Parametric Curves

Michael Kazhdan

(601.457/657)

Overview

What is a Spline?

Specific Examples:

- Hermite Splines
- Cardinal Splines
- Uniform Cubic B-Splines

Comparing Cardinal and Uniform Cubic B-Splines

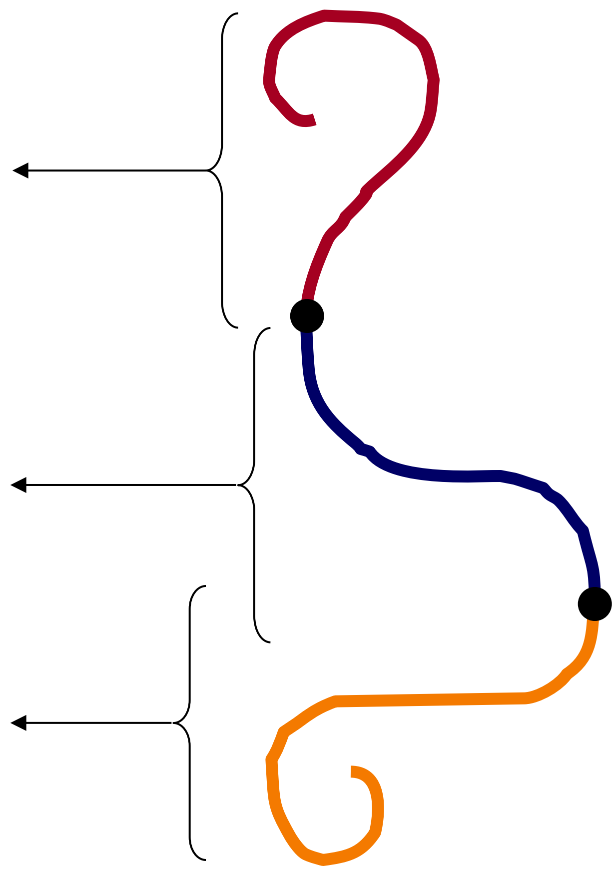
What is a Spline in CG?

A spline is a *piecewise* *polynomial function* whose derivatives satisfy *continuity constraints* at curve boundaries.

$\mathbf{P}_1(u) \quad u \in [0,1)$

$\mathbf{P}_2(u) \quad u \in [0,1)$

$\mathbf{P}_3(x) \quad u \in [0,1)$



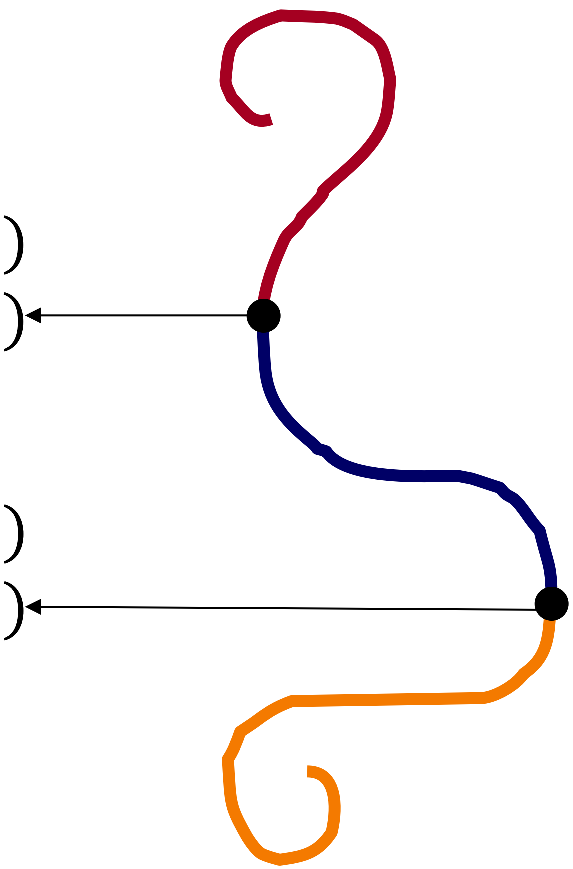
$$\mathbf{P}_k(u) = \sum_{j=0}^{n} \mathbf{a}_{kj} \cdot u^j \text{ with } \mathbf{a}_{kj} \in \mathbb{R}^d$$

What is a Spline in CG?

A spline is a *piecewise* *polynomial function* whose derivatives satisfy *continuity constraints* at curve boundaries.

$$\mathbf{P}_1(1) = \mathbf{P}_2(0)$$
$$\mathbf{P}_1'(1) = \mathbf{P}_2'(0)$$
$$\cdots$$

$$\mathbf{P}_2(1) = \mathbf{P}_3(0)$$
$$\mathbf{P}_2'(1) = \mathbf{P}_3'(0)$$
$$\cdots$$



$$\mathbf{P}_k(u) = \sum_{j=0}^{n} \mathbf{a}_{kj} \cdot u^j \text{ with } \mathbf{a}_{kj} \in \mathbb{R}^d$$

Overview

What is a Spline?

Specific Examples:

- Hermite Splines
- Cardinal Splines
- Uniform Cubic B-Splines

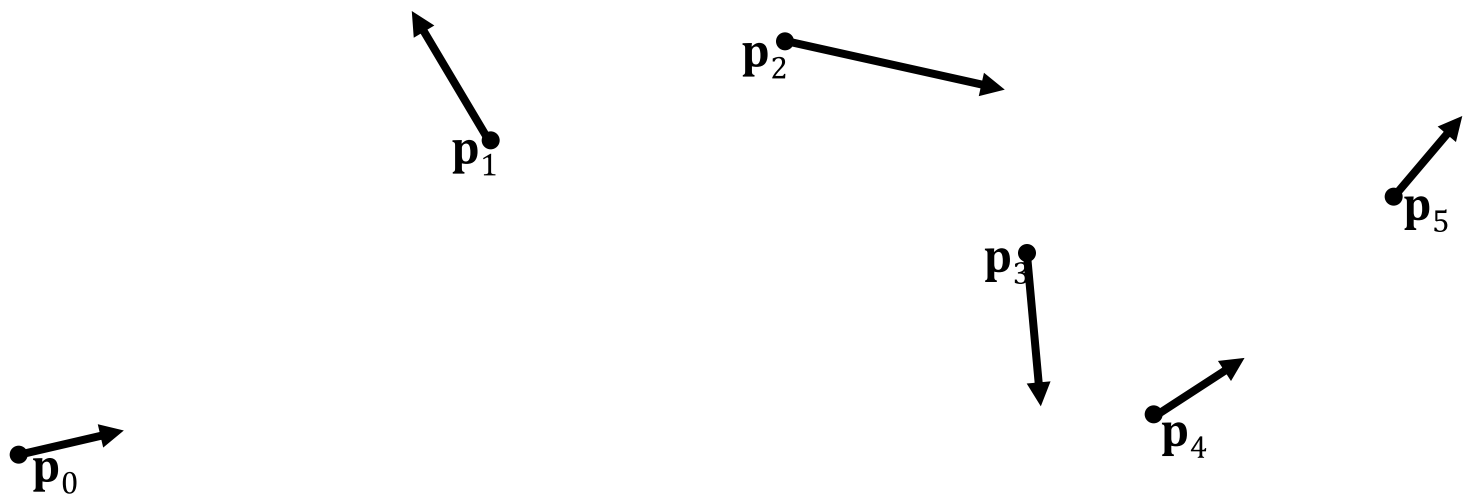
Comparing Cardinal and Uniform Cubic B-Splines

Specific Example: Hermite Splines

Interpolating piecewise *cubic* polynomial, each specified by:

- Start/end positions
- Start/end tangents

Iteratively construct the curve between adjacent end points that interpolate positions and tangents.

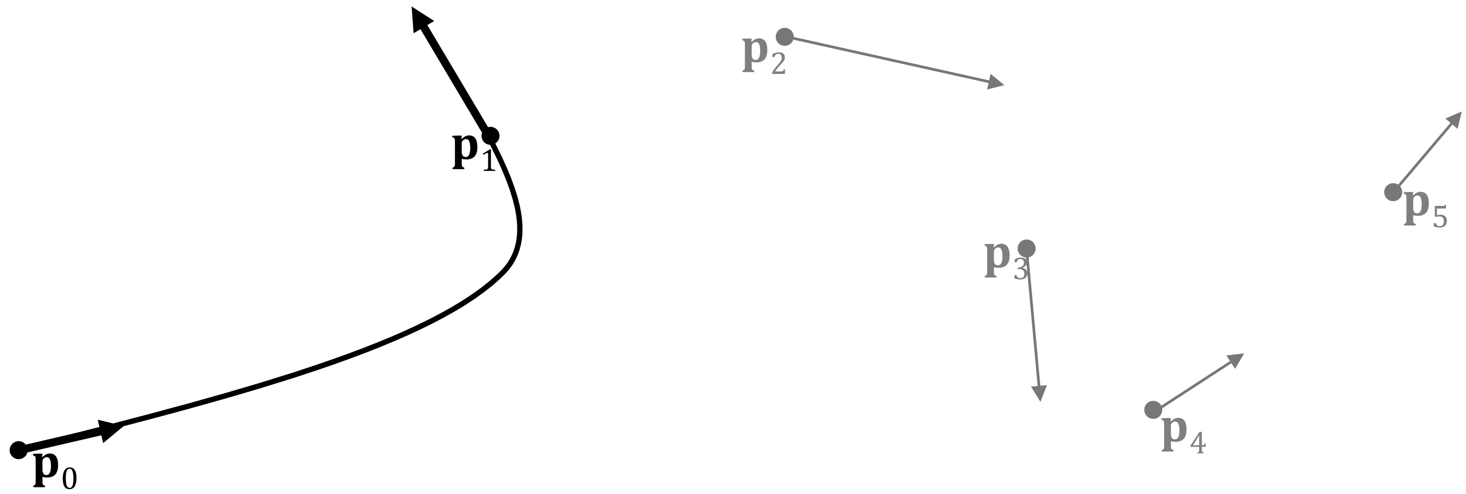


Specific Example: Hermite Splines

Interpolating piecewise *cubic* polynomial, each specified by:

- Start/end positions
- Start/end tangents

Iteratively construct the curve between adjacent end points that interpolate positions and tangents.

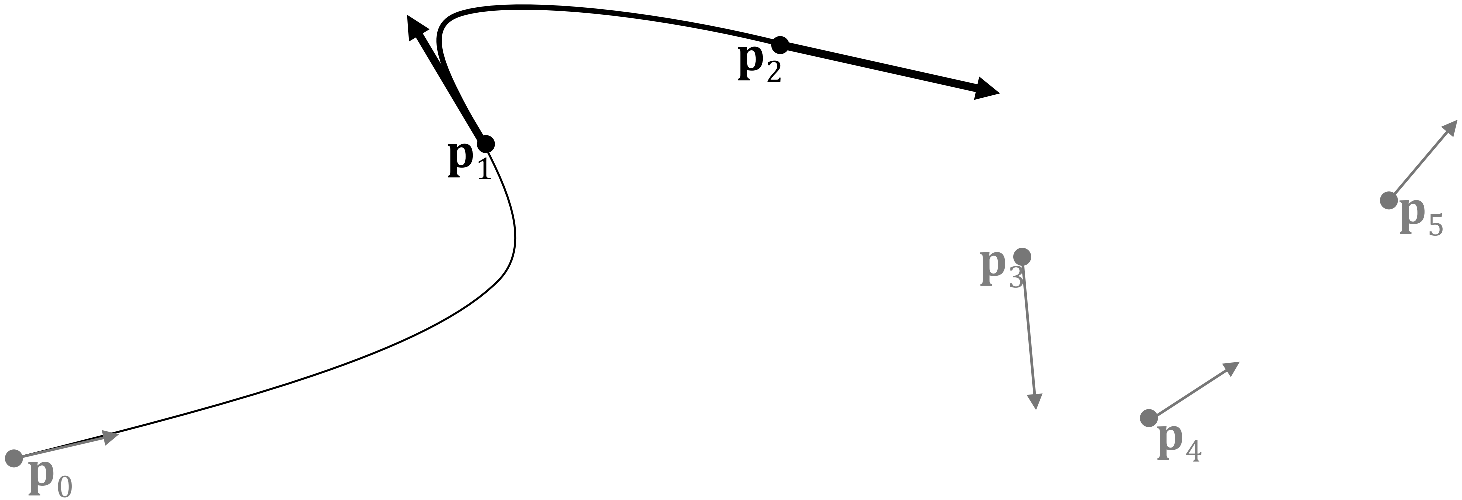


Specific Example: Hermite Splines

Interpolating piecewise *cubic* polynomial, each specified by:

- Start/end positions
- Start/end tangents

Iteratively construct the curve between adjacent end points that interpolate positions and tangents.

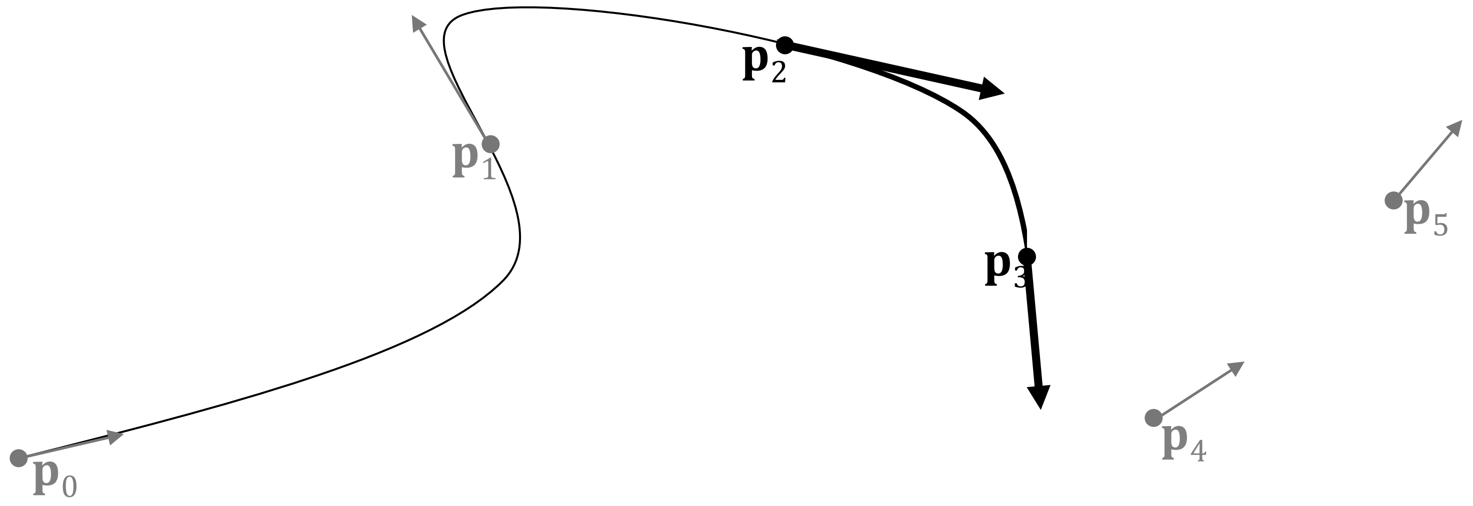


Specific Example: Hermite Splines

<u>Interpolating</u> piecewise *cubic* polynomial, each specified by:

- Start/end positions
- Start/end tangents

Iteratively construct the curve between adjacent end points that interpolate positions and tangents.

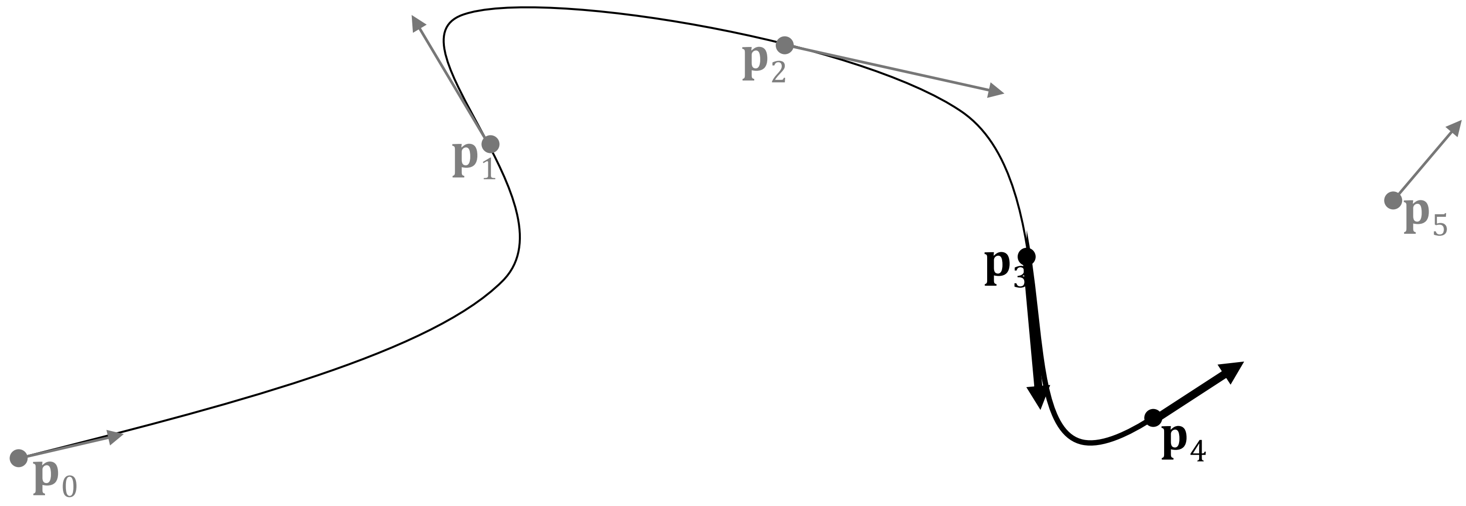


Specific Example: Hermite Splines

<u>Interpolating</u> piecewise *cubic* polynomial, each specified by:

- Start/end positions
- Start/end tangents

Iteratively construct the curve between adjacent end points that interpolate positions and tangents.

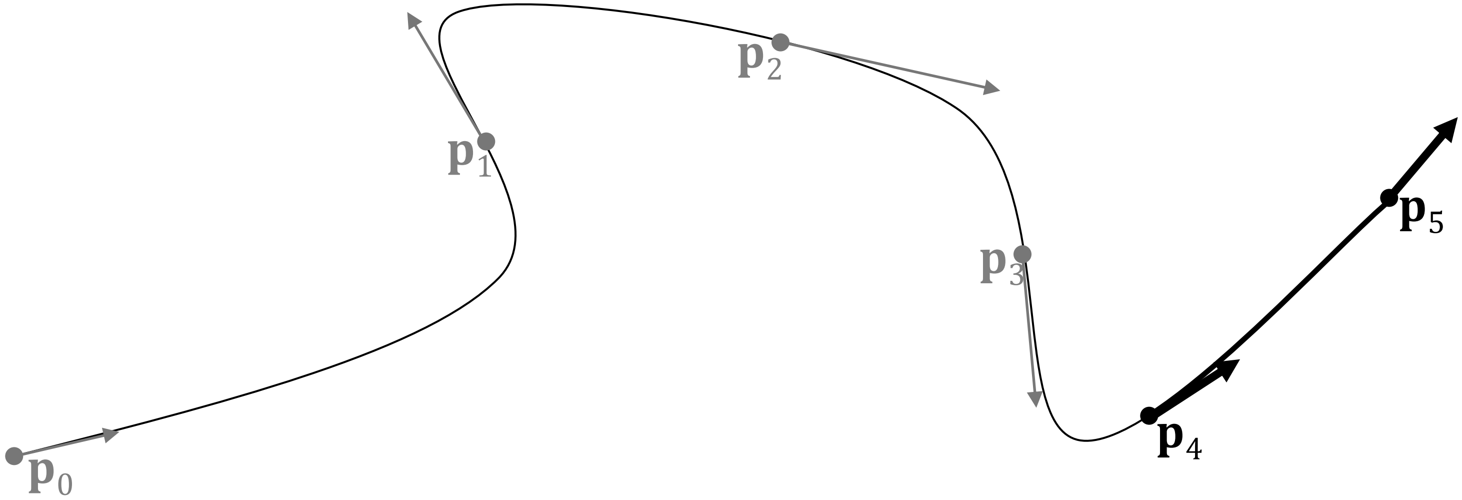


Specific Example: Hermite Splines

Interpolating piecewise *cubic* polynomial, each specified by:

- Start/end positions
- Start/end tangents

Iteratively construct the curve between adjacent end points that interpolate positions and tangents.

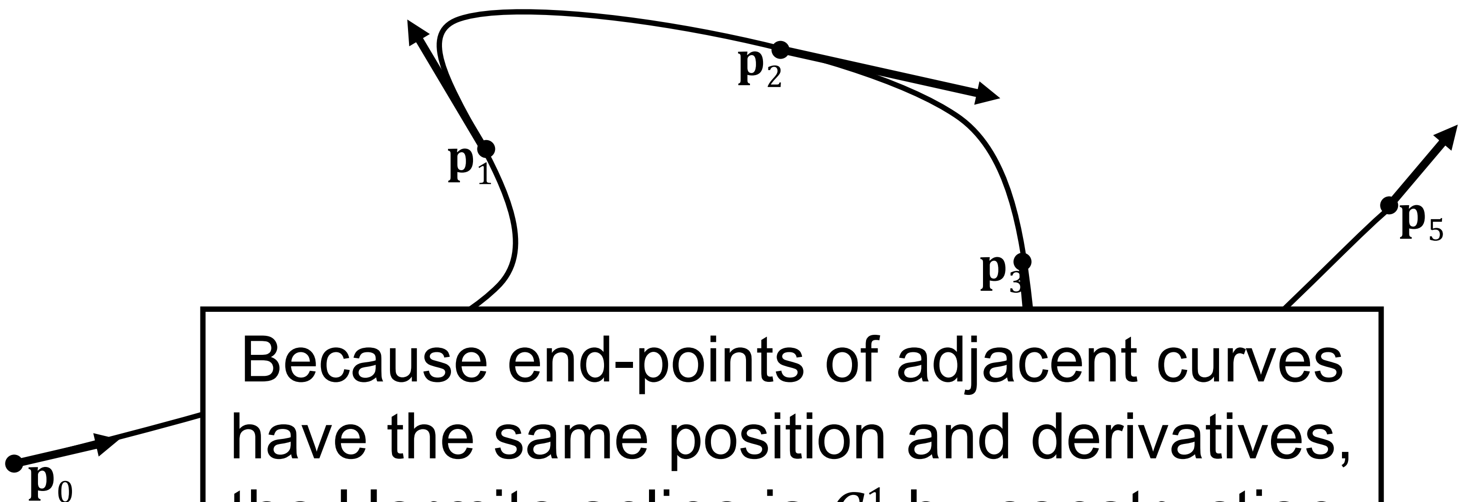


Specific Example: Hermite Splines

<u>Interpolating</u> piecewise *cubic* polynomial, each specified by:

- Start/end positions
- Start/end tangents

Iteratively construct the curve between adjacent end points that interpolate positions and tangents.



Because end-points of adjacent curves have the same position and derivatives, the Hermite spline is C^1 by construction.

