Topic 15:

Interpolating Curves

- Intro to curve interpolation & approximation
- Polynomial interpolation
- Cardinal splines

Interactive Design of 2D Curves

Goal: expand vocabulary of modeling primitives beyond global analytic shapes and 3D meshes

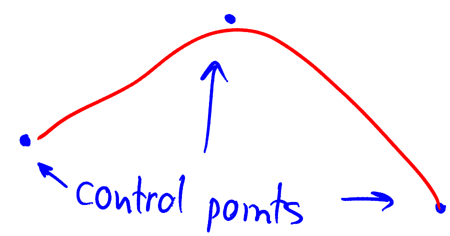
Criteria:

- Natural & intuitive interaction
- Controllable smoothness
- Adjustable resolution
- Analytic derivatives that are easy to compute
- Compact representation

Interactive Curve Design: Three Basic Tasks

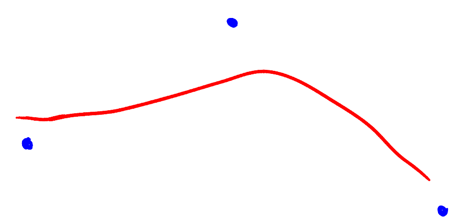
- Interpolation

  Curve goes through control points



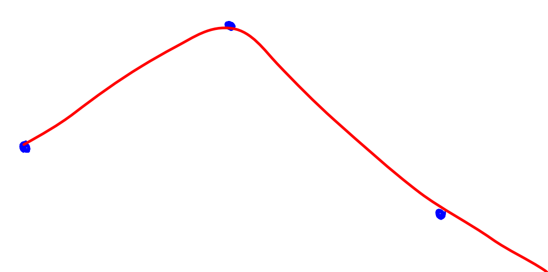
- Approximation

  Curve approximates but does not go through the control points



- Extrapolation

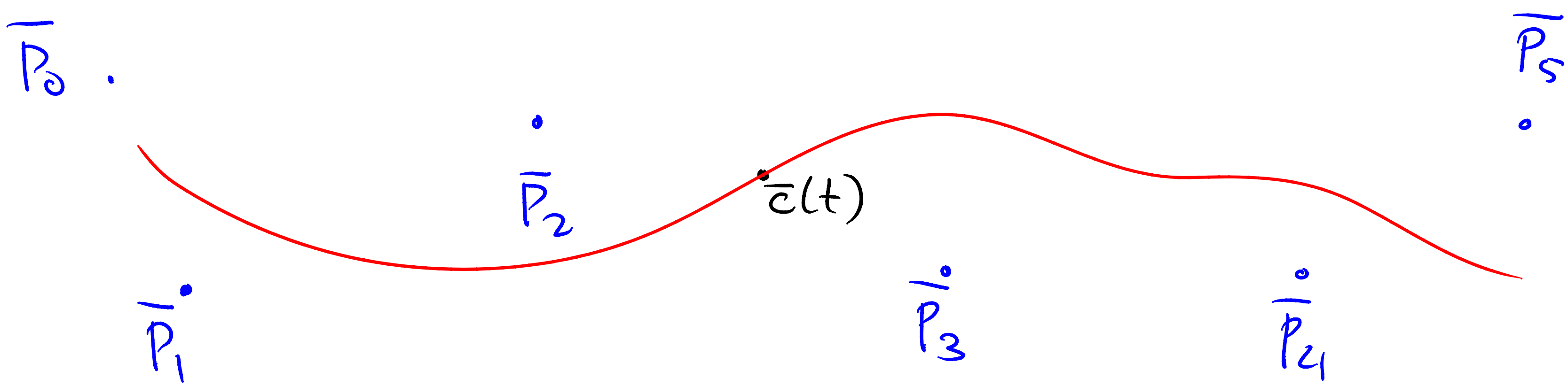
  Extend curve beyond the domain of the control points



2D Curve Design: General Problem Statement

Given N control points $\bar{p}_i$, $i=0,\ldots,N-1$

① define a curve $\bar{c}(t)$, $t \in [0, N-1]$ that interpolates/approximates them



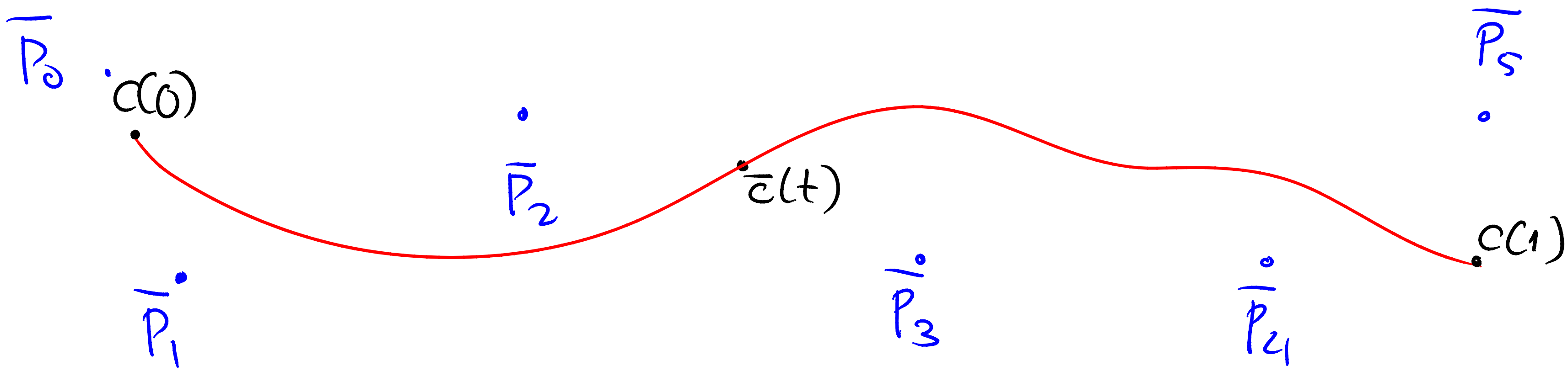
② compute its derivatives (and tangents, normals, etc)

2D Curve Design: General Problem Statement

Given N control points $\bar{p}_i$, $i=0,\dots,N-1$

① define a curve $\bar{c}(t)$, $t \in [0,1]$ that interpolates/approximates them

convention: we assume that t ranges from 0 to 1

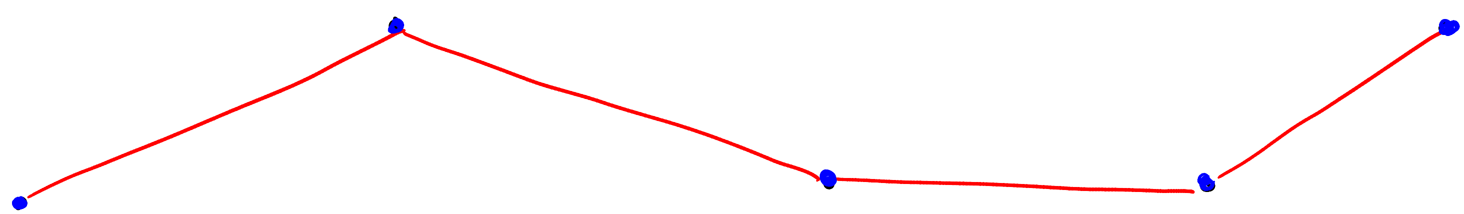


② compute its derivatives (and tangents, normals, etc)

Linear Interpolation

The simplest possible interpolation technique

Create a piecewise linear curve that connects the control points



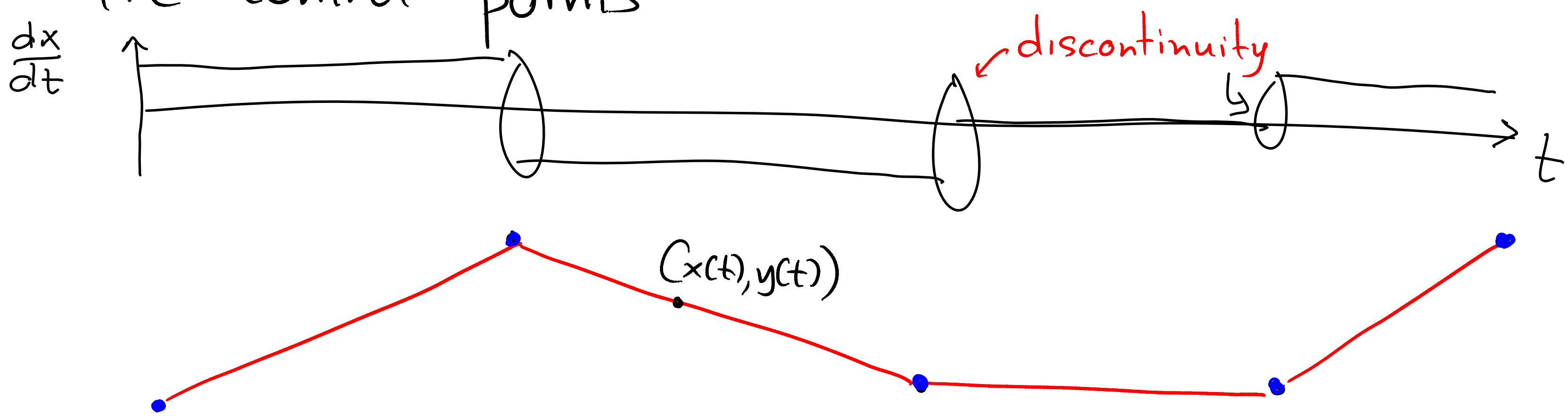
Q: What is the disadvantage of this technique?

