QUADRATIC VS. CUBIC

You can choose the order ($=\text{degree}+1$)

QUADRATIC VS. CUBIC

You can choose the order ($=\text{degree}+1$)

As opposed to Bézier curves, in B-splines the same $n + 1$ points can be used to construct curve of order 2 (linear), 3 (quadratic), 4 (cubic) ...

QUADRATIC VS. CUBIC

You can choose the order ($=\text{degree}+1$)

As opposed to Bézier curves, in B-splines the same $n + 1$ points can be used to construct curve of order 2 (linear), 3 (quadratic), 4 (cubic) ...

Example: use control points
 $\{(0, 0), (0, 1), (1, 1), (2, 1), (2, 0)\}$.

How would a quadratic
B-spline curve look like?

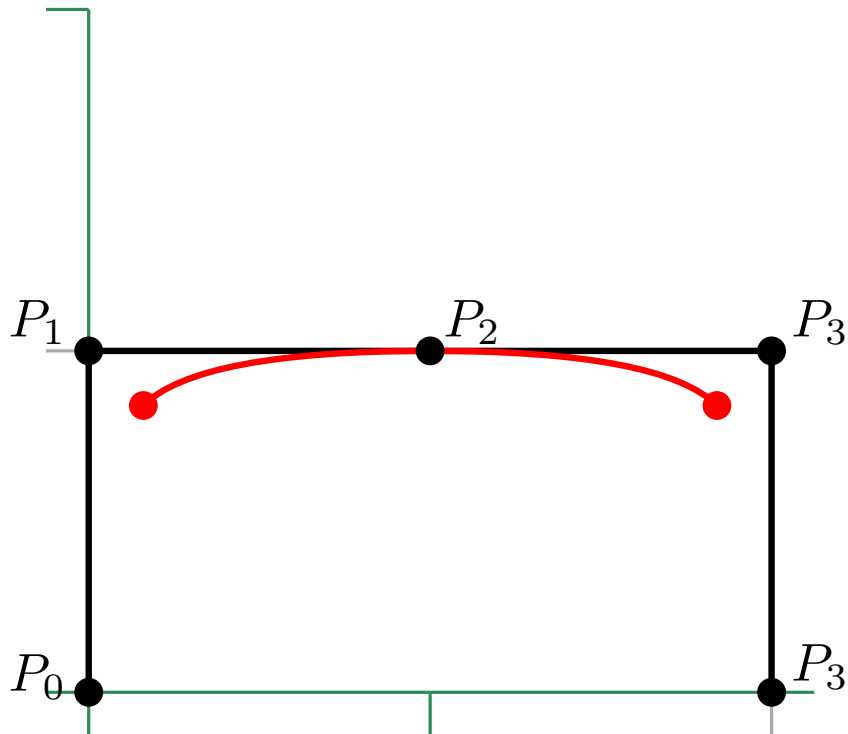
QUADRATIC VS. CUBIC

You can choose the order ($=\text{degree}+1$)

As opposed to Bézier curves, in B-splines the same $n + 1$ points can be used to construct curve of order 2 (linear), 3 (quadratic), 4 (cubic) ...

Example: use control points
 $\{(0, 0), (0, 1), (1, 1), (2, 1), (2, 0)\}$.

How would a quadratic B-spline curve look like?



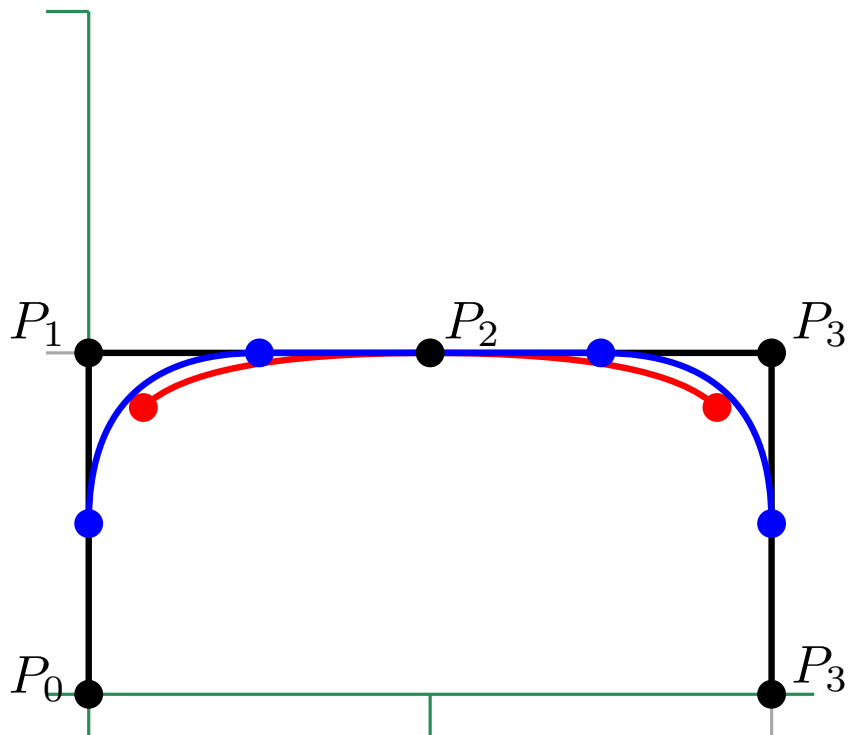
QUADRATIC VS. CUBIC

You can choose the order ($=\text{degree}+1$)

As opposed to Bézier curves, in B-splines the same $n + 1$ points can be used to construct curve of order 2 (linear), 3 (quadratic), 4 (cubic) ...

Example: use control points
 $\{(0, 0), (0, 1), (1, 1), (2, 1), (2, 0)\}$.

How would a quadratic B-spline curve look like?



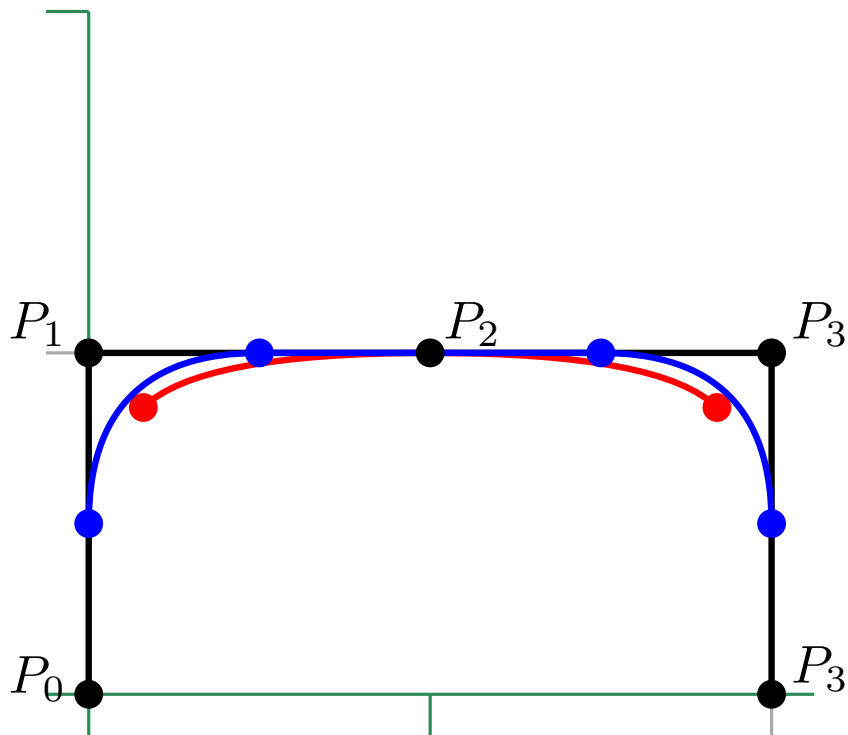
QUADRATIC VS. CUBIC

You can choose the order ($=\text{degree}+1$)

As opposed to Bézier curves, in B-splines the same $n + 1$ points can be used to construct curve of order 2 (linear), 3 (quadratic), 4 (cubic) ...

Example: use control points
 $\{(0, 0), (0, 1), (1, 1), (2, 1), (2, 0)\}$.

How would a quadratic
B-spline curve look like?



Quadratic
3 segments

Cubic
2 segments

INCREASING THE ORDER

Higher order, better fit

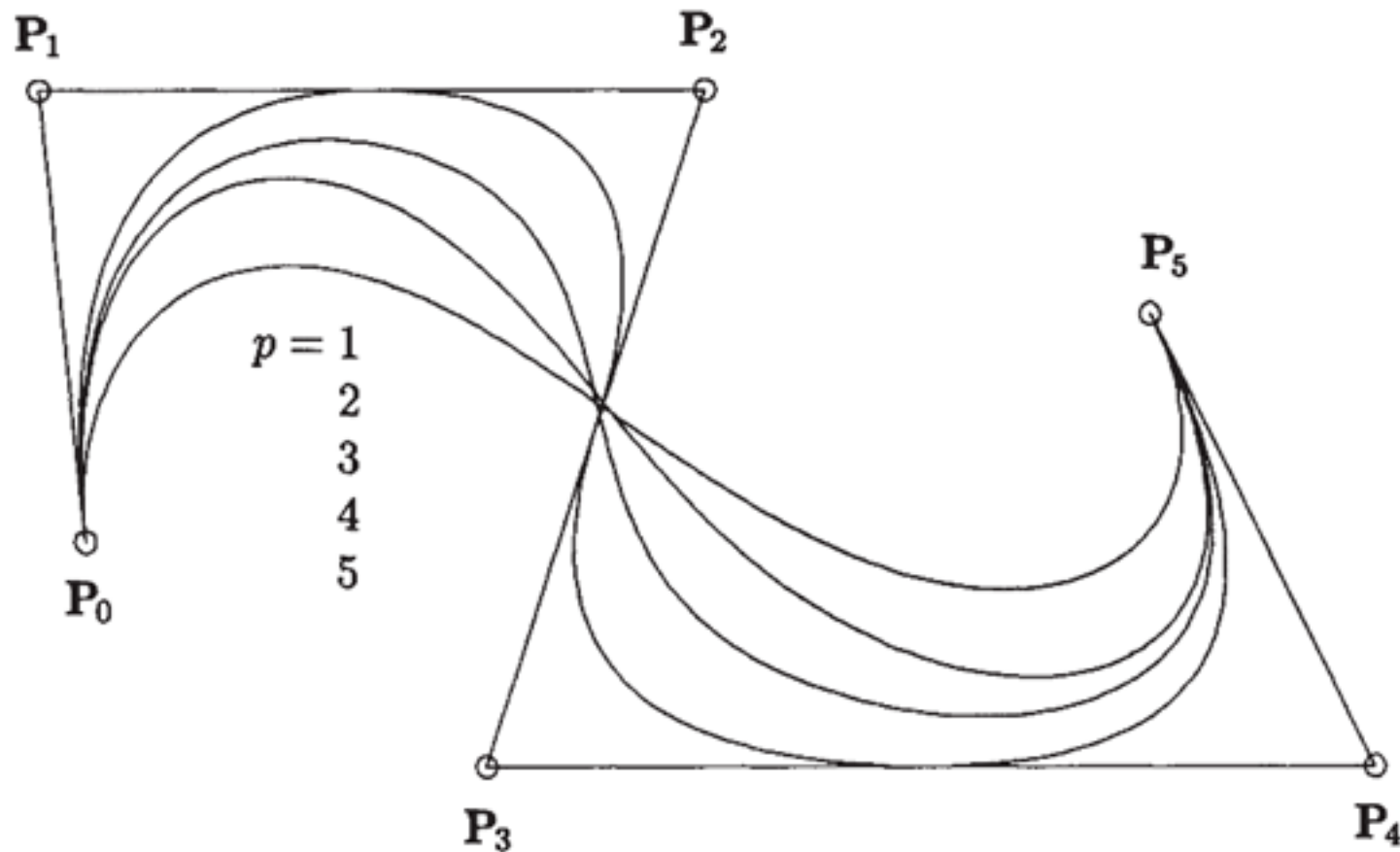


Figure 3.9. B-spline curves of different degree, using the same control polygon.

Where p is the degree of curve (i.e., $p = k - 1$)

Figure from [Piegl and Tiller]

B-SPLINE TO BEZIER

From a cubic B-Spline to a cubic Bézier curve

B-SPLINE TO BEZIER

From a cubic B-Spline to a cubic Bézier curve

Given a cubic B-spline segment $P(t)$ based on control points P_0, \dots, P_3 , how can we find a cubic Bézier curve $C(t)$ with the same shape?

B-SPLINE TO BEZIER

From a cubic B-Spline to a cubic Bézier curve

Given a cubic B-spline segment $P(t)$ based on control points P_0, \dots, P_3 , how can we find a cubic Bézier curve $C(t)$ with the same shape?

$$P(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

Matrix formulation of cubic B-spline

B-SPLINE TO BEZIER

From a cubic B-Spline to a cubic Bézier curve

Given a cubic B-spline segment $P(t)$ based on control points P_0, \dots, P_3 , how can we find a cubic Bézier curve $C(t)$ with the same shape?

$$P(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

Matrix formulation of cubic B-spline

$$C(t) = (t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} Q_0 \\ Q_1 \\ Q_2 \\ Q_3 \end{pmatrix}$$

Matrix formulation of cubic Bézier

B-SPLINE TO BEZIER

From a cubic B-Spline to a cubic Bézier curve

Given a cubic B-spline segment $P(t)$ based on control points P_0, \dots, P_3 , how can we find a cubic Bézier curve $C(t)$ with the same shape?

$$P(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

Matrix formulation of cubic B-spline

$$C(t) = (t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} Q_0 \\ Q_1 \\ Q_2 \\ Q_3 \end{pmatrix}$$

Matrix formulation of cubic Bézier

Unknowns

B-SPLINE TO BEZIER

From a cubic B-Spline to a cubic Bézier curve

Given a cubic B-spline segment $P(t)$ based on control points P_0, \dots, P_3 , how can we find a cubic Bézier curve $C(t)$ with the same shape?

$$P(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

Matrix formulation of cubic B-spline

$$C(t) = (t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} Q_0 \\ Q_1 \\ Q_2 \\ Q_3 \end{pmatrix}$$

Matrix formulation of cubic Bézier

Question: what is the solution to the problem?

Unknowns

B-SPLINE TO BEZIER

From a cubic B-Spline to a cubic Bézier curve

Given a cubic B-spline segment $P(t)$ based on control points P_0, \dots, P_3 , how can we find a cubic Bézier curve $C(t)$ with the same shape?

$$P(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

Matrix formulation of cubic B-spline

$$C(t) = (t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} Q_0 \\ Q_1 \\ Q_2 \\ Q_3 \end{pmatrix}$$

Matrix formulation of cubic Bézier

Question: what is the solution to the problem?

Unknowns

Solution: Solve $P(t) = C(t)$ for the unknowns Q_0, \dots, Q_3

B-SPLINE TO BEZIER

From a cubic B-Spline to a cubic Bézier curve

Given a cubic B-spline segment $P(t)$ based on control points P_0, \dots, P_3 , how can we find a cubic Bézier curve $C(t)$ with the same shape?

$$P(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

Matrix formulation of cubic B-spline

$$C(t) = (t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} Q_0 \\ Q_1 \\ Q_2 \\ Q_3 \end{pmatrix}$$

Matrix formulation of cubic Bézier

Question: what is the solution to the problem?