Specific Example: Hermite Splines

Given the polynomial:

$$\mathbf{P}_k(u) = \mathbf{a} \cdot u^3 + \mathbf{b} \cdot u^2 + \mathbf{c} \cdot u + \mathbf{d}$$

we can write its derivative as:

$$\mathbf{P}_k'(u) = 3 \cdot \mathbf{a} \cdot u^2 + 2 \cdot \mathbf{b} \cdot u + \mathbf{c}$$

Using the matrix representations:

$$\mathbf{P}_k(u) = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix} \qquad \mathbf{P}_k'(u) = \begin{pmatrix} 3 \cdot u^2 & 2 \cdot u & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix}$$

Specific Example: Hermite Splines

$$\mathbf{P}_k(u) = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix} \qquad \mathbf{P}'_k(u) = \begin{pmatrix} 3 \cdot u^2 & 2 \cdot u & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix}$$

The values/derivatives at the end-points are:

$$\mathbf{p}_k = \mathbf{P}_k(0) = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix} \qquad \vec{\mathbf{t}}_k = \mathbf{P}'_k(0) = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix}$$

$$\mathbf{p}_{k+1} = \mathbf{P}_k(1) = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix} \qquad \vec{\mathbf{t}}_{k+1} = \mathbf{P}'_k(1) = \begin{pmatrix} 3 & 2 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix}$$

Specific Example: Hermite Splines

$$\mathbf{p}_k = \mathbf{P}_k(0) = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix} \qquad \vec{\mathbf{t}}_k = \mathbf{P}_k'(0) = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix}$$

$$\mathbf{p}_{k+1} = \mathbf{P}_k(1) = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix} \qquad \vec{\mathbf{t}}_{k+1} = \mathbf{P}_k'(1) = \begin{pmatrix} 3 & 2 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix}$$

Combining into a single matrix expression:

$$\begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix}$$

Specific Example: Hermite Splines

$$\begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix}$$

Inverting:

$$\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix} = \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix}$$

Specific Example: Hermite Splines

$$\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix} = \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix}$$

Using the fact that:

$$\mathbf{P}_k(u) = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{pmatrix}$$

We get:

$$\mathbf{P}_k(u) = \underbrace{\begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix}}_{\text{parameters}} \underbrace{\begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}}_{\mathbf{M}_{\text{Hermite}}} \underbrace{\begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix}}_{\text{boundary info}}$$

Specific Example: Hermite Splines

$$\mathbf{P}_k(u) = \left(\begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix} \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \right) \begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix}$$

Pre-multiplying to get blending functions:

$$\mathbf{P}_k(u) = H_0(u) \cdot \mathbf{p}_k + H_1(u) \cdot \mathbf{p}_{k+1} + H_2(u) \cdot \vec{\mathbf{t}}_k + H_3(u)$$

with:

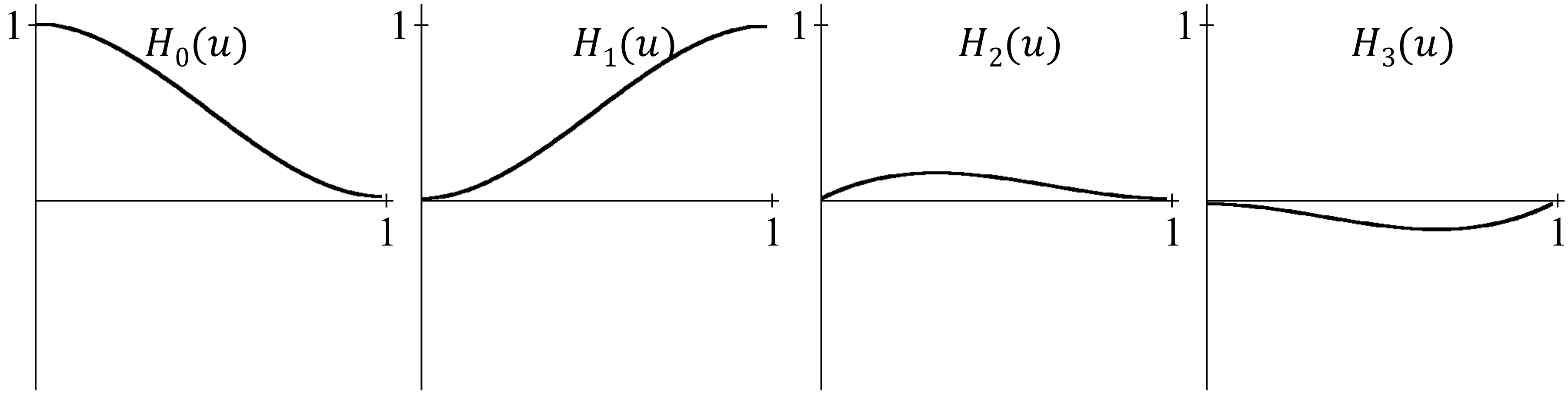
- $H_0(u) = 2u^3 - 3u^2 + 1$
- $H_1(u) = -2u^3 + 3u^2$
- $H_2(u) = u^3 - 2u^2 + u$
- $H_3(u) = u^3 - u^2$

Specific Example: Hermite Splines

Pre-multiplying to get blending functions:

- $H_0(u) = 2u^3 - 3u^2 + 1$
- $H_1(u) = -2u^3 + 3u^2$
- $H_2(u) = u^3 - 2u^2 + u$
- $H_3(u) = u^3 - u^2$

} Blending Functions



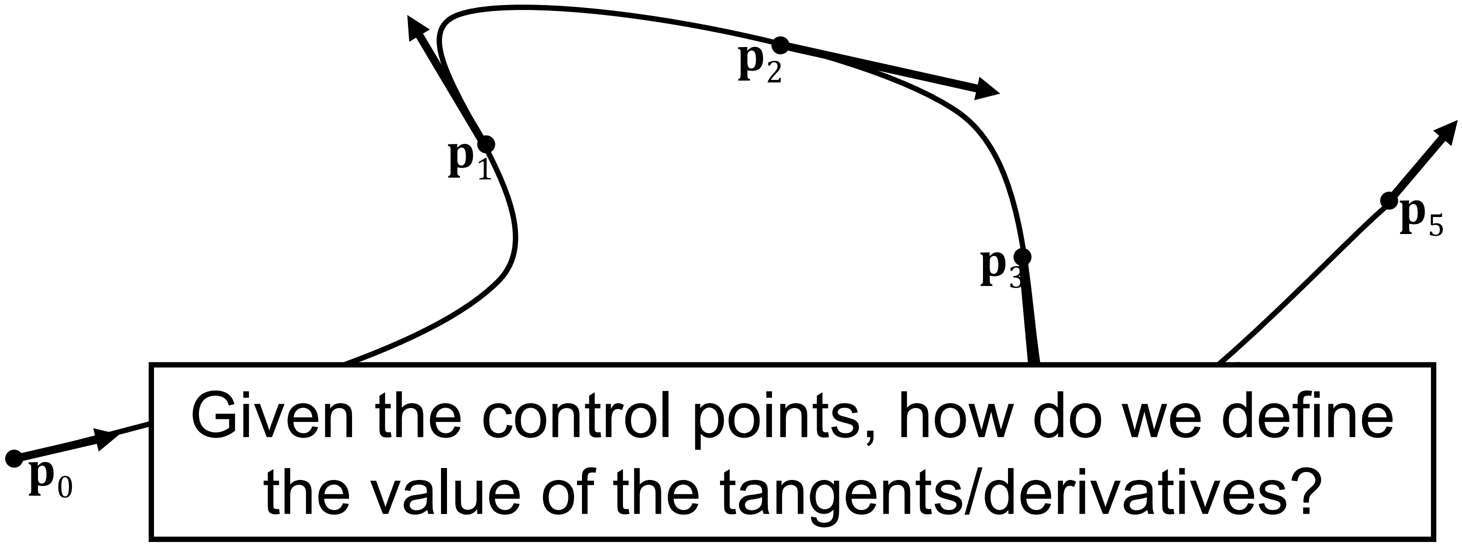
$$\mathbf{P}_k(u) = H_0(u) \cdot \mathbf{p}_k + H_1(u) \cdot \mathbf{p}_{k+1} + H_2(u) \cdot \vec{\mathbf{t}}_k + H_3(u) \cdot \vec{\mathbf{t}}_{k+1}$$

Specific Example: Hermite Splines

Interpolating piecewise *cubic* polynomial, each specified by:

- Start/end positions
- Start/end tangents

Iteratively construct the curve between adjacent end points that interpolate positions and tangents.



Given the control points, how do we define the value of the tangents/derivatives?

Overview

What is a Spline?

Specific Examples:

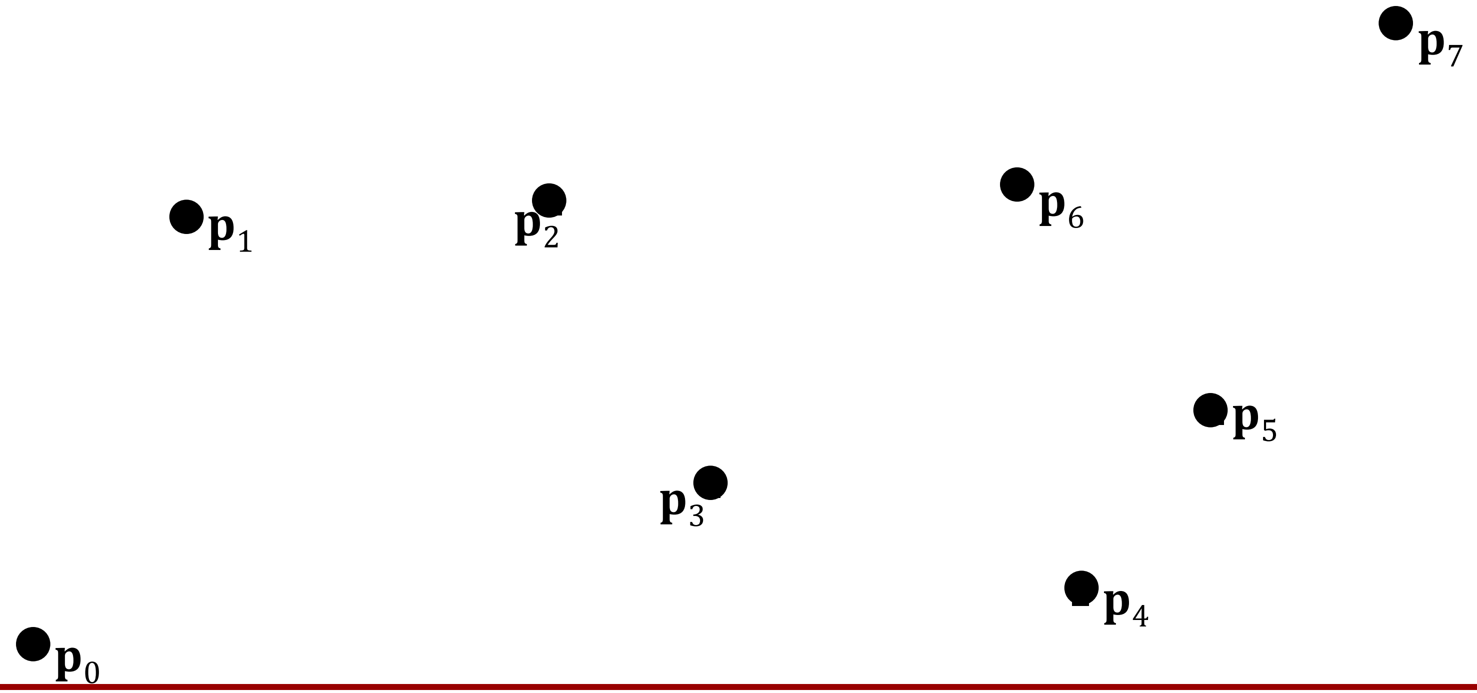
- Hermite Splines
- Cardinal Splines
- Uniform Cubic B-Splines

Comparing Cardinal and Uniform Cubic B-Splines

Specific Example: Cardinal Splines

<u>Interpolating</u> piecewise *cubic* polynomial, each specified by four control points.

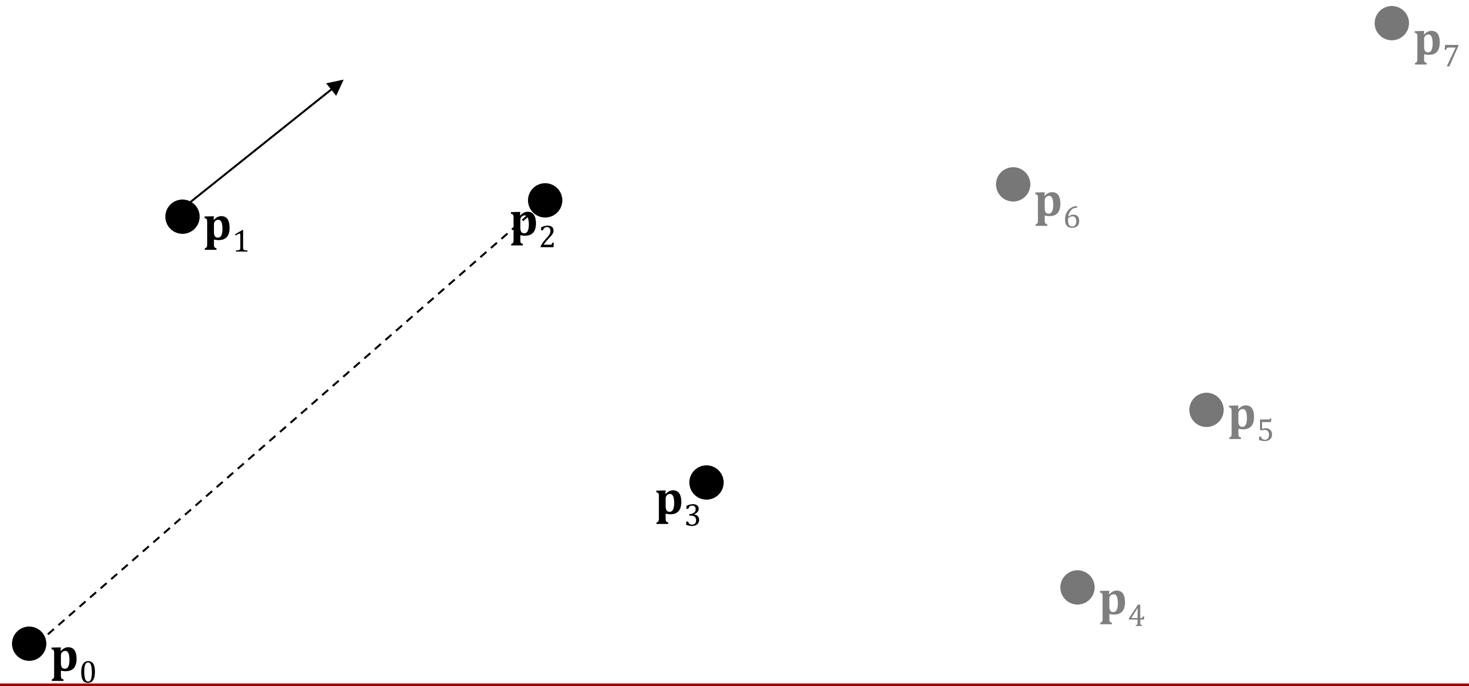
Iteratively construct the curve **through** middle two points **using adjacent points to define tangents**.



Specific Example: Cardinal Splines

Interpolating piecewise *cubic* polynomial, each specified by four control points.

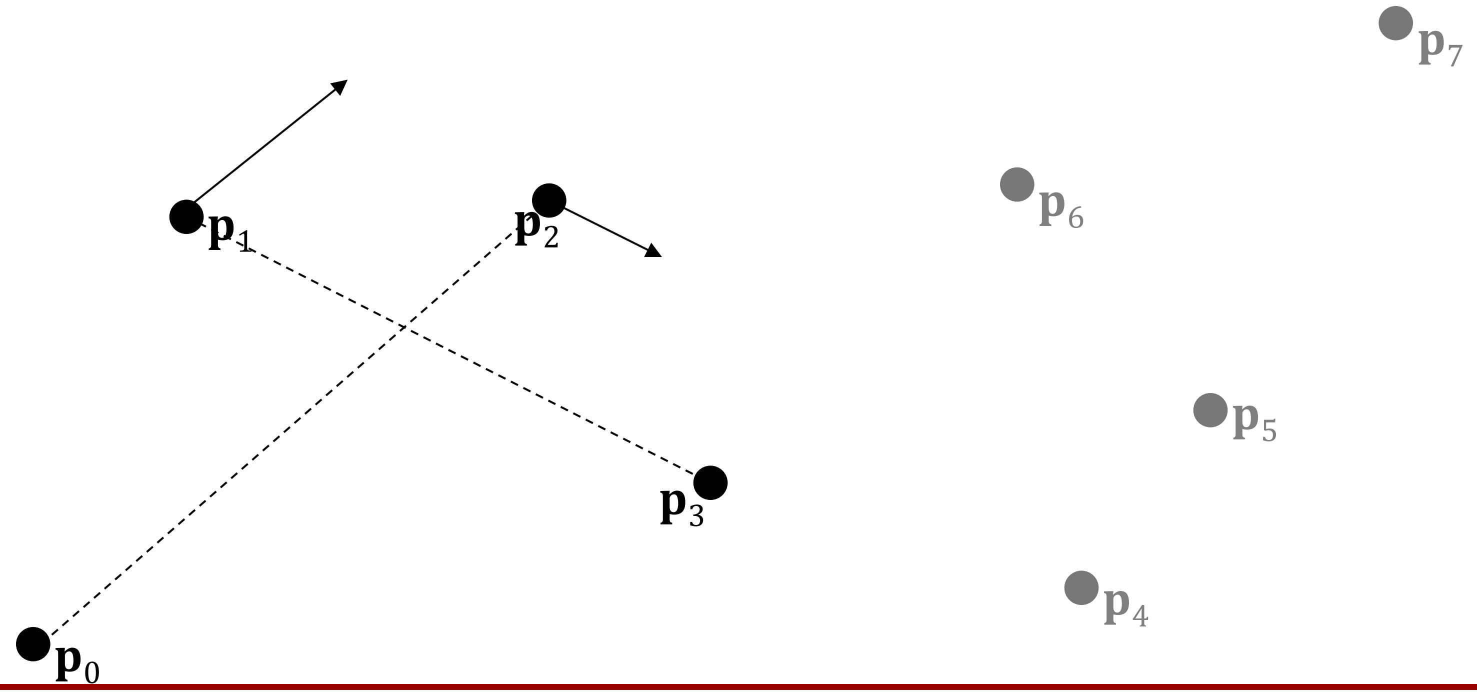
Iteratively construct the curve **through** middle two points **using adjacent points to define tangents**.



Specific Example: Cardinal Splines

Interpolating piecewise *cubic* polynomial, each specified by four control points.

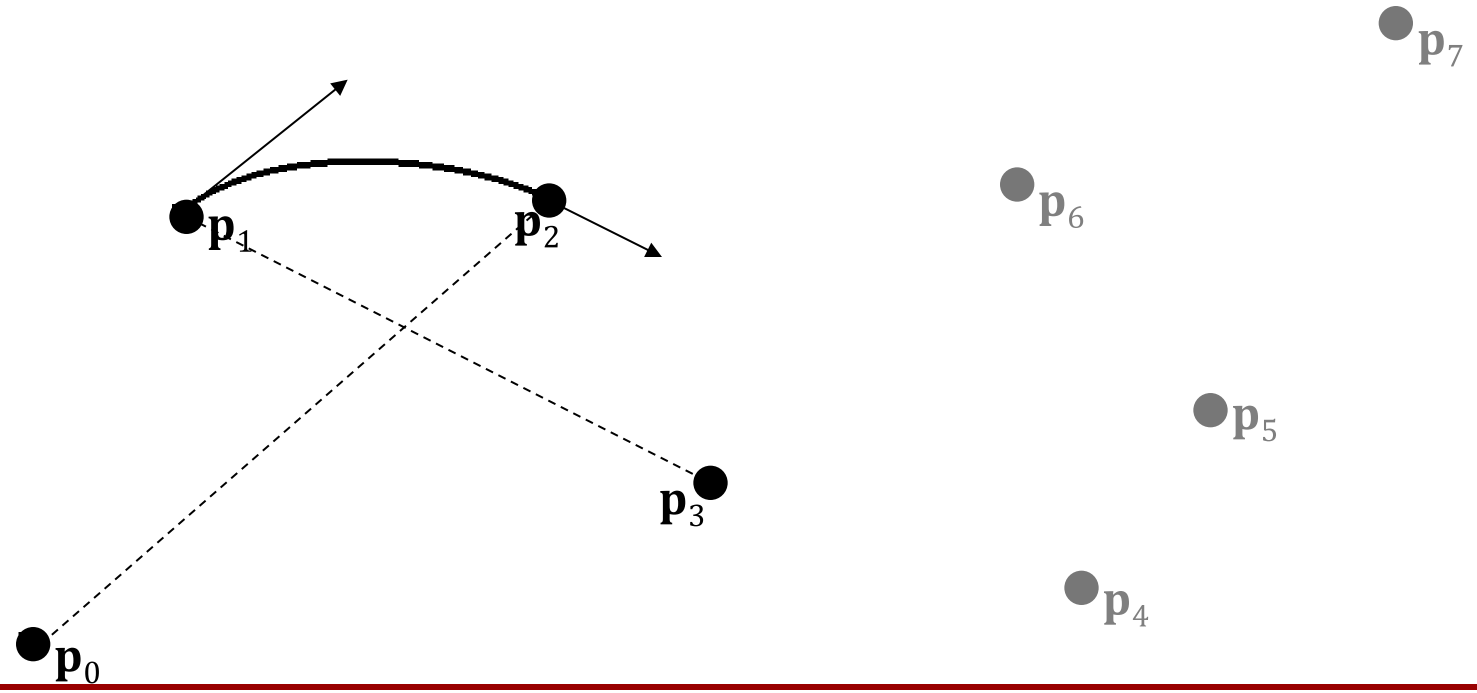
Iteratively construct the curve **through** middle two points **using adjacent points to define tangents**.



Specific Example: Cardinal Splines

Interpolating piecewise *cubic* polynomial, each specified by four control points.

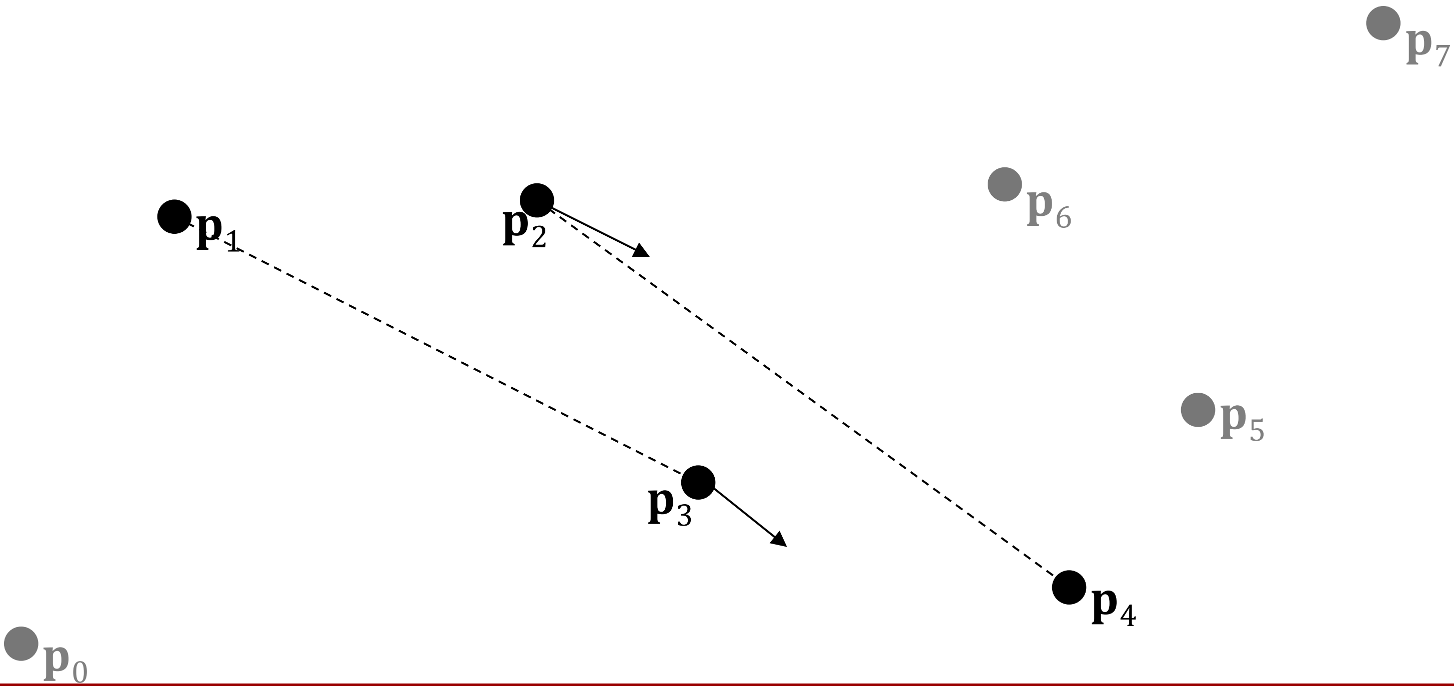
Iteratively construct the curve **through** middle two points **using adjacent points to define tangents**.



Specific Example: Cardinal Splines

Interpolating piecewise *cubic* polynomial, each specified by four control points.

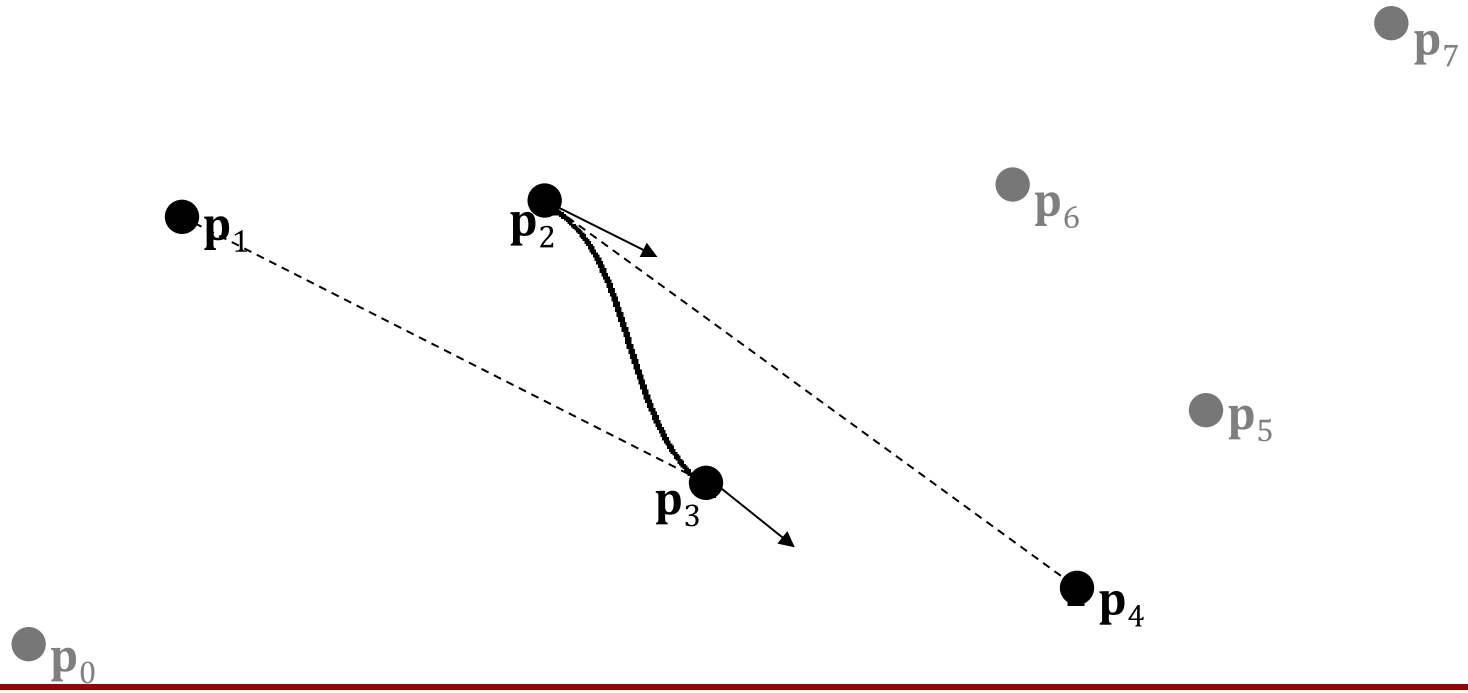
Iteratively construct the curve **through** middle two points **using adjacent points to define tangents**.