Ans: The curve may be continuous, but its derivatives are not!

Linear Interpolation

The simplest possible interpolation technique

Create a piecewise linear curve that connects the control points

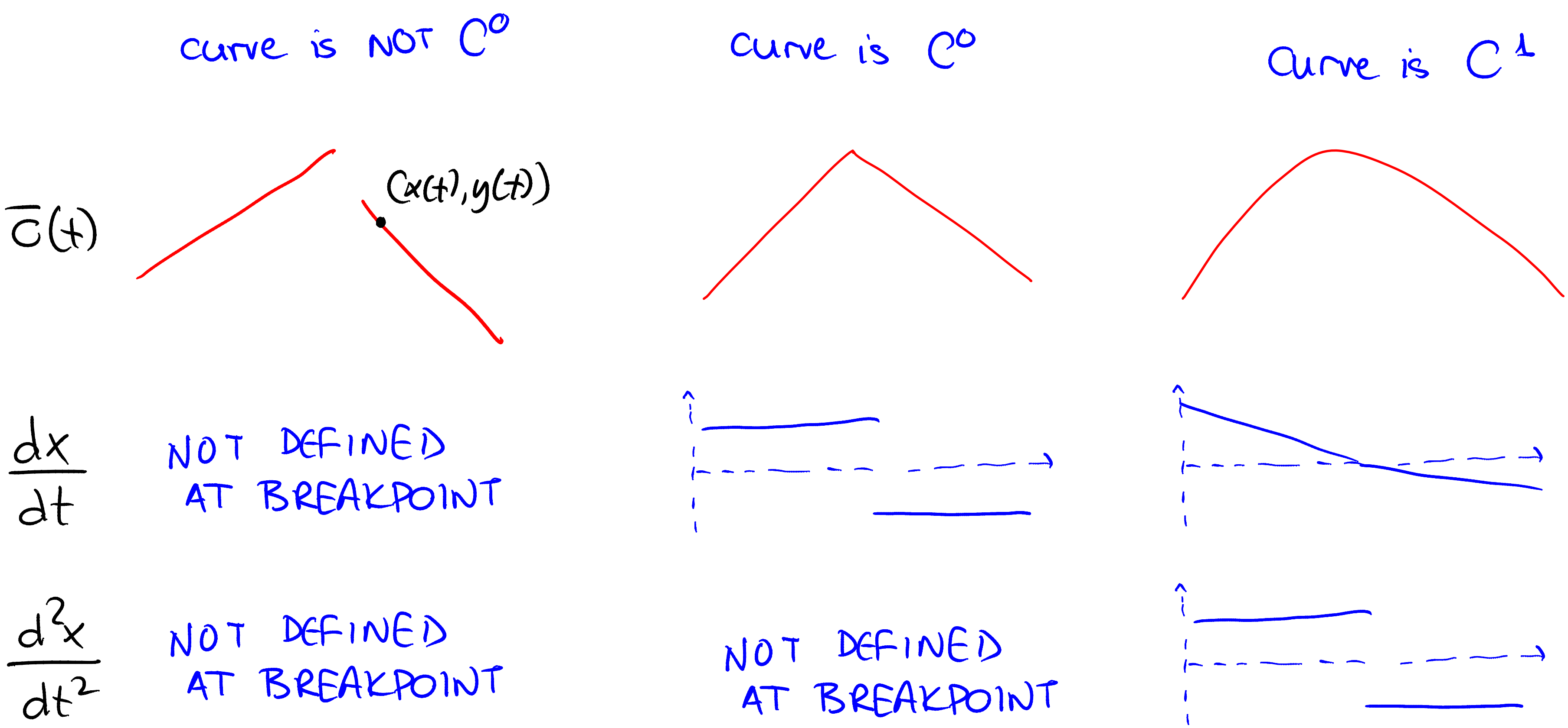


Q: What is the disadvantage of this technique?

Ans: The curve may be continuous, but its derivatives are not!

C^n Continuity

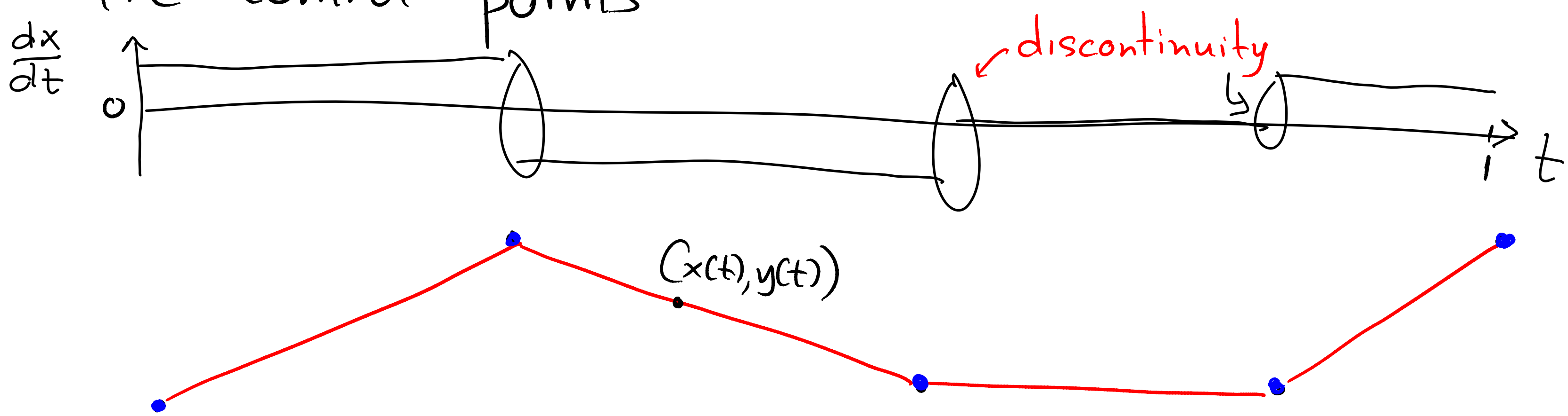
Definition: A function is called C^n if its n-th order derivative is continuous everywhere



Linear Interpolation

The simplest possible interpolation technique

Create a piecewise linear curve that connects the control points



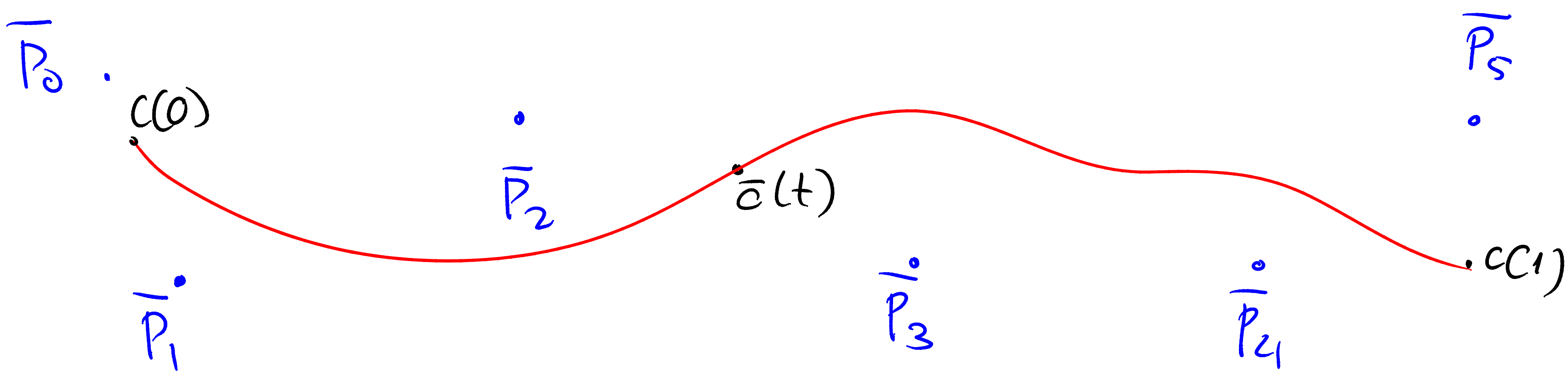
Q: What is the disadvantage of this technique?