Solution: Solve $P(t) = C(t)$ for the unknowns Q_0, \dots, Q_3

Result:

$$\begin{aligned} Q_0 &= \frac{1}{6}(P_0 + 4P_1 + P_2), \\ Q_1 &= \frac{1}{6}(4P_1 + 2P_2), \\ Q_2 &= \frac{1}{6}(2P_1 + 4P_2), \\ Q_3 &= \frac{1}{6}(P_1 + 4P_2 + P_3) \end{aligned}$$

Unknowns

HIGHER ORDER B-SPLINE CURVES

B-splines of order higher than four

HIGHER ORDER B-SPLINE CURVES

B-splines of order higher than four

The formulas for quadratic and cubic uniform B-splines generalize to arbitrary order

HIGHER ORDER B-SPLINE CURVES

B-splines of order higher than four

The formulas for quadratic and cubic uniform B-splines generalize to arbitrary order

$$P_i(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$

Matrix formulation of cubic
B-spline segment $P_i(t)$

HIGHER ORDER B-SPLINE CURVES

B-splines of order higher than four

The formulas for quadratic and cubic uniform B-splines generalize to arbitrary order

$$P_i(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix} \quad \text{Matrix formulation of cubic B-spline segment } P_i(t)$$

$$P(t) = (t^n, \dots, t^2, t, 1) M_n \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ \vdots \\ P_{i+n-1} \end{pmatrix} \quad \text{Matrix formulation of degree-}n \text{ B-spline segment } P_i(t)$$

HIGHER ORDER B-SPLINE CURVES

B-splines of order higher than four

The formulas for quadratic and cubic uniform B-splines generalize to arbitrary order

$$P_i(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix} \quad \text{Matrix formulation of cubic B-spline segment } P_i(t)$$

$$P(t) = (t^n, \dots, t^2, t, 1) M_n \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ \vdots \\ P_{i+n-1} \end{pmatrix} \quad \text{Matrix formulation of degree-}n \text{ B-spline segment } P_i(t)$$

$$m_{ij} = \frac{1}{n!} \binom{n}{i} \sum_{k=j}^n (n-k)^i (-1)^{k-j} \binom{n+1}{k-j}$$

HIGHER ORDER B-SPLINE CURVES

B-splines of order higher than four

The formulas for quadratic and cubic uniform B-splines generalize to arbitrary order

$$P_i(t) = \frac{1}{6}(t^3, t^2, t, 1) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}$$

$$P(t) = (t^n, \dots, t^2, t, 1) M_n \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ \vdots \\ P_{i+n-1} \end{pmatrix} \quad \text{Matrix for B-spline s}$$

$$m_{ij} = \frac{1}{n!} \binom{n}{i} \sum_{k=j}^n (n-k)^i (-1)^{k-j} \binom{n+1}{k-j}$$

$$M_3 = \frac{1}{6} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix}$$

$$M_4 = \frac{1}{24} \begin{pmatrix} -1 & -4 & 6 & -4 & 1 \\ -4 & 12 & -12 & 4 & 0 \\ 6 & -6 & -6 & 6 & 0 \\ -4 & -12 & 12 & 4 & 0 \\ 1 & 11 & 11 & 1 & 0 \end{pmatrix}$$

$$M_5 = \frac{1}{120} \begin{pmatrix} -1 & 5 & -10 & 10 & -5 & 1 \\ 5 & -20 & 30 & -20 & 5 & 0 \\ -10 & 20 & 0 & -20 & 10 & 0 \\ 10 & 20 & -60 & 20 & 10 & 0 \\ -5 & -50 & 0 & 50 & 5 & 0 \\ 1 & 26 & 66 & 26 & 1 & 0 \end{pmatrix}$$

INTERPOLATING B-SPLINES

How to build an interpolating cubic B-spline curve

INTERPOLATING B-SPLINES

How to build an interpolating cubic B-spline curve

B-spline curve approximates control points. How can we make it interpolate?

INTERPOLATING B-SPLINES

How to build an interpolating cubic B-spline curve

B-spline curve approximates control points. How can we make it interpolate?

B-spline curve: given $n + 1$ given control points, produces curve through $n - 1$ joints K_j

INTERPOLATING B-SPLINES

How to build an interpolating cubic B-spline curve

B-spline curve approximates control points. How can we make it interpolate?

B-spline curve: given $n + 1$ given control points, produces curve through $n - 1$ joints K_j

We want: given $n + 1$ data points K_0, \dots, K_n , produce n -segment curve **through them**

INTERPOLATING B-SPLINES

How to build an interpolating cubic B-spline curve

B-spline curve approximates control points. How can we make it interpolate?

B-spline curve: given $n + 1$ given control points, produces curve through $n - 1$ joints K_j

We want: given $n + 1$ data points K_0, \dots, K_n , produce n -segment curve **through them**

To produce n segments, a cubic B-spline requires $n + 3$ control points, P_{-1}, \dots, P_{n+1}

INTERPOLATING B-SPLINES

How to build an interpolating cubic B-spline curve Unknowns

B-spline curve approximates control points. How can we make it interpolate?

B-spline curve: given $n + 1$ given control points, produces curve through $n - 1$ joints K_j

We want: given $n + 1$ data points K_0, \dots, K_n , produce n -segment curve **through them**

To produce n segments, a cubic B-spline requires $n + 3$ control points, P_{-1}, \dots, P_{n+1}

INTERPOLATING B-SPLINES

How to build an interpolating cubic B-spline curve Unknowns

B-spline curve approximates control points. How can we make it interpolate?

B-spline curve: given $n + 1$ given control points, produces curve through $n - 1$ joints K_j

We want: given $n + 1$ data points K_0, \dots, K_n , produce n -segment curve **through them**

To produce n segments, a cubic B-spline requires $n + 3$ control points, P_{-1}, \dots, P_{n+1}

Recall formula of cubic B-spline:

$$P_i(t) = \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)P_{i-1} + \frac{1}{6}(3t^3 - 6t^2 + 4)P_i + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)P_{i+1} + \frac{t^3}{6}P_{i+2}$$

INTERPOLATING B-SPLINES

How to build an interpolating cubic B-spline curve Unknowns

B-spline curve approximates control points. How can we make it interpolate?

B-spline curve: given $n + 1$ given control points, produces curve through $n - 1$ joints K_j

We want: given $n + 1$ data points K_0, \dots, K_n , produce n -segment curve **through them**

To produce n segments, a cubic B-spline requires $n + 3$ control points, P_{-1}, \dots, P_{n+1}

Recall formula of cubic B-spline:

$$P_i(t) = \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)P_{i-1} + \frac{1}{6}(3t^3 - 6t^2 + 4)P_i + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)P_{i+1} + \frac{t^3}{6}P_{i+2}$$

$$P_i(1) = \frac{1}{6}(P_i + 4P_{i+1} + P_{i+2})$$

INTERPOLATING B-SPLINES

How to build an interpolating cubic B-spline curve Unknowns

B-spline curve approximates control points. How can we make it interpolate?

B-spline curve: given $n + 1$ given control points, produces curve through $n - 1$ joints K_j

We want: given $n + 1$ data points K_0, \dots, K_n , produce n -segment curve **through them**

To produce n segments, a cubic B-spline requires $n + 3$ control points, P_{-1}, \dots, P_{n+1}

Recall formula of cubic B-spline:

$$P_i(t) = \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)P_{i-1} + \frac{1}{6}(3t^3 - 6t^2 + 4)P_i + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)P_{i+1} + \frac{t^3}{6}P_{i+2}$$

$$P_i(1) = \frac{1}{6}(P_i + 4P_{i+1} + P_{i+2}) = K_{i+1}$$

we require this

INTERPOLATING B-SPLINES

How to build an interpolating cubic B-spline curve Unknowns

B-spline curve approximates control points. How can we make it interpolate?

B-spline curve: given $n + 1$ given control points, produces curve through $n - 1$ joints K_j

We want: given $n + 1$ data points K_0, \dots, K_n , produce n -segment curve **through them**

To produce n segments, a cubic B-spline requires $n + 3$ control points, P_{-1}, \dots, P_{n+1}

Recall formula of cubic B-spline:

$$P_i(t) = \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)P_{i-1} + \frac{1}{6}(3t^3 - 6t^2 + 4)P_i + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)P_{i+1} + \frac{t^3}{6}P_{i+2}$$

$$P_i(1) = \frac{1}{6}(P_i + 4P_{i+1} + P_{i+2}) = K_{i+1}$$

we require this → this gives one equation for each K_j ($n + 1$ in total)

INTERPOLATING B-SPLINES

How to build an interpolating cubic B-spline curve Unknowns

B-spline curve approximates control points. How can we make it interpolate?

B-spline curve: given $n + 1$ given control points, produces curve through $n - 1$ joints K_j

We want: given $n + 1$ data points K_0, \dots, K_n , produce n -segment curve **through them**

To produce n segments, a cubic B-spline requires $n + 3$ control points, P_{-1}, \dots, P_{n+1}

Recall formula of cubic B-spline:

$$P_i(t) = \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)P_{i-1} + \frac{1}{6}(3t^3 - 6t^2 + 4)P_i + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)P_{i+1} + \frac{t^3}{6}P_{i+2}$$

$$P_i(1) = \frac{1}{6}(P_i + 4P_{i+1} + P_{i+2}) = K_{i+1}$$

we require this → this gives one equation for each K_j ($n + 1$ in total)

We need $n + 3$ equations, so we need two more: we fix extreme tangent vectors T_1 and T_n

INTERPOLATING B-SPLINES

How to build an interpolating cubic B-spline curve Unknowns

B-spline curve approximates control points. How can we make it interpolate?

B-spline curve: given $n + 1$ given control points, produces curve through $n - 1$ joints K_j

We want: given $n + 1$ data points K_0, \dots, K_n , produce n -segment curve **through them**

To produce n segments, a cubic B-spline requires $n + 3$ control points, P_{-1}, \dots, P_{n+1}

Recall formula of cubic B-spline:

$$P_i(t) = \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)P_{i-1} + \frac{1}{6}(3t^3 - 6t^2 + 4)P_i + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)P_{i+1} + \frac{t^3}{6}P_{i+2}$$

$$P_i(1) = \frac{1}{6}(P_i + 4P_{i+1} + P_{i+2}) = K_{i+1}$$

we require this → this gives one equation for each K_j ($n + 1$ in total)

We need $n + 3$ equations, so we need two more: we fix extreme tangent vectors T_1 and T_n

$$P'_0(0) = \frac{1}{2}(P_1 - P_{-1})$$

$$P'_n(1) = \frac{1}{2}(P_{n+1} - P_{n-1})$$

INTERPOLATING B-SPLINES

How to build an interpolating cubic B-spline curve Unknowns

B-spline curve approximates control points. How can we make it interpolate?

B-spline curve: given $n + 1$ given control points, produces curve through $n - 1$ joints K_j

We want: given $n + 1$ data points K_0, \dots, K_n , produce n -segment curve **through them**

To produce n segments, a cubic B-spline requires $n + 3$ control points, P_{-1}, \dots, P_{n+1}

Recall formula of cubic B-spline:

$$P_i(t) = \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)P_{i-1} + \frac{1}{6}(3t^3 - 6t^2 + 4)P_i + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)P_{i+1} + \frac{t^3}{6}P_{i+2}$$

$$P_i(1) = \frac{1}{6}(P_i + 4P_{i+1} + P_{i+2}) = K_{i+1}$$

we require this \longrightarrow this gives one equation for each K_j ($n + 1$ in total)

We need $n + 3$ equations, so we need two more: we fix extreme tangent vectors T_1 and T_n

$$P'_0(0) = \frac{1}{2}(P_1 - P_{-1}) = T_1$$
$$P'_n(1) = \frac{1}{2}(P_{n+1} - P_{n-1}) = T_n$$

two more equations (T_1 and T_2 given by user)

INTERPOLATING B-SPLINES

How to build an interpolating cubic B-spline curve Unknowns

B-spline curve approximates control points. How can we make it interpolate?

B-spline curve: given $n + 1$ given control points, produces curve through $n - 1$ joints K_j

We want: given $n + 1$ data points K_0, \dots, K_n , produce n -segment curve **through them**

To produce n segments, a cubic B-spline requires $n + 3$ control points, P_{-1}, \dots, P_{n+1}

Recall formula of cubic B-spline:

$$P_i(t) = \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)P_{i-1} + \frac{1}{6}(3t^3 - 6t^2 + 4)P_i + \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)P_{i+1} + \frac{t^3}{6}P_{i+2}$$

$$P_i(1) = \frac{1}{6}(P_i + 4P_{i+1} + P_{i+2}) = K_{i+1}$$

we require this \longrightarrow this gives one equation for each K_j ($n + 1$ in total)

We need $n + 3$ equations, so we need two more: we fix extreme tangent vectors T_1 and T_n

$$P'_0(0) = \frac{1}{2}(P_1 - P_{-1}) = T_1$$
$$P'_n(1) = \frac{1}{2}(P_{n+1} - P_{n-1}) = T_n$$

two more equations (T_1 and T_2 given by user)

System with $n + 3$ unknowns and $n + 3$ equations that (one can check) is nonsingular

KNOT VECTOR-BASED APPROACH

An different way to look at B-splines

KNOT VECTOR-BASED APPROACH

An different way to look at B-splines

Recall Bézier curves: $P(t) = \sum_{i=0}^n P_i B_{n,i}(t) \quad t \in [0, 1]$

KNOT VECTOR-BASED APPROACH

An different way to look at B-splines

Recall Bézier curves: $P(t) = \sum_{i=0}^n P_i B_{n,i}(t) \quad t \in [0, 1]$

We can do the same with B-splines! Approach based on using a *knot vector*

KNOT VECTOR-BASED APPROACH

An different way to look at B-splines

Recall Bézier curves: $P(t) = \sum_{i=0}^n P_i B_{n,i}(t) \quad t \in [0, 1]$

We can do the same with B-splines! Approach based on using a *knot vector*

For now, we consider **cubic uniform B-splines** (order 4)

└── Example: 5 control points, so two cubic segments

KNOT VECTOR-BASED APPROACH

An different way to look at B-splines

Recall Bézier curves: $P(t) = \sum_{i=0}^n P_i B_{n,i}(t) \quad t \in [0, 1]$

We can do the same with B-splines! Approach based on using a *knot vector*

For now, we consider **cubic uniform B-splines** (order 4)

└── Example: 5 control points, so two cubic segments

→ cubic = order 4

We need to find five weight (basis) functions $B_{4,0}(t), \dots, B_{4,4}(t)$

- In this approach we assume each cubic segment is defined for one interval of one unit $[u, u + 1]$
- Each integer value u is called a *knot* (the sequence of knot values is called *knot vector*)
- In *uniform* B-splines, knots are equally spaced, e.g., $0, 1, 2, \dots$

KNOT VECTOR-BASED APPROACH

An different way to look at B-splines

Recall Bézier curves: $P(t) = \sum_{i=0}^n P_i B_{n,i}(t) \quad t \in [0, 1]$

We can do the same with B-splines! Approach based on using a *knot vector*

For now, we consider **cubic uniform B-splines** (order 4)

└── Example: 5 control points, so two cubic segments

→ cubic = order 4

We need to find five weight (basis) functions $B_{4,0}(t), \dots, B_{4,4}(t)$

- In this approach we assume each cubic segment is defined for one interval of one unit $[u, u + 1]$
- Each integer value u is called a *knot* (the sequence of knot values is called *knot vector*)
- In *uniform* B-splines, knots are equally spaced, e.g., $0, 1, 2, \dots$

└── in our case, since there are two segments, the parameter for the first one lives in $[0, 1]$ and for the other in $[1, 2]$

KNOT VECTOR-BASED APPROACH

The cubic B-spline basis functions

The cubic B-spline basis functions

Each function should:

- Be a cubic polynomial
- Have its maximum near “its” control point
- Drop to zero away from “its” control point

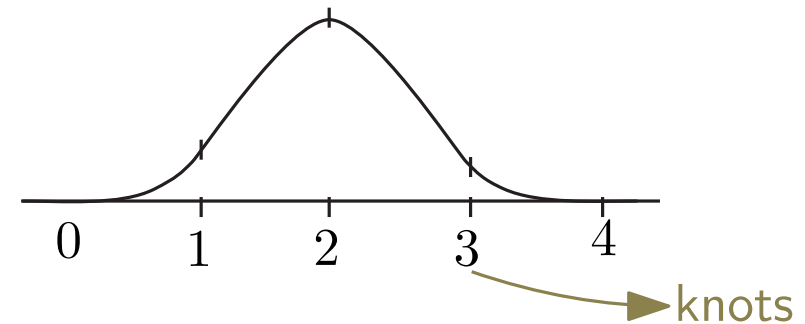
KNOT VECTOR-BASED APPROACH

The cubic B-spline basis functions

Each function should:

- Be a cubic polynomial
- Have its maximum near “its” control point
- Drop to zero away from “its” control point

Overall shape:



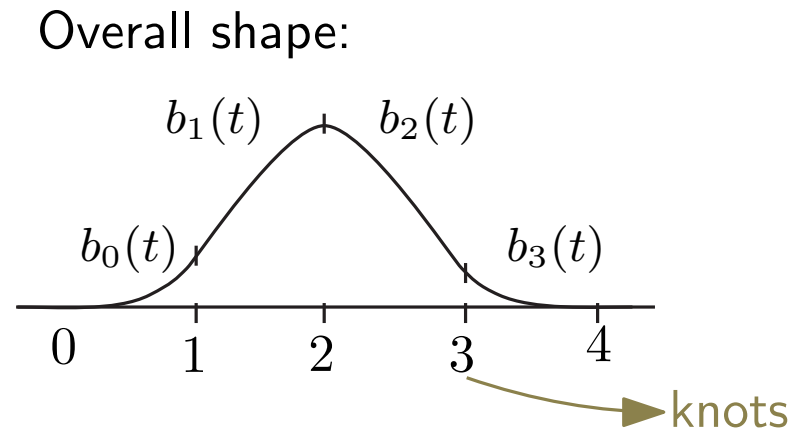
KNOT VECTOR-BASED APPROACH

The cubic B-spline basis functions

Each function should:

- Be a cubic polynomial
- Have it maximum near “its” control point
- Drop to zero away from “its” control point