Specific Example: Cardinal Splines

Interpolating piecewise *cubic* polynomial, each specified by four control points.

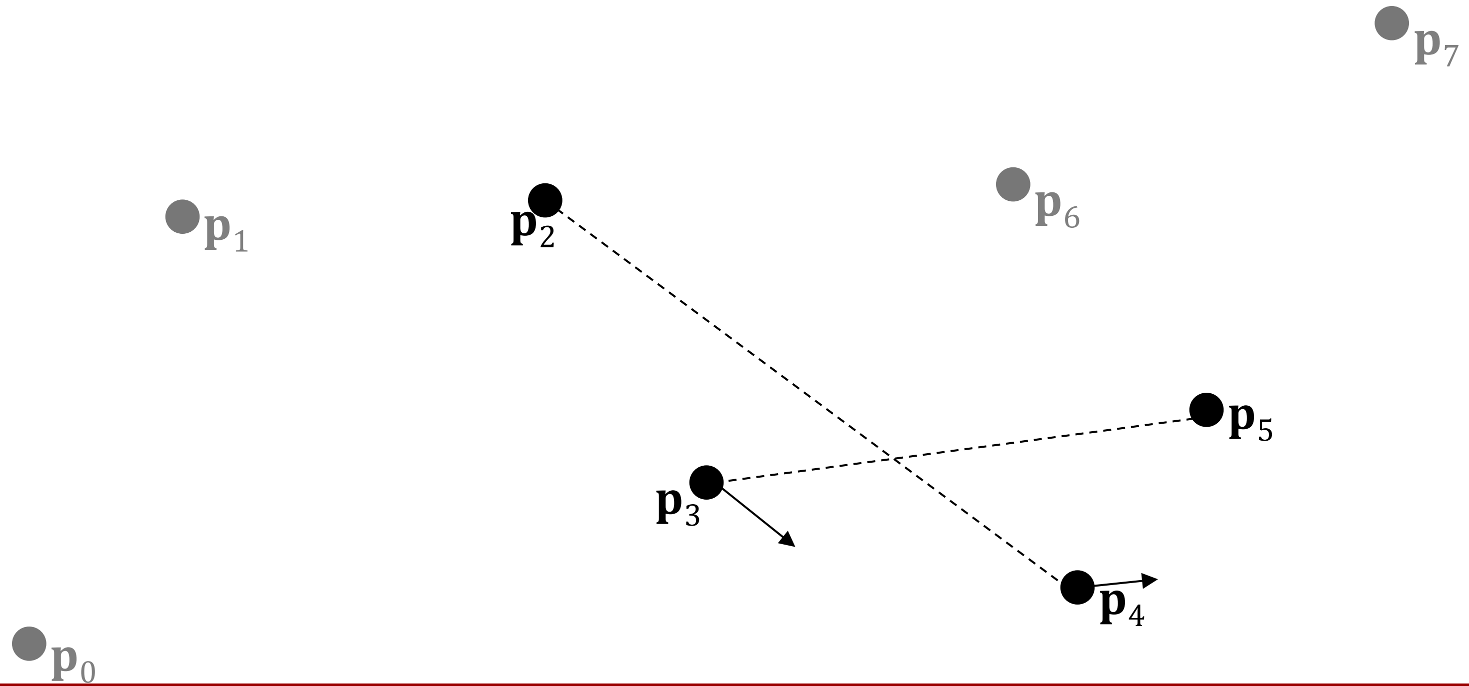
Iteratively construct the curve **through** middle two points **using adjacent points to define tangents**.



Specific Example: Cardinal Splines

Interpolating piecewise *cubic* polynomial, each specified by four control points.

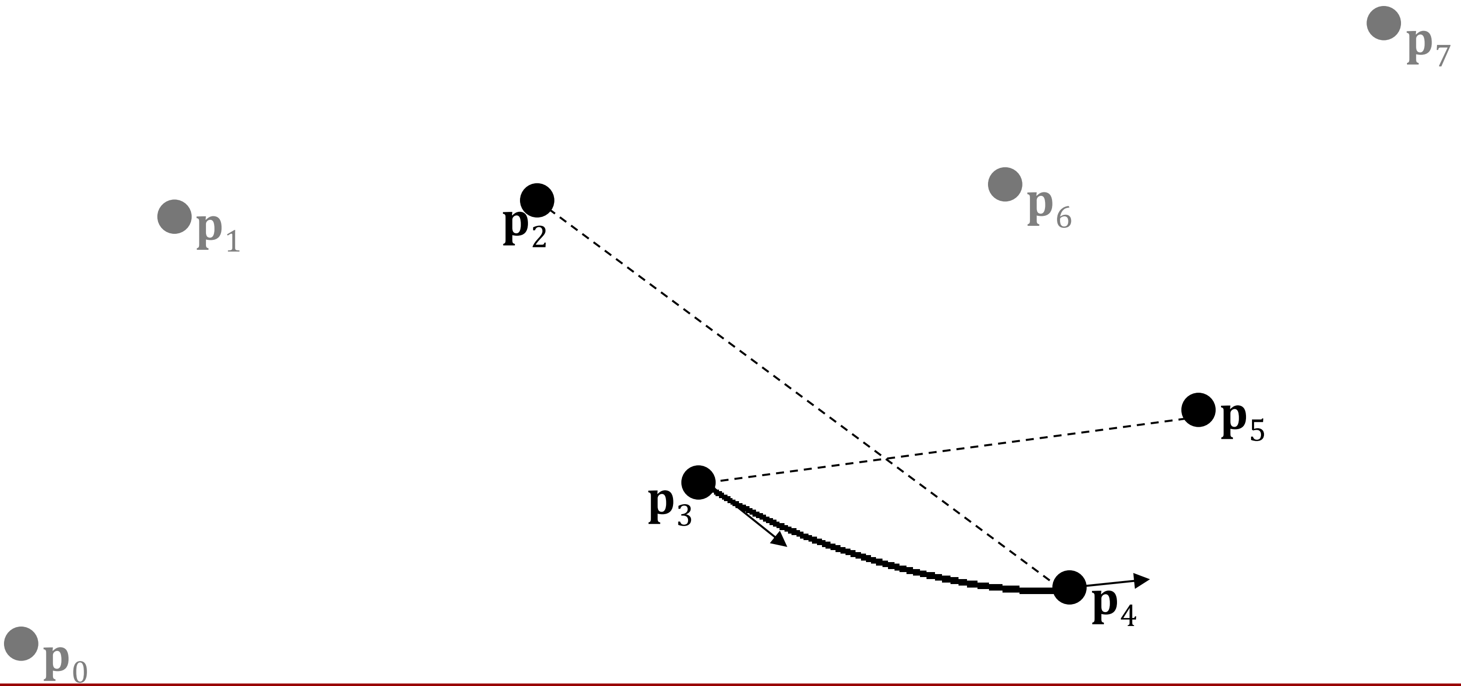
Iteratively construct the curve **through** middle two points **using adjacent points to define tangents**.



Specific Example: Cardinal Splines

Interpolating piecewise *cubic* polynomial, each specified by four control points.

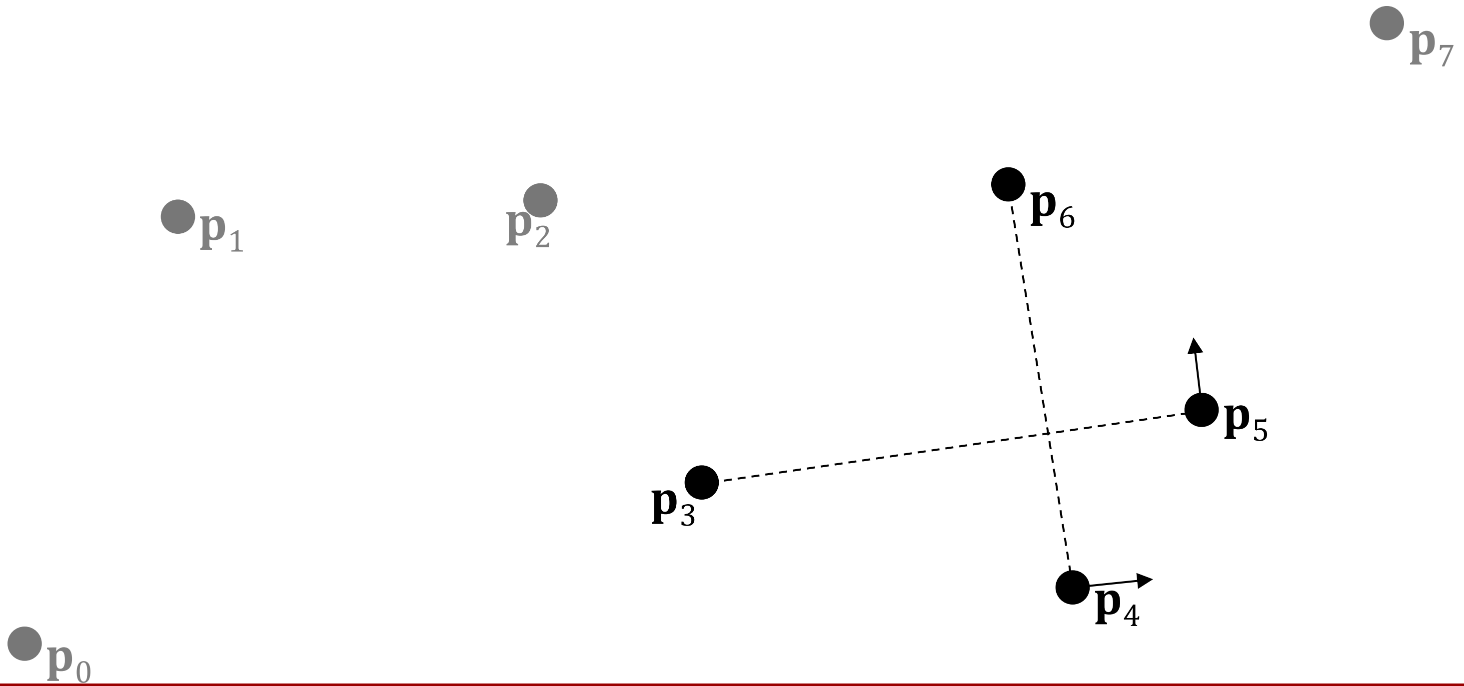
Iteratively construct the curve **through** middle two points **using adjacent points to define tangents**.



Specific Example: Cardinal Splines

<u>Interpolating</u> piecewise *cubic* polynomial, each specified by four control points.

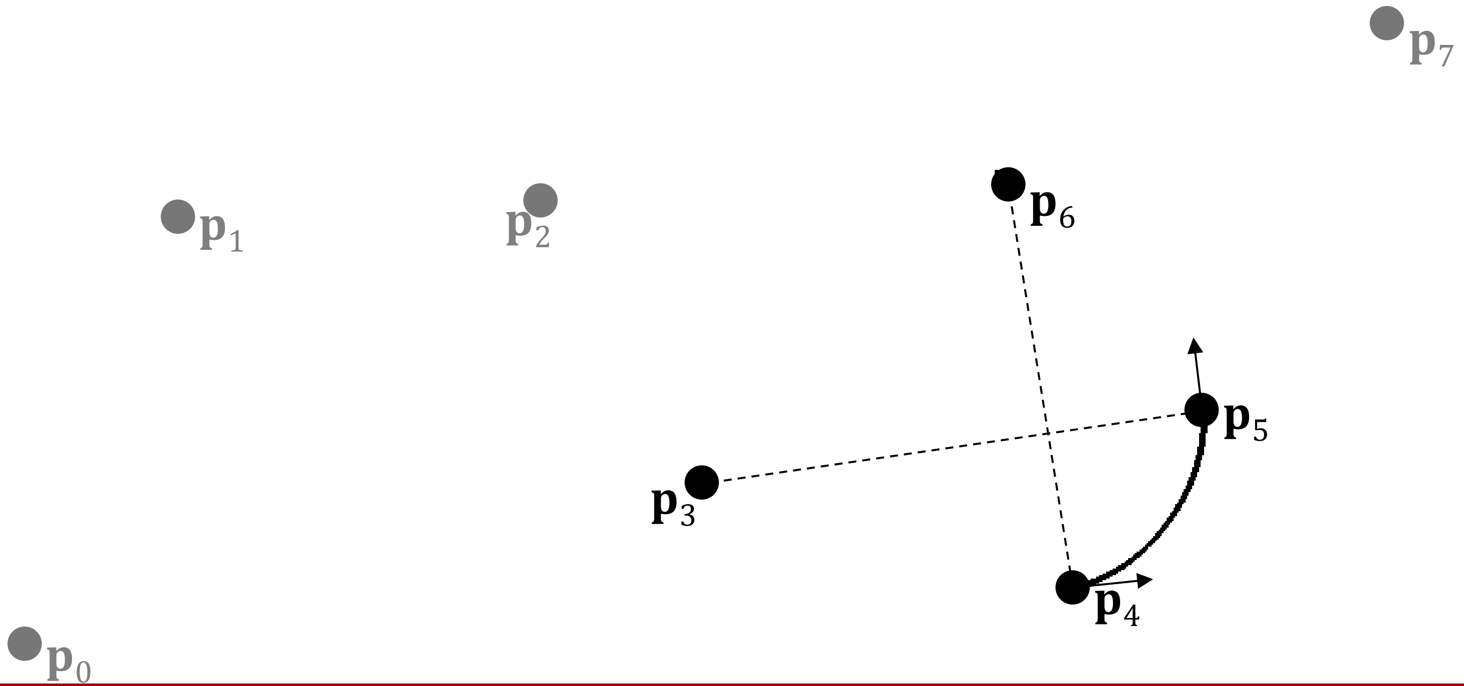
Iteratively construct the curve **through** middle two points **using adjacent points to define tangents**.



Specific Example: Cardinal Splines

Interpolating piecewise *cubic* polynomial, each specified by four control points.

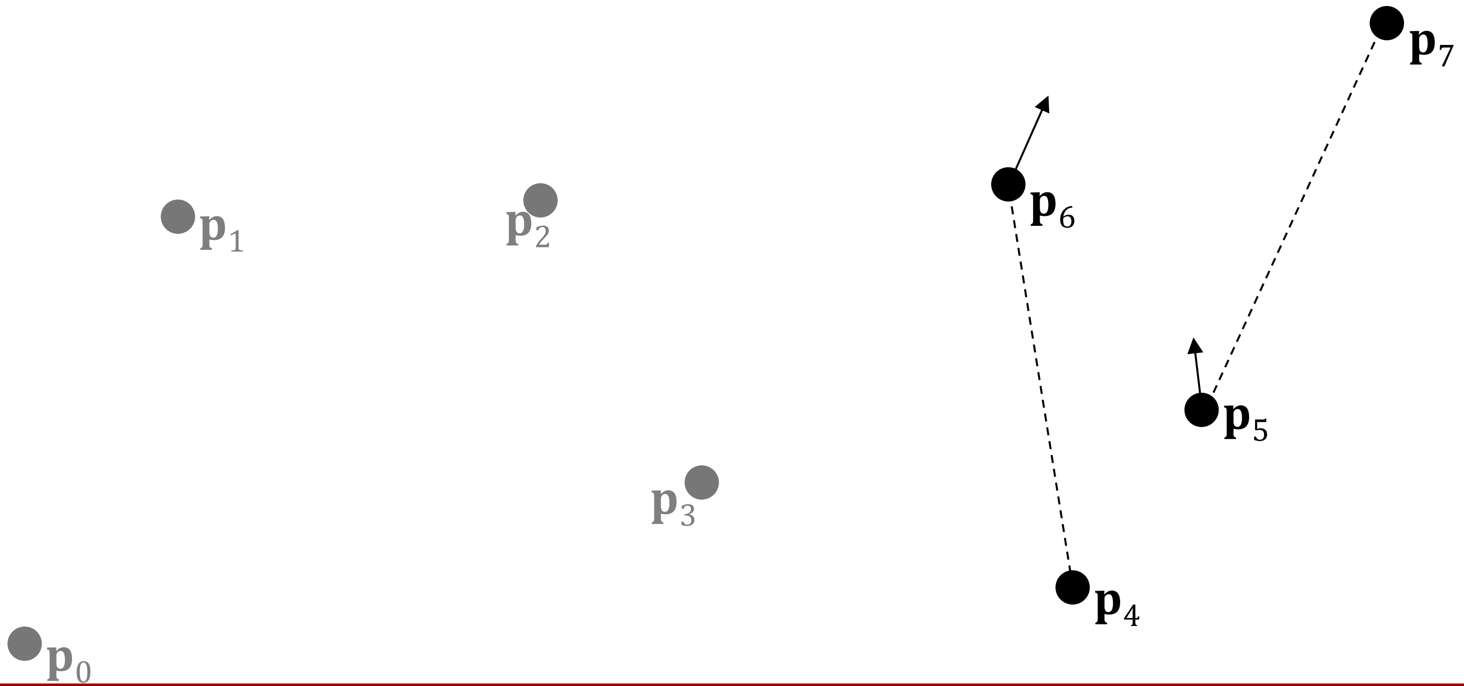
Iteratively construct the curve **through** middle two points **using adjacent points to define tangents**.



Specific Example: Cardinal Splines

Interpolating piecewise *cubic* polynomial, each specified by four control points.

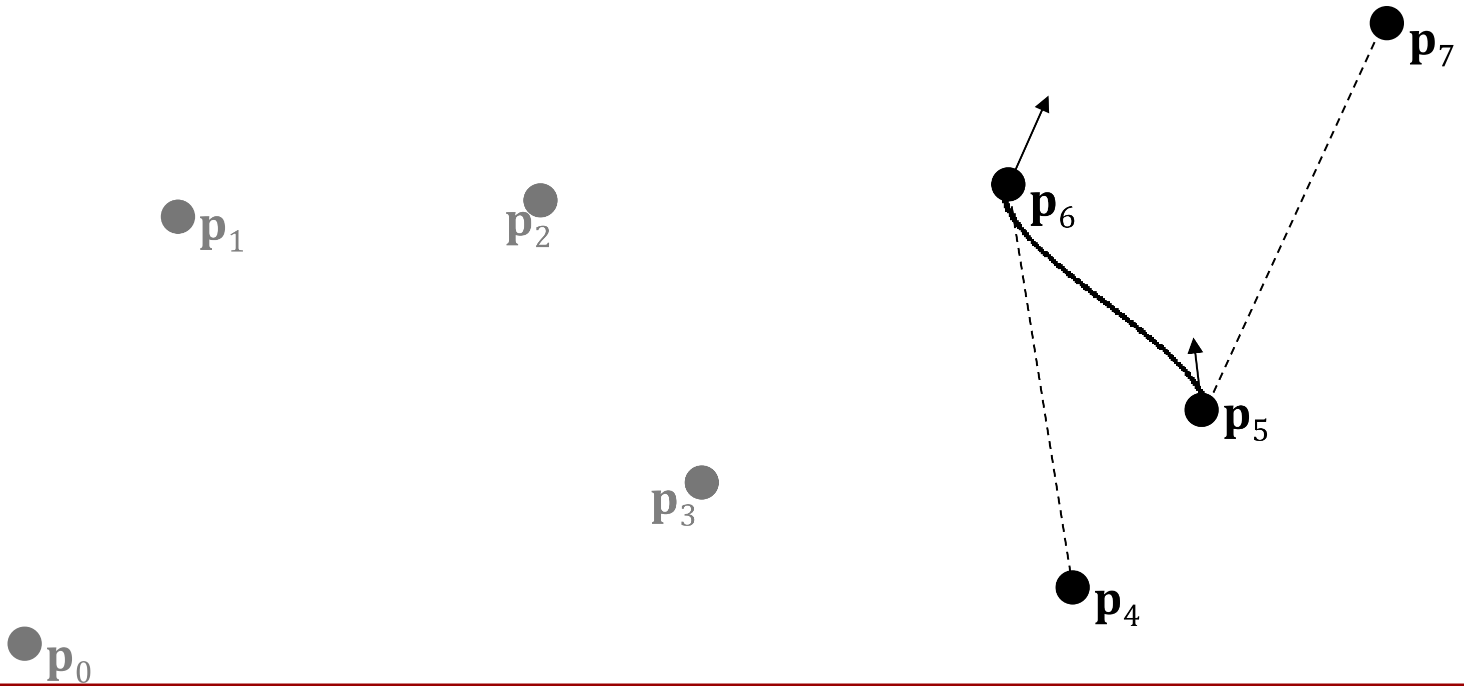
Iteratively construct the curve **through** middle two points **using adjacent points to define tangents**.



Specific Example: Cardinal Splines

<u>Interpolating</u> piecewise *cubic* polynomial, each specified by four control points.

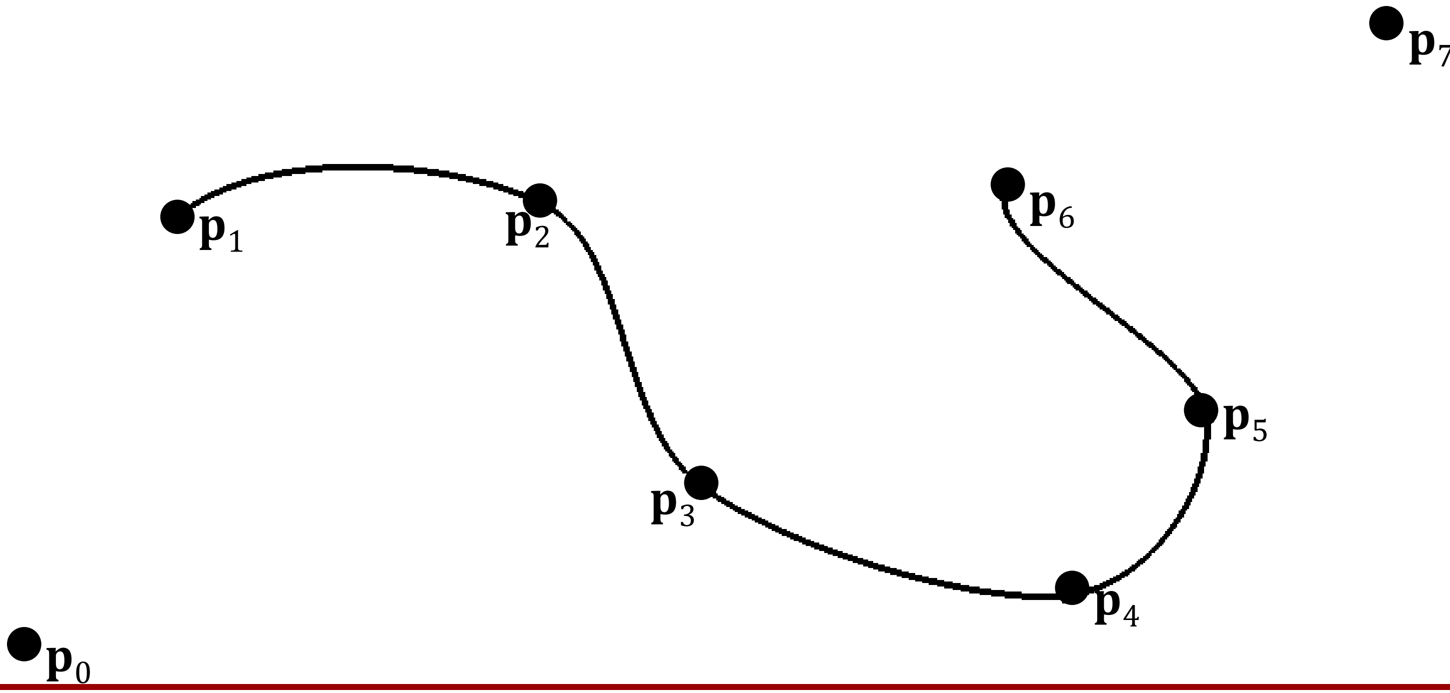
Iteratively construct the curve **through** middle two points **using adjacent points to define tangents**.



Specific Example: Cardinal Splines

Interpolating piecewise *cubic* polynomial, each specified by four control points.

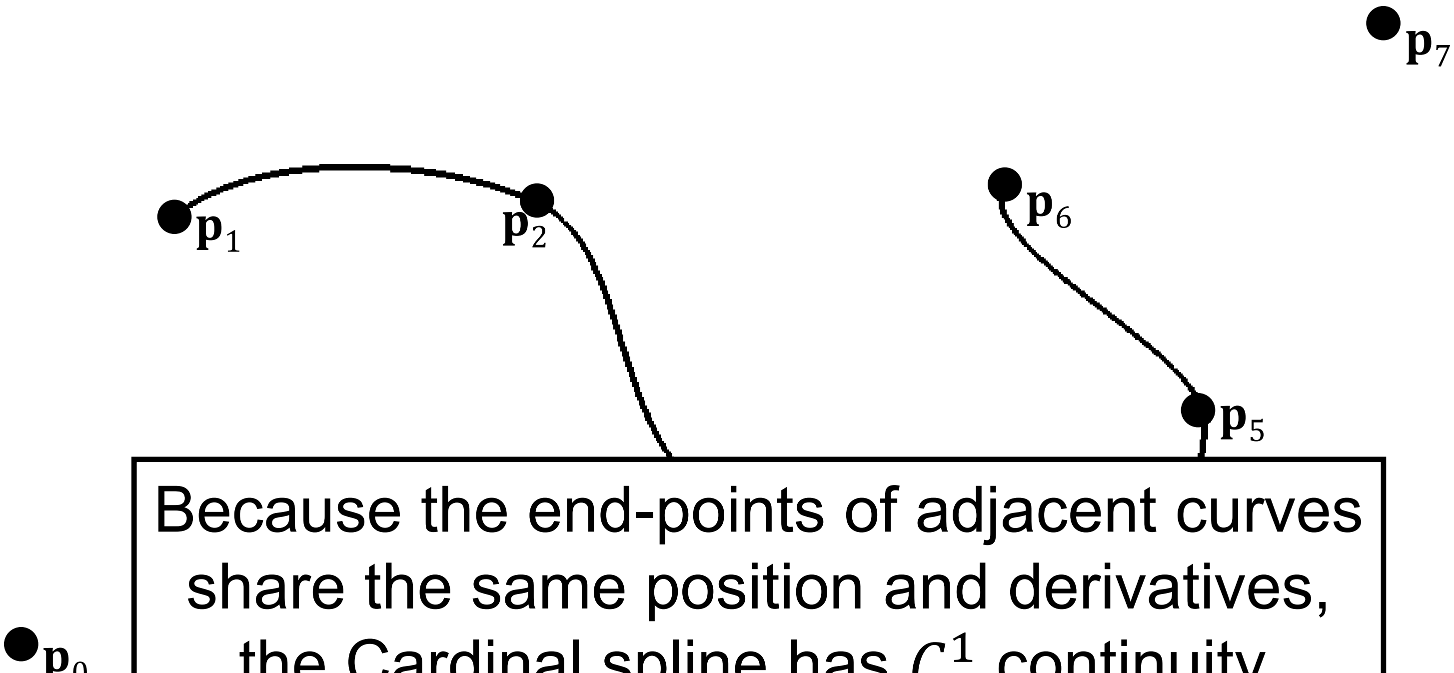
Iteratively construct the curve **through** middle two points **using adjacent points to define tangents**.