Ans: Curve only has C^0 continuity

2D Curve Design: General Problem Statement

Given N control points $\bar{p}_i$, $i=0,\dots,N-1$

① define a curve $\bar{c}(t)$, $t \in [0,1]$ that interpolates/approximates them



② compute its derivatives (and tangents, normals, etc)

* We will seek functions that are at least C^1

Topic 15:

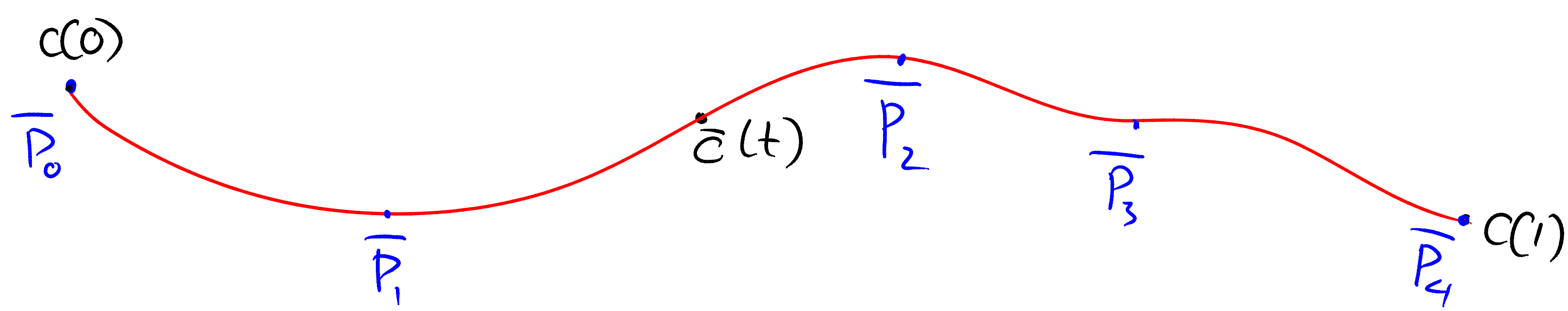
Interpolating Curves

- Intro to curve interpolation & approximation
- Polynomial interpolation
- Bézier curves
- Cardinal splines

General Polynomial Interpolation

Given N control points $\bar{P}_i = (x_i, y_i)$ $i=0,\dots,N-1$

① define $(N-1)$ – degree polynomials $x(t), y(t)$ such that $x\left(\frac{i}{N-1}\right) = x_i$, $y\left(\frac{i}{N-1}\right) = y_i$ for $i=0,\dots,N-1$

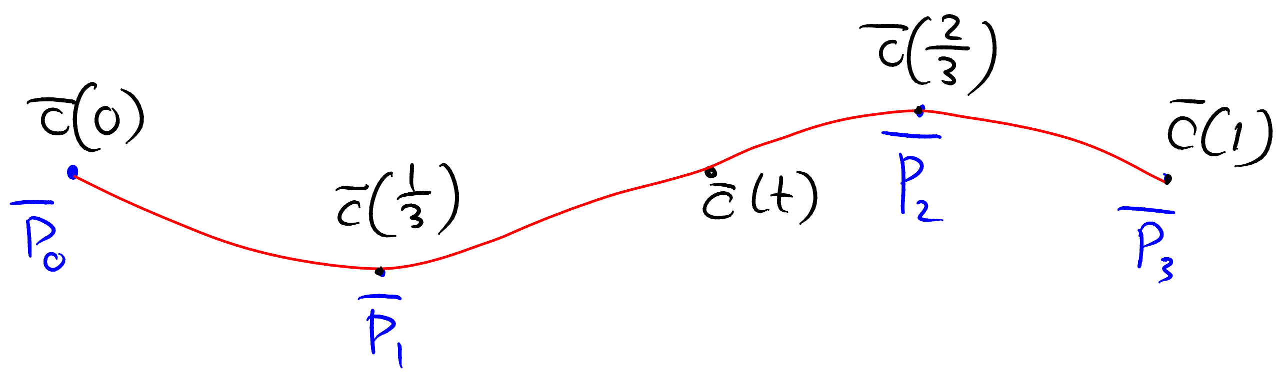


② compute its derivatives (and tangents, normals, etc)

Cubic Interpolation

Given 4 control points $\bar{P}_i = (x_i, y_i)$ $i=0,\ldots,N-1$

① define 3rd-degree polynomials $x(t), y(t)$ such that $x(\frac{i}{3}) = x_i$, $y(\frac{i}{3}) = y_i$ for $i=0,\ldots,3$



② compute its derivatives (and tangents, normals, etc)

Cubic Interpolation: Basic Equations

$$\left.\begin{aligned} x(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ y(t) &= b_0 + b_1 t + b_2 t^2 + b_3 t^2 \end{aligned}\right\}$$

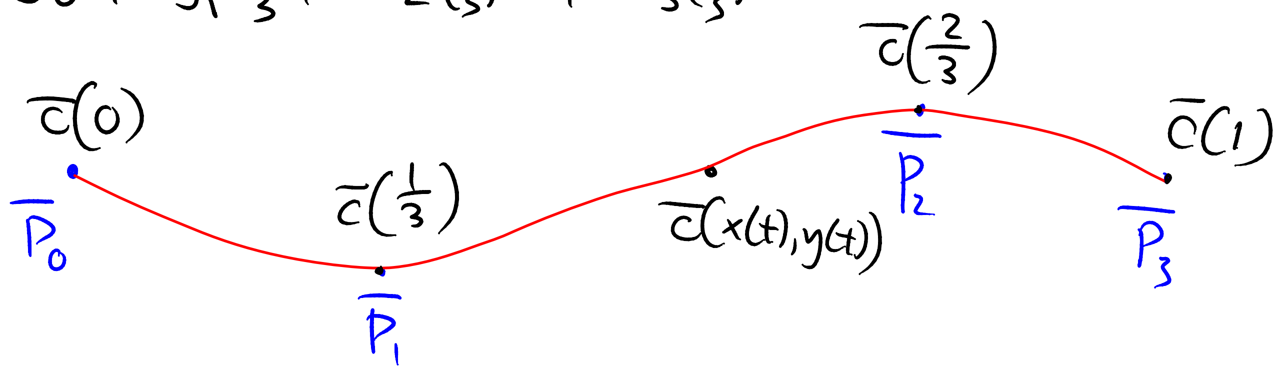
given $\overline{P_1}, \overline{P_2}, \overline{P_3}, \overline{P_4}$

compute a_i, b_i

Equations for one control point

$$x_1 = a_0 + a_1 \cdot \tfrac{1}{3} + a_2 \left(\tfrac{1}{3}\right)^2 + a_3 \left(\tfrac{1}{3}\right)^3$$

$$y_1 = b_0 + b_1 \cdot \tfrac{1}{3} + b_2 \left(\tfrac{1}{3}\right)^2 + b_3 \left(\tfrac{1}{3}\right)^3$$



Equations in matrix form:

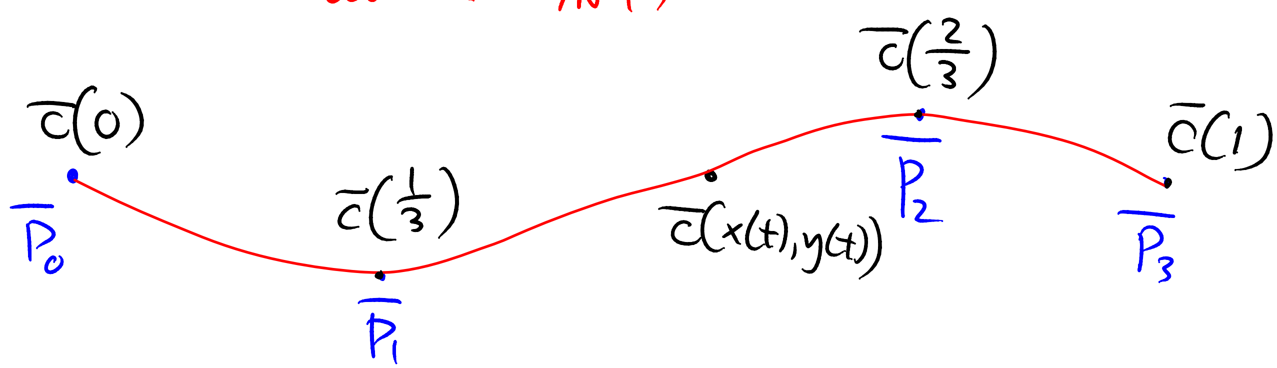
$$\begin{bmatrix} x_1 & y_1 \end{bmatrix} = \begin{bmatrix} 1 & \tfrac{1}{3} & \left(\tfrac{1}{3}\right)^2 & \left(\tfrac{1}{3}\right)^3 \end{bmatrix} \begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}$$

Cubic Interpolation: Computing the Coeffs

$$\left.\begin{aligned} x(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ y(t) &= b_0 + b_1 t + b_2 t^2 + b_3 t^2 \end{aligned}\right\}$$

given $\overline{P_1}, \overline{P_2}, \overline{P_3}, \overline{P_4}$
compute a_i, b_i

$$\underbrace{[x_i \;\; y_i]}_{\text{known}} = \underbrace{\begin{bmatrix} 1 & t_i & (t_i)^2 & (t_i)^3 \end{bmatrix}}_{\text{known } (t_i = i/_{N-1})} \underbrace{\begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}}_{\text{unknown}}$$



Equations in matrix form:

$$[x_1 \;\; y_1] = \begin{bmatrix} 1 & \frac{1}{3} & \left(\frac{1}{3}\right)^2 & \left(\frac{1}{3}\right)^3 \end{bmatrix} \begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}$$

Cubic Interpolation: Computing the Coeffs

$$\left.\begin{aligned} x(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ y(t) &= b_0 + b_1 t + b_2 t^2 + b_3 t^2 \end{aligned}\right\}$$

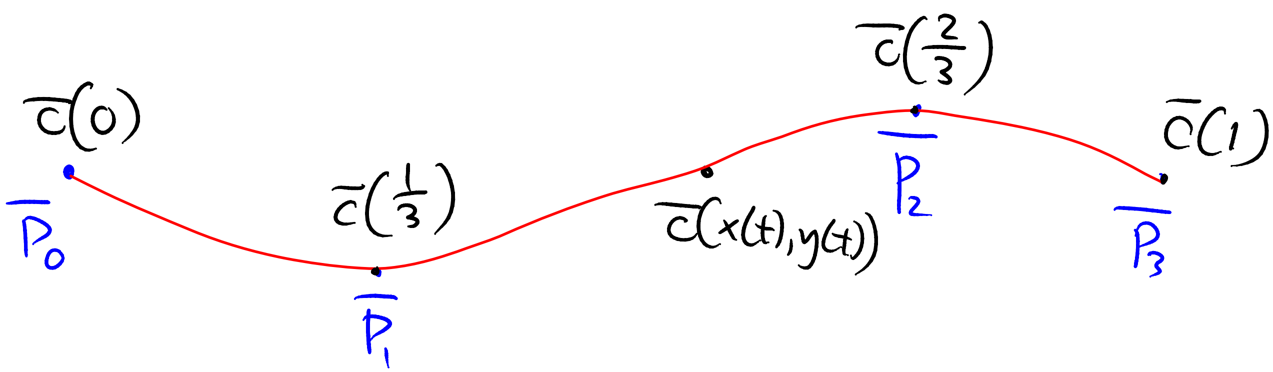
given $\overline{P_1}, \overline{P_2}, \overline{P_3}, \overline{P_4}$
compute a_i, b_i