We write the basis function as the union of **four parts**



KNOT VECTOR-BASED APPROACH

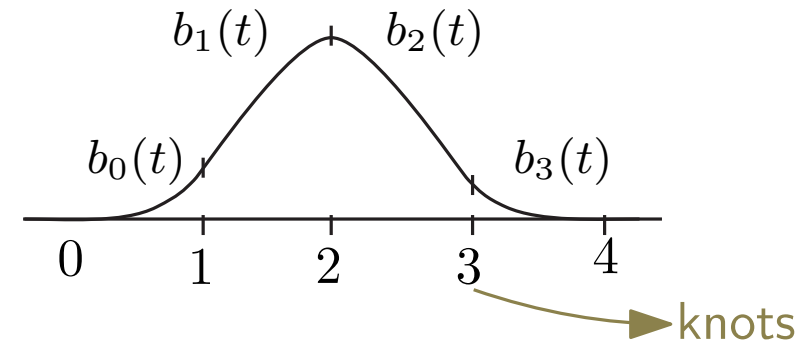
The cubic B-spline basis functions

Each function should:

- Be a cubic polynomial
- Have its maximum near “its” control point
- Drop to zero away from “its” control point

We write the basis function as the union of **four parts** —→ each $b_i(t)$ is a cubic polynomial defined over $[0, 1]$

Overall shape:



KNOT VECTOR-BASED APPROACH

The cubic B-spline basis functions

Each function should:

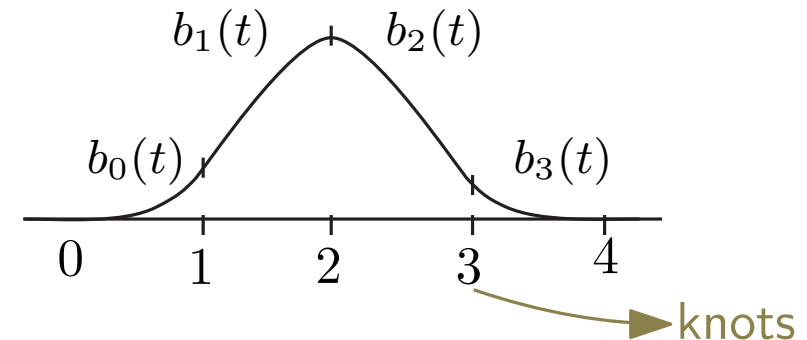
- Be a cubic polynomial
- Have it maximum near “its” control point
- Drop to zero away from “its” control point

We write the basis function as the union of **four parts** → each $b_i(t)$ is a cubic polynomial defined over $[0, 1]$

Conditions sought for the $b_i(t)$ functions:

- Affine invariant
- C^2 -continuous at three joints
- $b_0(t)$, $b'_0(t)$, $b''_0(t)$ should be zero at the start point $b_0(0)$
- $b_3(t)$, $b'_3(t)$, $b''_3(t)$ should be zero at the end point $b_3(1)$

Overall shape:



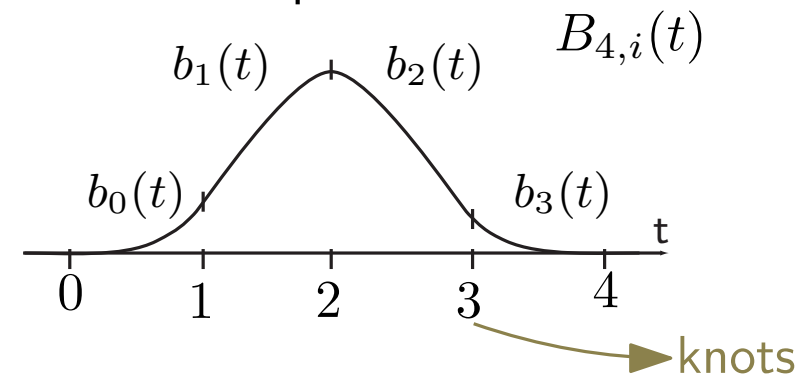
KNOT VECTOR-BASED APPROACH

The cubic B-spline basis functions

Each function should:

- Be a cubic polynomial
- Have it maximum near “its” control point
- Drop to zero away from “its” control point

Overall shape:



We write the basis function as the union of **four parts** → each $b_i(t)$ is a cubic polynomial defined over $[0, 1]$

Conditions sought for the $b_i(t)$ functions:

- Affine invariant
- C^2 -continuous at three joints
- $b_0(t)$, $b'_0(t)$, $b''_0(t)$ should be zero at the start point $b_0(0)$
- $b_3(t)$, $b'_3(t)$, $b''_3(t)$ should be zero at the end point $b_3(1)$

How many equations on the coefficients of the $b_i(t)$ functions do we obtain from these conditions?

The cubic B-spline basis functions

Solution to the equations:

$$b_0(t) = \frac{1}{6}t^3$$

$$b_1(t) = \frac{1}{6}(1 + 3t + 3t^2 - 3t^3)$$

$$b_2(t) = \frac{1}{6}(4 - 6t^2 + 3t^3)$$

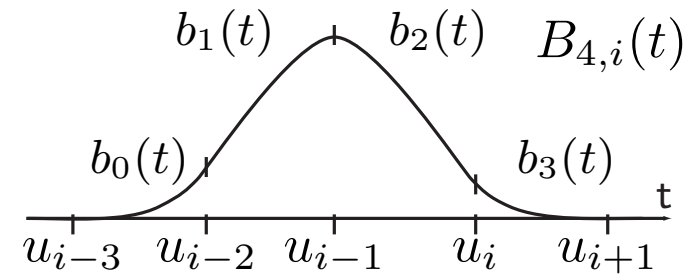
$$b_3(t) = \frac{1}{6}(1 - 3t + 3t^2 - t^3)$$

KNOT VECTOR-BASED APPROACH

The cubic B-spline basis functions

Solution to the equations:

$$\mathbf{N}_{i,4}(\mathbf{t}) = B_{4,i}(t) = \begin{cases} b_0(t) = \frac{1}{6}t^3 & u_{i-3} \leq t \leq u_{i-2} \\ b_1(t) = \frac{1}{6}(1 + 3t + 3t^2 - 3t^3) & u_{i-2} \leq t \leq u_{i-1} \\ b_2(t) = \frac{1}{6}(4 - 6t^2 + 3t^3) & u_{i-1} \leq t \leq u_i \\ b_3(t) = \frac{1}{6}(1 - 3t + 3t^2 - t^3) & u_i \leq t \leq u_{i+1} \end{cases}$$



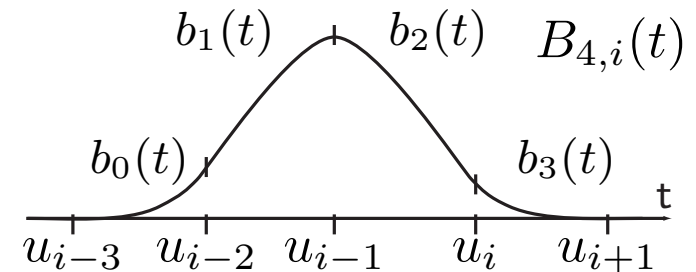
Each control point P_i gets multiplied by a shifted copy of the basis function:

KNOT VECTOR-BASED APPROACH

The cubic B-spline basis functions

Solution to the equations:

$$\mathbf{N}_{i,4}(\mathbf{t}) = B_{4,i}(t) = \begin{cases} b_0(t) = \frac{1}{6}t^3 & u_{i-3} \leq t \leq u_{i-2} \\ b_1(t) = \frac{1}{6}(1 + 3t + 3t^2 - 3t^3) & u_{i-2} \leq t \leq u_{i-1} \\ b_2(t) = \frac{1}{6}(4 - 6t^2 + 3t^3) & u_{i-1} \leq t \leq u_i \\ b_3(t) = \frac{1}{6}(1 - 3t + 3t^2 - t^3) & u_i \leq t \leq u_{i+1} \end{cases}$$



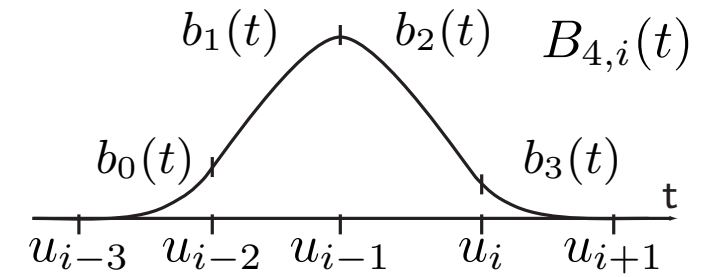
Each control point P_i gets multiplied by a shifted copy of the basis function: $P(t) = \sum_{i=0}^4 P_i B_{4,i}(t)$

KNOT VECTOR-BASED APPROACH

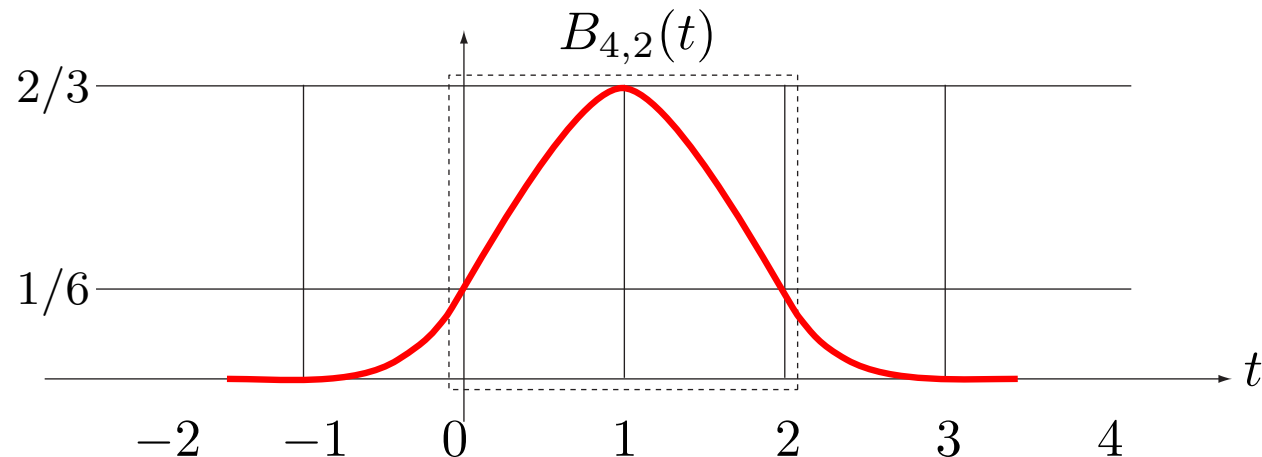
The cubic B-spline basis functions

Solution to the equations:

$$\mathbf{N}_{i,4}(\mathbf{t}) = \begin{cases} b_0(t) = \frac{1}{6}t^3 & u_{i-3} \leq t \leq u_{i-2} \\ b_1(t) = \frac{1}{6}(1 + 3t + 3t^2 - 3t^3) & u_{i-2} \leq t \leq u_{i-1} \\ b_2(t) = \frac{1}{6}(4 - 6t^2 + 3t^3) & u_{i-1} \leq t \leq u_i \\ b_3(t) = \frac{1}{6}(1 - 3t + 3t^2 - t^3) & u_i \leq t \leq u_{i+1} \end{cases}$$



Each control point P_i gets multiplied by a shifted copy of the basis function: $P(t) = \sum_{i=0}^4 P_i B_{4,i}(t)$

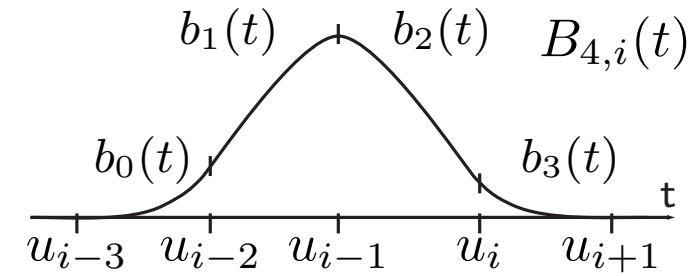


KNOT VECTOR-BASED APPROACH

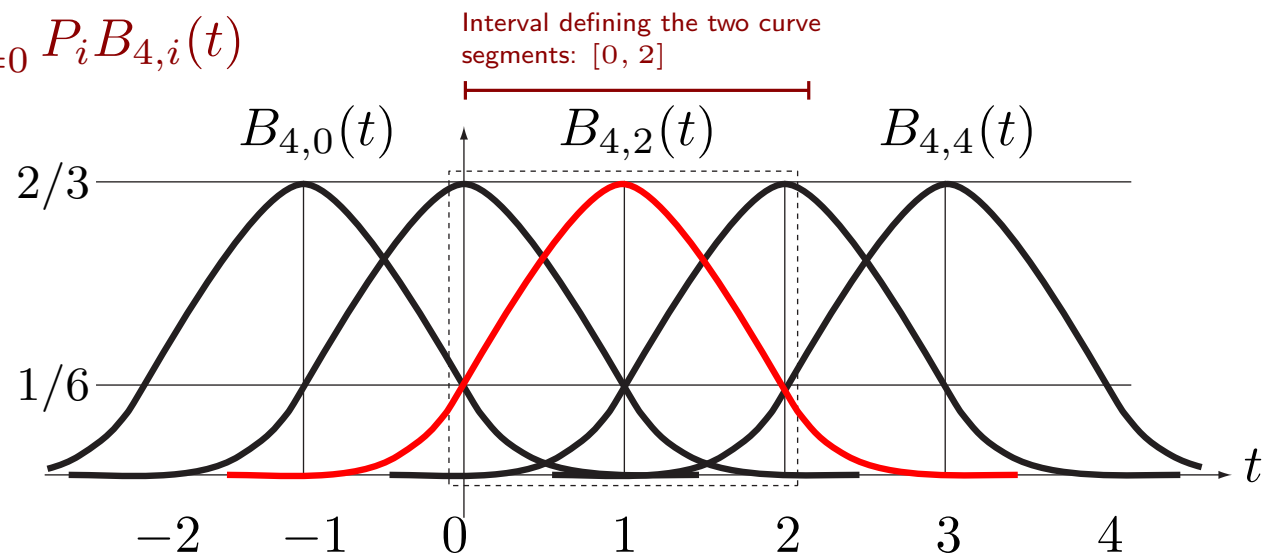
The cubic B-spline basis functions

Solution to the equations:

$$\mathbf{N}_{i,4}(\mathbf{t}) = B_{4,i}(t) = \begin{cases} b_0(t) = \frac{1}{6}t^3 & u_{i-3} \leq t \leq u_{i-2} \\ b_1(t) = \frac{1}{6}(1 + 3t + 3t^2 - 3t^3) & u_{i-2} \leq t \leq u_{i-1} \\ b_2(t) = \frac{1}{6}(4 - 6t^2 + 3t^3) & u_{i-1} \leq t \leq u_i \\ b_3(t) = \frac{1}{6}(1 - 3t + 3t^2 - t^3) & u_i \leq t \leq u_{i+1} \end{cases}$$



Each control point P_i gets multiplied by a shifted copy of the basis function: $P(t) = \sum_{i=0}^4 P_i B_{4,i}(t)$