Specific Example: Cardinal Splines

<u>Interpolating</u> piecewise *cubic* polynomial, each specified by four control points.

Iteratively construct the curve **through** middle two points **using adjacent points to define tangents**.



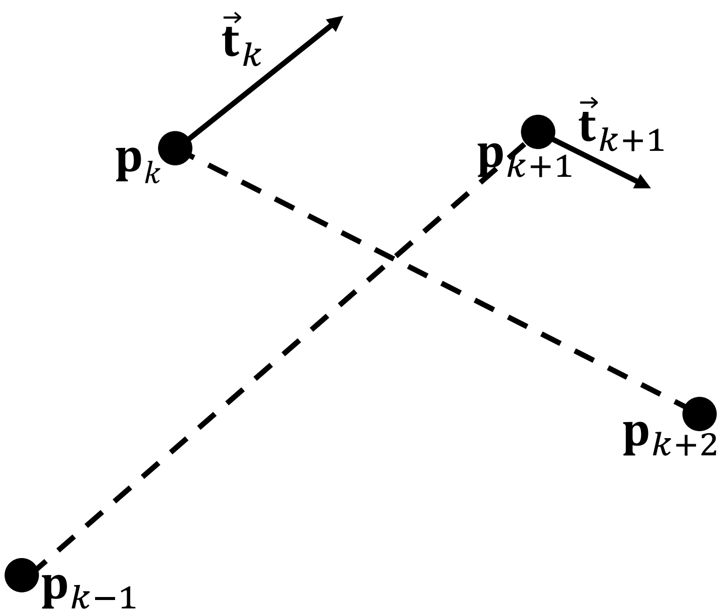
Because the end-points of adjacent curves share the same position and derivatives, the Cardinal spline has C^1 continuity.

Specific Example: Cardinal Splines

Using Hermite splines, we have:

$$\mathbf{P}_k(u) = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix} \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix}$$

$\mathbf{M}_{\text{Hermite}}$



$$\vec{\mathbf{t}}_k = \tau(\mathbf{p}_{k+1} - \mathbf{p}_{k-1})$$

$$\vec{\mathbf{t}}_{k+1} = \tau(\mathbf{p}_{k+2} - \mathbf{p}_k)$$

Specific Example: Cardinal Splines

We can express the boundary constraints as:

$$\begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \tau(\mathbf{p}_{k+1} - \mathbf{p}_{k-1}) \\ \tau(\mathbf{p}_{k+2} - \mathbf{p}_k) \end{pmatrix}$$

⇒ Using Hermite splines, we get:

$$\mathbf{P}_k(u) = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix} \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix}$$

$\mathbf{M}_{\text{Hermite}}$

Specific Example: Cardinal Splines

We can express the boundary constraints as:

$$\begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \tau(\mathbf{p}_{k+1} - \mathbf{p}_{k-1}) \\ \tau(\mathbf{p}_{k+2} - \mathbf{p}_k) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\tau & 0 & \tau & 0 \\ 0 & -\tau & 0 & \tau \end{pmatrix} \begin{pmatrix} \mathbf{p}_{k-1} \\ \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{p}_{k+2} \end{pmatrix}$$

⇒ Using Hermite splines, we get:

$$\mathbf{P}_k(u) = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix} \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix}$$

$\mathbf{M}_{\text{Hermite}}$

Specific Example: Cardinal Splines

We can express the boundary constraints as:

$$\begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \tau(\mathbf{p}_{k+1} - \mathbf{p}_{k-1}) \\ \tau(\mathbf{p}_{k+2} - \mathbf{p}_k) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\tau & 0 & \tau & 0 \\ 0 & -\tau & 0 & \tau \end{pmatrix} \begin{pmatrix} \mathbf{p}_{k-1} \\ \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{p}_{k+2} \end{pmatrix}$$

⇒ Using Hermite splines, we get:

$$\mathbf{P}_k(u) = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix} \underbrace{\begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}}_{\mathbf{M}_{\text{Hermite}}} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\tau & 0 & \tau & 0 \\ 0 & -\tau & 0 & \tau \end{pmatrix} \begin{pmatrix} \mathbf{p}_{k-1} \\ \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{p}_{k+2} \end{pmatrix}$$

Specific Example: Cardinal Splines

$$\mathbf{P}_k(u) = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix} \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\tau & 0 & \tau & 0 \\ 0 & -\tau & 0 & \tau \end{pmatrix} \begin{pmatrix} \mathbf{p}_{k-1} \\ \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{p}_{k+2} \end{pmatrix}$$

Multiplying, we get the Cardinal matrix representation:

$$\mathbf{P}_k(u) = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix} \begin{pmatrix} -\tau & 2-\tau & \tau-2 & \tau \\ 2\tau & \tau-3 & 3-2\tau & -\tau \\ -\tau & 0 & \tau & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{p}_{k-1} \\ \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{p}_{k+2} \end{pmatrix}$$

$\mathbf{M}_{\text{Cardinal}}$

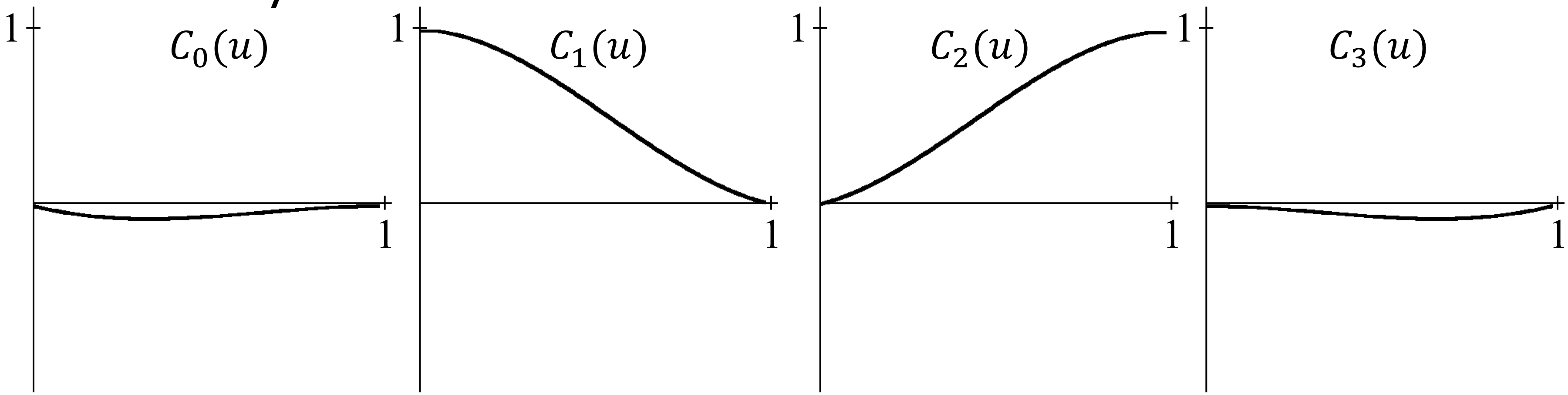
Specific Example: Cardinal Splines

Setting:

- $C_0(u) = -\tau u^3 + 2\tau u^2 - \tau u$
- $C_1(u) = (2-\tau)u^3 + (\tau - 3)u^2 + 1$
- $C_2(u) = (\tau - 2)u^3 + (3 - 2\tau)u^2 + \tau u$
- $C_3(u) = \tau u^3 - \tau u^2$

} Blending Functions

For $\tau = 1/2$:



$$\mathbf{P}_k(u) = C_0(u) \cdot \mathbf{p}_{k-1} + C_1(u) \cdot \mathbf{p}_k + C_2(u) \cdot \mathbf{p}_{k+1} + C_3(u) \cdot \mathbf{p}_{k+2}$$

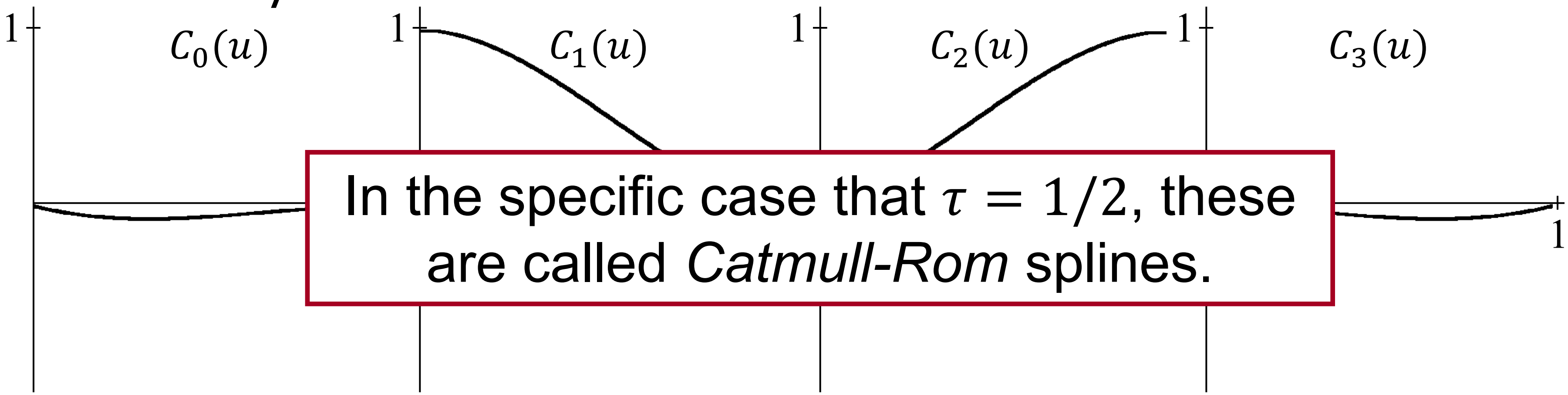
Specific Example: Cardinal Splines

Setting:

- $C_0(u) = -\tau u^3 + 2\tau u^2 - \tau u$
- $C_1(u) = (2-\tau)u^3 + (\tau - 3)u^2 + 1$
- $C_2(u) = (\tau - 2)u^3 + (3 - 2\tau)u^2 + \tau u$
- $C_3(u) = \tau u^3 - \tau u^2$

} Blending Functions

For $\tau = 1/2$:



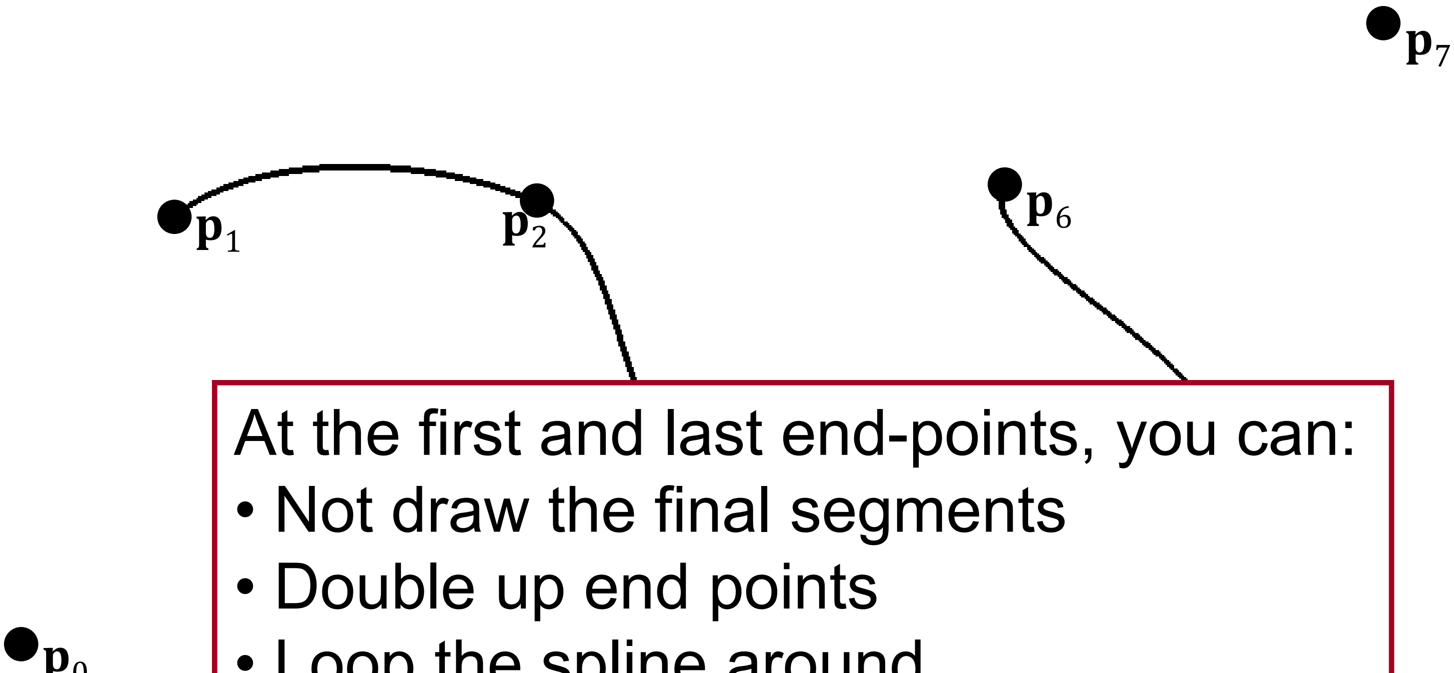
In the specific case that $\tau = 1/2$, these are called *Catmull-Rom* splines.

$$\mathbf{P}_k(u) = C_0(u) \cdot \mathbf{p}_{k-1} + C_1(u) \cdot \mathbf{p}_k + C_2(u) \cdot \mathbf{p}_{k+1} + C_3(u) \cdot \mathbf{p}_{k+2}$$

Specific Example: Cardinal Splines

<u>Interpolating</u> piecewise *cubic* polynomial, each specified by four control points.

Iteratively construct the curve **through** middle two points **using adjacent points to define tangents**.



At the first and last end-points, you can:

- Not draw the final segments
- Double up end points
- Loop the spline around

Overview

What is a Spline?

Specific Examples:

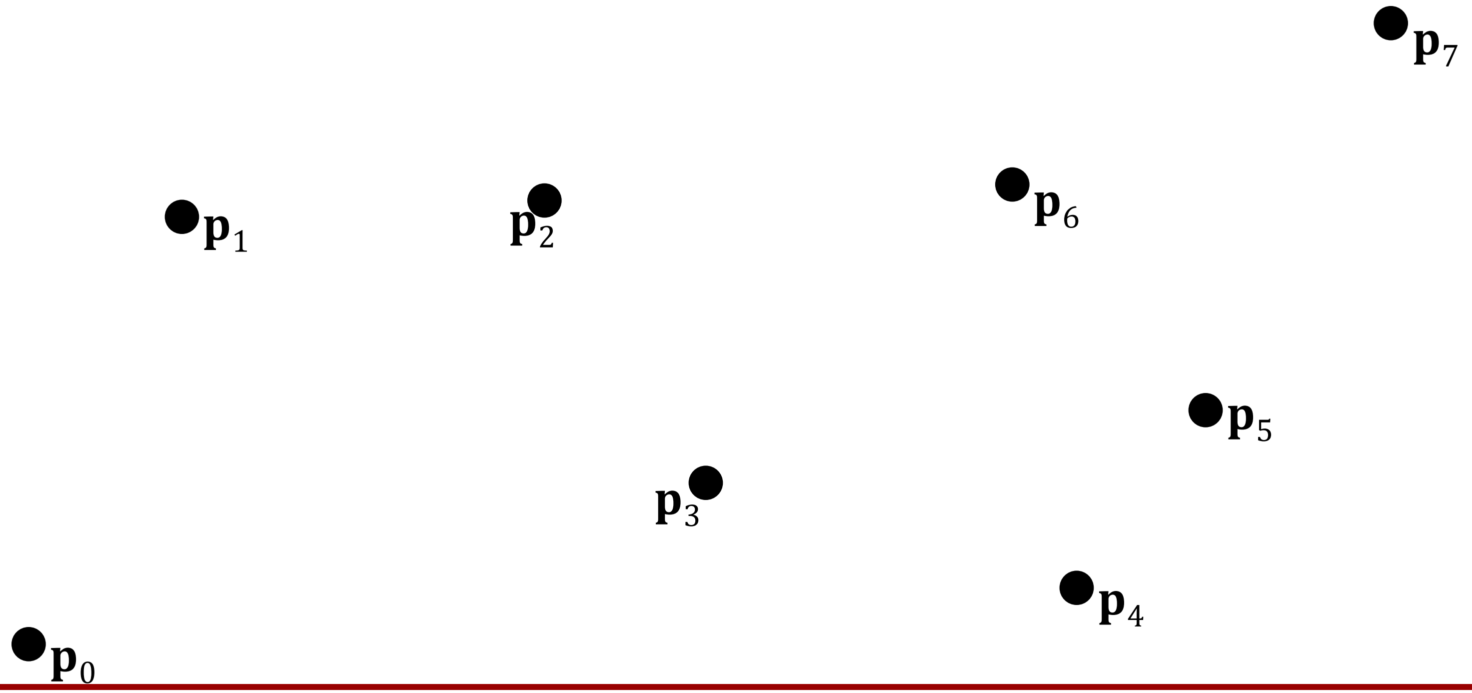
- Hermite Splines
- Cardinal Splines
- Uniform Cubic B-Splines

Comparing Cardinal and Uniform Cubic B-Splines

Specific Example: Uniform Cubic B-Splines

Approximating piecewise *cubic* polynomial, each specified by four control points.

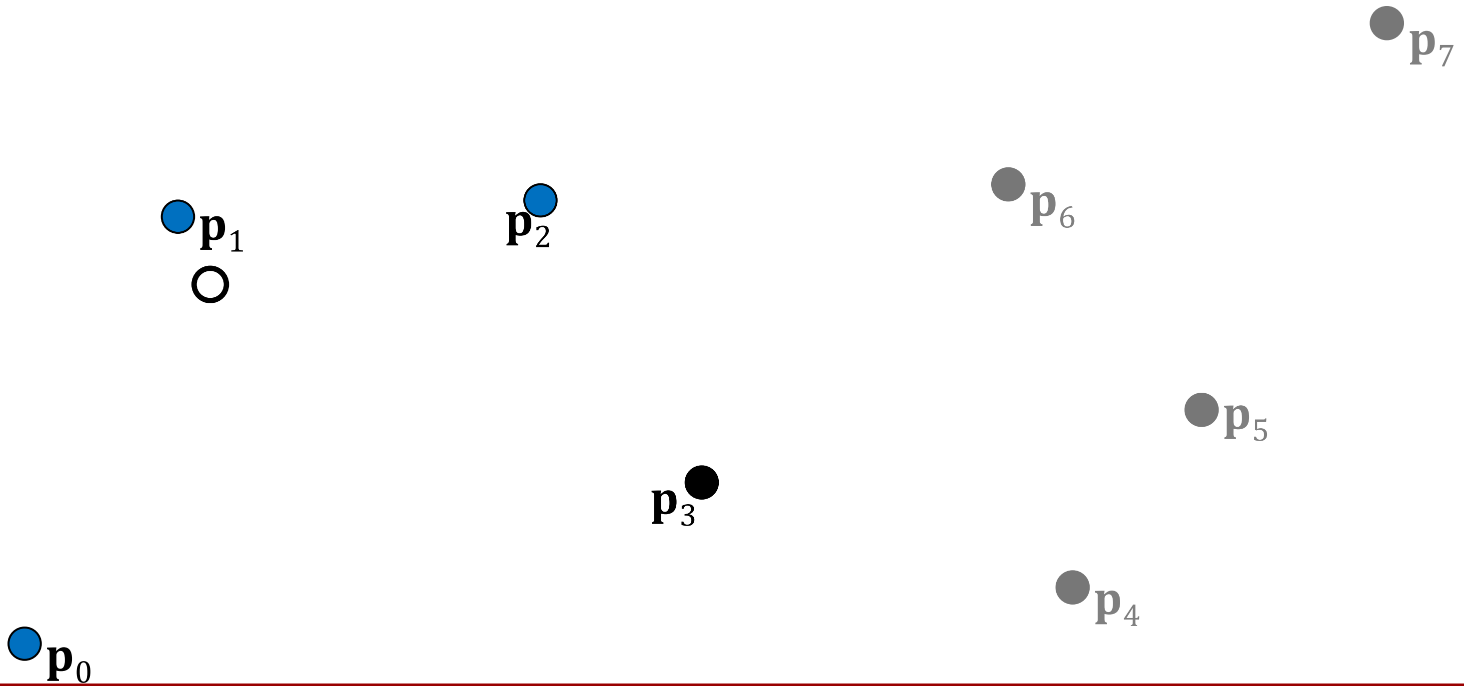
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

Approximating piecewise *cubic* polynomial, each specified by four control points.

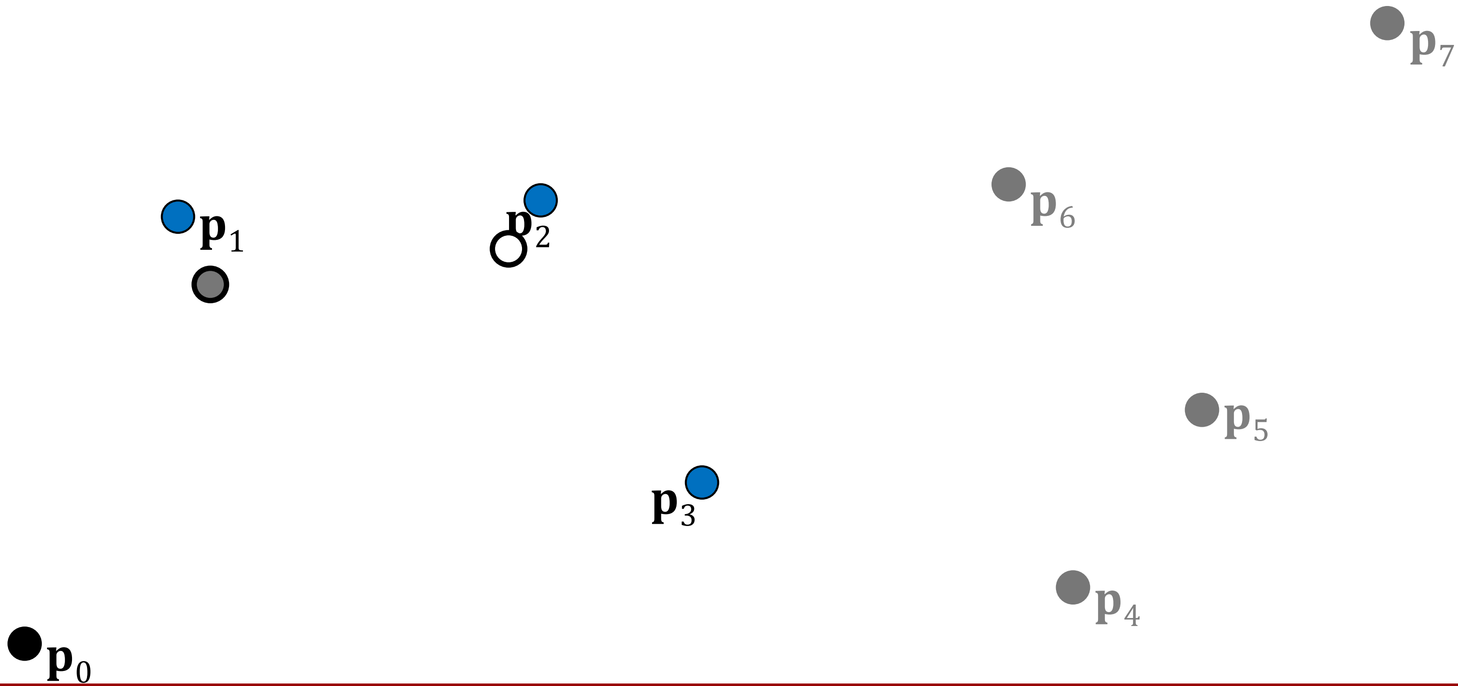
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

Approximating piecewise *cubic* polynomial, each specified by four control points.