Eqs for 4 control points:

$$\underbrace{\overset{C}{\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}}}_{\text{known}} = \underbrace{\overset{A}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1/3 & (1/3)^2 & (1/3)^3 \\ 1 & 2/3 & (2/3)^2 & (2/3)^3 \\ 1 & 1 & 1 & 1 \end{bmatrix}}}_{\text{known}} \underbrace{\overset{X}{\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \\ a_4 & b_4 \end{bmatrix}}}_{\text{unknown}}$$

$\Rightarrow$ solve system in terms of unknown matrix

$$X = A^{-1} C$$



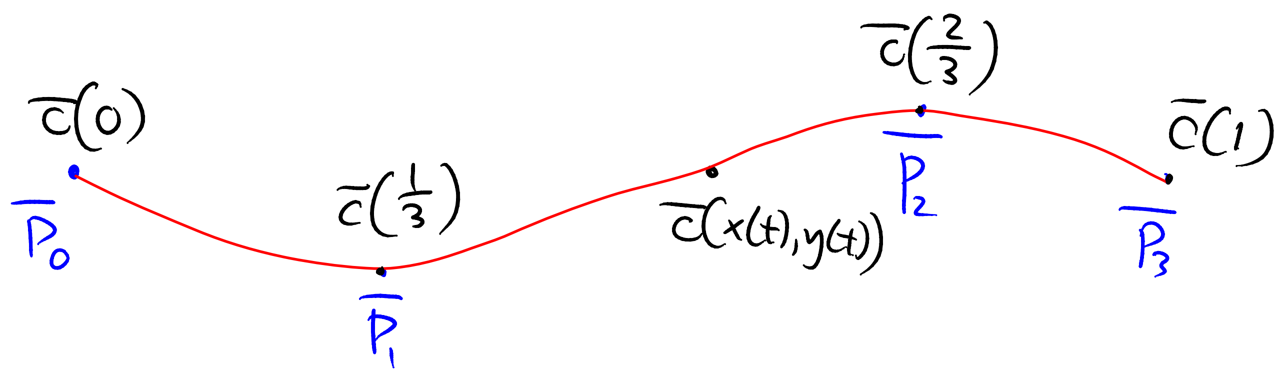
Cubic Interpolation: Computing the Coeffs

$$\left.\begin{aligned} x(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ y(t) &= b_0 + b_1 t + b_2 t^2 + b_3 t^2 \end{aligned}\right\}$$

given $\bar{P}_1, \bar{P}_2, \bar{P}_3, \bar{P}_4$
compute a_i, b_i

Coefficients of interpolating polynomial computed by

$$\boxed{\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \\ a_4 & b_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1/3 & (1/3)^2 & (1/3)^3 \\ 1 & 2/3 & (2/3)^2 & (2/3)^3 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}}$$

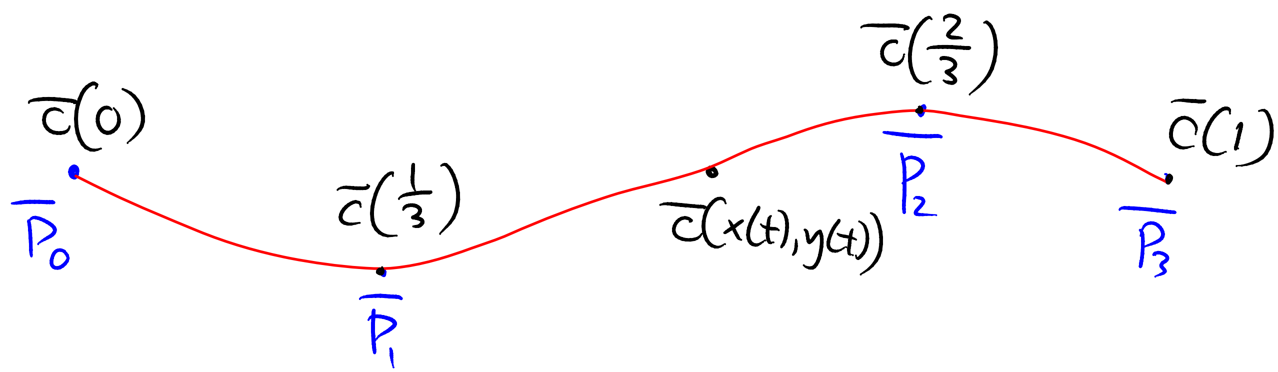


Cubic Interpolation: Evaluating the Polynomial

$$\left.\begin{aligned} x(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ y(t) &= b_0 + b_1 t + b_2 t^2 + b_3 t^2 \end{aligned}\right\}$$

given $\bar{P}_1, \bar{P}_2, \bar{P}_3, \bar{P}_4$
compute a_i, b_i

$$\boxed{\begin{bmatrix} x(t) & y(t) \end{bmatrix} = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}}$$

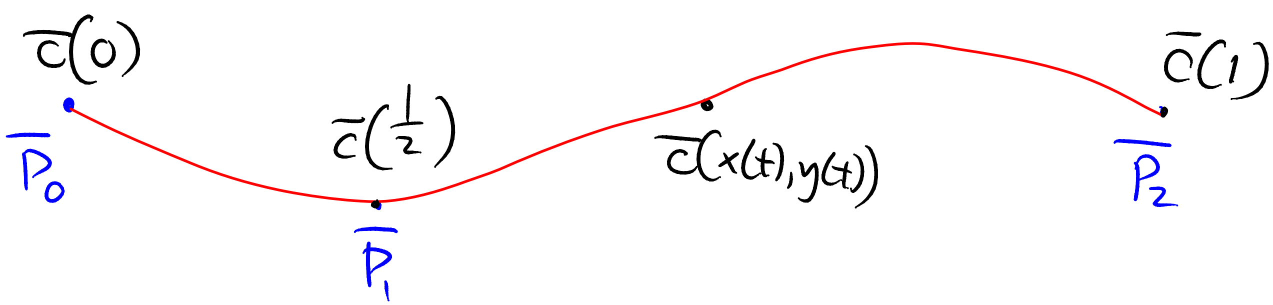


Cubic Interpolation: What If < 4 Control Points?

$$\left.\begin{aligned} x(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ y(t) &= b_0 + b_1 t + b_2 t^2 + b_3 t^2 \end{aligned}\right\} \quad \text{given } \bar{P}_1, \bar{P}_2, \bar{P}_3, \bar{P}_4 \text{ compute } a_i, b_i$$

Coefficients of interpolating polynomial computed by

$$\begin{array}{c} \uparrow \\ \text{degree} \\ +1 \\ \downarrow \end{array} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \\ a_4 & b_4 \end{bmatrix} = \underbrace{\begin{bmatrix} \text{more Eqs than} \\ \text{unknowns,} \\ \text{cannot compute} \\ \text{inverse} \end{bmatrix}^{-1}}_{\leftarrow \#\text{ control points} \rightarrow} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}$$

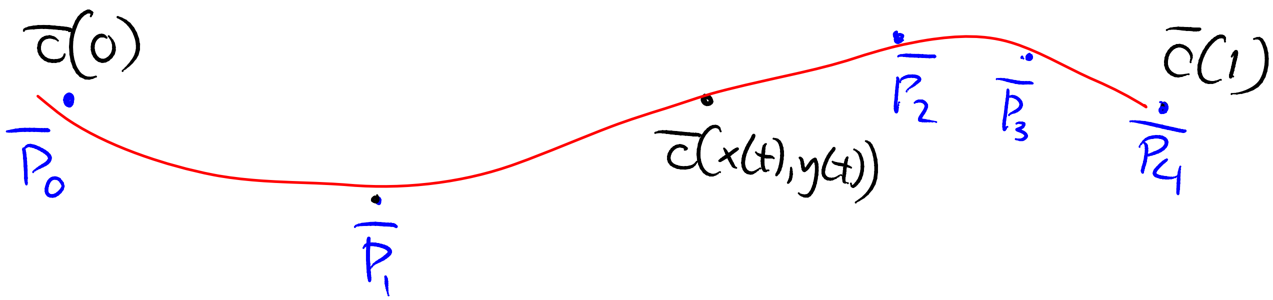


Cubic Interpolation: What If > 4 Control Points?

$$\left.\begin{aligned} x(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ y(t) &= b_0 + b_1 t + b_2 t^2 + b_3 t^2 \end{aligned}\right\}$$ given $\bar{P}_1, \bar{P}_2, \bar{P}_3, \bar{P}_4$ compute a_i, b_i