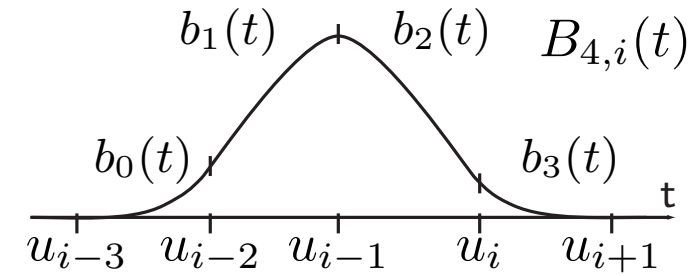


KNOT VECTOR-BASED APPROACH

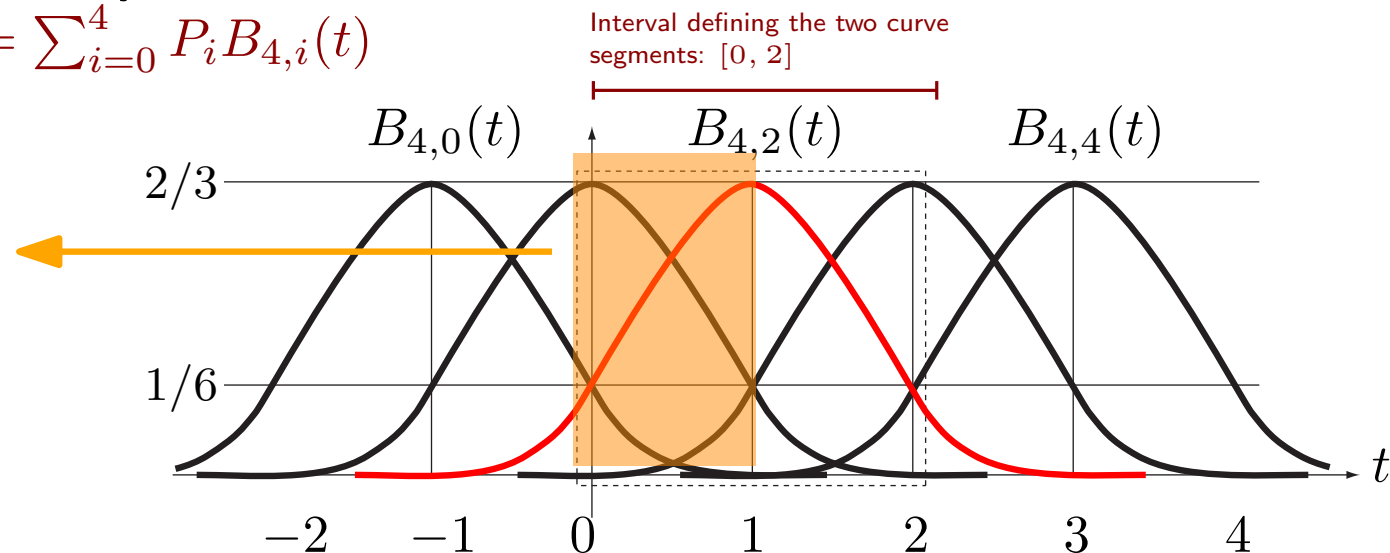
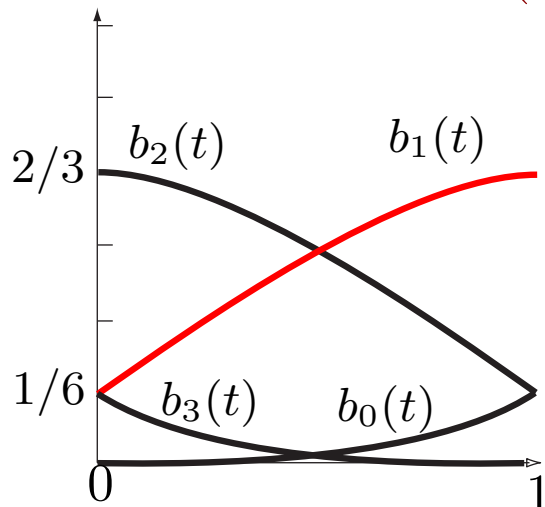
The cubic B-spline basis functions

Solution to the equations:

$$\mathbf{N}_{i,4}(\mathbf{t}) = B_{4,i}(t) = \begin{cases} b_0(t) = \frac{1}{6}t^3 & u_{i-3} \leq t \leq u_{i-2} \\ b_1(t) = \frac{1}{6}(1 + 3t + 3t^2 - 3t^3) & u_{i-2} \leq t \leq u_{i-1} \\ b_2(t) = \frac{1}{6}(4 - 6t^2 + 3t^3) & u_{i-1} \leq t \leq u_i \\ b_3(t) = \frac{1}{6}(1 - 3t + 3t^2 - t^3) & u_i \leq t \leq u_{i+1} \end{cases}$$



Each control point P_i gets multiplied by a shifted copy of the basis function: $P(t) = \sum_{i=0}^4 P_i B_{4,i}(t)$

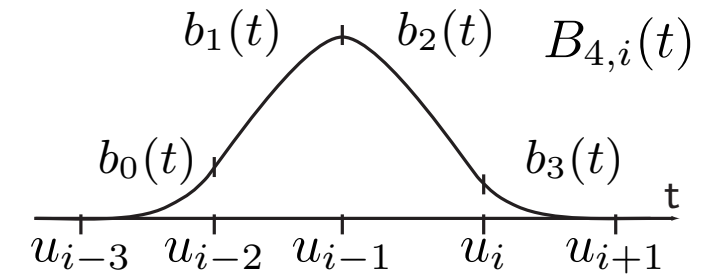


KNOT VECTOR-BASED APPROACH

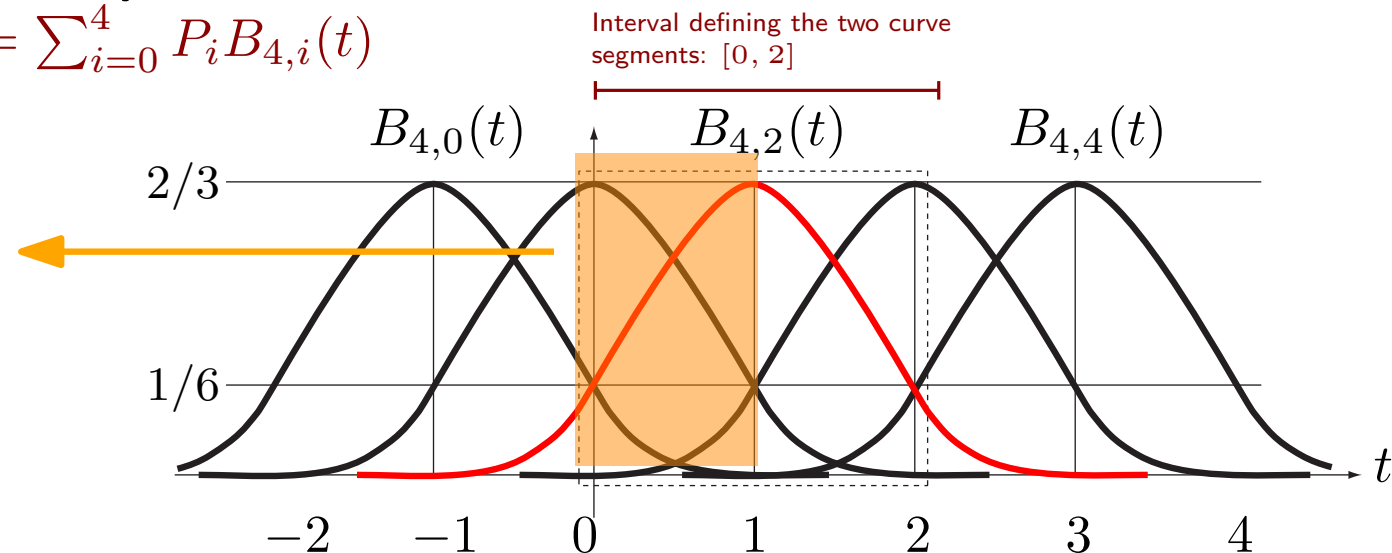
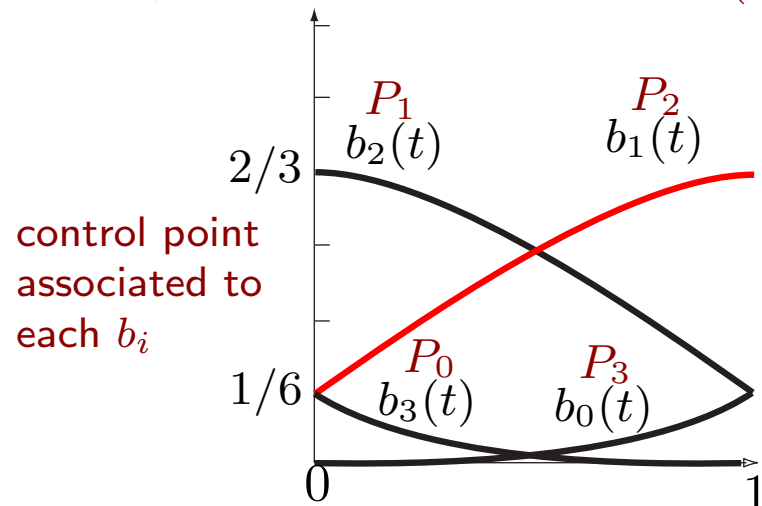
The cubic B-spline basis functions

Solution to the equations:

$$\mathbf{N}_{i,4}(\mathbf{t}) = \begin{cases} b_0(t) = \frac{1}{6}t^3 & u_{i-3} \leq t \leq u_{i-2} \\ b_1(t) = \frac{1}{6}(1 + 3t + 3t^2 - 3t^3) & u_{i-2} \leq t \leq u_{i-1} \\ b_2(t) = \frac{1}{6}(4 - 6t^2 + 3t^3) & u_{i-1} \leq t \leq u_i \\ b_3(t) = \frac{1}{6}(1 - 3t + 3t^2 - t^3) & u_i \leq t \leq u_{i+1} \end{cases}$$



Each control point P_i gets multiplied by a shifted copy of the basis function: $P(t) = \sum_{i=0}^4 P_i B_{4,i}(t)$



KNOT VECTOR-BASED APPROACH

General B-spline basis functions

This can be generalized to $(n + 1)$ control points and order k as follows:

KNOT VECTOR-BASED APPROACH

General B-spline basis functions

This can be generalized to $(n + 1)$ control points and order k as follows:

$$P(t) = \sum_{i=0}^n P_i N_{i,k}(t)$$

KNOT VECTOR-BASED APPROACH

General B-spline basis functions

This can be generalized to $(n + 1)$ control points and order k as follows:

$$P(t) = \sum_{i=0}^n P_i \boxed{N_{i,k}(t)} \rightarrow \text{B-spline basis function}$$

KNOT VECTOR-BASED APPROACH

General B-spline basis functions

This can be generalized to $(n + 1)$ control points and order k as follows:

$$P(t) = \sum_{i=0}^n P_i \boxed{N_{i,k}(t)} \rightarrow \text{B-spline basis function}$$

Given: control points P_0, \dots, P_n , knots: $t_0 \leq t_1 \leq \dots \leq t_{n+k}$, order: k

KNOT VECTOR-BASED APPROACH

General B-spline basis functions

This can be generalized to $(n + 1)$ control points and order k as follows:

$$P(t) = \sum_{i=0}^n P_i \boxed{N_{i,k}(t)} \rightarrow \text{B-spline basis function}$$

Given: control points P_0, \dots, P_n , knots: $t_0 \leq t_1 \leq \dots \leq t_{n+k}$, order: k

Note that # knots depends on # control points and order

KNOT VECTOR-BASED APPROACH

General B-spline basis functions

This can be generalized to $(n + 1)$ control points and order k as follows:

$$P(t) = \sum_{i=0}^n P_i \boxed{N_{i,k}(t)} \rightarrow \text{B-spline basis function}$$

Given: control points P_0, \dots, P_n , knots: $t_0 \leq t_1 \leq \dots \leq t_{n+k}$, order: k

Order 1 ($k = 1$, degree 0)

Note that # knots depends on # control points and order

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

KNOT VECTOR-BASED APPROACH

General B-spline basis functions

This can be generalized to $(n + 1)$ control points and order k as follows:

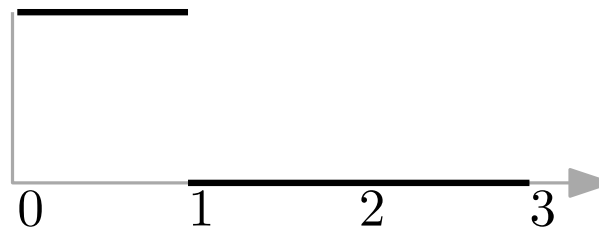
$$P(t) = \sum_{i=0}^n P_i \boxed{N_{i,k}(t)} \rightarrow \text{B-spline basis function}$$

Given: control points P_0, \dots, P_n , knots: $t_0 \leq t_1 \leq \dots \leq t_{n+k}$, order: k

Order 1 ($k = 1$, degree 0)

Note that # knots depends on # control points and order

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



$N_{0,1}(t)$

$t_i \mathbf{s} = [0, 1, 2, 3, \dots]$

KNOT VECTOR-BASED APPROACH

General B-spline basis functions

This can be generalized to $(n + 1)$ control points and order k as follows:

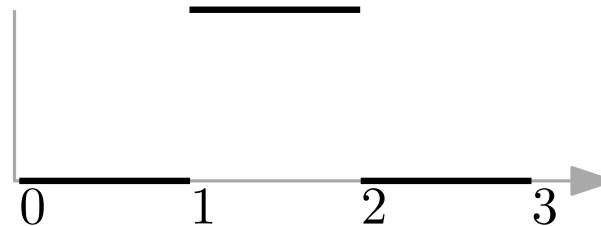
$$P(t) = \sum_{i=0}^n P_i \boxed{N_{i,k}(t)} \rightarrow \text{B-spline basis function}$$

Given: control points P_0, \dots, P_n , knots: $t_0 \leq t_1 \leq \dots \leq t_{n+k}$, order: k

Order 1 ($k = 1$, degree 0)

Note that # knots depends on # control points and order

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



$N_{1,1}(t)$

t_i s = $[0, 1, 2, 3, \dots]$

KNOT VECTOR-BASED APPROACH

General B-spline basis functions

This can be generalized to $(n + 1)$ control points and order k as follows:

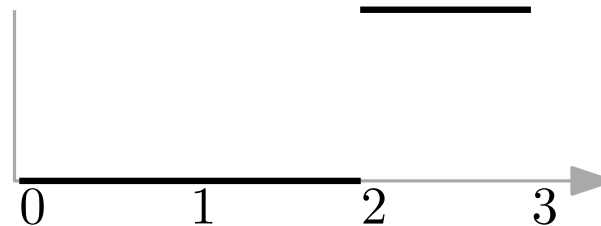
$$P(t) = \sum_{i=0}^n P_i \boxed{N_{i,k}(t)} \rightarrow \text{B-spline basis function}$$

Given: control points P_0, \dots, P_n , knots: $t_0 \leq t_1 \leq \dots \leq t_{n+k}$, order: k

Order 1 ($k = 1$, degree 0)

Note that # knots depends on # control points and order

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



$N_{2,1}(t)$

$t_i \mathbf{s} = [0, 1, 2, 3, \dots]$

KNOT VECTOR-BASED APPROACH

General B-spline basis functions

This can be generalized to $(n + 1)$ control points and order k as follows:

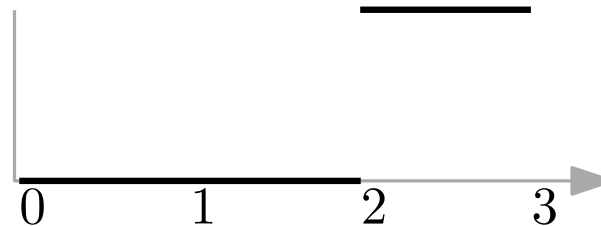
$$P(t) = \sum_{i=0}^n P_i \boxed{N_{i,k}(t)} \rightarrow \text{B-spline basis function}$$

Given: control points P_0, \dots, P_n , knots: $t_0 \leq t_1 \leq \dots \leq t_{n+k}$, order: k

Order 1 ($k = 1$, degree 0)

Note that # knots depends on # control points and order

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



$N_{2,1}(t)$
 t_i s = $[0, 1, 2, 3, \dots]$

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

(taking $0/0$ as 0)

KNOT VECTOR-BASED APPROACH

Examples of B-spline basis functions

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

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

KNOT VECTOR-BASED APPROACH

Examples of B-spline basis functions

$$N_{i,1} = \begin{cases} 1, & \text{if } t \in [t_i, t_{i+1}) \\ 0, & \text{otherwise} \end{cases} \quad N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t)$$

Order 2 ($k = 2$, degree 1)

Question: how do the basis functions look for order 2 ($k = 2$)? (for uniform knots)

KNOT VECTOR-BASED APPROACH

Examples of B-spline basis functions

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

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

Order 2 ($k = 2$, degree 1)

Question: how do the basis functions look for order 2 ($k = 2$)? (for uniform knots)

Order 3 ($k = 3$, degree 2)

KNOT VECTOR-BASED APPROACH

Examples of B-spline basis functions

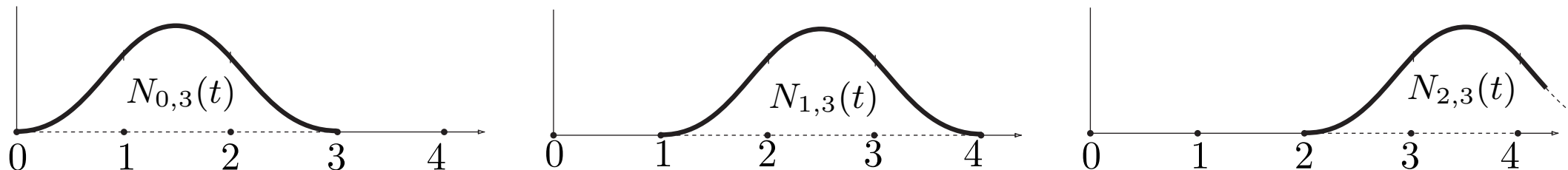
$$N_{i,1} = \begin{cases} 1, & \text{if } t \in [t_i, t_{i+1}) \\ 0, & \text{otherwise} \end{cases}$$

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

Order 2 ($k = 2$, degree 1)

Question: how do the basis functions look for order 2 ($k = 2$)? (for uniform knots)

Order 3 ($k = 3$, degree 2)