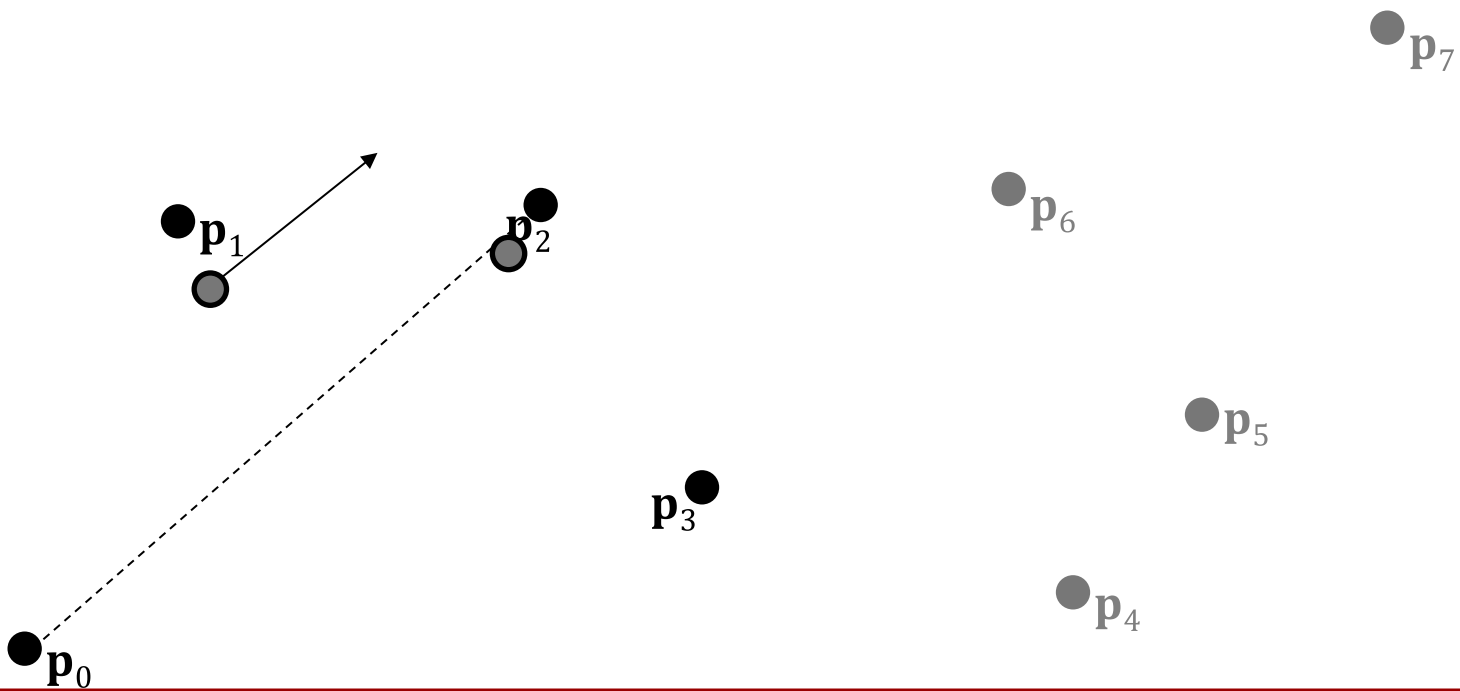
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

<u>Approximating</u> piecewise *cubic* polynomial, each specified by four control points.

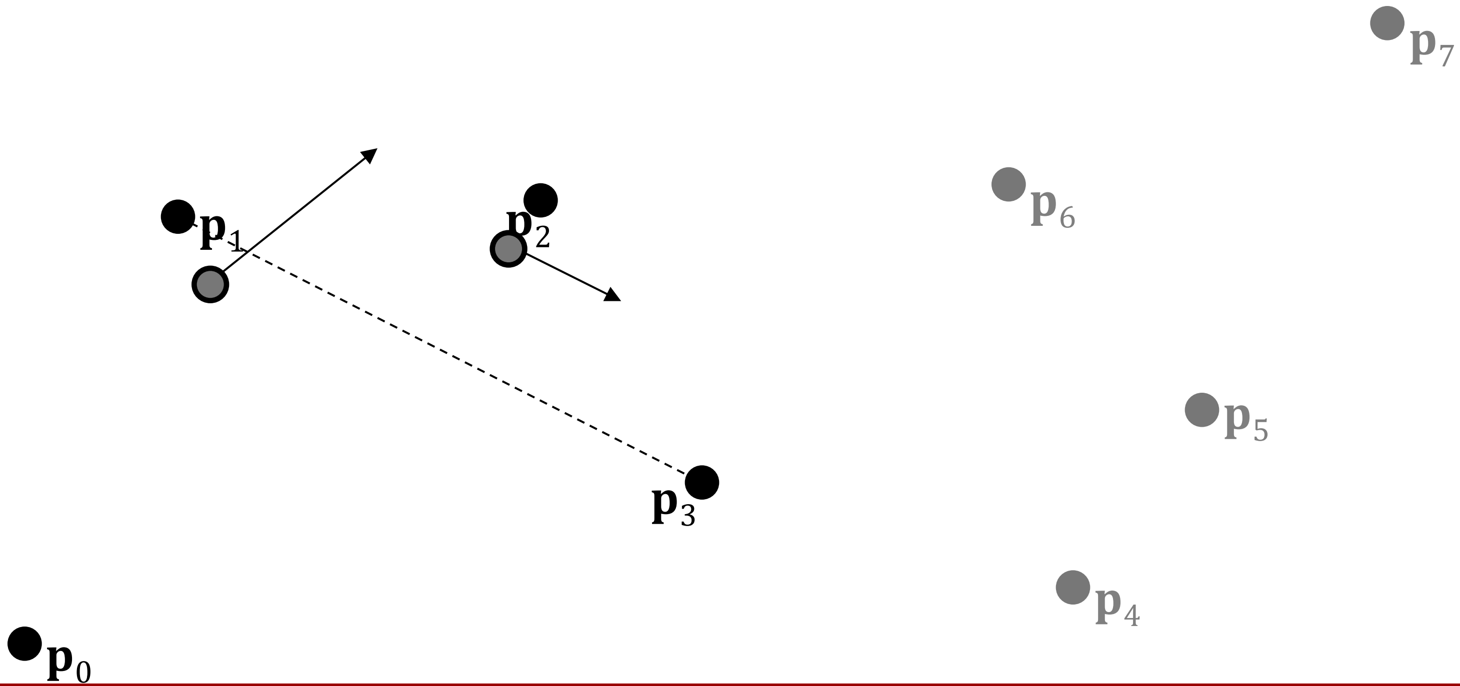
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

<u>Approximating</u> piecewise *cubic* polynomial, each specified by four control points.

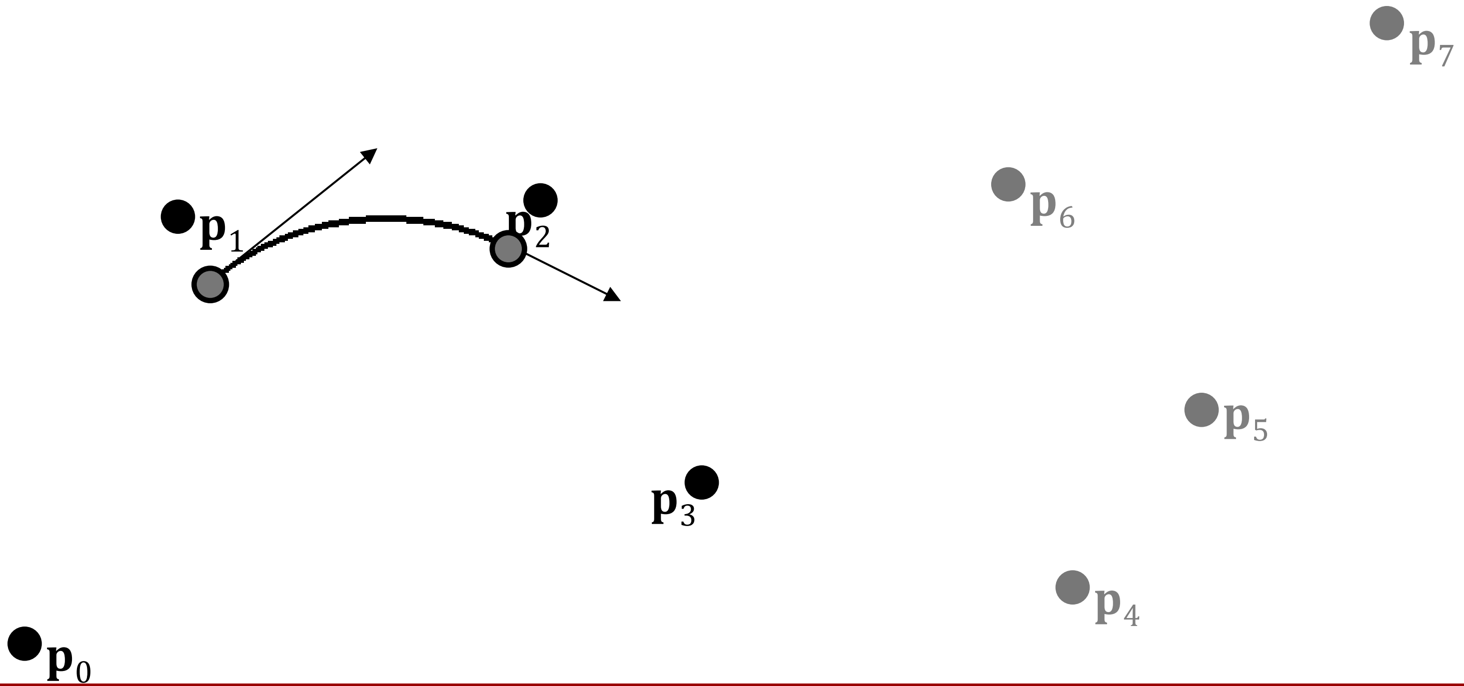
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

Approximating piecewise *cubic* polynomial, each specified by four control points.

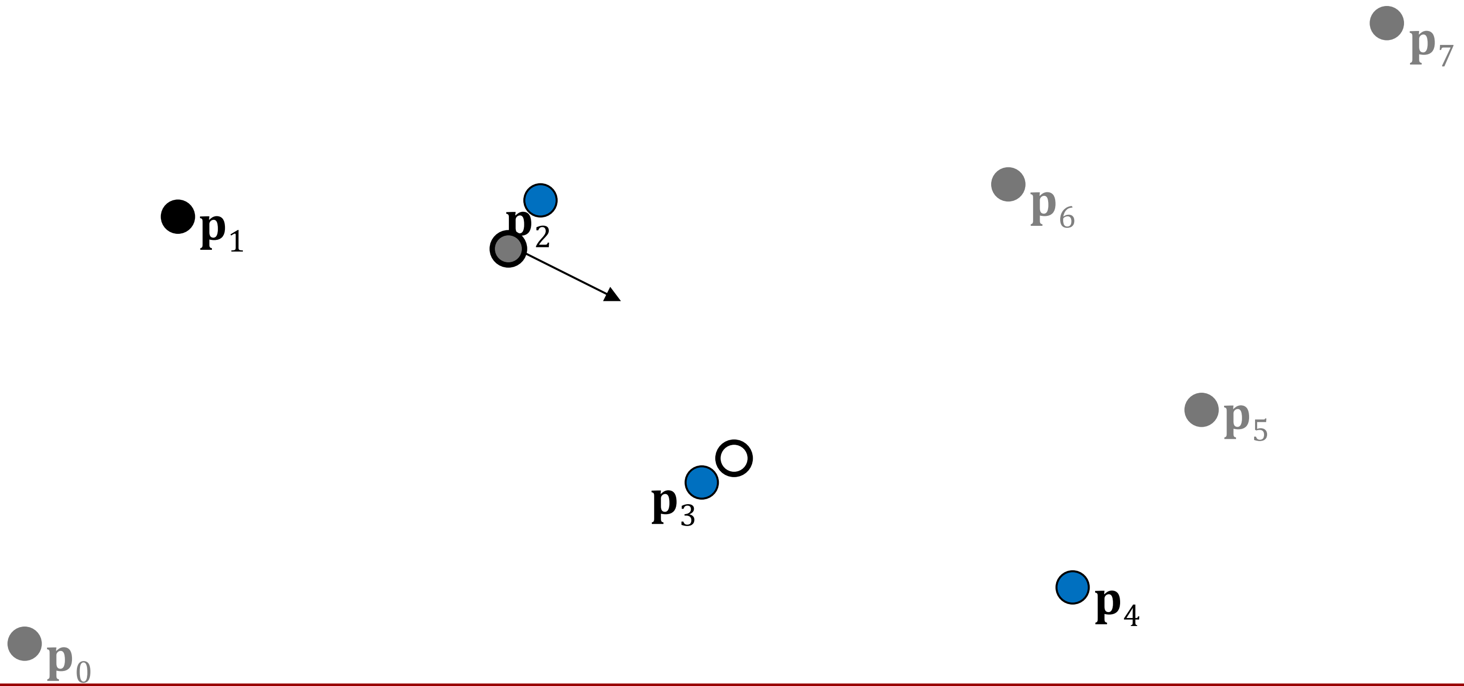
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

<u>Approximating</u> piecewise *cubic* polynomial, each specified by four control points.

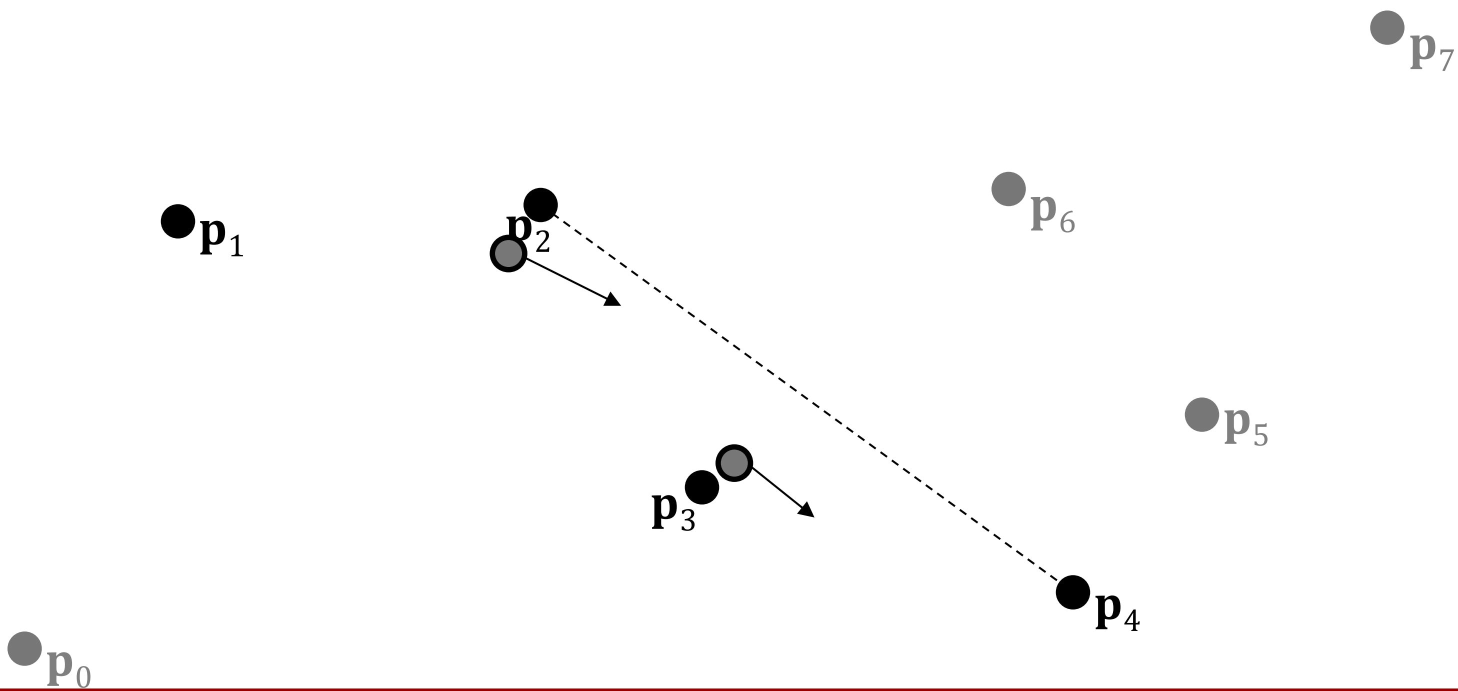
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

Approximating piecewise *cubic* polynomial, each specified by four control points.

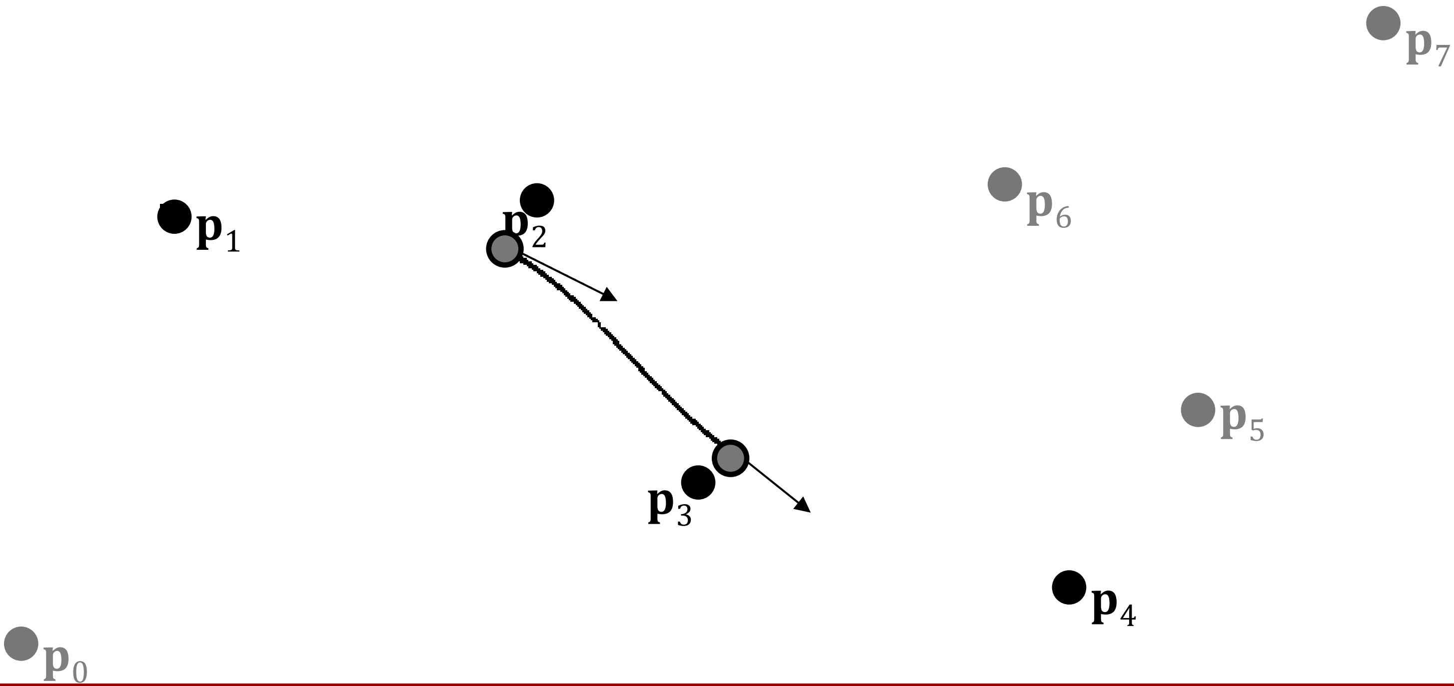
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

<u>Approximating</u> piecewise *cubic* polynomial, each specified by four control points.

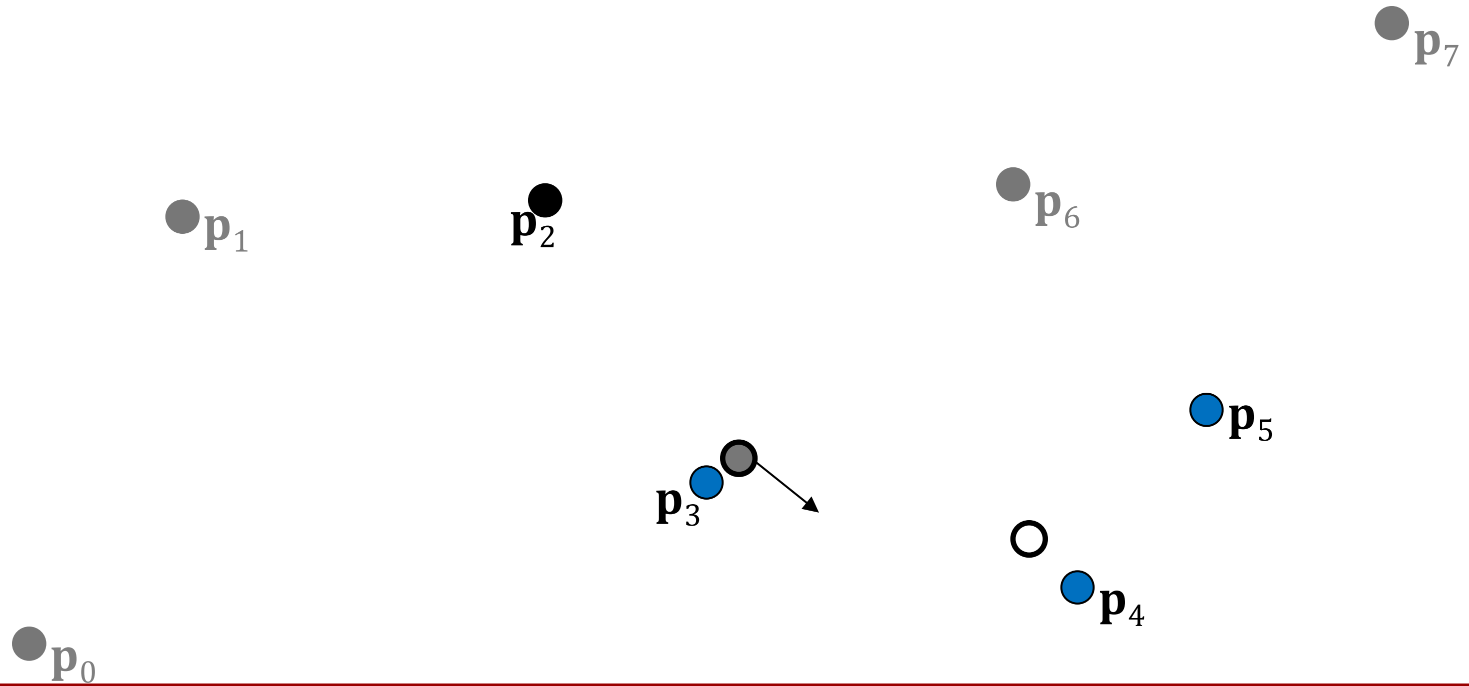
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

<u>Approximating</u> piecewise *cubic* polynomial, each specified by four control points.

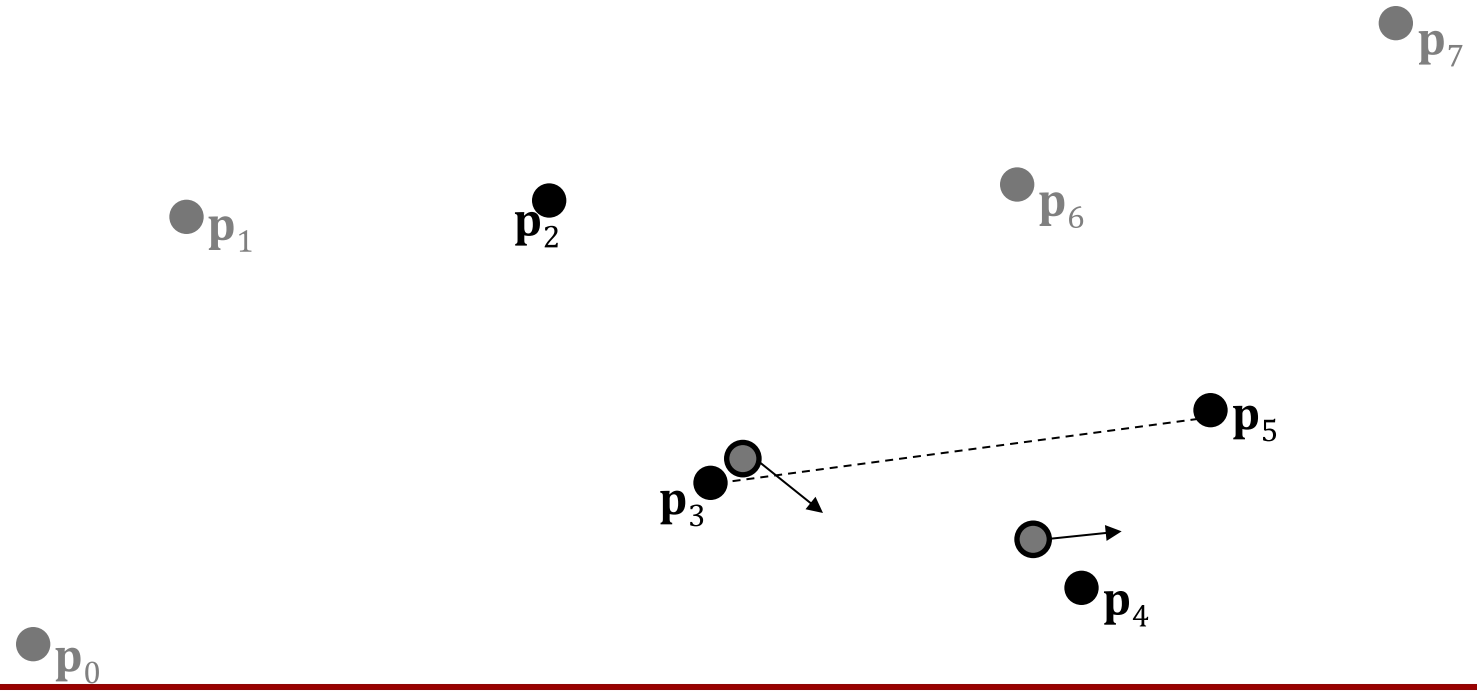
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

<u>Approximating</u> piecewise *cubic* polynomial, each specified by four control points.