Coefficients of interpolating polynomial computed by

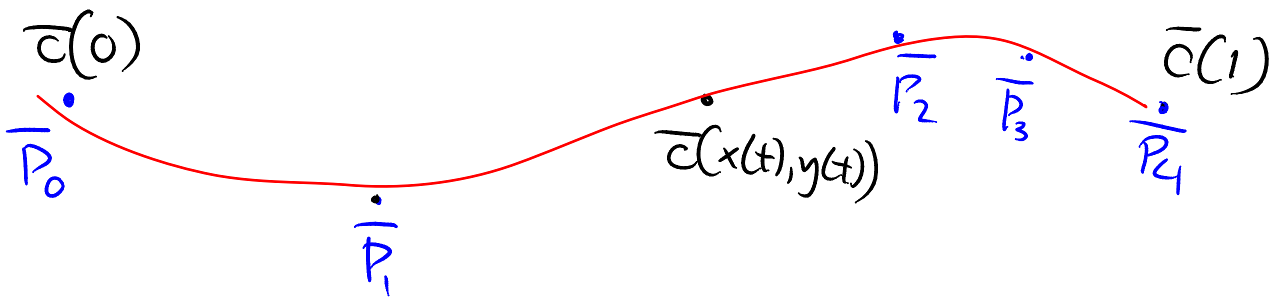
$$\begin{array}{c} \text{degree} \\ +1 \end{array} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \\ a_4 & b_4 \end{bmatrix} = \underbrace{\begin{bmatrix} \text{over-determined} \\ \text{linear system} \\ \Rightarrow \\ \text{poly cannot pass} \\ \text{through all pts} \end{bmatrix}^{-1}}_{\#\text{control points}} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}$$



Exact Interpolation of N Points

⇒ To interpolate N points perfectly with a single polynomial we need a polynomial of degree N-1

$$\begin{array}{cc} \text{degree } +1 & \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \\ a_4 & b_4 \end{bmatrix} \end{array} = \underbrace{\begin{bmatrix} \text{N×N matrix} \\ \#\text{ constraints } = \\ \#\text{ unknown coeffs} \end{bmatrix}^{-1}}_{\#\text{ control points.}} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}$$

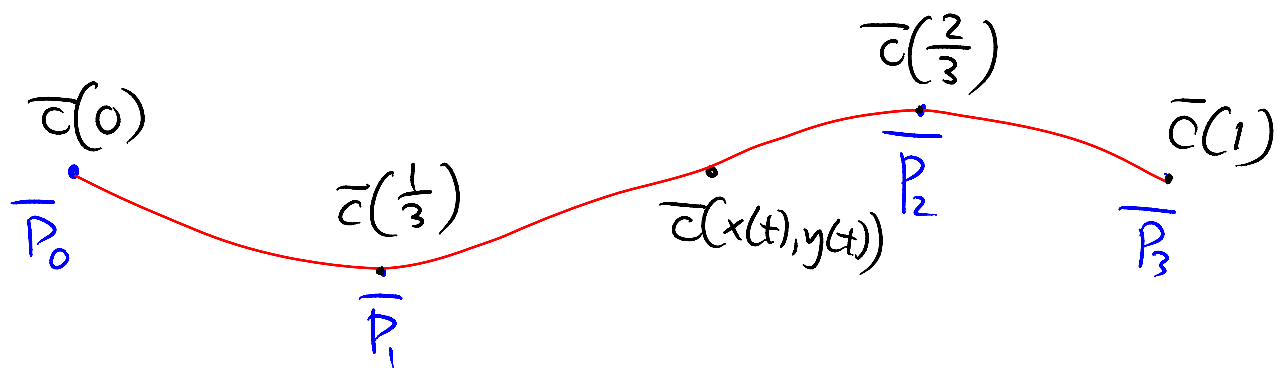


Cubic Interpolation: Evaluating Derivatives

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$$\frac{dx}{dt}(t) = a_1 + 2a_2 t + 3a_3 t^2$$

$$\boxed{\begin{bmatrix} \frac{dx}{dt}(t) & \frac{dy}{dt}(t) \end{bmatrix} = \begin{bmatrix} 1 & 2t & 3t^2 \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}}$$

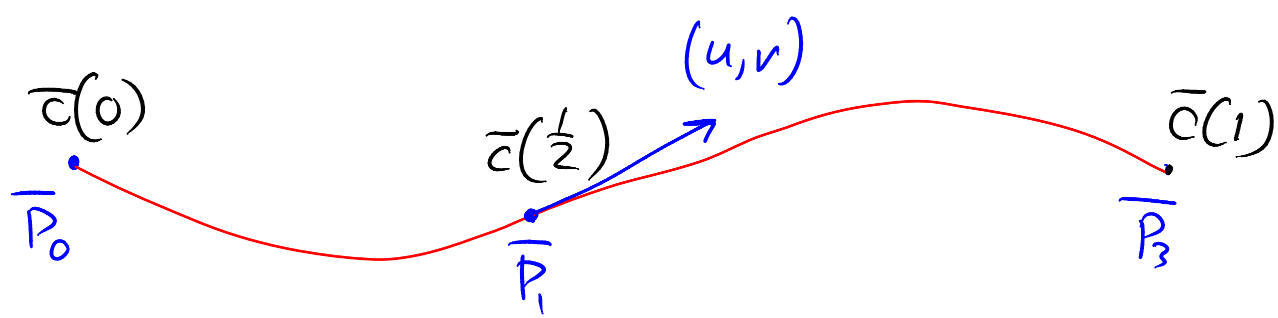


Specifying the Poly via Tangent Constraints

Instead of specifying 4 control points, we could specify 3 points and a derivative

$\Rightarrow$ replace 4th pair of eqs with

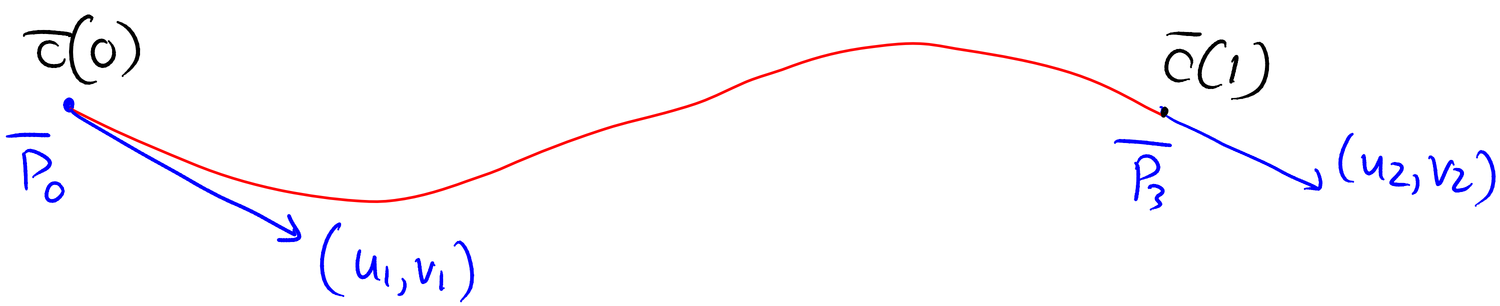
$$\boxed{\begin{gathered}\begin{bmatrix} u & v \end{bmatrix} = \begin{bmatrix} 1 & 1 & 3\left(\tfrac{1}{2}\right)^2 \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix} \\[1em] \begin{bmatrix} \dfrac{dx}{dt}(t) & \dfrac{dy}{dt}(t) \end{bmatrix} = \begin{bmatrix} 1 & 2t & 3t^2 \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}\end{gathered}}$$



Specifying the Poly via Tangent Constraints

Instead of specifying 4 control points, we could specify 2 points and 2 derivatives (we will use this later today!)

$$\boxed{\begin{bmatrix} \frac{dx}{dt}(t) & \frac{dy}{dt}(t) \end{bmatrix} = \begin{bmatrix} 1 & 2t & 3t^2 \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}}$$

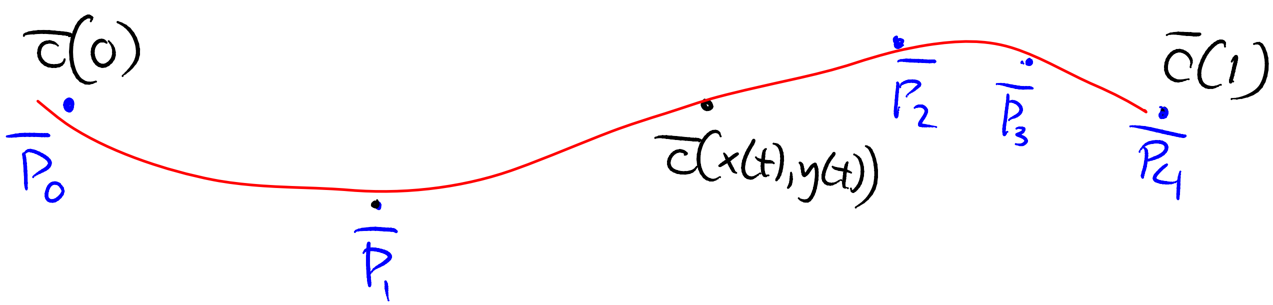


Degree-N Poly Interpolation: Major Drawback

⇒ To interpolate N points perfectly with a single polynomial we need a polynomial of degree N-1

Major drawback: it is a global interpolation scheme

i.e. moving one control point changes the interpolation of all points, often in unexpected/unintuitive ways