B-SPLINE BASIS FUNCTIONS

Some important properties

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

B-SPLINE BASIS FUNCTIONS

Some important properties

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

1) Shifted basis $N_{i,k}(t) = N_{0,k}(t - i)$

B-SPLINE BASIS FUNCTIONS

Some important properties

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

1) Shifted basis $N_{i,k}(t) = N_{0,k}(t - i)$

2) Local support $N_{i,k}(t) \neq 0$ only for $t \in [t_i, t_{i+k})$

B-SPLINE BASIS FUNCTIONS

Some important properties

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

1) Shifted basis $N_{i,k}(t) = N_{0,k}(t - i)$

2) Local support $N_{i,k}(t) \neq 0$ only for $t \in [t_i, t_{i+k})$

3) Non-zero bases for fixed t

In any interval $[t_j, t_{j+1})$, at most k of the $N_{i,k}$ are non-zero: $N_{(j-k+1),k}, \dots, N_{j,k}$

B-SPLINE BASIS FUNCTIONS

Some important properties

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

1) Shifted basis $N_{i,k}(t) = N_{0,k}(t - i)$

2) Local support $N_{i,k}(t) \neq 0$ only for $t \in [t_i, t_{i+k})$

3) Non-zero bases for fixed t

In any interval $[t_j, t_{j+1})$, at most k of the $N_{i,k}$ are non-zero: $N_{(j-k+1),k}, \dots, N_{j,k}$

4) Non-negativity $N_{i,k}(t) \geq 0$ for all i, k, t

B-SPLINE BASIS FUNCTIONS

Some important properties

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

1) Shifted basis $N_{i,k}(t) = N_{0,k}(t - i)$

2) Local support $N_{i,k}(t) \neq 0$ only for $t \in [t_i, t_{i+k})$

3) Non-zero bases for fixed t

In any interval $[t_j, t_{j+1})$, at most k of the $N_{i,k}$ are non-zero: $N_{(j-k+1),k}, \dots, N_{j,k}$

4) Non-negativity $N_{i,k}(t) \geq 0$ for all i, k, t

5) Affine invariance

For any $t \in [t_j, t_{j+1})$, $\sum_{i=0}^n N_{i,k}(t) = 1$

B-SPLINE BASIS FUNCTIONS

Some important properties

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

1) Shifted basis $N_{i,k}(t) = N_{0,k}(t - i)$

2) Local support $N_{i,k}(t) \neq 0$ only for $t \in [t_i, t_{i+k})$

3) Non-zero bases for fixed t

In any interval $[t_j, t_{j+1})$, at most k of the $N_{i,k}$ are non-zero: $N_{(j-k+1),k}, \dots, N_{j,k}$

4) Non-negativity $N_{i,k}(t) \geq 0$ for all i, k, t

5) Affine invariance

For any $t \in [t_j, t_{j+1})$, $\sum_{i=0}^n N_{i,k}(t) = 1$

6) Continuity

For uniform knots, the curve and its $k - 1$ derivatives are continuous
(Non-uniform B-Splines can have discontinuities at knot values!)

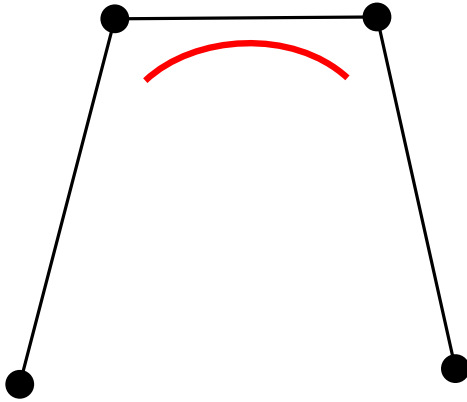
UNDERSTANDING KNOT VECTORS

Open (or clamped) uniform B-Splines

UNDERSTANDING KNOT VECTORS

Open (or clamped) uniform B-Splines

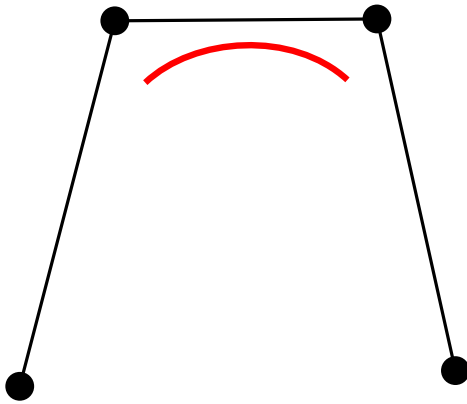
Uniform knot vector except at ends: at the beginning and end knot values are repeated k times



UNDERSTANDING KNOT VECTORS

Open (or clamped) uniform B-Splines

Uniform knot vector except at ends: at the beginning and end knot values are repeated k times



$n = 3$ (4 control points)

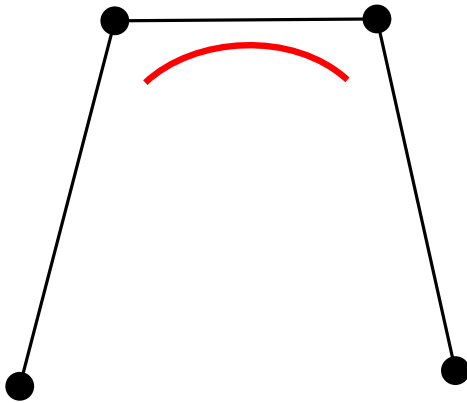
$k = 4$ (cubic B-spline)

uniform knot vector: $(0, 1/7, 2/7, 3/7, 4/7, 5/7, 6/7, 1)$

UNDERSTANDING KNOT VECTORS

Open (or clamped) uniform B-Splines

Uniform knot vector except at ends: at the beginning and end knot values are repeated k times



$n = 3$ (4 control points)

$k = 4$ (cubic B-spline)

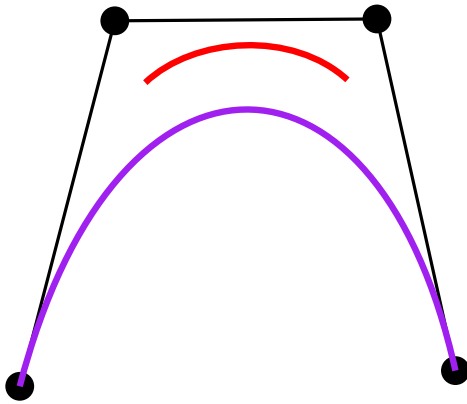
uniform knot vector: $(0, 1/7, 2/7, 3/7, 4/7, 5/7, 6/7, 1)$

“open” knot vector: $(0, 0, 0, 0, 1, 1, 1, 1)$

UNDERSTANDING KNOT VECTORS

Open (or clamped) uniform B-Splines

Uniform knot vector except at ends: at the beginning and end knot values are repeated k times



$n = 3$ (4 control points)

$k = 4$ (cubic B-spline)

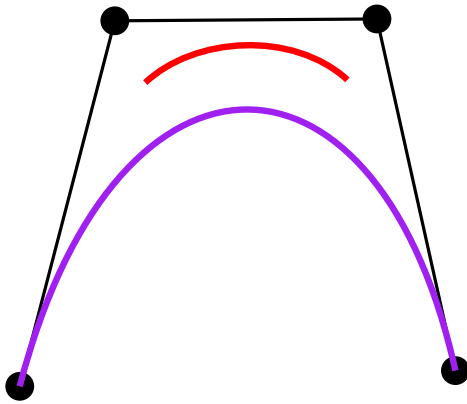
uniform knot vector: $(0, 1/7, 2/7, 3/7, 4/7, 5/7, 6/7, 1)$

“open” knot vector: $(0, 0, 0, 0, 1, 1, 1, 1)$

UNDERSTANDING KNOT VECTORS

Open (or clamped) uniform B-Splines

Uniform knot vector except at ends: at the beginning and end knot values are repeated k times



$n = 3$ (4 control points)

$k = 4$ (cubic B-spline)

uniform knot vector: $(0, 1/7, 2/7, 3/7, 4/7, 5/7, 6/7, 1)$

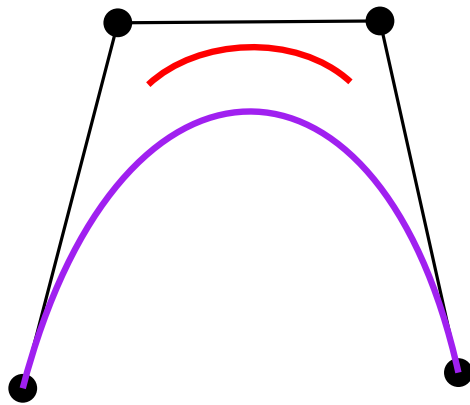
“open” knot vector: $(0, 0, 0, 0, 1, 1, 1, 1)$

Cubic Bézier curve?

UNDERSTANDING KNOT VECTORS

Open (or clamped) uniform B-Splines

Uniform knot vector except at ends: at the beginning and end knot values are repeated k times



$n = 3$ (4 control points)

$k = 4$ (cubic B-spline)

uniform knot vector: $(0, 1/7, 2/7, 3/7, 4/7, 5/7, 6/7, 1)$

“open” knot vector: $(0, 0, 0, 0, 1, 1, 1, 1)$

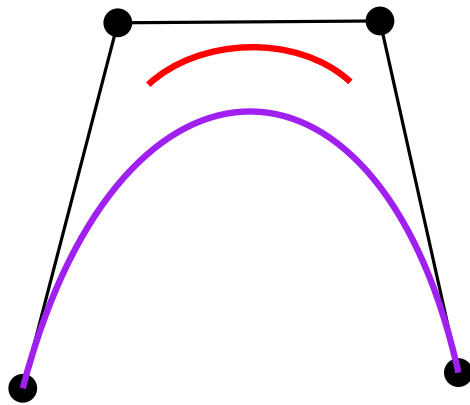
Cubic Bézier curve?

Example: compute open basis functions for
 $n = 2$ and $k = 3$ (quadratic) B-splines

UNDERSTANDING KNOT VECTORS

Open (or clamped) uniform B-Splines

Uniform knot vector except at ends: at the beginning and end knot values are repeated k times



$n = 3$ (4 control points)

$k = 4$ (cubic B-spline)

uniform knot vector: $(0, 1/7, 2/7, 3/7, 4/7, 5/7, 6/7, 1)$

“open” knot vector: $(0, 0, 0, 0, 1, 1, 1, 1)$

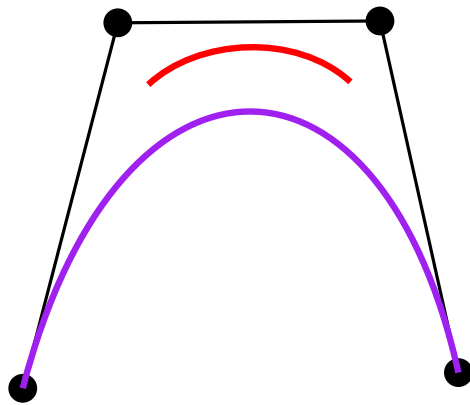
Cubic Bézier curve? \longrightarrow **Always the case when $k = n + 1$**

Example: compute open basis functions for
 $n = 2$ and $k = 3$ (quadratic) B-splines

UNDERSTANDING KNOT VECTORS

Open (or clamped) uniform B-Splines

Uniform knot vector except at ends: at the beginning and end knot values are repeated k times



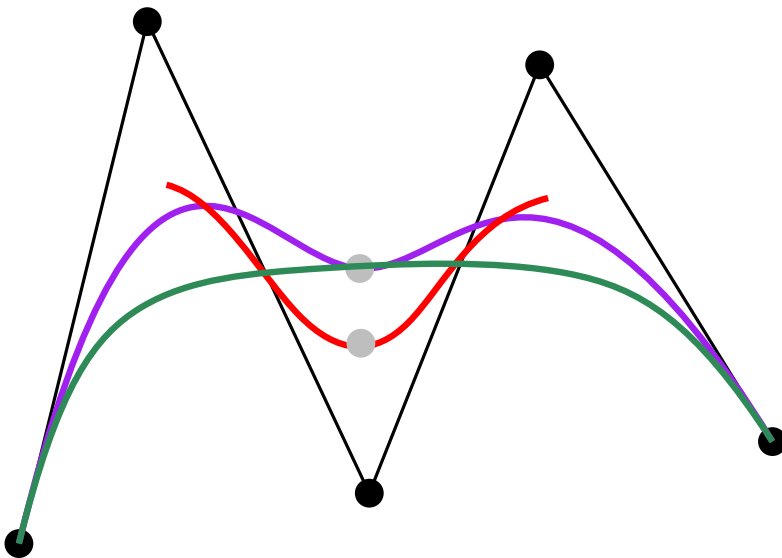
$n = 3$ (4 control points)

$k = 4$ (cubic B-spline)

uniform knot vector: $(0, 1/7, 2/7, 3/7, 4/7, 5/7, 6/7, 1)$

“open” knot vector: $(0, 0, 0, 0, 1, 1, 1, 1)$

Cubic Bézier curve?



$n = 4$ (5 control points)

$k = 4$ (cubic B-spline)

uniform knot vector: $(0, 1/8, 2/8, 3/8, 4/8, 5/8, 6/8, 7/8, 1)$

“open” knot vector: $(0, 0, 0, 0, 0.5, 1, 1, 1, 1)$

degree-4 Bézier—knot vector: $(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)$

Open uniform B-spline curves always start at P_0 and end at P_n . Tangents are also like in Bézier curves

UNDERSTANDING KNOT VECTORS

Example for quadratic open B-Splines

Example: compute basis functions for 5 control points ($n = 4$) and $k = 3$ (i.e., quadratic open B-splines)

UNDERSTANDING KNOT VECTORS

Example for quadratic open B-Splines

Example: compute basis functions for 5 control points ($n = 4$) and $k = 3$ (i.e., quadratic open B-splines)

Recall:

- knot vector: $(0, 0, 0, 1, 2, 3, 3, 3)$
- t goes from $t_{k-1} = t_2 = 0$ to $t_{n+1} = t_5 = 3$
- need to compute 5 bases: $N_{0,3}(t)$ to $N_{4,3}(t)$

$$N_{i,1} = \begin{cases} 1, & \text{if } t \in [t_i, t_{i+1}) \\ 0, & \text{otherwise} \end{cases} \quad N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t)$$

UNDERSTANDING KNOT VECTORS

More examples

Bézier vs open B-Spline of order 3
where $n = 9$ and $k = 3$

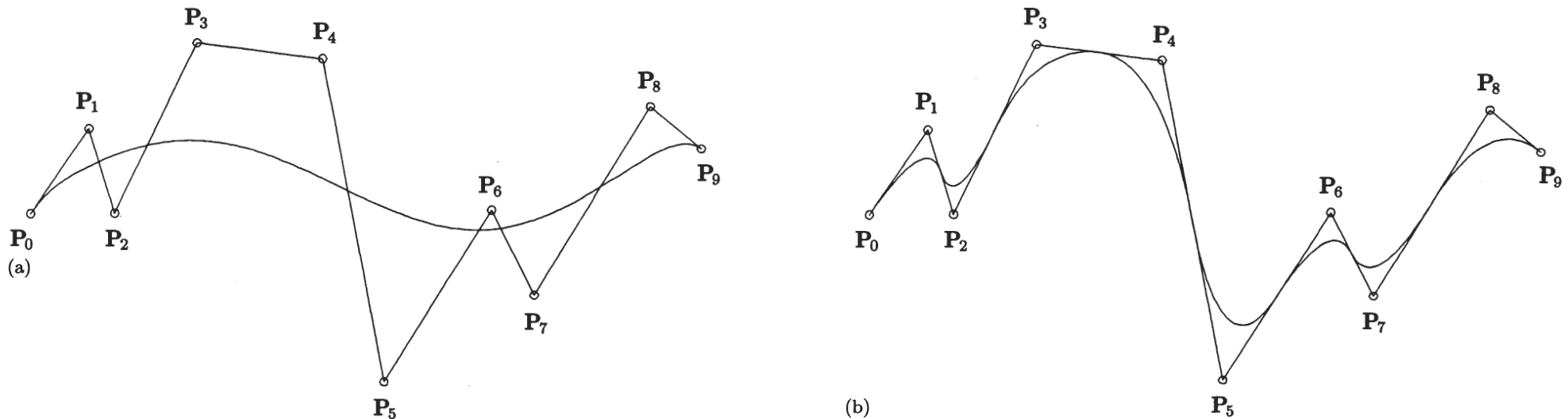


Figure 3.8. B-spline curves. (a) A ninth-degree Bézier curve on the knot vector $U = \{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$; (b) a quadratic curve using the same control polygon defined on $U = \{0, 0, 0, 1/8, 2/8, 3/8, 4/8, 5/8, 6/8, 7/8, 1, 1, 1\}$.