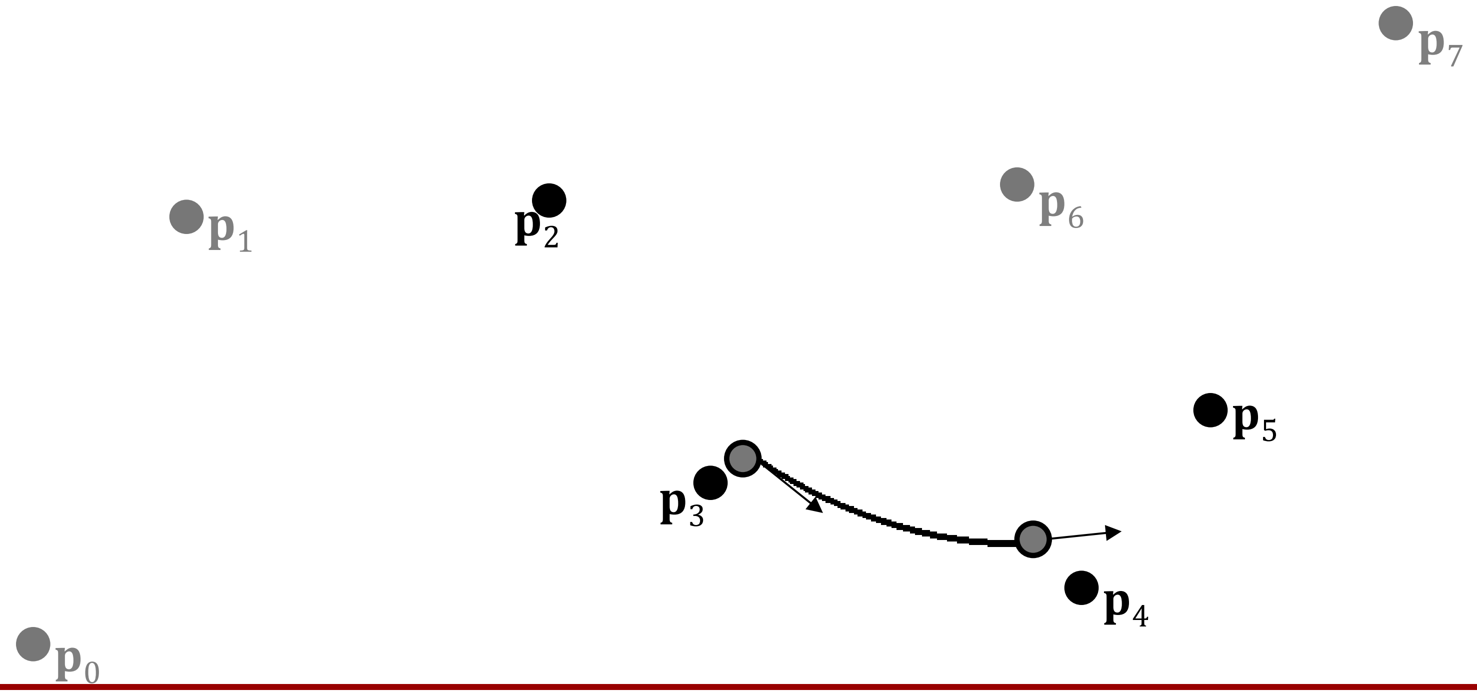
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

Approximating piecewise *cubic* polynomial, each specified by four control points.

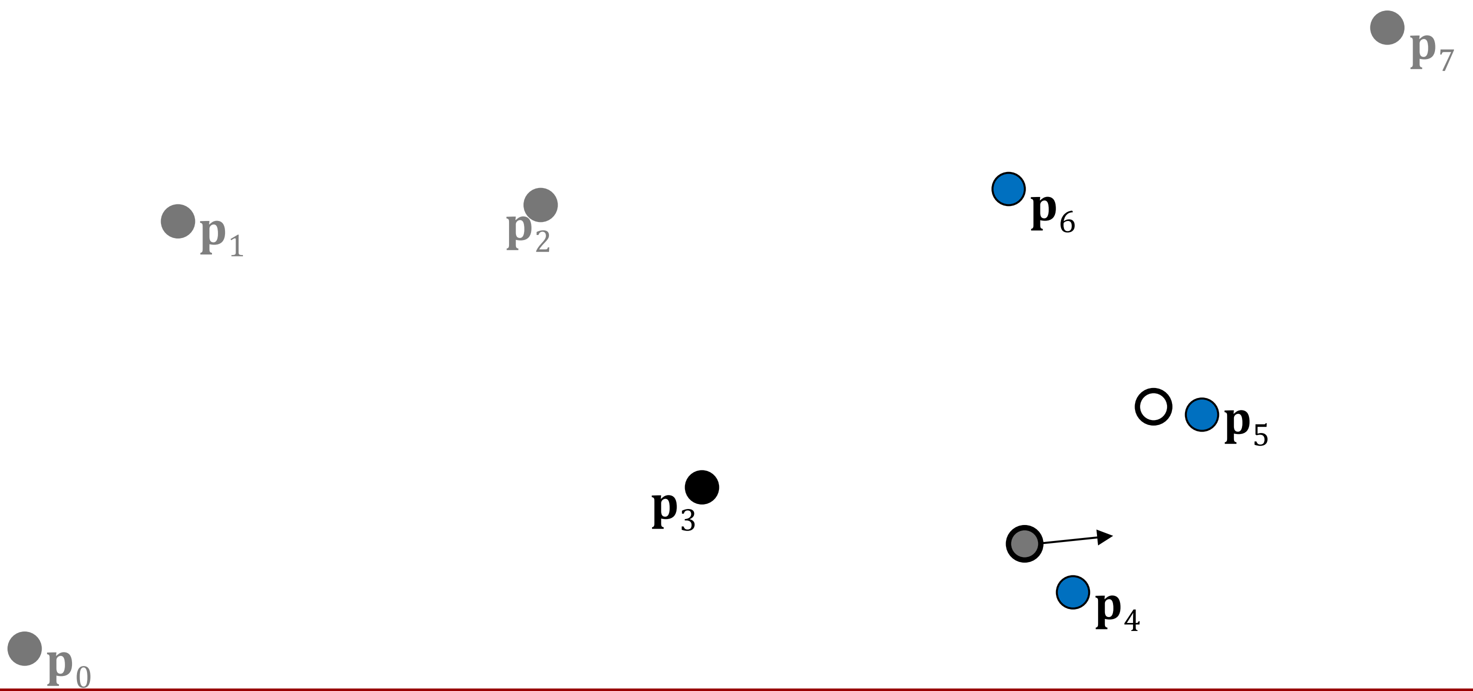
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

Approximating piecewise *cubic* polynomial, each specified by four control points.

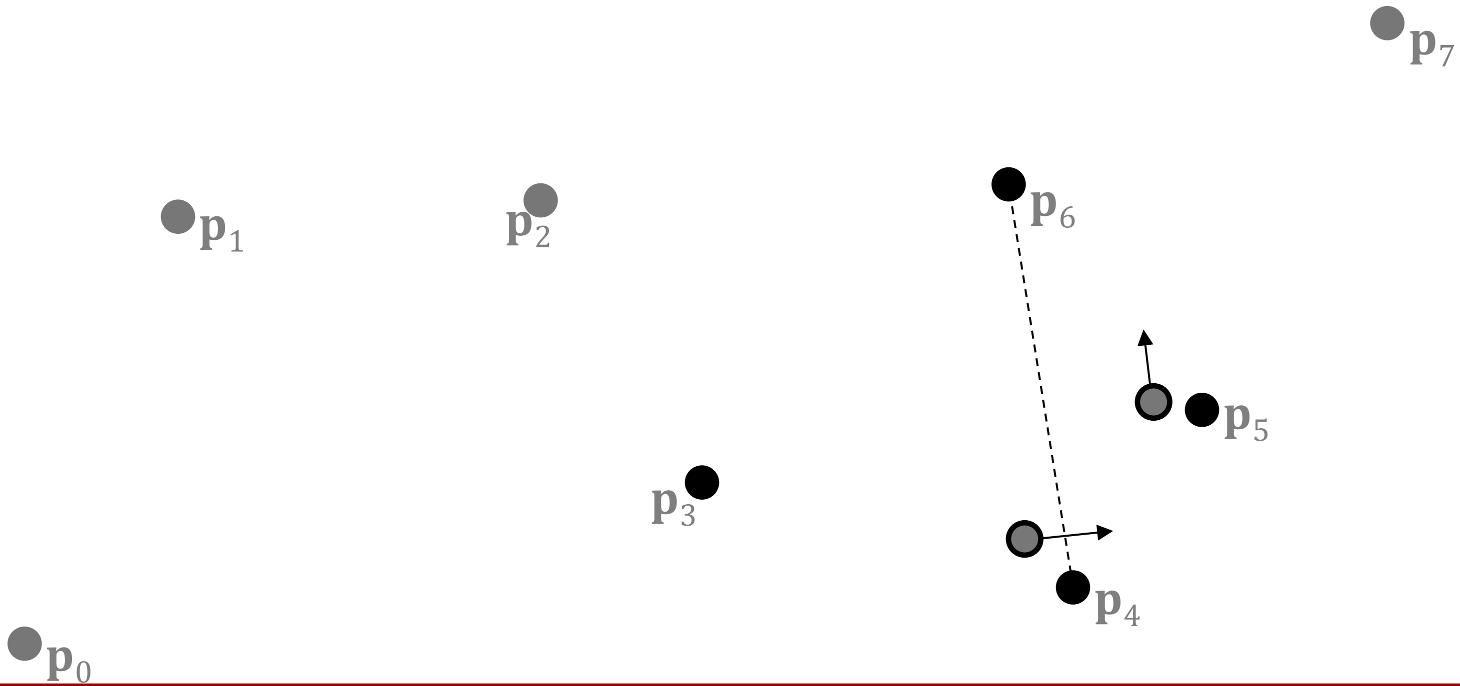
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

Approximating piecewise *cubic* polynomial, each specified by four control points.

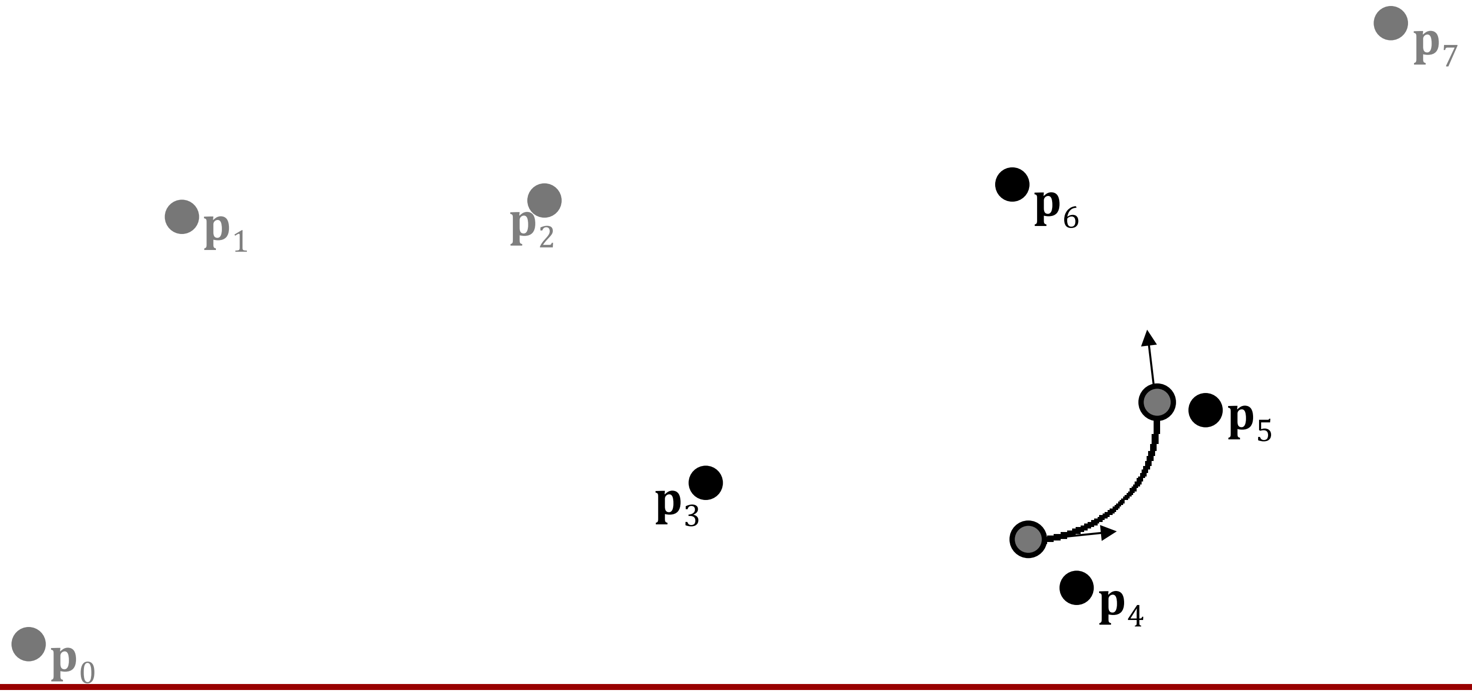
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

Approximating piecewise *cubic* polynomial, each specified by four control points.

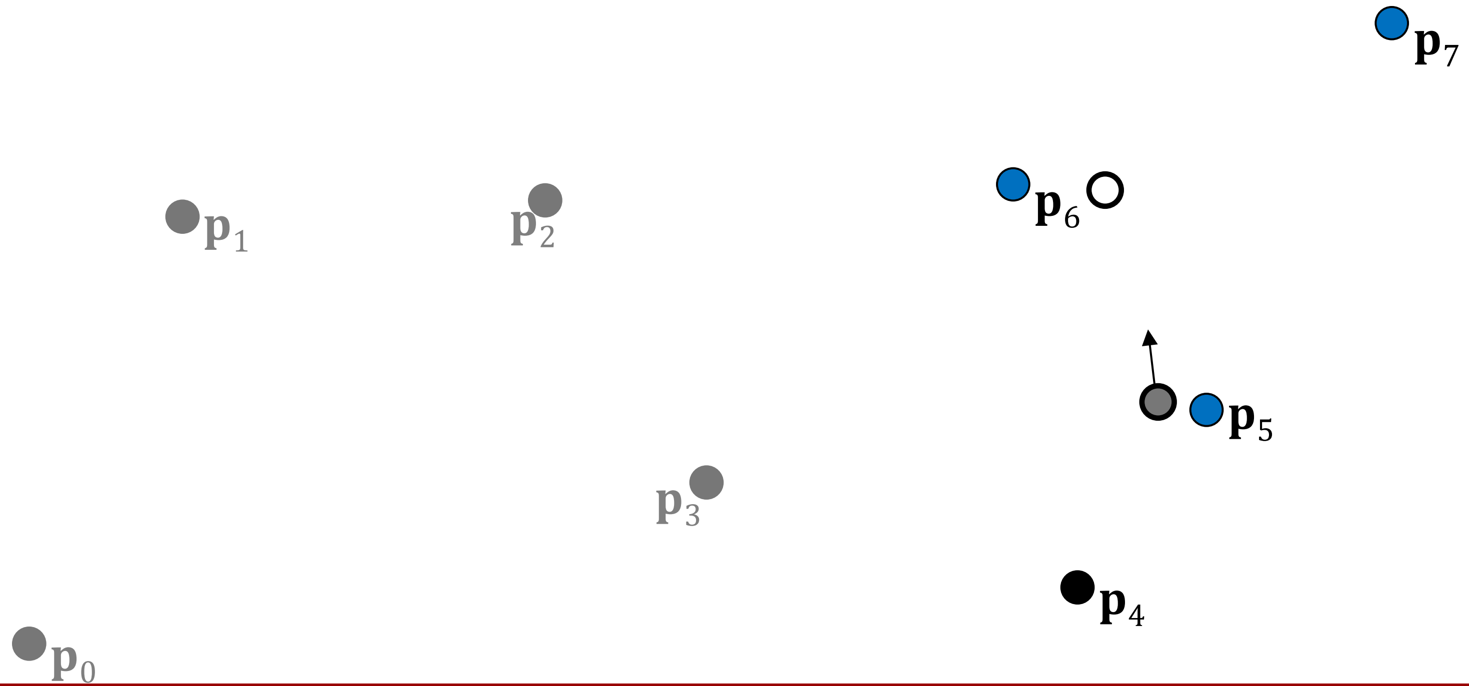
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

Approximating piecewise *cubic* polynomial, each specified by four control points.

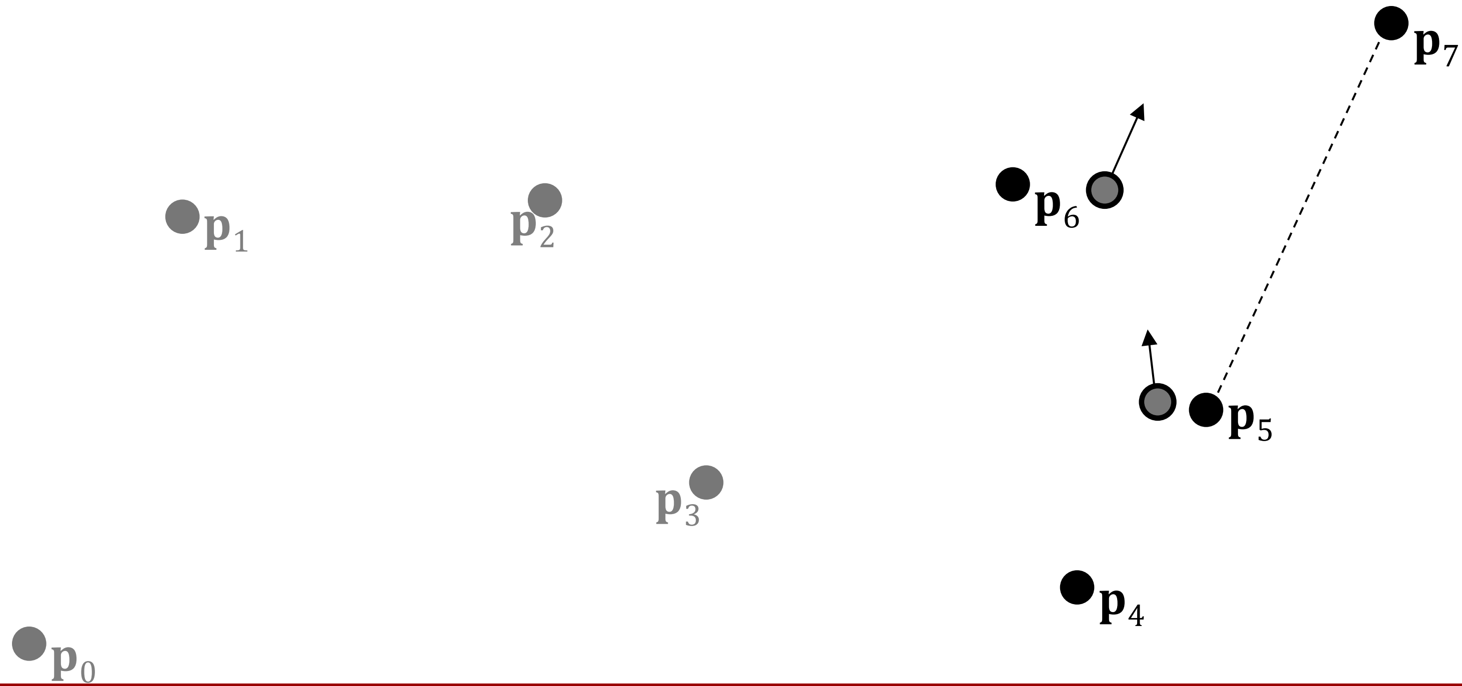
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

Approximating piecewise *cubic* polynomial, each specified by four control points.

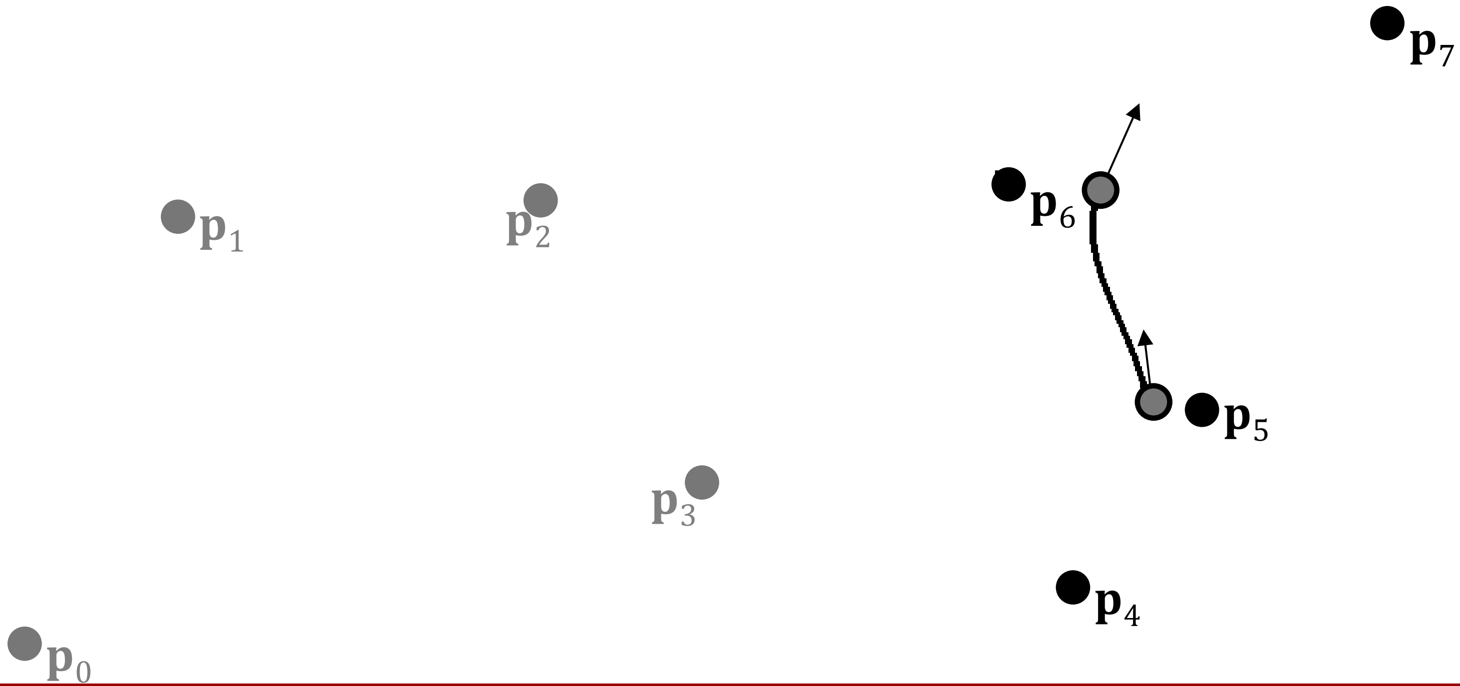
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

Approximating piecewise *cubic* polynomial, each specified by four control points.

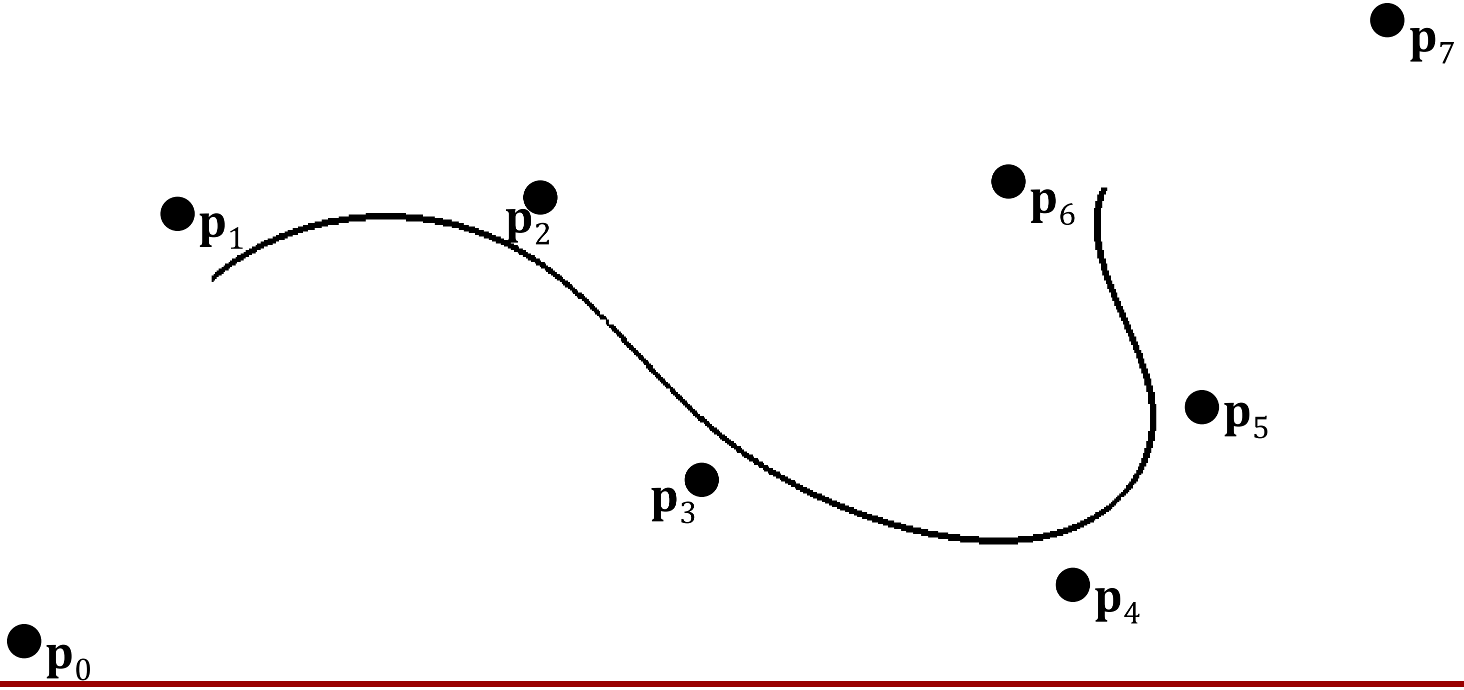
Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

<u>Approximating</u> piecewise *cubic* polynomial, each specified by four control points.

Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



Specific Example: Uniform Cubic B-Splines

Using Hermite splines, we have:

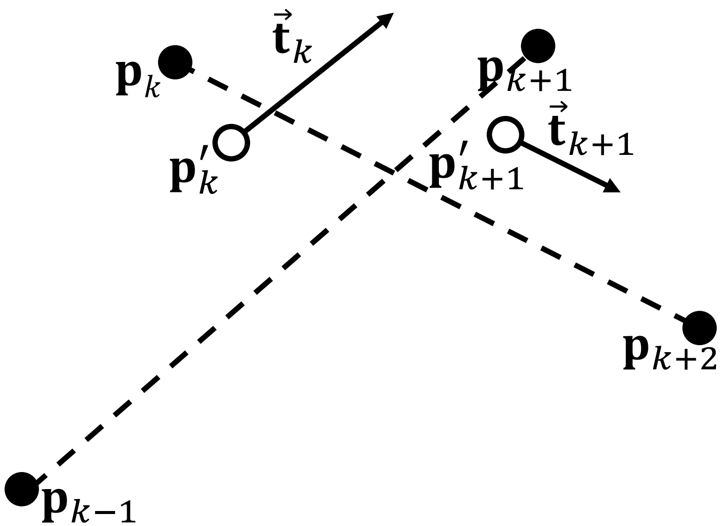
$$\mathbf{P}_k(u) = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix} \underbrace{\begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}}_{\mathbf{M}_{\text{Hermite}}} \begin{pmatrix} \mathbf{p}'_k \\ \mathbf{p}'_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix}$$

$$\mathbf{p}'_k = \frac{(\mathbf{p}_{k-1} + 4\mathbf{p}_k + \mathbf{p}_{k+1})}{6}$$

$$\mathbf{p}'_{k+1} = \frac{(\mathbf{p}_k + 4\mathbf{p}_{k+1} + \mathbf{p}_{k+2})}{6}$$

$$\vec{\mathbf{t}}_k = \frac{(\mathbf{p}_{k+1} - \mathbf{p}_{k-1})}{2}$$

$$\vec{\mathbf{t}}_{k+1} = \frac{(\mathbf{p}_{k+2} - \mathbf{p}_k)}{2}$$



Specific Example: Uniform Cubic B-Splines

We can express the boundary constraints as:

$$\begin{pmatrix} \mathbf{p}'_k \\ \mathbf{p}'_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix} = \frac{1}{6} \begin{pmatrix} \mathbf{p}_{k-1} + 4\mathbf{p}_k + \mathbf{p}_{k+1} \\ \mathbf{p}_k + 4\mathbf{p}_{k+1} + \mathbf{p}_{k+2} \\ 6s(\mathbf{p}_{k+1} - \mathbf{p}_{k-1}) \\ 6s(\mathbf{p}_{k+2} - \mathbf{p}_k) \end{pmatrix}$$

⇒ Using Hermite splines, we get:

$$\mathbf{P}_k(u) = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix} \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{p}'_k \\ \mathbf{p}'_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix}$$

$\mathbf{M}_{\text{Hermite}}$

Specific Example: Uniform Cubic B-Splines

We can express the boundary constraints as:

$$\begin{pmatrix} \mathbf{p}'_k \\ \mathbf{p}'_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix} = \frac{1}{6}\begin{pmatrix} \mathbf{p}_{k-1} + 4\mathbf{p}_k + \mathbf{p}_{k+1} \\ \mathbf{p}_k + 4\mathbf{p}_{k+1} + \mathbf{p}_{k+2} \\ 6s(\mathbf{p}_{k+1} - \mathbf{p}_{k-1}) \\ 6s(\mathbf{p}_{k+2} - \mathbf{p}_k) \end{pmatrix} = \frac{1}{6}\begin{pmatrix} 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ -3 & 0 & 3 & 0 \\ 1 & -3 & 0 & 3 \end{pmatrix}\begin{pmatrix} \mathbf{p}_{k-1} \\ \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{p}_{k+2} \end{pmatrix}$$

⇒ Using Hermite splines, we get:

$$\mathbf{P}_k(u) = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix}\begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} \mathbf{p}'_k \\ \mathbf{p}'_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix}$$

$\mathbf{M}_{\text{Hermite}}$

Specific Example: Uniform Cubic B-Splines

We can express the boundary constraints as:

$$\begin{pmatrix} \mathbf{p}'_k \\ \mathbf{p}'_{k+1} \\ \vec{\mathbf{t}}_k \\ \vec{\mathbf{t}}_{k+1} \end{pmatrix} = \frac{1}{6}\begin{pmatrix} \mathbf{p}_{k-1} + 4\mathbf{p}_k + \mathbf{p}_{k+1} \\ \mathbf{p}_k + 4\mathbf{p}_{k+1} + \mathbf{p}_{k+2} \\ 6s(\mathbf{p}_{k+1} - \mathbf{p}_{k-1}) \\ 6s(\mathbf{p}_{k+2} - \mathbf{p}_k) \end{pmatrix} = \frac{1}{6}\begin{pmatrix} 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ -3 & 0 & 3 & 0 \\ 1 & -3 & 0 & 3 \end{pmatrix}\begin{pmatrix} \mathbf{p}_{k-1} \\ \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{p}_{k+2} \end{pmatrix}$$

⇒ Using Hermite splines, we get:

$$\mathbf{P}_k(u) = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix}\begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\frac{1}{6}\begin{pmatrix} 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ -3 & 0 & 3 & 0 \\ 1 & -3 & 0 & 3 \end{pmatrix}\begin{pmatrix} \mathbf{p}_{k-1} \\ \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{p}_{k+2} \end{pmatrix}$$

↑ $\mathbf{M}_{\text{Hermite}}$

Specific Example: Uniform Cubic B-Splines

$$\mathbf{P}_k(u) = (u^3 \quad u^2 \quad u \quad 1)\begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\frac{1}{6}\begin{pmatrix} 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ -3 & 0 & 3 & 0 \\ 1 & -3 & 0 & 3 \end{pmatrix}\begin{pmatrix} \mathbf{p}_{k-1} \\ \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{p}_{k+2} \end{pmatrix}$$

Multiplying, we get the uniform cubic B-spline matrix representation:

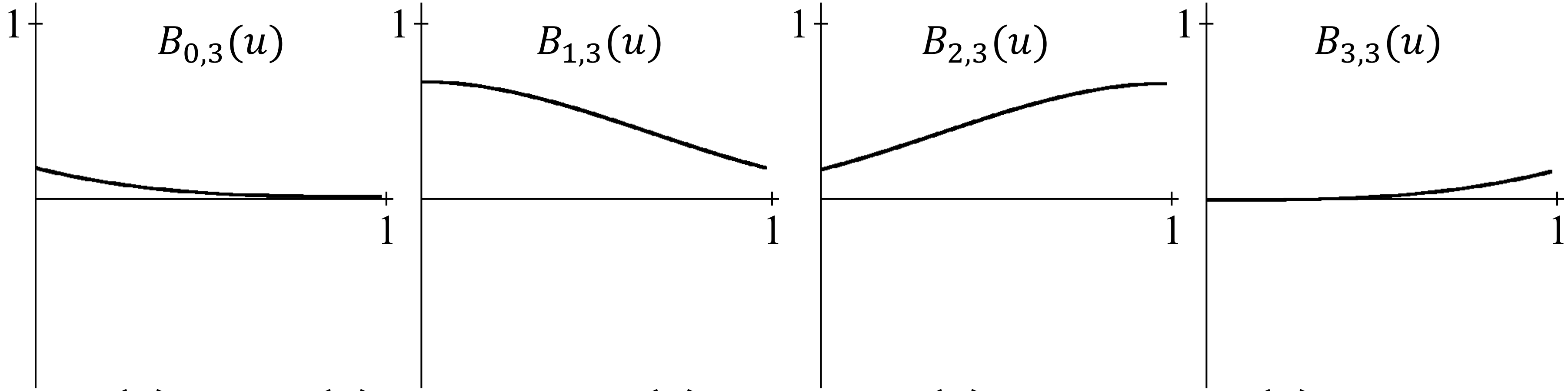
$$\mathbf{P}_k(u) = (u^3 \quad u^2 \quad u \quad 1)\frac{1}{6}\begin{pmatrix} -1 & -3 & -3 & 1 \\ 1 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix}\begin{pmatrix} \mathbf{p}_{k-1} \\ \mathbf{p}_k \\ \mathbf{p}_{k+1} \\ \mathbf{p}_{k+2} \end{pmatrix}$$

$\mathbf{M}_{\text{BSpline}}$

Specific Example: Uniform Cubic B-Splines

Setting the blending functions to:

- $B_{0,3}(u) = (^1/_6)u^3 + (^1/_2)u^2 - (^1/_2)u + {^1/_6}$
- $B_{1,3}(u) = (^1/_2)u^3 - u^2 + {^2/_3}$
- $B_{2,3}(u) = -(^1/_2)u^3 + (^1/_2)u^2 + (^1/_2)u + {^1/_6}$
- $B_{3,3}(u) = (^1/_6)u^3$

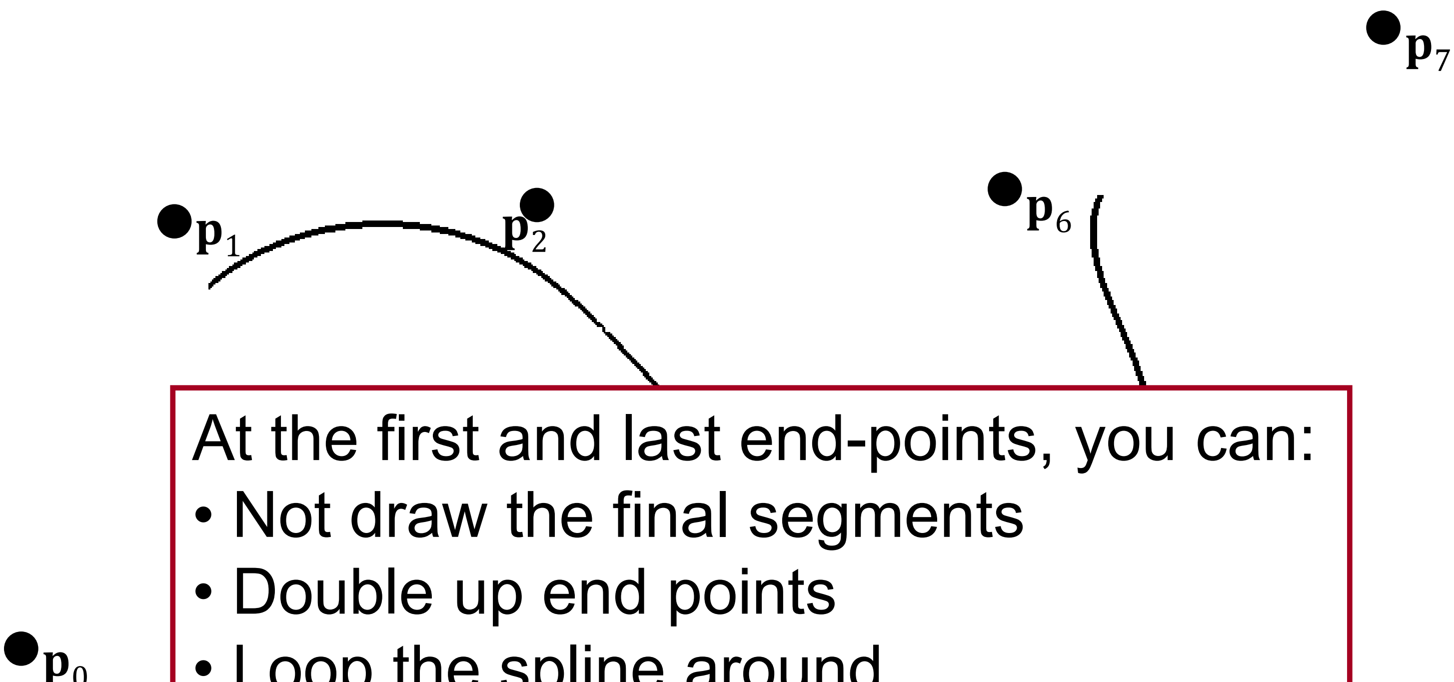


$$\mathbf{P}_k(u) = B_{0,3}(u)\cdot\mathbf{p}_{k-1} + B_{1,3}(u)\cdot\mathbf{p}_k + B_{2,3}(u)\cdot\mathbf{p}_{k+1} + B_{3,3}(u)\cdot\mathbf{p}_{k+2}$$

Specific Example: Uniform Cubic B-Splines

Approximating piecewise *cubic* polynomial, each specified by four control points.

Iteratively construct the curve **near** middle two points **using adjacent points to define tangents**.



At the first and last end-points, you can:

- Not draw the final segments
- Double up end points
- Loop the spline around

Overview

What is a Spline?

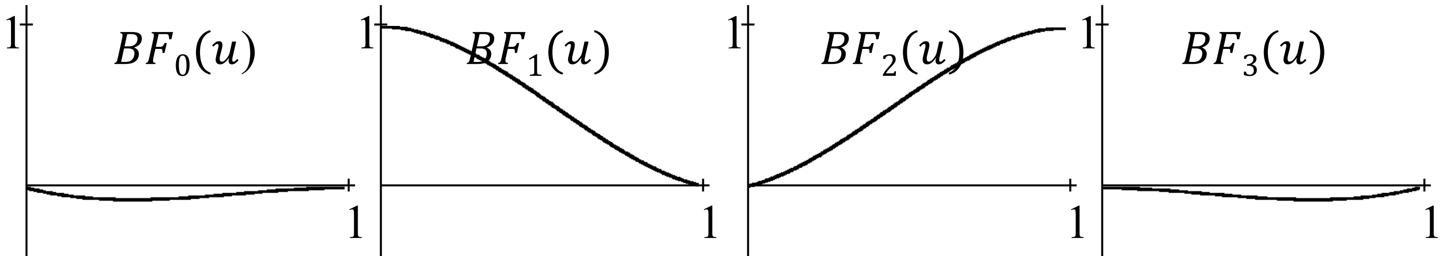
Specific Examples:

- Hermite Splines
- Cardinal Splines
- Uniform Cubic B-Splines

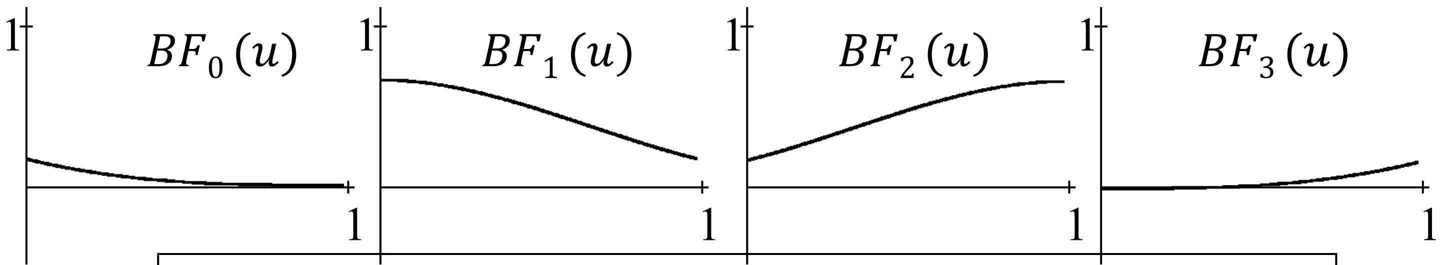
Comparing Catmull-Rom (Cardinal with $\tau = 1/2$) and Uniform Cubic B-Splines

Blending Functions

Blending functions provide a way for expressing the functions $\mathbf{P}_k(u)$ as a weighted sum of the four control points $\mathbf{p}_{k-1}$, $\mathbf{p}_k$, $\mathbf{p}_{k+1}$, and $\mathbf{p}_{k+2}$:



Cardinal Blending Functions ($s = 1/2$)



Uniform Cubic B-Spline Blending Functions ($s = 1/2$)

$$\mathbf{P}_k(u) = BF_0(u) \cdot \mathbf{p}_{\mathrm{k}-1} + BF_1(u) \cdot \mathbf{p}_k + BF_2(u) \cdot \mathbf{p}_{k+1} + BF_3(u) \cdot \mathbf{p}_{k+2}$$

Blending Functions

Properties:

Translation Equivariance:

- If we translate all the control points by the same vector $\mathbf{q}$, the position of the new curve at value u should be the position of the old curve at u, translated by $\mathbf{q}$.

⇒ Given control points $\{\mathbf{p}_{k-1}, \mathbf{p}_k, \mathbf{p}_{k+1}, \mathbf{p}_{k+2}\}$ and translation vector $\mathbf{q}$:
Let $\mathbf{P}_k(u)$ be the curve defined by $\{\mathbf{p}_{k-1}, \mathbf{p}_k, \mathbf{p}_{k+1}, \mathbf{p}_{k+2}\}$.
Let $\mathbf{Q}_k(u)$ be the curve defined by $\{\mathbf{q}+\mathbf{p}_{k-1}, \mathbf{q}+\mathbf{p}_k, \mathbf{q}+\mathbf{p}_{k+1}, \mathbf{q}+\mathbf{p}_{k+2}\}$.
We want:

$$\mathbf{Q}_k(u) = \mathbf{q} + \mathbf{P}_k(u)$$

⇒ Expanding $\mathbf{Q}_k(u)$, we have:

$$\begin{aligned}\mathbf{Q}_k(u) &= BF_0(u)(\mathbf{q}+\mathbf{p}_{k-1}) + BF_1(u)(\mathbf{q}+\mathbf{p}_k) + BF_2(u)(\mathbf{q}+\mathbf{p}_{k+1}) + BF_3(u)(\mathbf{q}+\mathbf{p}_{k+2}) \\ &= \big(BF_0(u) + BF_1(u) + BF_2(u) + BF_3(u)\big)\mathbf{q} + \mathbf{P}_k(u)\end{aligned}$$

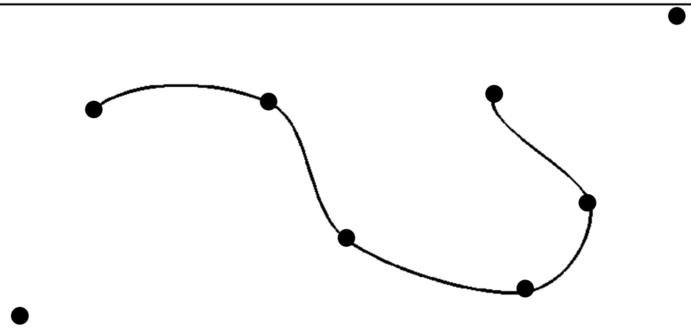
⇒ To satisfy translation equivariance, we must have:

$$BF_0(u) + BF_1(u) + BF_2(u) + BF_3(u) = 1$$

$$\mathbf{P}_k(u) = BF_0(u)\cdot\mathbf{p}_{\mathrm{k}-1} + BF_1(u)\cdot\mathbf{p}_k + BF_2(u)\cdot\mathbf{p}_{k+1} + BF_3(u)\cdot\mathbf{p}_{k+2}$$

Comparison: Catmull vs. Uniform

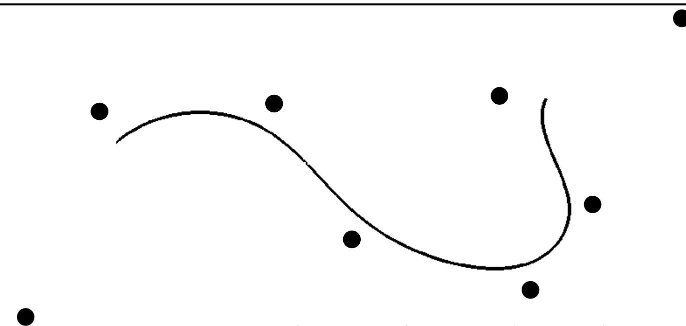
Catmull-Rom Splines



$$BF_0(u) = -\tfrac{1}{2}u^3 + \quad u^2 - \tfrac{1}{2}u$$
$$BF_1(u) = \quad \tfrac{3}{2}u^3 - \tfrac{5}{2}u^2 + \quad +1$$
$$BF_2(u) = -\tfrac{3}{2}u^3 + 2u^2 + \tfrac{1}{2}u$$
$$BF_3(u) = \quad \tfrac{1}{2}u^3 - \tfrac{1}{2}u^2$$

$$BF_0(u) + BF_1(u) + BF_2(u) + BF_3(u) = 1$$

Uniform Cubic B Splines



$$BF_0(u) = -\tfrac{1}{6}u^3 + \tfrac{1}{2}u^2 - \tfrac{1}{2}u + \tfrac{1}{6}$$
$$BF_1(u) = \quad \tfrac{1}{2}u^3 - \quad u^2 + \quad + \tfrac{2}{3}$$
$$BF_2(u) = -\tfrac{1}{2}u^3 + \tfrac{1}{2}u^2 + \tfrac{1}{2}u + \tfrac{1}{6}$$
$$BF_3(u) = \quad \tfrac{1}{6}u^3$$

$$BF_0(u) + BF_1(u) + BF_2(u) + BF_3(u) = 1$$

$$\mathbf{P}_k(u) = BF_0(u)\cdot\mathbf{p}_{k-1} + BF_1(u)\cdot\mathbf{p}_k + BF_2(u)\cdot\mathbf{p}_{k+1} + BF_3(u)\cdot\mathbf{p}_{k+2}$$

Blending Functions

Properties:

Translation Equivariance:

$BF_0(u) + BF_1(u) + BF_2(u) + BF_3(u) = 1$ for all $0 \le u \le 1$.

Continuity:

- We need the curve $\mathbf{P}_{k+1}(u)$ to begin where $\mathbf{P}_k(u)$ ended.

⇒ Taking the difference, we get:

$$0 = \mathbf{P}_{k+1}(0) - \mathbf{P}_k(1)$$

⇒ Expanding we get:

$$\begin{aligned} 0 = &\big(\qquad\qquad -BF_0(1)\big)\mathbf{p}_{k-1} \\ &+ \big(BF_0(0) - BF_1(1)\big)\mathbf{p}_k \\ &+ \big(BF_1(0) - BF_2(1)\big)\mathbf{p}_{k+1} \\ &+ \big(BF_2(0) - BF_3(1)\big)\mathbf{p}_{k+2} \\ &+ \big(BF_3(0) \qquad\qquad\big)\mathbf{p}_{k+3} \end{aligned}$$

⇒ For this to be true for all control points $\{\mathbf{p}_{k-1}, \mathbf{p}_k, \mathbf{p}_{k+1}, \mathbf{p}_{k+2}, \mathbf{p}_{k+3}\}$, we must have:

$$\begin{aligned} 0 &= BF_0(1) \\ BF_0(0) &= BF_1(1) \\ BF_1(0) &= BF_2(1) \\ BF_2(0) &= BF_3(1) \\ BF_3(0) &= 0 \end{aligned}$$

$$\mathbf{P}_k(u) = BF_0(u)\cdot\mathbf{p}_{\mathrm{k}-1} + BF_1(u)\cdot\mathbf{p}_k + BF_2(u)\cdot\mathbf{p}_{k+1} + BF_3(u)\cdot\mathbf{p}_{k+2}$$

Blending Functions

Properties:

More Generally, for the spline to have continuous n-th order derivatives, the blending functions need to satisfy:

$$0 = BF_0^{(n)}(1)$$
$$BF_0^{(n)}(0) = BF_1^{(n)}(1)$$
$$BF_1^{(n)}(0) = BF_2^{(n)}(1)$$
$$BF_2^{(n)}(0) = BF_3^{(n)}(1)$$
$$BF_3^{(n)}(0) = 0$$

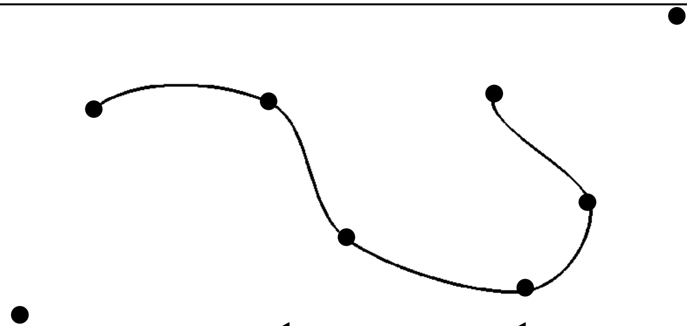
⇒ For this to be true for all control points $\{\mathbf{p}_{k-1}, \mathbf{p}_k, \mathbf{p}_{k+1}, \mathbf{p}_{k+2}, \mathbf{p}_{k+3}\}$, we must have:

$$0 = BF_0(1)$$
$$BF_0(0) = BF_1(1)$$
$$BF_1(0) = BF_2(1)$$
$$BF_2(0) = BF_3(1)$$
$$BF_3(0) = 0$$

$$\mathbf{P}_k(u) = BF_0(u) \cdot \mathbf{p}_{\mathrm{k}-1} + BF_1(u) \cdot \mathbf{p}_k + BF_2(u) \cdot \mathbf{p}_{k+1} + BF_3(u) \cdot \mathbf{p}_{k+2}$$

Comparison: Catmull vs. Uniform

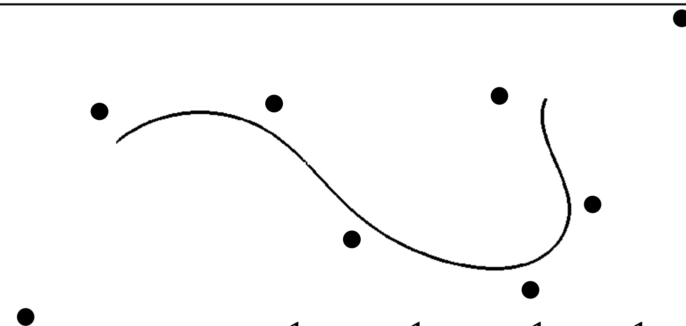
Catmull-Rom Splines



$$BF_0(u) = -\tfrac{1}{2}u^3 + u^2 - \tfrac{1}{2}u$$
$$BF_1(u) = \tfrac{3}{2}u^3 - \tfrac{5}{2}u^2 + 1$$
$$BF_2(u) = -\tfrac{3}{2}u^3 + 2u^2 + \tfrac{1}{2}u$$
$$BF_3(u) = \tfrac{1}{2}u^3 - \tfrac{1}{2}u^2$$