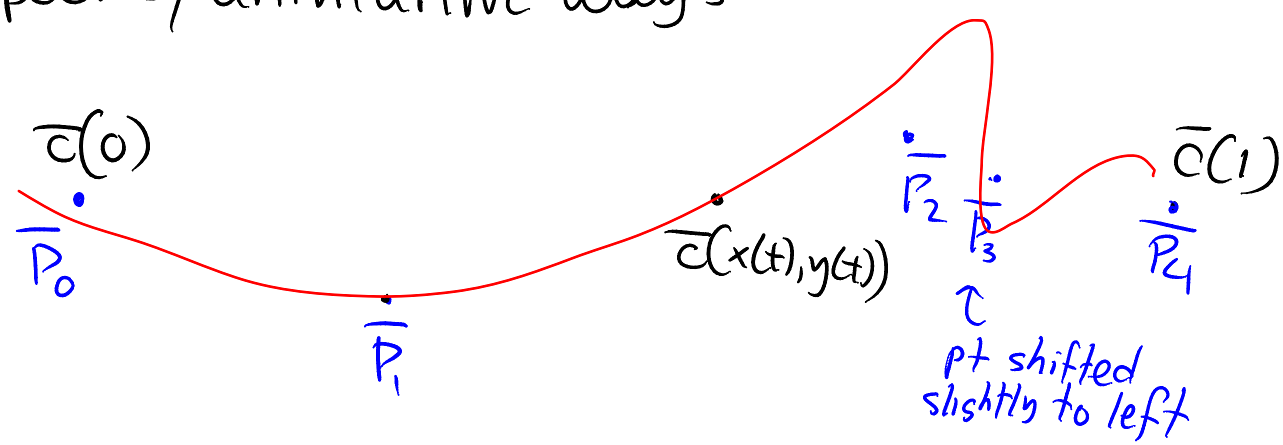


Degree-N Poly Interpolation: Major Drawback

⇒ To interpolate N points perfectly with a single polynomial we need a polynomial of degree N-1

Major drawback: it is a global interpolation scheme

i.e. moving one control point changes the interpolation of all points, often in unexpected/unintuitive ways



Topic 15:

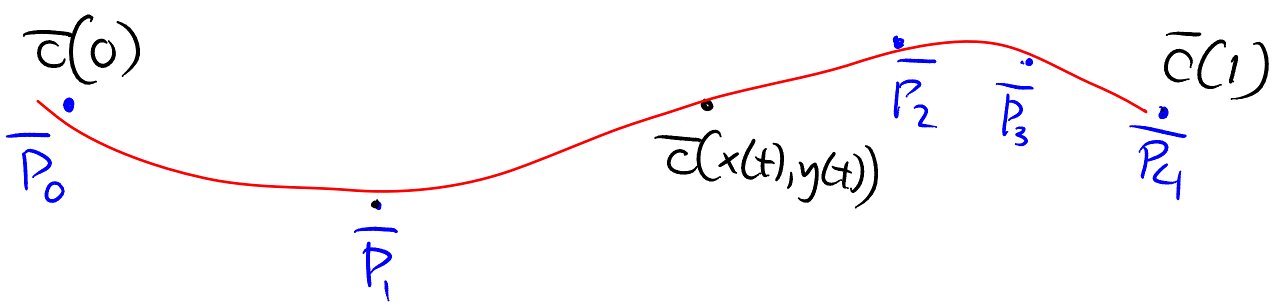
Interpolating Curves

- Intro to curve interpolation & approximation
- Polynomial interpolation
- Cardinal splines

Exact Interpolation of N Points

To interpolate N points perfectly with a single polynomial we need a polynomial of degree N-1

$$\text{degree}+1 \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \\ a_4 & b_4 \end{bmatrix} = \underbrace{\begin{bmatrix} \text{NxN matrix} \\ \#\text{ constraints} = \\ \#\text{ unknown coeffs} \end{bmatrix}^{-1}}_{\#\text{ control points}} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}$$

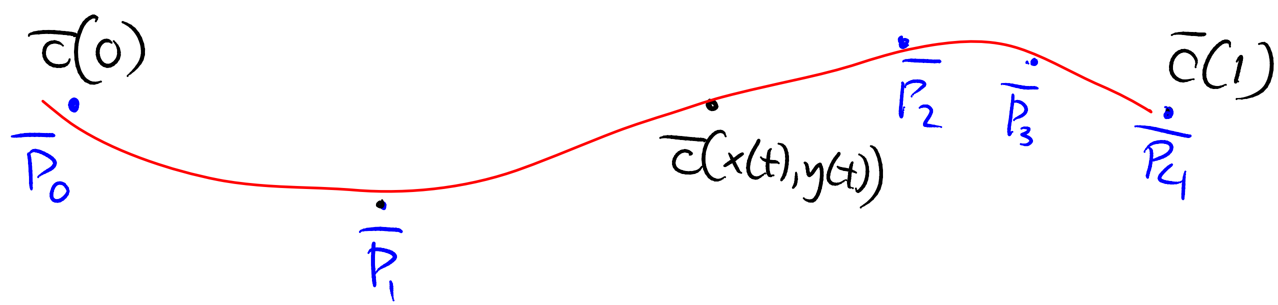


Degree-N Poly Interpolation: Major Drawback

⇒ To interpolate N points perfectly with a single polynomial we need a polynomial of degree N-1

Major drawback: It is a global interpolation scheme

i.e. moving one control point changes the interpolation of all points, often in unexpected/unintuitive ways

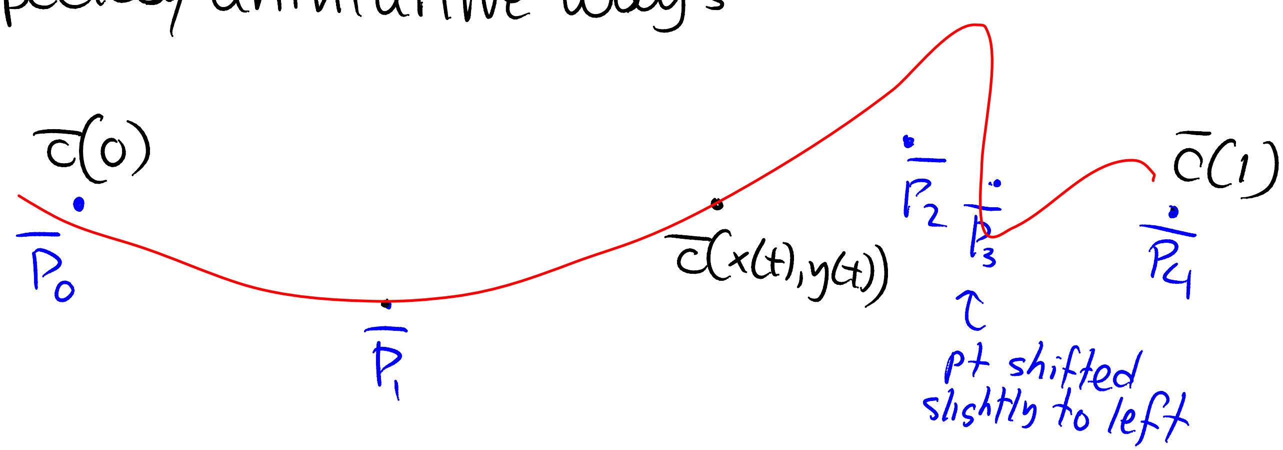


Degree-N Poly Interpolation: Major Drawback

⇒ To interpolate N points perfectly with a single polynomial we need a polynomial of degree N-1

Major drawback: it is a global interpolation scheme

i.e. moving one control point changes the interpolation of all points, often in unexpected/unintuitive ways

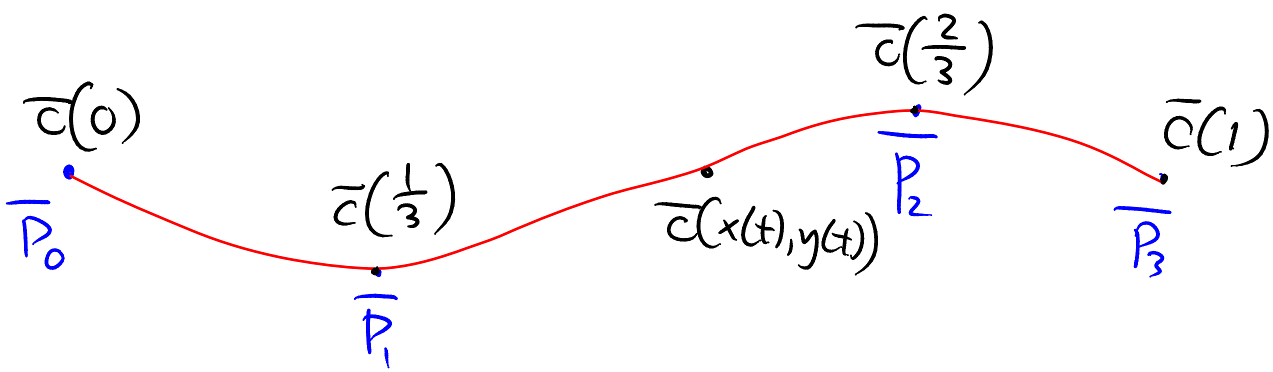


Cubic Interpolation: Evaluating Derivatives

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$$\frac{dx}{dt}(t) = a_1 + 2a_2 t + 3a_3 t^2$$

$$\boxed{\begin{bmatrix} \frac{dx}{dt}(t) & \frac{dy}{dt}(t) \end{bmatrix} = \begin{bmatrix} 1 & 2t & 3t^2 \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}}$$