NON-UNIFORM B-SPLINES

When knots are *not* equally spaced

NON-UNIFORM B-SPLINES

When knots are *not* equally spaced

└─▶ ignoring the first and last k knots, in case of open b-splines

NON-UNIFORM B-SPLINES

When knots are *not* equally spaced

└─▶ ignoring the first and last k knots, in case of open b-splines

- Only restriction: non-decreasing knots

NON-UNIFORM B-SPLINES

When knots are *not* equally spaced

└─▶ ignoring the first and last k knots, in case of open b-splines

- Only restriction: non-decreasing knots
- Knots can have multiplicity larger than one

NON-UNIFORM B-SPLINES

When knots are *not* equally spaced

└─▶ ignoring the first and last k knots, in case of open b-splines

- Only restriction: non-decreasing knots
- Knots can have multiplicity larger than one

Effect of knot multiplicity for $k = 4$ (cubic)

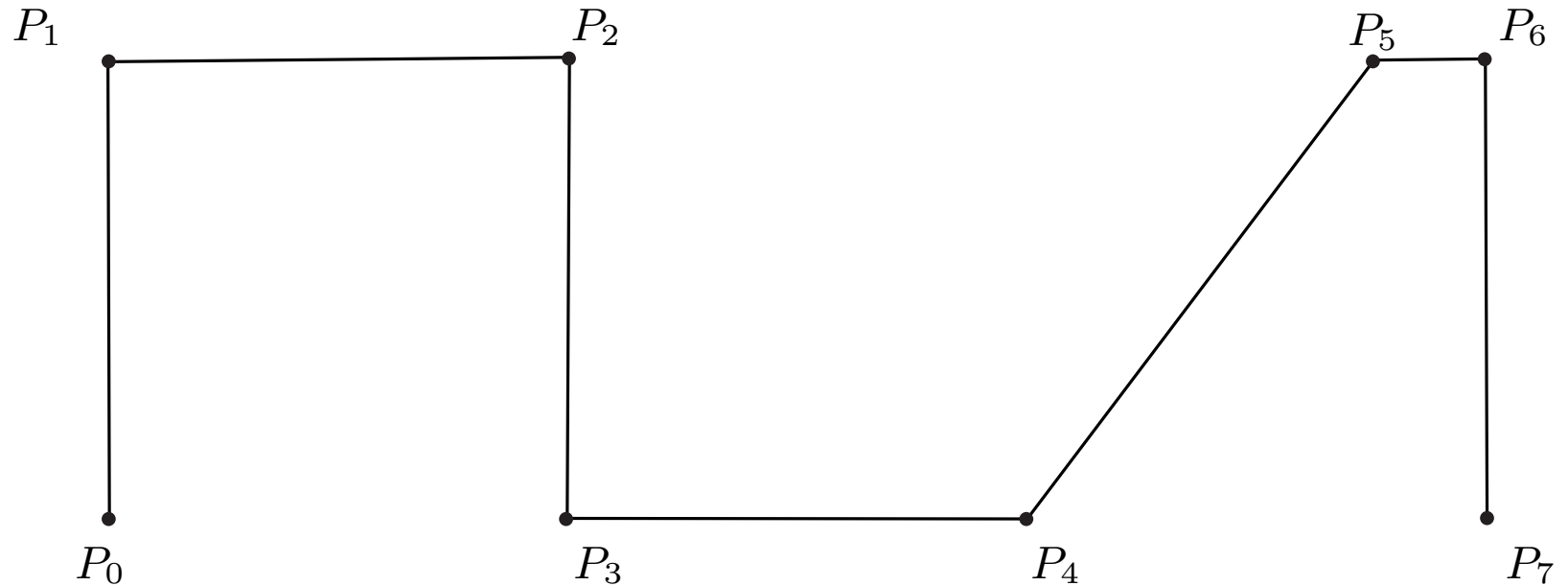
NON-UNIFORM B-SPLINES

When knots are *not* equally spaced

└─→ ignoring the first and last k knots, in case of open b-splines

- Only restriction: non-decreasing knots
- Knots can have multiplicity larger than one

Effect of knot multiplicity for $k = 4$ (cubic)



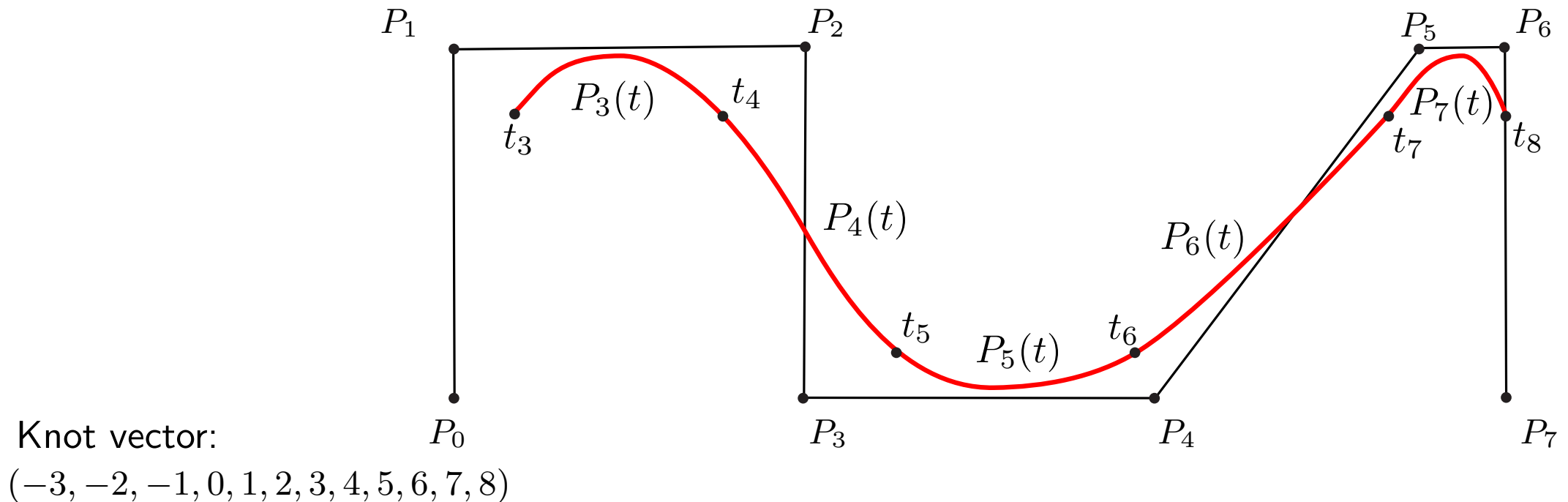
NON-UNIFORM B-SPLINES

When knots are *not* equally spaced

→ ignoring the first and last k knots, in case of open b-splines

- Only restriction: non-decreasing knots
- Knots can have multiplicity larger than one

Effect of knot multiplicity for $k = 4$ (cubic)



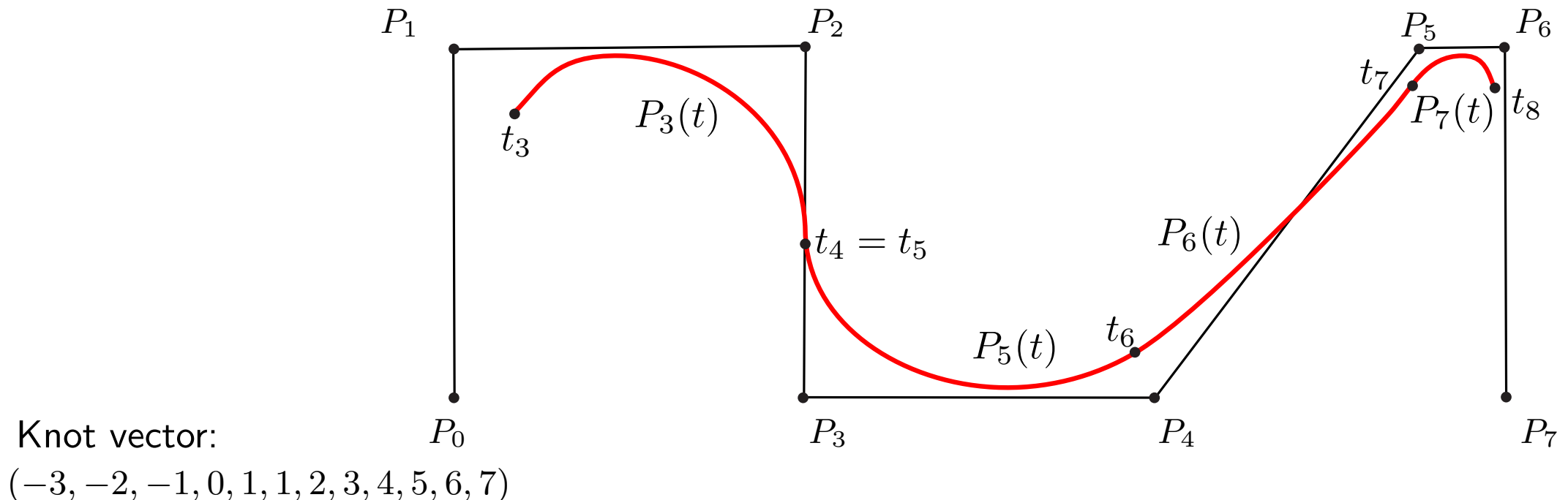
NON-UNIFORM B-SPLINES

When knots are *not* equally spaced

→ ignoring the first and last k knots, in case of open b-splines

- Only restriction: non-decreasing knots
- Knots can have multiplicity larger than one

Effect of knot multiplicity for $k = 4$ (cubic)



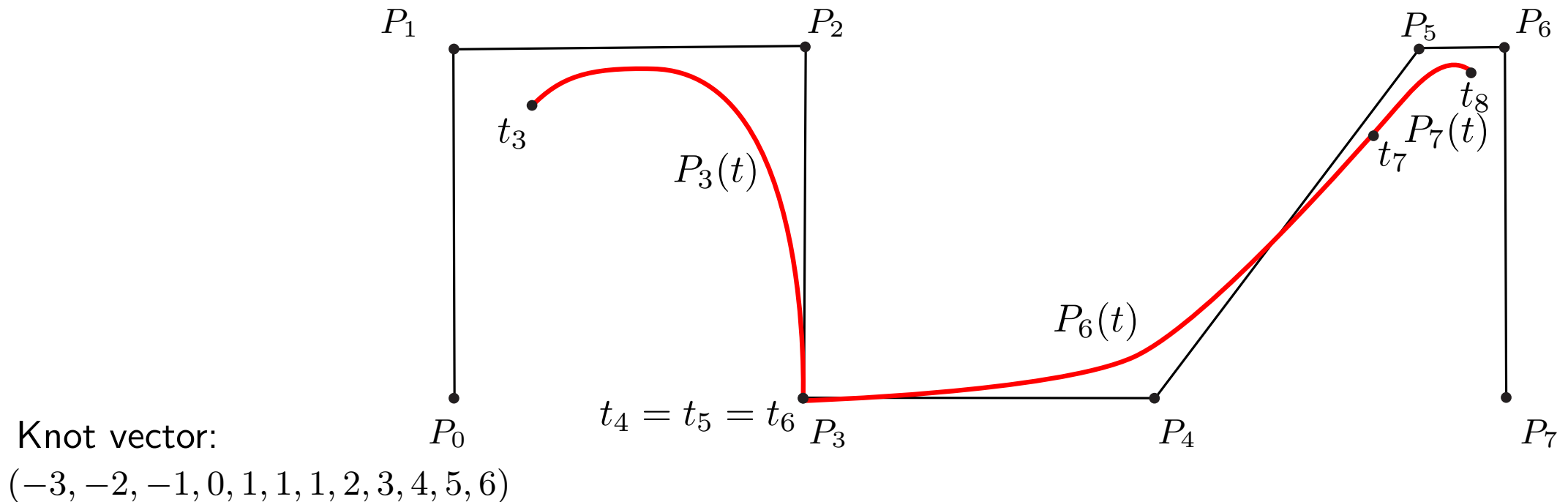
NON-UNIFORM B-SPLINES

When knots are *not* equally spaced

→ ignoring the first and last k knots, in case of open b-splines

- Only restriction: non-decreasing knots
- Knots can have multiplicity larger than one

Effect of knot multiplicity for $k = 4$ (cubic)



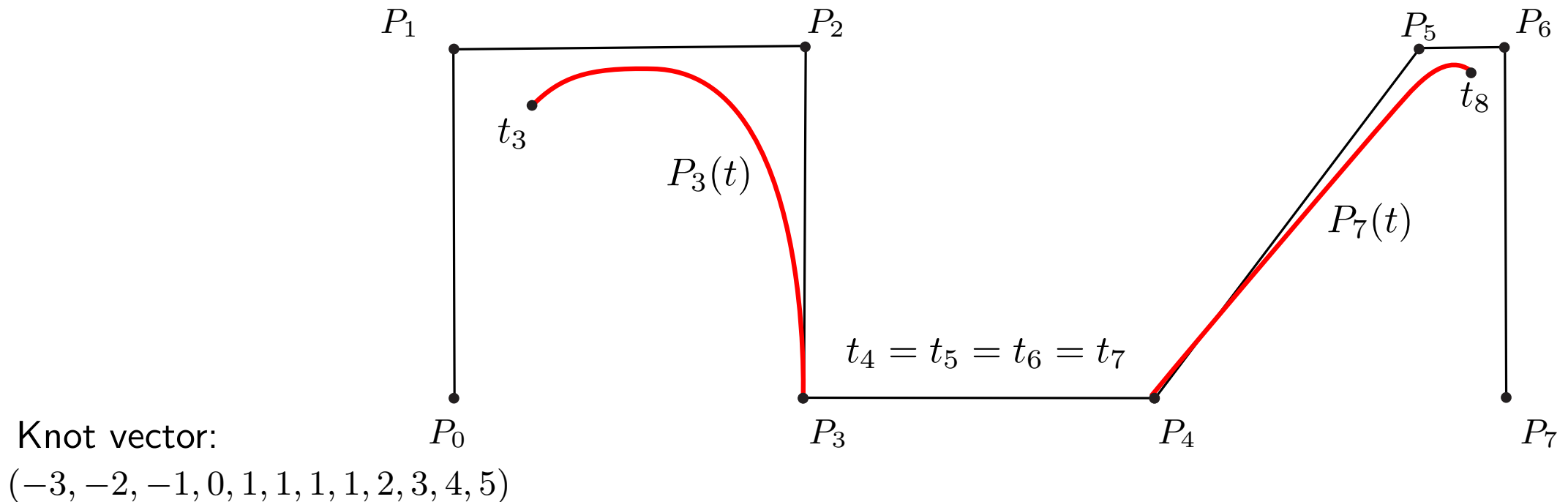
NON-UNIFORM B-SPLINES

When knots are *not* equally spaced

→ ignoring the first and last k knots, in case of open b-splines

- Only restriction: non-decreasing knots
- Knots can have multiplicity larger than one

Effect of knot multiplicity for $k = 4$ (cubic)



NON-UNIFORM B-SPLINES

Understanding knot vectors

NON-UNIFORM B-SPLINES

Understanding knot vectors

Open uniform B-splines interpolate the first and last control points due to the knot multiplicity

In general: continuity at the knots depends on multiplicity

$N_{i,k}(t)$ is $(k - m - 1)$ times continuously differentiable, where m is the multiplicity of the knot

NON-UNIFORM B-SPLINES

Understanding knot vectors

Open uniform B-splines interpolate the first and last control points due to the knot multiplicity

In general: continuity at the knots depends on multiplicity

$N_{i,k}(t)$ is $(k - m - 1)$ times continuously differentiable, where m is the multiplicity of the knot
(m =number of repetitions of knot value)

Examples:

- If all knots are different, a cubic ($k = 4$) B-spline is C^2 -continuous at every knot

NON-UNIFORM B-SPLINES

Understanding knot vectors

Open uniform B-splines interpolate the first and last control points due to the knot multiplicity

In general: continuity at the knots depends on multiplicity

$N_{i,k}(t)$ is $(k - m - 1)$ times continuously differentiable, where m is the multiplicity of the knot
(m =number of repetitions of knot value)

Examples:

- If all knots are different, a cubic ($k = 4$) B-spline is C^2 -continuous at every knot
- If a knot appears twice, the cubic B-spline will be only C^1 -continuous there

NON-UNIFORM B-SPLINES

Understanding knot vectors

Open uniform B-splines interpolate the first and last control points due to the knot multiplicity

In general: continuity at the knots depends on multiplicity

$N_{i,k}(t)$ is $(k - m - 1)$ times continuously differentiable, where m is the multiplicity of the knot
(m =number of repetitions of knot value)

Examples:

- If all knots are different, a cubic ($k = 4$) B-spline is C^2 -continuous at every knot
- If a knot appears twice, the cubic B-spline will be only C^1 -continuous there
- If a knot appears three times, the cubic B-spline will be only C^0 -continuous there

See example is
<http://geometrie.foretnik.net/files/NURBS-en.swf>

NON-UNIFORM B-SPLINES

Matrix form

Matrix-based expressions to compute non-uniform B-splines also exist

NON-UNIFORM B-SPLINES

Matrix form

Matrix-based expressions to compute non-uniform B-splines also exist

Linear case ($k = 2$)

$$N_{i2} = \frac{u - u_i}{u_{i+1} - u_i} N_{i1}(u) + \frac{u_{i+2} - u}{u_{i+2} - u_{i+1}} N_{i+1,1}(u)$$
$$= \begin{cases} \frac{u - u_i}{u_{i+1} - u_i} & \text{for } u \in [u_i, u_{i+1}), \\ \frac{u_{i+2} - u}{u_{i+2} - u_{i+1}} & \text{for } u \in [u_{i+1}, u_{i+2}), \\ 0 & \text{otherwise.} \end{cases}$$