$BF_0(0) = 0$	$BF_0(1) = \boxed{0}$
$BF_1(0) = 1$	$BF_1(1) = 0$
$BF_2(0) = 0$	$BF_2(1) = 1$
$BF_3(0) = \boxed{0}$	$BF_3(1) = 0$

Uniform Cubic B Splines



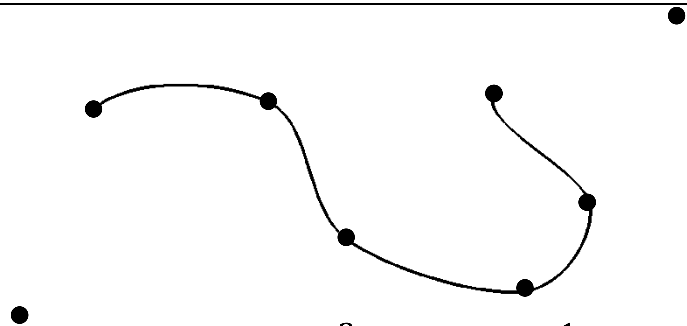
$$BF_0(u) = -\tfrac{1}{6}u^3 + \tfrac{1}{2}u^2 - \tfrac{1}{2}u + \tfrac{1}{6}$$
$$BF_1(u) = \tfrac{1}{2}u^3 - u^2 + \tfrac{2}{3}$$
$$BF_2(u) = -\tfrac{1}{2}u^3 + \tfrac{1}{2}u^2 + \tfrac{1}{2}u + \tfrac{1}{6}$$
$$BF_3(u) = \tfrac{1}{6}u^3$$

$BF_0(0) = \tfrac{1}{6}$	$BF_0(1) = \boxed{0}$
$BF_1(0) = \tfrac{2}{3}$	$BF_1(1) = \tfrac{1}{6}$
$BF_2(0) = \tfrac{1}{6}$	$BF_2(1) = \tfrac{2}{3}$
$BF_3(0) = \boxed{0}$	$BF_3(1) = \tfrac{1}{6}$

$$\mathbf{P}_k(u) = BF_0(u)\cdot\mathbf{p}_{k-1} + BF_1(u)\cdot\mathbf{p}_k + BF_2(u)\cdot\mathbf{p}_{k+1} + BF_3(u)\cdot\mathbf{p}_{k+2}$$

Comparison: Catmull vs. Uniform

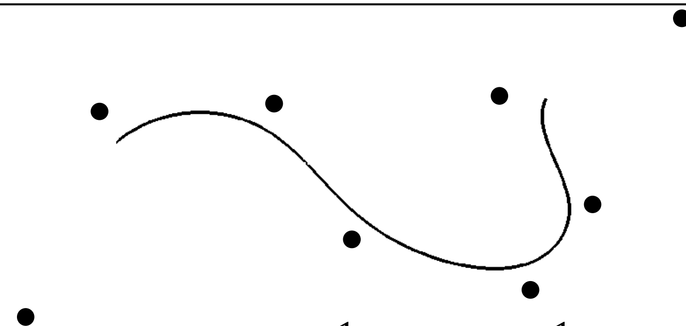
Catmull-Rom Splines



$$BF'_0(u) = -\tfrac{3}{2}u^2 + 2u - \tfrac{1}{2}$$
$$BF'_1(u) = \tfrac{9}{2}u^2 - 5u$$
$$BF'_2(u) = -\tfrac{9}{2}u^2 + 4u + \tfrac{1}{2}$$
$$BF'_3(u) = \tfrac{3}{2}u^2 - u$$

$$BF'_0(0) = -\tfrac{1}{2} \qquad BF'_0(1) = \boxed{0}$$
$$BF'_1(0) = 0 \qquad BF'_1(1) = -\tfrac{1}{2}$$
$$BF'_2(0) = \tfrac{1}{2} \qquad BF'_2(1) = 0$$
$$BF'_3(0) = \boxed{0} \qquad BF'_3(1) = \tfrac{1}{2}$$

Uniform Cubic B Splines



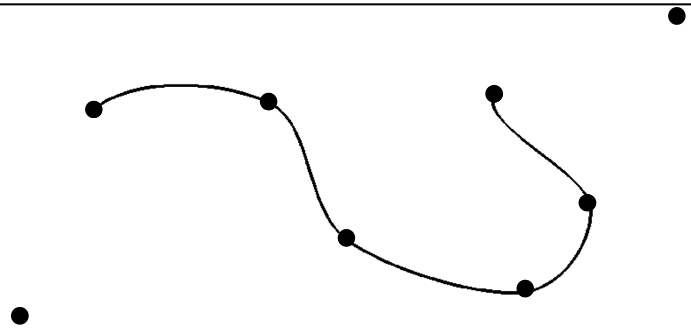
$$BF'_0(u) = -\tfrac{1}{2}u^2 + u - \tfrac{1}{2}$$
$$BF'_1(u) = \tfrac{3}{2}u^2 - 2u$$
$$BF'_2(u) = -\tfrac{3}{2}u^2 + u + \tfrac{1}{2}$$
$$BF'_3(u) = \tfrac{1}{2}u^2$$

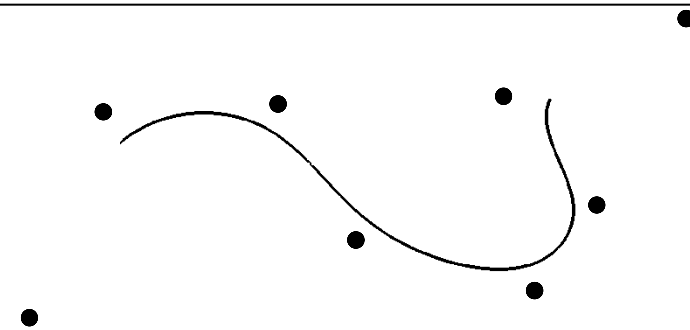
$$BF'_0(0) = -\tfrac{1}{2} \qquad BF'_0(1) = \boxed{0}$$
$$BF'_1(0) = 0 \qquad BF'_1(1) = -\tfrac{1}{2}$$
$$BF'_2(0) = \tfrac{1}{2} \qquad BF'_2(1) = 0$$
$$BF'_3(0) = \boxed{0} \qquad BF''_3(1) = \tfrac{1}{2}$$

$$\mathbf{P}_k(u) = BF_0(u)\cdot\mathbf{p}_{k-1} + BF_1(u)\cdot\mathbf{p}_k + BF_2(u)\cdot\mathbf{p}_{k+1} + BF_3(u)\cdot\mathbf{p}_{k+2}$$

Comparison: Catmull vs. Uniform

Catmull-Rom Splines	Uniform Cubic B Splines





Catmull-Rom Splines

$$\begin{aligned} BF_0''(u) &= -3u + 2 \\ BF_1''(u) &= 9u - 5 \\ BF_2''(u) &= -9u + 4 \\ BF_3''(u) &= 3u - 1 \end{aligned}$$

$$\begin{aligned} BF_0''(0) &= 2 & BF_0''(1) &= \boxed{-1} \\ BF_1''(0) &= -5 & BF_1''(1) &= 4 \\ BF_2''(0) &= 4 & BF_2''(1) &= -5 \\ BF_3''(0) &= \boxed{-1} & BF_3''(1) &= 2 \end{aligned}$$

Uniform Cubic B Splines

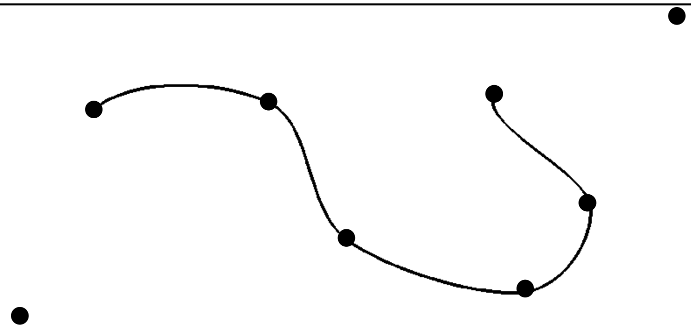
$$\begin{aligned} BF_0''(u) &= -u + 1 \\ BF_1''(u) &= 3u - 2 \\ BF_2''(u) &= -3u + 1 \\ BF_3''(u) &= u \end{aligned}$$

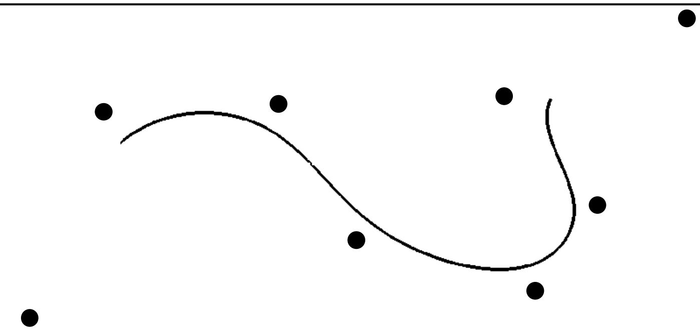
$$\begin{aligned} BF_0''(0) &= 1 & BF_0''(1) &= \boxed{0} \\ BF_1''(0) &= -2 & BF_1''(1) &= 1 \\ BF_2''(0) &= 1 & BF_2''(1) &= -2 \\ BF_3''(0) &= \boxed{0} & BF_3''(1) &= 1 \end{aligned}$$

$$\mathbf{P}_k(u) = BF_0(u) \cdot \mathbf{p}_{k-1} + BF_1(u) \cdot \mathbf{p}_k + BF_2(u) \cdot \mathbf{p}_{k+1} + BF_3(u) \cdot \mathbf{p}_{k+2}$$

Comparison: Catmull vs. Uniform

Catmull-Rom Splines	Uniform Cubic B Splines





$$\begin{aligned} BF_0'''(u) &= -1 \\ BF_1'''(u) &= 3 \\ BF_2'''(u) &= -3 \\ BF_3'''(u) &= 1 \end{aligned}$$

$$\begin{aligned} BF_0'''(0) &= -1 \\ BF_1'''(0) &= 3 \\ BF_2'''(0) &= -3 \\ BF_3'''(0) &= \boxed{1} \end{aligned} \qquad \begin{aligned} BF_0'''(1) &= \boxed{-1} \\ BF_1'''(1) &= 3 \\ BF_2'''(1) &= -3 \\ BF_3'''(1) &= 1 \end{aligned}$$

$$\mathbf{P}_k(u) = BF_0(u)\cdot\mathbf{p}_{k-1} + BF_1(u)\cdot\mathbf{p}_k + BF_2(u)\cdot\mathbf{p}_{k+1} + BF_3(u)\cdot\mathbf{p}_{k+2}$$

Blending Functions

Properties:

Translation Equivariance:

$BF_0(u) + BF_1(u) + BF_2(u) + BF_3(u) = 1$ for all $0 \leq u \leq 1$.

Continuity:

$0 = BF_0(1),\ BF_0(0) = BF_1(1),\ BF_1(0) = BF_2(1),\ BF_2(0) = BF_3(1),\ BF_3(0) = 0$

Convex Hull Containment:

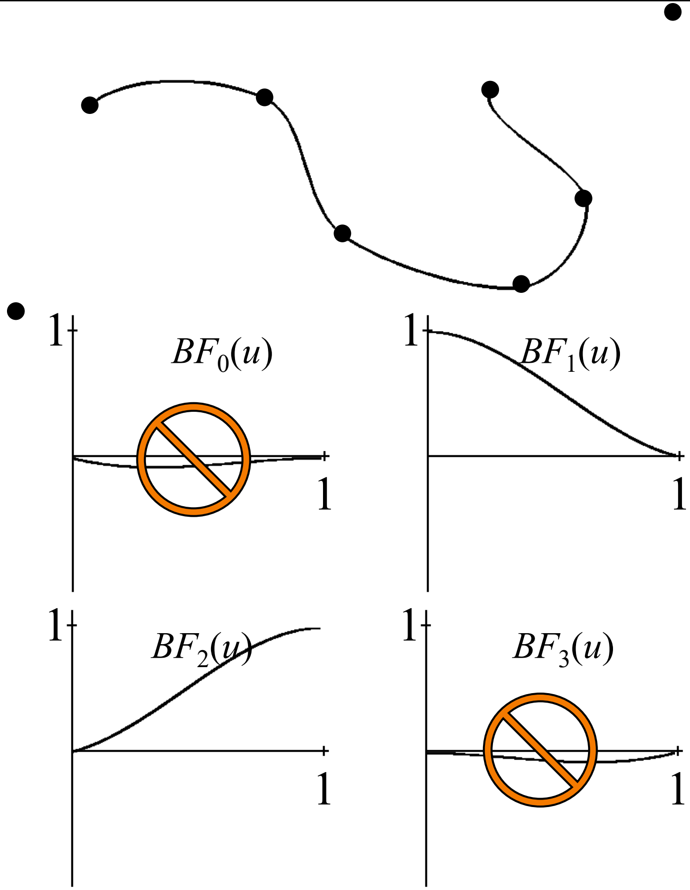
- A point is inside the convex hull of a collection of points if and only if it can be expressed as the weighted average of the points, <u>where all the weights are non-negative</u>.

$\Rightarrow$ $BF_0(u), BF_1(u), BF_2(u), BF_3(u) \geq 0$, for all $0 \leq u \leq 1$.

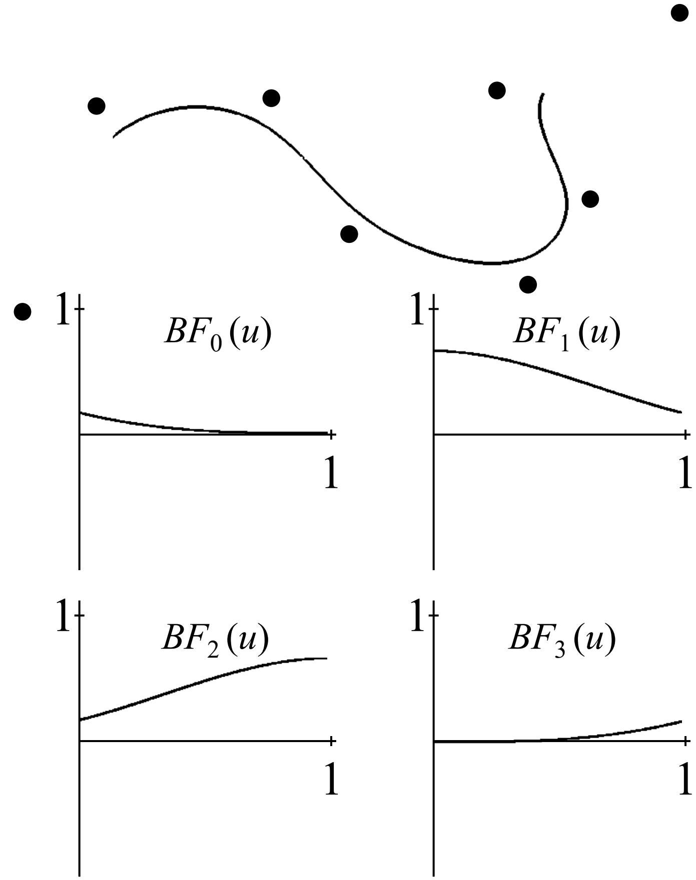
$$\mathbf{P}_k(u) = BF_0(u) \cdot \mathbf{p}_{k-1} + BF_1(u) \cdot \mathbf{p}_k + BF_2(u) \cdot \mathbf{p}_{k+1} + BF_3(u) \cdot \mathbf{p}_{k+2}$$

Comparison: Catmull vs. Uniform

Catmull-Rom Splines

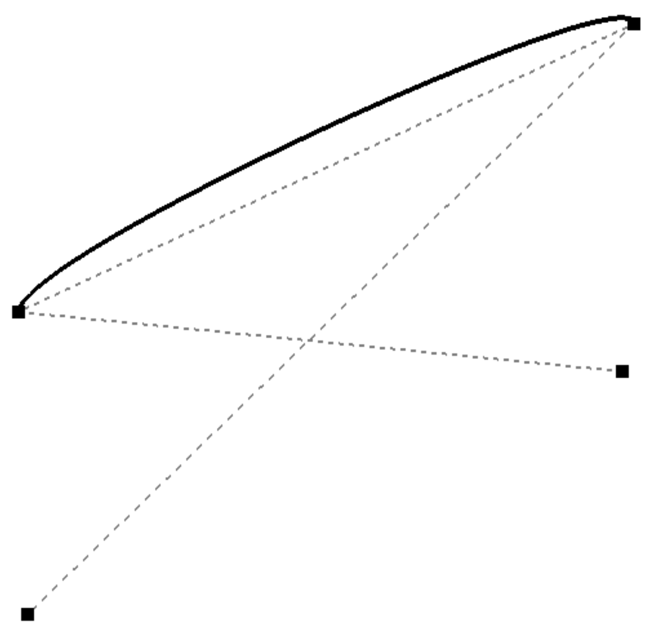
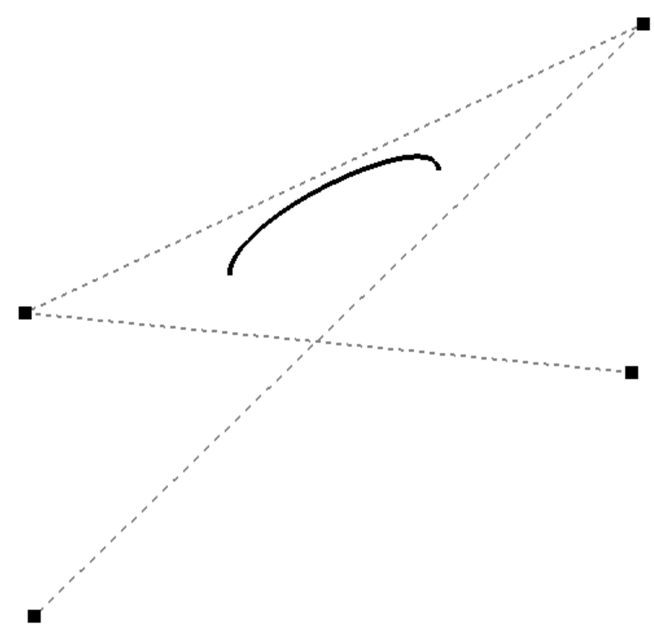


Uniform Cubic B Splines



$$\mathbf{P}_k(u) = BF_0(u)\cdot\mathbf{p}_{k-1} + BF_1(u)\cdot\mathbf{p}_k + BF_2(u)\cdot\mathbf{p}_{k+1} + BF_3(u)\cdot\mathbf{p}_{k+2}$$

Comparison: Catmull vs. Uniform

Catmull-Rom Splines	Uniform Cubic B Splines
	

$$\mathbf{P}_k(u) = BF_0(u)\cdot \mathbf{p}_{\mathrm{k}-1} + BF_1(u)\cdot \mathbf{p}_k + BF_2(u)\cdot \mathbf{p}_{k+1} + BF_3(u)\cdot \mathbf{p}_{k+2}$$

Blending Functions

Properties:

Translation Equivariance :

$BF_0(u) + BF_1(u) + BF_2(u) + BF_3(u) = 1$ for all $0 \leq u \leq 1$.

Continuity:

$0 = BF_0(1),\ BF_0(0) = BF_1(1),\ BF_1(0) = BF_2(1),\ BF_2(0) = BF_3(1),\ BF_3(0) = 0$

Convex Hull Containment:

$BF_0(u), BF_1(u), BF_2(u), BF_3(u) \geq 0$, for all $0 \leq u \leq 1$.

Interpolation:

- We want the spline segments to satisfy:

$$\mathbf{P}_k(0) = \mathbf{p}_k \quad \text{and} \quad \mathbf{P}_k(1) = \mathbf{p}_{k+1}$$

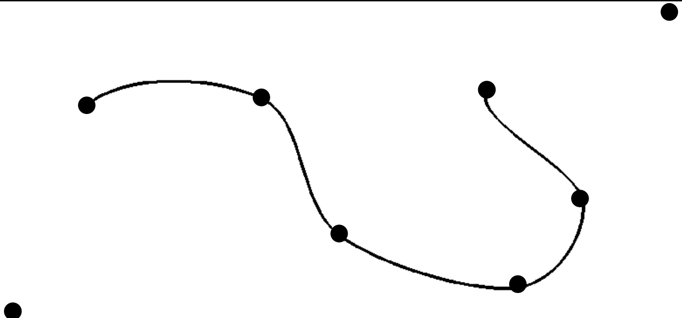
⇒ At the end-points, the blending functions satisfy:

$$\begin{matrix} BF_0(0) \\ BF_1(0) \\ BF_2(0) \\ BF_3(0) \end{matrix} = \begin{matrix} 0 \\ 1 \\ 0 \\ 0 \end{matrix} \quad \text{and} \quad \begin{matrix} BF_0(1) \\ BF_1(1) \\ BF_2(1) \\ BF_3(1) \end{matrix} = \begin{matrix} 0 \\ 0 \\ 1 \\ 0 \end{matrix}$$

$$\mathbf{P}_k(u) = BF_0(u) \cdot \mathbf{p}_{\mathrm{k}-1} + BF_1(u) \cdot \mathbf{p}_k + BF_2(u) \cdot \mathbf{p}_{k+1} + BF_3(u) \cdot \mathbf{p}_{k+2}$$

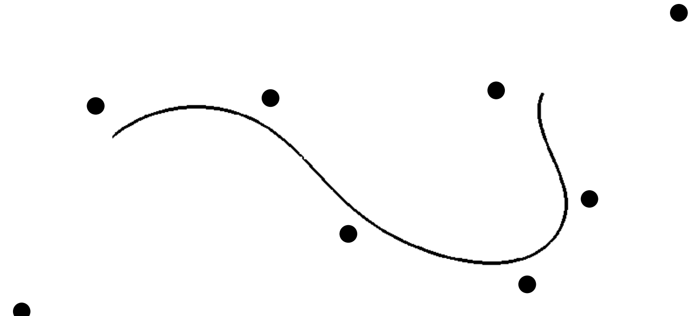
Comparison: Catmull vs. Uniform

Catmull-Rom Splines	Uniform Cubic B Splines



$$BF_0(u) = -\tfrac{1}{2}u^3 + u^2 - \tfrac{1}{2}u$$
$$BF_1(u) = \tfrac{3}{2}u^3 - \tfrac{5}{2}u^2 + 1$$
$$BF_2(u) = -\tfrac{3}{2}u^3 + 2u^2 + \tfrac{1}{2}u$$
$$BF_3(u) = \tfrac{1}{2}u^3 - \tfrac{1}{2}u^2$$

$BF_0(0) = 0$ | $BF_0(1) = 0$
$BF_1(0) = 1$ | $BF_1(1) = 0$
$BF_2(0) = 0$ | $BF_2(1) = 1$
$BF_3(0) = 0$ | $BF_3(1) = 0$



$$BF_0(u) = -\tfrac{1}{6}u^3 + \tfrac{1}{2}u^2 - \tfrac{1}{2}u + \tfrac{1}{6}$$
$$BF_1(u) = \tfrac{1}{2}u^3 - u^2 + \tfrac{2}{3}$$
$$BF_2(u) = -\tfrac{1}{2}u^3 + \tfrac{1}{2}u^2 + \tfrac{1}{2}u + \tfrac{1}{6}$$
$$BF_3(u) = \tfrac{1}{6}u^3$$

$BF_0(0) = \tfrac{1}{6}$ | $BF_0(1) = 0$
$BF_1(0) = \tfrac{2}{3}$ | $BF_1(1) = \tfrac{1}{6}$
$BF_2(0) = \tfrac{1}{6}$ | $BF_2(1) = \tfrac{2}{3}$
$BF_3(0) = 0$ | $BF_3(1) = \tfrac{1}{6}$

$$\mathbf{P}_k(u) = BF_0(u)\cdot\mathbf{p}_{k-1} + BF_1(u)\cdot\mathbf{p}_k + BF_2(u)\cdot\mathbf{p}_{k+1} + BF_3(u)\cdot\mathbf{p}_{k+2}$$

Blending Functions

Properties:

Translation Equivariance :

$$BF_0(u) + BF_1(u) + BF_2(u) + BF_3(u) = 1 \text{ for all } 0 \le u \le 1.$$

Continuity:

$$\begin{aligned} 0 &= BF_0(1) \\ BF_0(0) &= BF_1(1) \\ BF_1(0) &= BF_2(1) \\ BF_2(0) &= BF_3(1) \\ BF_3(0) &= 0 \end{aligned}$$

Required Conditions

Convex Hull Containment:

$$BF_0(u), BF_1(u), BF_2(u), BF_3(u) \ge 0, \text{ for all } 0 \le u \le 1.$$

Interpolation:

$$\begin{pmatrix} BF_0(0) \\ BF_1(0) \\ BF_2(0) \\ BF_3(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \text{ and } \begin{pmatrix} BF_0(1) \\ BF_1(1) \\ BF_2(1) \\ BF_3(1) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Desirable Conditions

$$\mathbf{P}_k(u) = BF_0(u) \cdot \mathbf{p}_{k-1} + BF_1(u) \cdot \mathbf{p}_k + BF_2(u) \cdot \mathbf{p}_{k+1} + BF_3(u) \cdot \mathbf{p}_{k+2}$$

Summary

A spline is a *piecewise* *polynomial function* whose derivatives satisfy some *continuity constraints* across curve junctions.

Looked at specification for 3 splines:

- Hermite
- Cardinal
- Uniform Cubic B-Spline

Hermite, Cardinal: Interpolating, cubic, C^1

Uniform Cubic B-Spline: Approximating, convex-hull containment, cubic, C^2

Spline Demo $(t = 1 - 2\tau)$