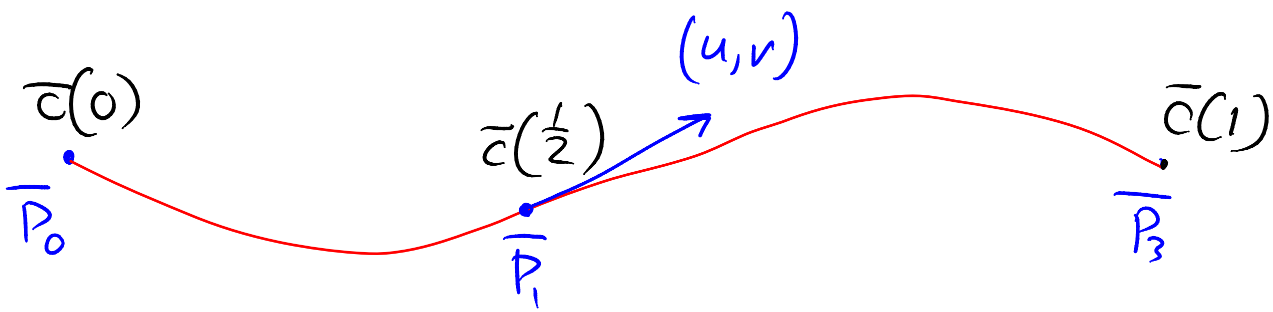


Specifying the Poly via Tangent Constraints

Instead of specifying 4 control points, we could specify 3 points and a derivative

$\Rightarrow$ replace 4th pair of eqs with

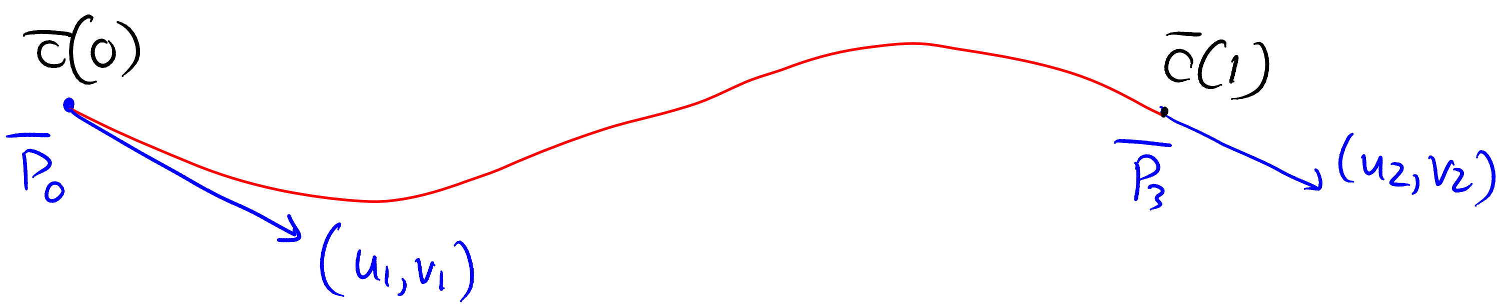
$$\boxed{\begin{gathered}\begin{bmatrix} u & v \end{bmatrix} = \begin{bmatrix} 1 & 1 & 3\left(\tfrac{1}{2}\right)^2 \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix} \\ \begin{bmatrix} \frac{dx}{dt}(t) & \frac{dy}{dt}(t) \end{bmatrix} = \begin{bmatrix} 1 & 2t & 3t^2 \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}\end{gathered}}$$



Specifying the Poly via Tangent Constraints

Instead of specifying 4 control points, we could specify 2 points and 2 derivatives

$$\boxed{\begin{bmatrix} \frac{dx}{dt}(t) & \frac{dy}{dt}(t) \end{bmatrix} = \begin{bmatrix} 1 & 2t & 3t^2 \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}}$$



Topic 15:

Interpolating Curves

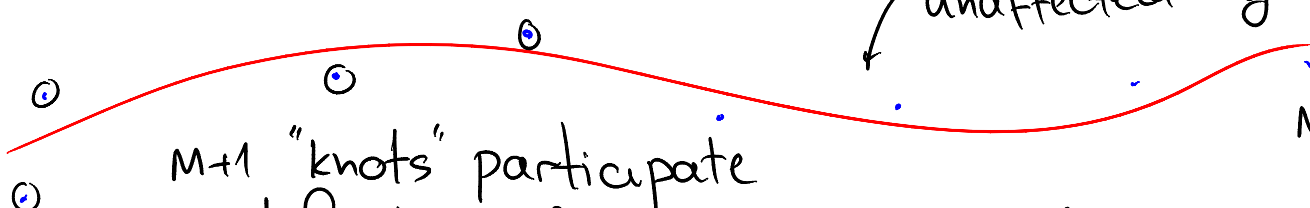
- Intro to curve interpolation & approximation
- Polynomial interpolation
- Cardinal splines

Splines: Local Control + Reduced Continuity

- Idea: provide local control of curve approximating N points by sacrificing C^N continuity

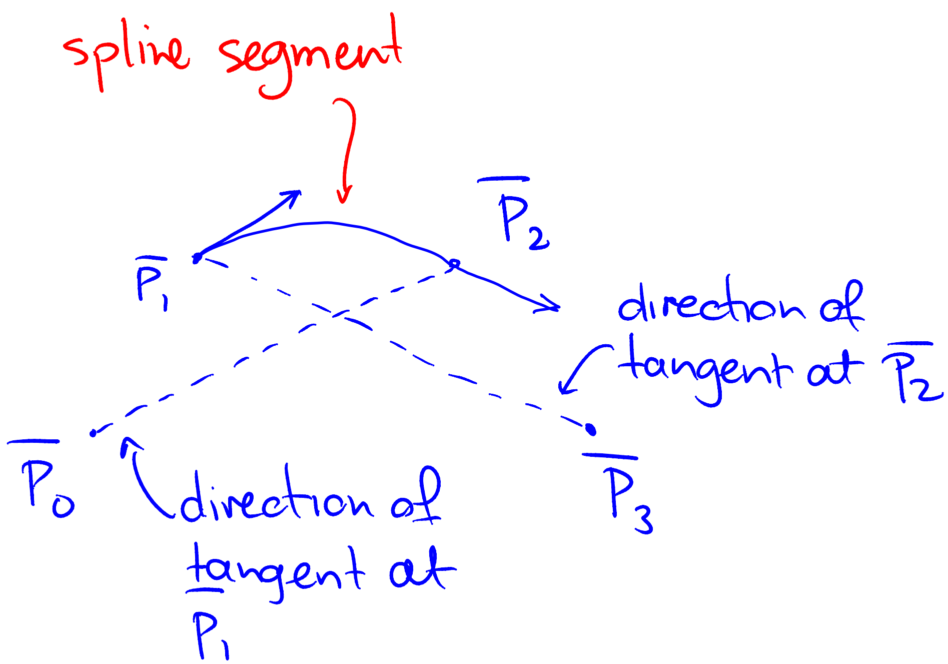
- M-degree spline: a piecewise-polynomial curve of degree M

- Splines often defined to have C^{M-2} continuity at the "knot" points (a.k.a. control points)



Cubic Cardinal Splines

- Idea: provide local control of curve interpolating N-2 points, with C^1 continuity
- Cubic polynomials used as basic building block



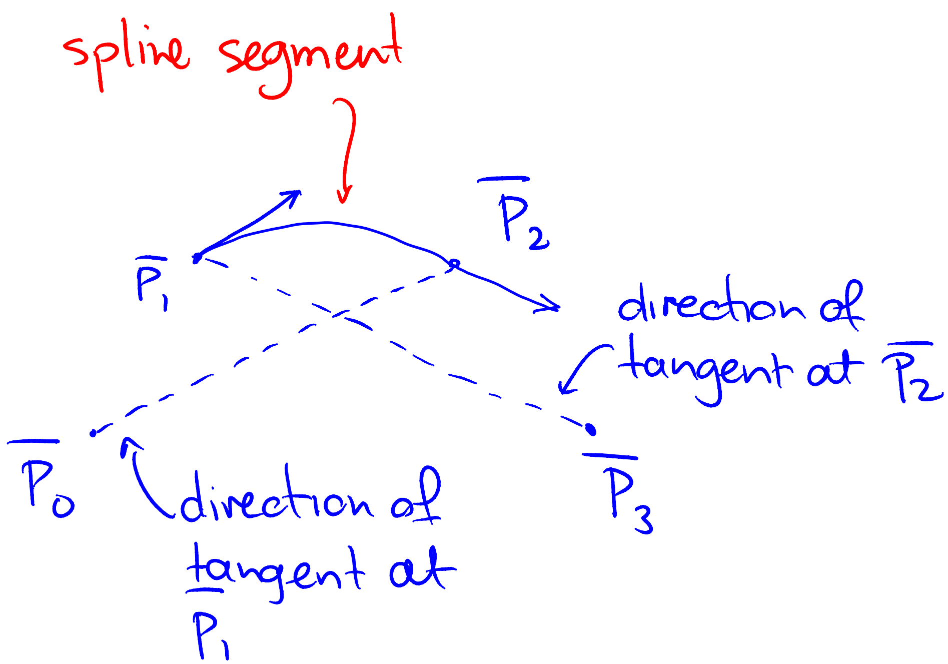
$\bar{P}_1, \bar{P}_2$ act as endpoint constraints

$\bar{P}_0, \bar{P}_3$ define derivative constraints

Cubic Cardinal Splines: Specifying the Curve

Approach: ① A user only specifies points $\bar{P}_0, \bar{P}_1, \ldots$

② tangent at $\bar{P}_i$ set to be parallel to vector connecting $\bar{P}_{i-1}$ and $\bar{P}_{i+1}$