For $i = 0$, this becomes

$$N_{02} = \begin{cases} \frac{u - u_0}{u_1 - u_0} & \text{for } u \in [u_0, u_1), \\ \frac{u_2 - u}{u_2 - u_1} & \text{for } u \in [u_1, u_2), \\ 0 & \text{otherwise.} \end{cases}$$

$$\Delta = u_2 - u_1$$

$$t = \frac{u - u_1}{\Delta} = \frac{u - u_1}{u_2 - u_1}.$$

$$\mathbf{P}(t) = (t, 1) \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \end{pmatrix}$$

$$t \in [0, 1]$$

$N_{12}(u)$ is obtained by incrementing all the indices

NON-UNIFORM B-SPLINES

Matrix form

Matrix-based expressions to compute non-uniform B-splines also exist

Quadratic case ($k = 3$)

$$N_{03}(u) = \begin{cases} \frac{u - u_0}{u_2 - u_0} \cdot \frac{u - u_0}{u_1 - u_0} & \text{for } u \in [u_0, u_1), \\ \frac{u - u_0}{u_2 - u_0} \cdot \frac{u_2 - u}{u_2 - u_1} + \frac{u_3 - u}{u_3 - u_1} \cdot \frac{u - u_1}{u_2 - u_1} & \text{for } u \in [u_1, u_2), \\ \frac{u_3 - u}{u_3 - u_1} \cdot \frac{u_3 - u}{u_3 - u_2} & \text{for } u \in [u_2, u_3), \\ 0 & \text{otherwise.} \end{cases}$$

→ $N_{13}(u)$ and $N_{23}(u)$ are obtained by incrementing all the indices over subinterval $[u_2, u_3)$

$$\begin{aligned} N_{03}(u) &= \frac{u_3 - u}{u_3 - u_1} \cdot \frac{u_3 - u}{u_3 - u_2}, \\ N_{13}(u) &= \frac{u - u_1}{u_3 - u_1} \cdot \frac{u_3 - u}{u_3 - u_2} + \frac{u_4 - u}{u_4 - u_2} \cdot \frac{u - u_2}{u_3 - u_2}, \\ N_{23}(u) &= \frac{u - u_2}{u_4 - u_2} \cdot \frac{u - u_2}{u_3 - u_2}. \end{aligned}$$

Need notation for difference between consecutive knots:

$$\Delta_1 = u_2 - u_1, \quad \Delta_2 = u_3 - u_2, \quad \Delta_3 = u_4 - u_3.$$

We also define $t = (u - u_2)/\Delta_2$, which implies

$$\begin{aligned} u - u_1 &= t\Delta_2 + \Delta_1, \\ u - u_2 &= t\Delta_2, \\ u - u_3 &= (t - 1)\Delta_2, \\ u - u_4 &= t\Delta_2 - (\Delta_2 + \Delta_3). \end{aligned}$$

$$\mathbf{P}(t) = (t^2, t, 1) \begin{pmatrix} a & -a - b & b \\ -2a & 2a & 0 \\ a & 1 - a & 0 \end{pmatrix} \begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \end{pmatrix}$$

$$a = \frac{\Delta_2}{\Delta_1 + \Delta_2}, \quad b = \frac{\Delta_2}{\Delta_2 + \Delta_3},$$

$$t \in [0, 1]$$

NON-UNIFORM B-SPLINES

Matrix form

Matrix-based expressions to compute non-uniform B-splines also exist

Cubic case ($k = 4$)

$$N_{04}(u) = \begin{cases} \frac{u-u_0}{u_3-u_0} \cdot \frac{u-u_0}{u_2-u_0} \cdot \frac{u-u_0}{u_1-u_0} & \text{for } u \in [u_0, u_1), \\ \frac{u-u_0}{u_3-u_0} \cdot \frac{u-u_0}{u_2-u_0} \cdot \frac{u_2-u_1}{u_2-u} \\ + \frac{u_3-u_0}{u-u_0} \cdot \frac{u_2-u_0}{u_3-u} \cdot \frac{u-u_1}{u-u_1} \\ + \frac{u_3-u_0}{u_4-u} \cdot \frac{u_3-u_1}{u-u_1} \cdot \frac{u_2-u_1}{u-u_1} & \text{for } u \in [u_1, u_2), \\ + \frac{u_4-u_1}{u-u_0} \cdot \frac{u_3-u_1}{u_3-u} \cdot \frac{u_2-u_1}{u_3-u} \\ \frac{u_3-u_0}{u_4-u} \cdot \frac{u_3-u_1}{u-u_1} \cdot \frac{u_3-u_2}{u_3-u} \\ + \frac{u_4-u_1}{u_4-u} \cdot \frac{u_3-u_1}{u-u_1} \cdot \frac{u_3-u_2}{u-u_2} \\ + \frac{u_4-u_1}{u_4-u} \cdot \frac{u_4-u_2}{u_4-u} \cdot \frac{u_3-u_2}{u-u_2} & \text{for } u \in [u_2, u_3), \\ \frac{u_4-u}{u_4-u_1} \cdot \frac{u_4-u}{u_4-u_2} \cdot \frac{u_4-u}{u_4-u_3} & \text{for } u \in [u_3, u_4), \\ 0 & \text{otherwise.} \end{cases}$$

$N_{14}(u)$, $N_{24}(u)$ and $N_{34}(u)$ are obtained by incrementing all the indices

Only in $[u_3, u_4)$ all four are nonzero, with values:

$$N_{04}(u) = \frac{u_4-u}{u_4-u_1} \cdot \frac{u_4-u}{u_4-u_2} \cdot \frac{u_4-u}{u_4-u_3},$$

$$N_{14}(u) = \frac{u-u_1}{u_4-u_1} \cdot \frac{u_4-u}{u_4-u_2} \cdot \frac{u_4-u}{u_4-u_3} + \frac{u_5-u}{u_5-u_2} \cdot \frac{u-u_2}{u_4-u_2} \cdot \frac{u_4-u}{u_4-u_3} \\ + \frac{u_5-u}{u_5-u_2} \cdot \frac{u_5-u}{u_5-u_3} \cdot \frac{u-u_3}{u_4-u_3},$$

$$N_{24}(u) = \frac{u-u_2}{u_5-u_2} \cdot \frac{u-u_2}{u_4-u_2} \cdot \frac{u_4-u}{u_4-u_3} + \frac{u-u_2}{u_5-u_2} \cdot \frac{u_5-u}{u_5-u_3} \cdot \frac{u-u_3}{u_4-u_3} \\ + \frac{u_6-u}{u_6-u_3} \cdot \frac{u-u_3}{u_5-u_3} \cdot \frac{u-u_3}{u_4-u_3},$$

$$N_{34}(u) = \frac{u-u_3}{u_6-u_3} \cdot \frac{u-u_3}{u_5-u_3} \cdot \frac{u-u_3}{u_4-u_3}.$$

take $\Delta_1 = u_2 - u_1, \quad \Delta_2 = u_3 - u_2, \quad \Delta_3 = u_4 - u_3,$
 $\Delta_4 = u_5 - u_4, \quad \Delta_5 = u_6 - u_5, \quad t = (u - u_3)/\Delta_3.$

NON-UNIFORM B-SPLINES

Matrix form

Matrix-based expressions to compute non-uniform B-splines also exist

Cubic case ($k = 4$)

$$N_{04}(u) = \begin{cases} \frac{u-u_0}{u_3-u_0} \cdot \frac{u-u_0}{u_2-u_0} \cdot \frac{u-u_0}{u_1-u_0} \\ \frac{u_3-u_0}{u-u_0} \cdot \frac{u_2-u_0}{u-u_0} \cdot \frac{u_1-u_0}{u_2-u} \\ \frac{u_3-u_0}{u-u_0} \cdot \frac{u_2-u_0}{u_3-u} \cdot \frac{u_2-u_1}{u-u_1} \\ \quad + \frac{u_3-u_0}{u_4-u} \cdot \frac{u_3-u_1}{u-u_1} \cdot \frac{u_2-u_1}{u-u_1} \\ \quad + \frac{u_4-u_1}{u-u_0} \cdot \frac{u_3-u_1}{u_3-u} \cdot \frac{u_2-u_1}{u_3-u} \\ \frac{u_3-u_0}{u_4-u} \cdot \frac{u_3-u_1}{u-u_1} \cdot \frac{u_3-u_2}{u_3-u} \\ \quad + \frac{u_4-u_1}{u_4-u} \cdot \frac{u_3-u_1}{u_4-u} \cdot \frac{u_3-u_2}{u-u_2} \\ \quad + \frac{u_4-u_1}{u_4-u} \cdot \frac{u_4-u_2}{u_4-u} \cdot \frac{u_3-u_2}{u_4-u} \\ \frac{u_4-u}{u_4-u_1} \cdot \frac{u_4-u}{u_4-u_2} \cdot \frac{u_4-u}{u_4-u_3} \\ 0 \end{cases}$$

We obtain:

$$\mathbf{P}(t) = (t^3, t^2, t, 1) \begin{pmatrix} -a & a+b+c & -b-c-d & d \\ 3a & -3a-3b & 3b & 0 \\ -3a & 3a-3e & 3e & 0 \\ a & 1-a-f & f & 0 \end{pmatrix} \begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix},$$

where

$$\begin{aligned} a &= \frac{\Delta_3^2}{(\Delta_1 + \Delta_2 + \Delta_3)(\Delta_2 + \Delta_3)}, & d &= \frac{\Delta_3^2}{(\Delta_3 + \Delta_4 + \Delta_5)(\Delta_3 + \Delta_4)}, \\ b &= \frac{\Delta_3^2}{(\Delta_2 + \Delta_3 + \Delta_4)(\Delta_2 + \Delta_3)}, & e &= \frac{\Delta_2 \Delta_3}{(\Delta_2 + \Delta_3 + \Delta_4)(\Delta_2 + \Delta_3)}, \\ c &= \frac{\Delta_3^2}{(\Delta_2 + \Delta_3 + \Delta_4)(\Delta_3 + \Delta_4)}, & f &= \frac{\Delta_2^2}{(\Delta_2 + \Delta_3 + \Delta_4)(\Delta_2 + \Delta_3)}. \end{aligned}$$

$$\begin{aligned} &\text{for } u \in [u_3, u_4), \quad N_{34}(u) = \frac{u}{u_6-u_3} \cdot \frac{u}{u_5-u_3} \cdot \frac{u}{u_4-u_3}. \\ &\text{otherwise.} \end{aligned}$$

$N_{14}(u)$, $N_{24}(u)$ and $N_{34}(u)$ are obtained by incrementing all the indices

$$\begin{aligned} \text{take } \Delta_1 &= u_2 - u_1, & \Delta_2 &= u_3 - u_2, & \Delta_3 &= u_4 - u_3, \\ \Delta_4 &= u_5 - u_4, & \Delta_5 &= u_6 - u_5, & t &= (u - u_3)/\Delta_3. \end{aligned}$$

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

The most general parametric curve

Same idea as for rational Bézier: each control point P_i has a weight, $w_i \geq 0$. This gives even more flexibility to shape the curve

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

The most general parametric curve

Same idea as for rational Bézier: each control point P_i has a weight, $w_i \geq 0$. This gives even more flexibility to shape the curve

Advantages

- Invariant under projections

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

The most general parametric curve

Same idea as for rational Bézier: each control point P_i has a weight, $w_i \geq 0$. This gives even more flexibility to shape the curve

Advantages

- Invariant under projections

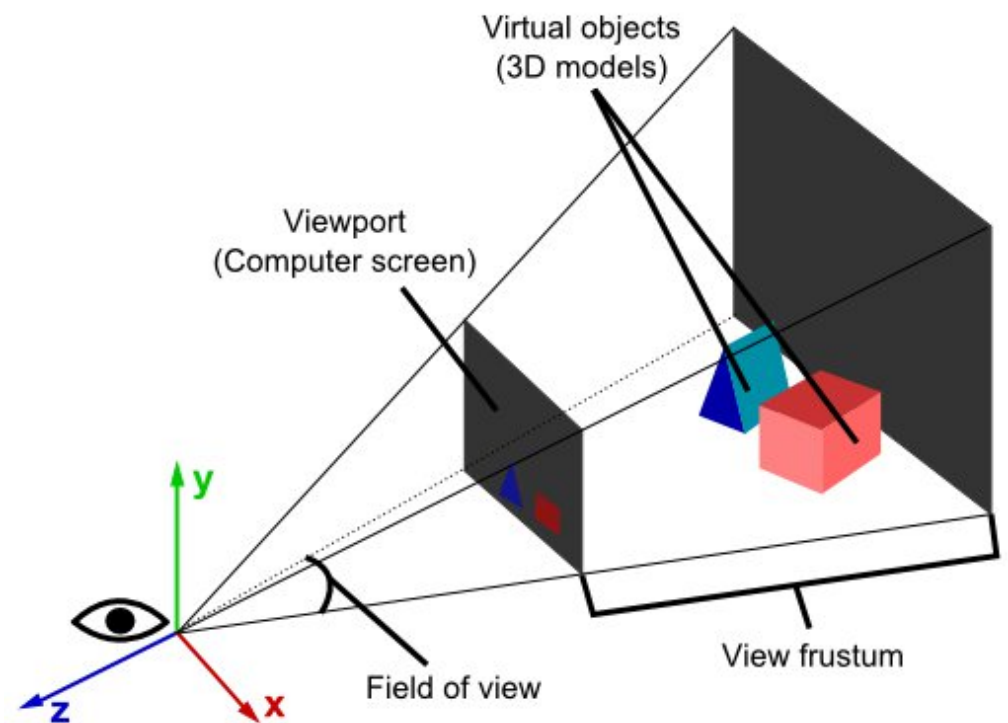


Figure from real3dtutorials.com

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

The most general parametric curve

Same idea as for rational Bézier: each control point P_i has a weight, $w_i \geq 0$. This gives even more flexibility to shape the curve

Advantages

- Invariant under projections
- It can represent conic curves exactly (e.g., segments of circles, ellipses, hyperbolas and parabolas)

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

The most general parametric curve

Same idea as for rational Bézier: each control point P_i has a weight, $w_i \geq 0$. This gives even more flexibility to shape the curve

Advantages

- Invariant under projections
- It can represent conic curves exactly (e.g., segments of circles, ellipses, hyperbolas and parabolas)
- It is more general, so it includes as particular cases all other B-splines and Bézier curves

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Recall: homogeneous coordinates

Any 2D point (x, y) is equivalent to a 3D point: (wx, wy, w)

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Recall: homogeneous coordinates

Any 2D point (x, y) is equivalent to a 3D point: (wx, wy, w)

└─► the projection of 3D point (a, b, w) to 2D is $(a/w, b/w)$

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Recall: homogeneous coordinates

Any 2D point (x, y) is equivalent to a 3D point: (wx, wy, w)

└─► the projection of 3D point (a, b, w) to 2D is $(a/w, b/w)$

Suppose your control points Q_i have one extra dimension $P_i \in \mathbb{R}^2 \rightarrow Q_i \in \mathbb{R}^3$

└─► e.g., each point $P_i = (x_i, y_i)$, becomes $Q_i = (w_i x_i, w_i y_i, w_i)$, for some $w_i \geq 0$

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Recall: homogeneous coordinates

Any 2D point (x, y) is equivalent to a 3D point: (wx, wy, w)

└─► the projection of 3D point (a, b, w) to 2D is $(a/w, b/w)$

Suppose your control points Q_i have one extra dimension $P_i \in \mathbb{R}^2 \rightarrow Q_i \in \mathbb{R}^3$

└─► e.g., each point $P_i = (x_i, y_i)$, becomes $Q_i = (w_i x_i, w_i y_i, w_i)$, for some $w_i \geq 0$

Compute a 3D B-spline curve in the usual (non-rational) way: $P_{\text{nr}}(t) = \sum_{i=0}^n Q_i N_{i,k}(t)$

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Recall: homogeneous coordinates

Any 2D point (x, y) is equivalent to a 3D point: (wx, wy, w)

└─► the projection of 3D point (a, b, w) to 2D is $(a/w, b/w)$

Suppose your control points Q_i have one extra dimension $P_i \in \mathbb{R}^2 \rightarrow Q_i \in \mathbb{R}^3$

└─► e.g., each point $P_i = (x_i, y_i)$, becomes $Q_i = (w_i x_i, w_i y_i, w_i)$, for some $w_i \geq 0$

Compute a 3D B-spline curve in the usual (non-rational) way: $P_{nr}(t) = \sum_{i=0}^n Q_i N_{i,k}(t)$

Now, we can project any point on the 3D curve $P_{nr}(t)$ to 2D:

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Recall: homogeneous coordinates

Any 2D point (x, y) is equivalent to a 3D point: (wx, wy, w)

└─► the projection of 3D point (a, b, w) to 2D is $(a/w, b/w)$

Suppose your control points Q_i have one extra dimension $P_i \in \mathbb{R}^2 \rightarrow Q_i \in \mathbb{R}^3$

└─► e.g., each point $P_i = (x_i, y_i)$, becomes $Q_i = (w_i x_i, w_i y_i, w_i)$, for some $w_i \geq 0$

Compute a 3D B-spline curve in the usual (non-rational) way: $P_{nr}(t) = \sum_{i=0}^n Q_i N_{i,k}(t)$

Now, we can project any point on the 3D curve $P_{nr}(t)$ to 2D:

└─► we need to isolate the coefficients w_i multiplying each point and divide by them

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Recall: homogeneous coordinates

Any 2D point (x, y) is equivalent to a 3D point: (wx, wy, w)

└─► the projection of 3D point (a, b, w) to 2D is $(a/w, b/w)$

Suppose your control points Q_i have one extra dimension $P_i \in \mathbb{R}^2 \rightarrow Q_i \in \mathbb{R}^3$

└─► e.g., each point $P_i = (x_i, y_i)$, becomes $Q_i = (w_i x_i, w_i y_i, w_i)$, for some $w_i \geq 0$

Compute a 3D B-spline curve in the usual (non-rational) way: $P_{nr}(t) = \sum_{i=0}^n Q_i N_{i,k}(t)$

Now, we can project any point on the 3D curve $P_{nr}(t)$ to 2D:

└─► we need to isolate the coefficients w_i multiplying each point and divide by them

$$P_r(t) = \frac{\sum_{i=0}^n w_i P_i N_{i,k}(t)}{\sum_{j=0}^n w_j N_{j,k}(t)} = \sum_{i=0}^n P_i \left(\frac{w_i N_{i,k}(t)}{\sum_{j=0}^n w_j N_{j,k}(t)} \right) = \sum_{i=0}^n P_i R_{i,k}(t)$$

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Recall: homogeneous coordinates

Any 2D point (x, y) is equivalent to a 3D point: (wx, wy, w)

└─► the projection of 3D point (a, b, w) to 2D is $(a/w, b/w)$

Suppose your control points Q_i have one extra dimension $P_i \in \mathbb{R}^2 \rightarrow Q_i \in \mathbb{R}^3$

└─► e.g., each point $P_i = (x_i, y_i)$, becomes $Q_i = (w_i x_i, w_i y_i, w_i)$, for some $w_i \geq 0$

Compute a 3D B-spline curve in the usual (non-rational) way: $P_{nr}(t) = \sum_{i=0}^n Q_i N_{i,k}(t)$

Now, we can project any point on the 3D curve $P_{nr}(t)$ to 2D:

└─► we need to isolate the coefficients w_i multiplying each point and divide by them

$$P_r(t) = \frac{\sum_{i=0}^n w_i P_i N_{i,k}(t)}{\sum_{j=0}^n w_j N_{j,k}(t)} = \sum_{i=0}^n P_i \underbrace{\left(\frac{w_i N_{i,k}(t)}{\sum_{j=0}^n w_j N_{j,k}(t)} \right)}_{\text{rational blending functions}} = \sum_{i=0}^n P_i \underbrace{R_{i,k}(t)}_{\text{rational blending functions}}$$

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Rational curves as curves in projective space

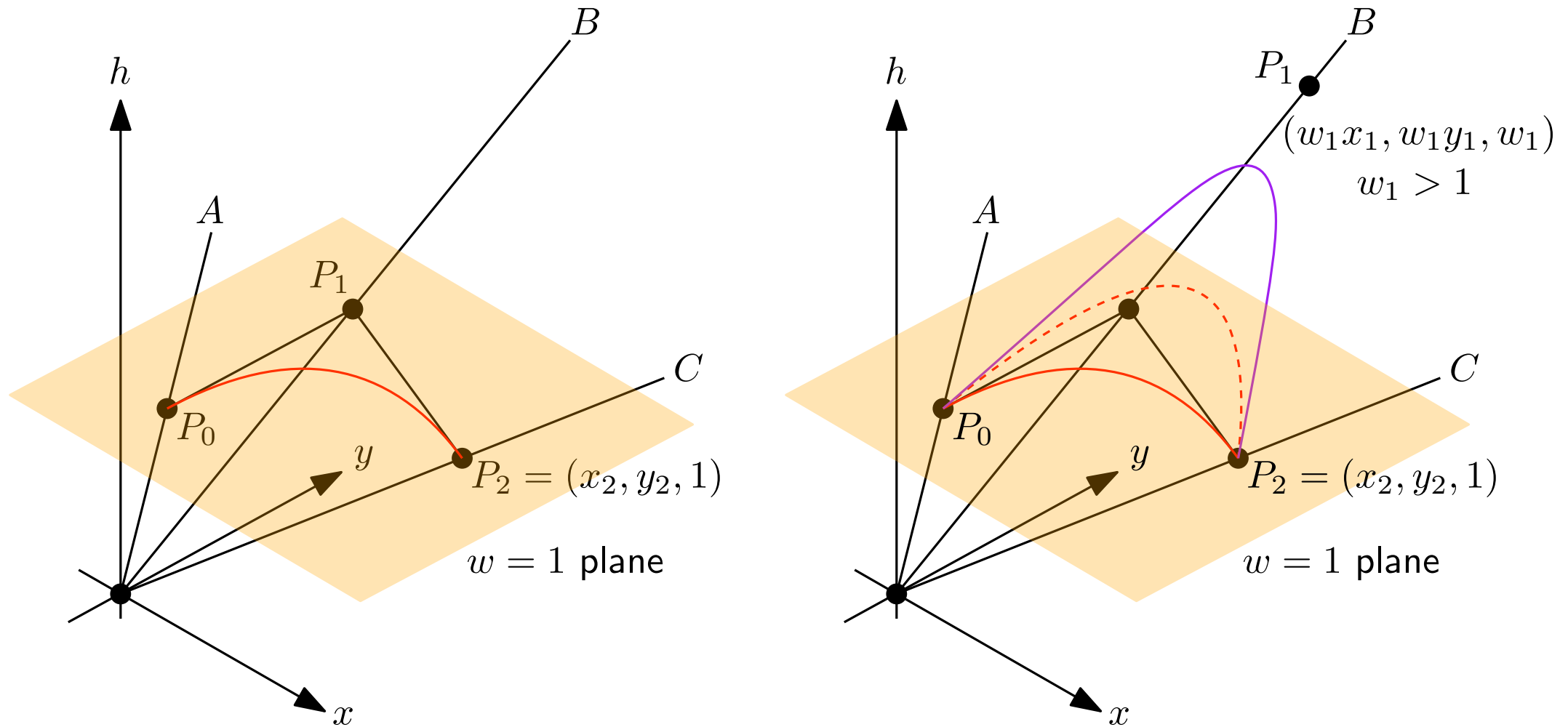


Figure adapted from book by Mortenson

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Rational curves as curves in projective space

Another picture

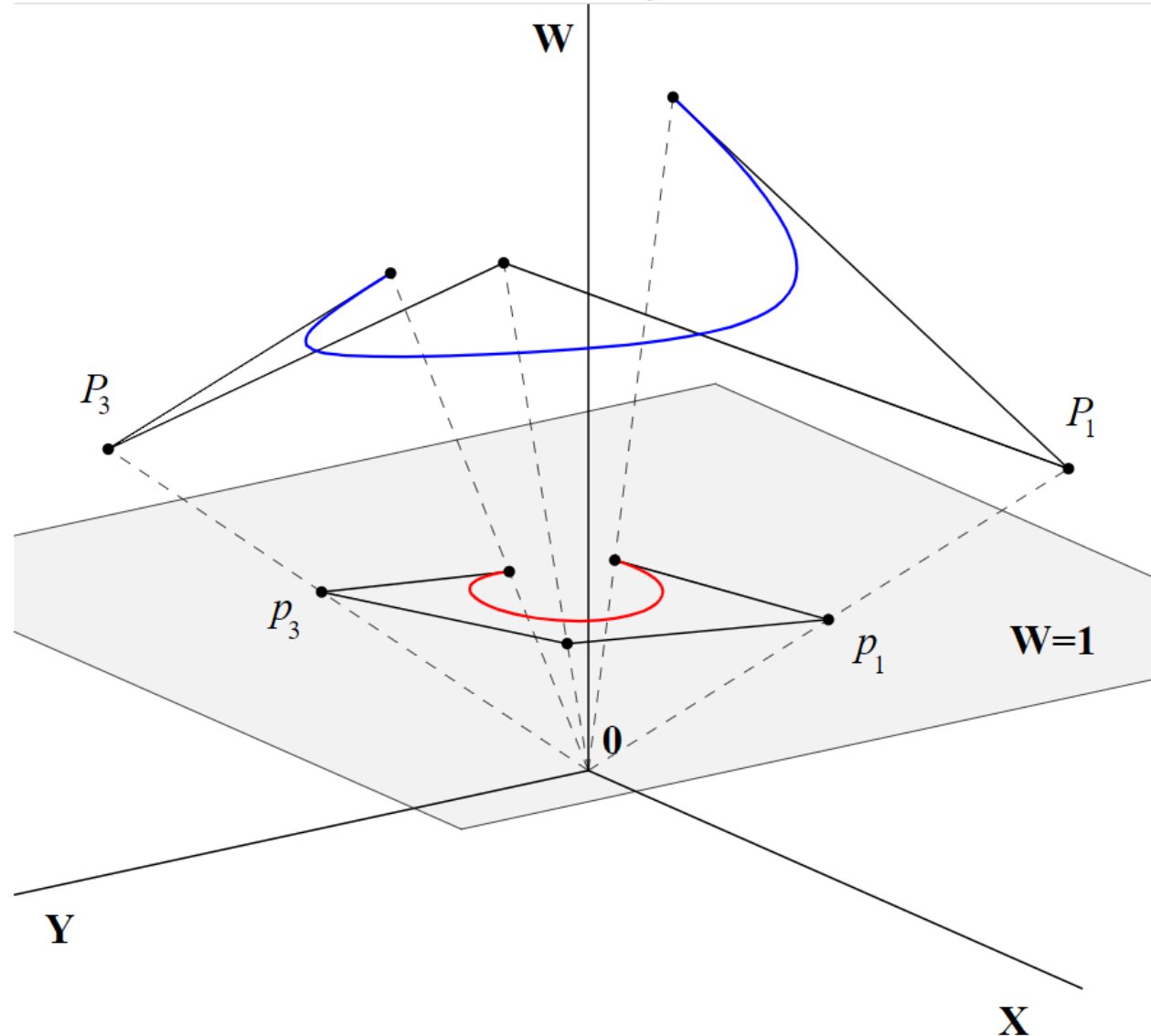


Figure by Wojciech Muła - Own work (Python script, final touches Inkscape), CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=1196334>

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Properties of rational basis functions and NURBS

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Properties of rational basis functions and NURBS

Has most properties of the non-rational basis functions, plus a few more:

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Properties of rational basis functions and NURBS

Has most properties of the non-rational basis functions, plus a few more:

- Non-negativity, partition of unity, unimodality, local support (for control point position and weight), convex hull property, etc..

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Properties of rational basis functions and NURBS

Has most properties of the non-rational basis functions, plus a few more:

- Non-negativity, partition of unity, unimodality, local support (for control point position and weight), convex hull property, etc..
- Effect of changing weight: increasing weight moves curve closer to P_i

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Properties of rational basis functions and NURBS

Has most properties of the non-rational basis functions, plus a few more:

- Non-negativity, partition of unity, unimodality, local support (for control point position and weight), convex hull property, etc..
- Effect of changing weight: increasing weight moves curve closer to P_i

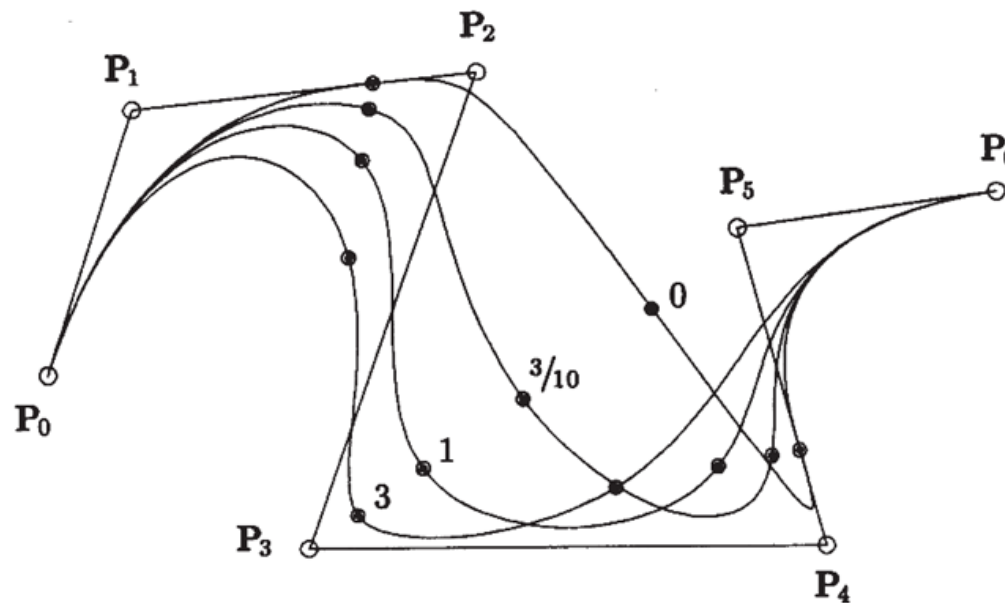


Figure 4.2. Rational cubic B-spline curves, with w_3 varying.

Figure from [Piegl and Tiller]

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Properties of rational basis functions and NURBS

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Properties of rational basis functions and NURBS

- When all weights are the same, the curve becomes non-rational

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

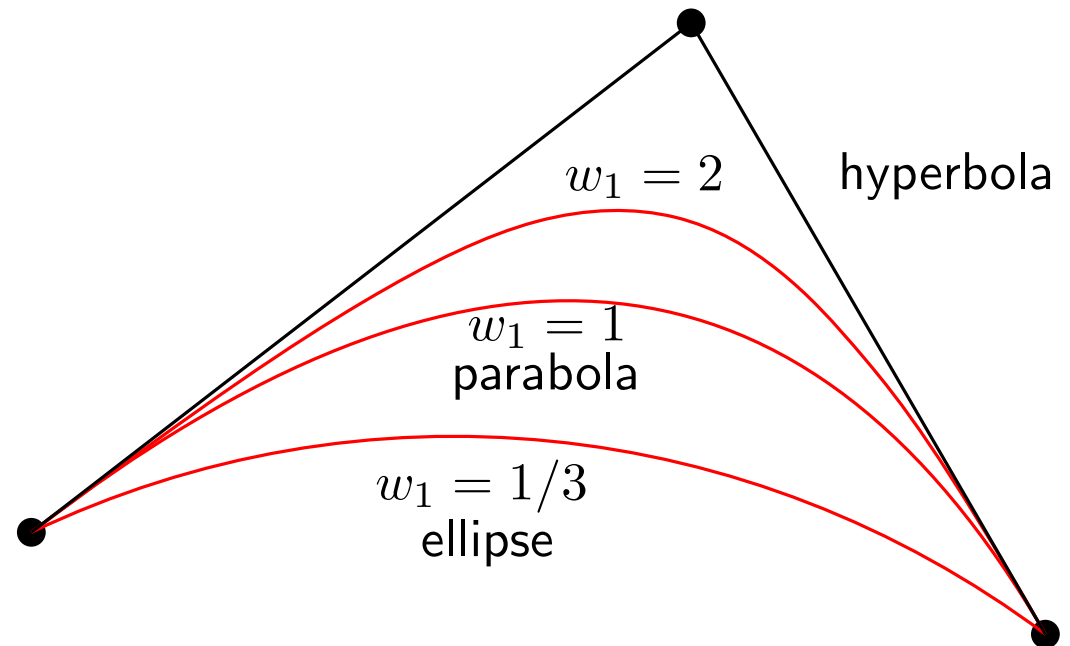
Properties of rational basis functions and NURBS

- When all weights are the same, the curve becomes non-rational
- Curve is invariant under projective transformations

NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Properties of rational basis functions and NURBS

- When all weights are the same, the curve becomes non-rational
- Curve is invariant under projective transformations
- Conic sections can be represented *exactly* (same as with rational Bézier curves)



NON-UNIFORM RATIONAL B-SPLINES (NURBS)

Properties of rational basis functions and NURBS

- When all weights are the same, the curve becomes non-rational
- Curve is invariant under projective transformations
- Conic sections can be represented *exactly* (same as with rational Bézier curves)