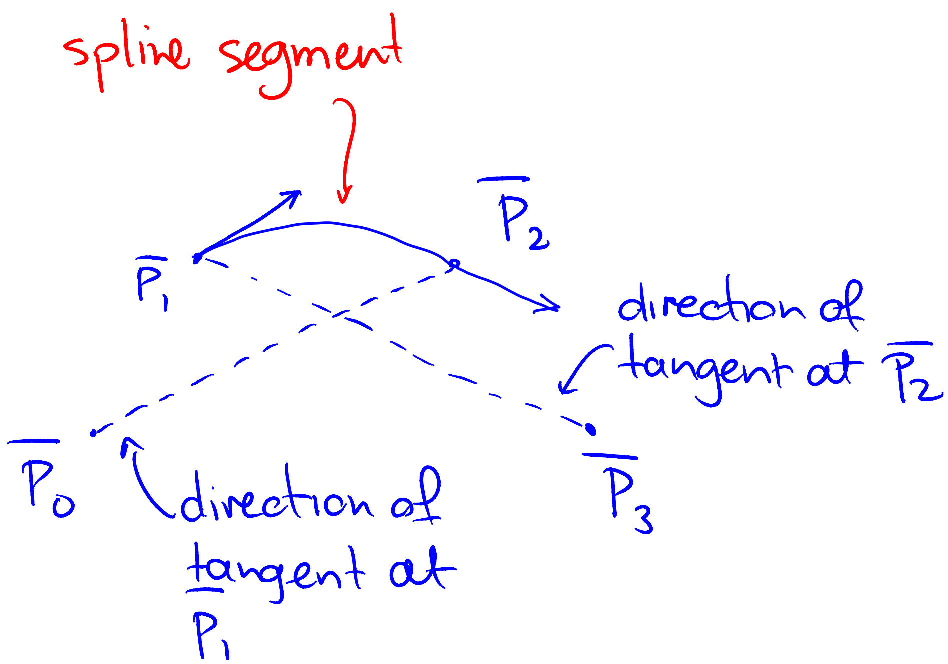
$\bar{P}_1, \bar{P}_2$ act as endpoint constraints

$\bar{P}_0, \bar{P}_3$ define derivative constraints

Cubic Cardinal Splines: Defining 1st Segment

Approach: ① A user only specifies points $\bar{P}_0, \bar{P}_1, \ldots$

② tangent at $\bar{P}_i$ set to be parallel to vector connecting $\bar{P}_{i-1}$ and $\bar{P}_{i+1}$



$\bar{P}_1, \bar{P}_2$ act as endpoint constraints

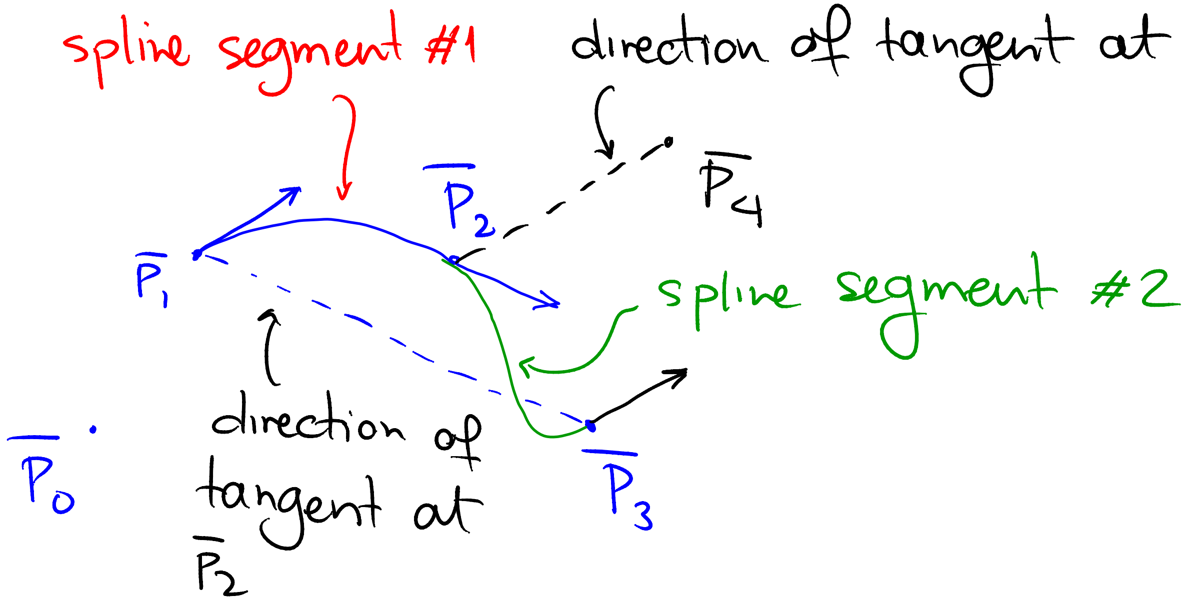
$\bar{P}_0, \bar{P}_3$ define derivative constraints

Cubic Cardinal Splines: Defining 2nd Segment

Approach: ① A user only specifies points $\overline{P}_0, \overline{P}_1, \ldots$

② tangent at $\overline{P}_i$ set to be parallel to vector connecting $\overline{P}_{i-1}$ and $\overline{P}_{i+1}$

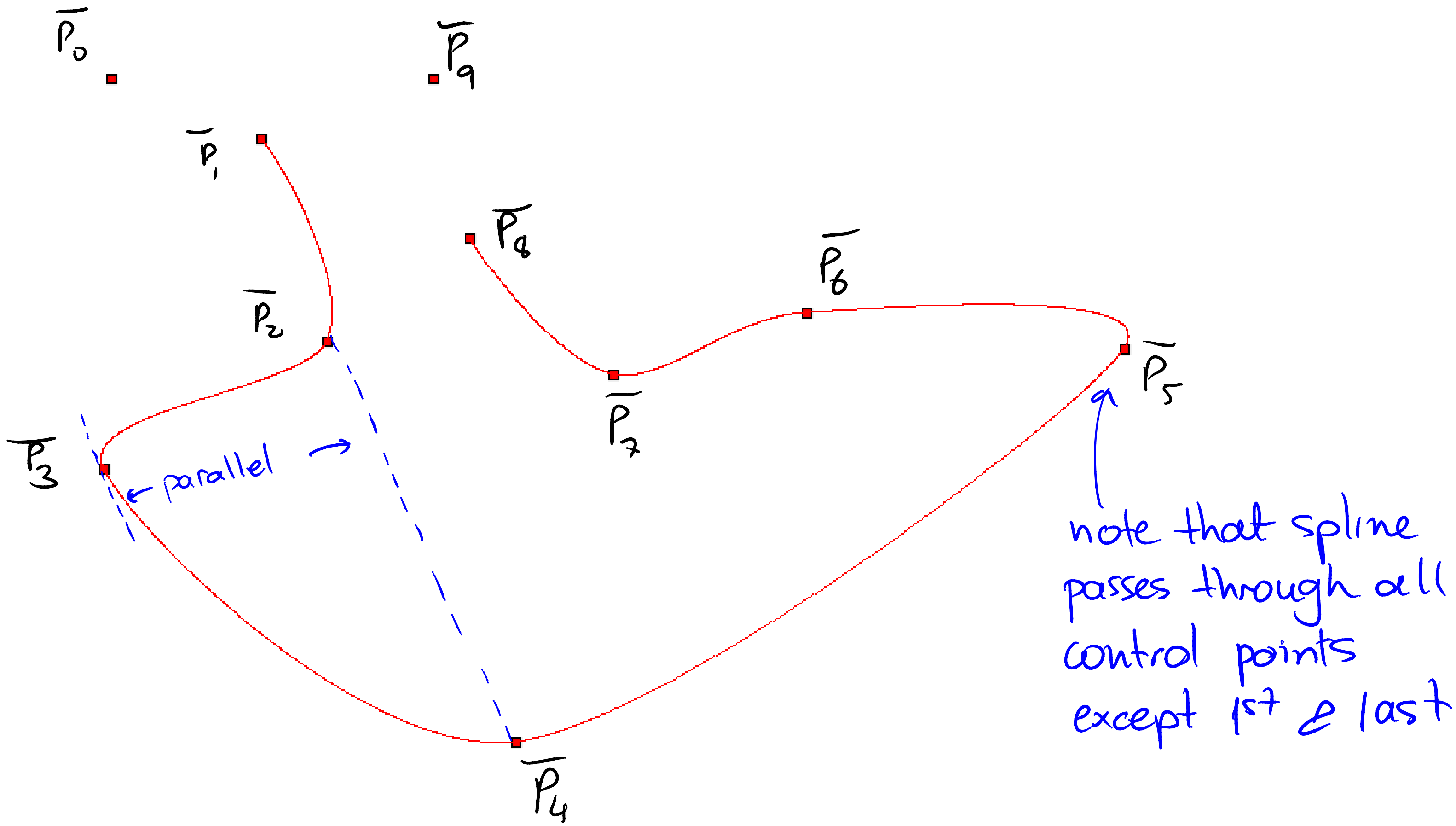
Example: Adding a fifth point adds a new segment



Cubic Cardinal Splines: General Case

Approach: ① A user only specifies points $\bar{P}_0, \bar{P}_1, \ldots$

② tangent at $\bar{P}_i$ set to be parallel to vector connecting $\bar{P}_{i-1}$ and $\bar{P}_{i+1}$



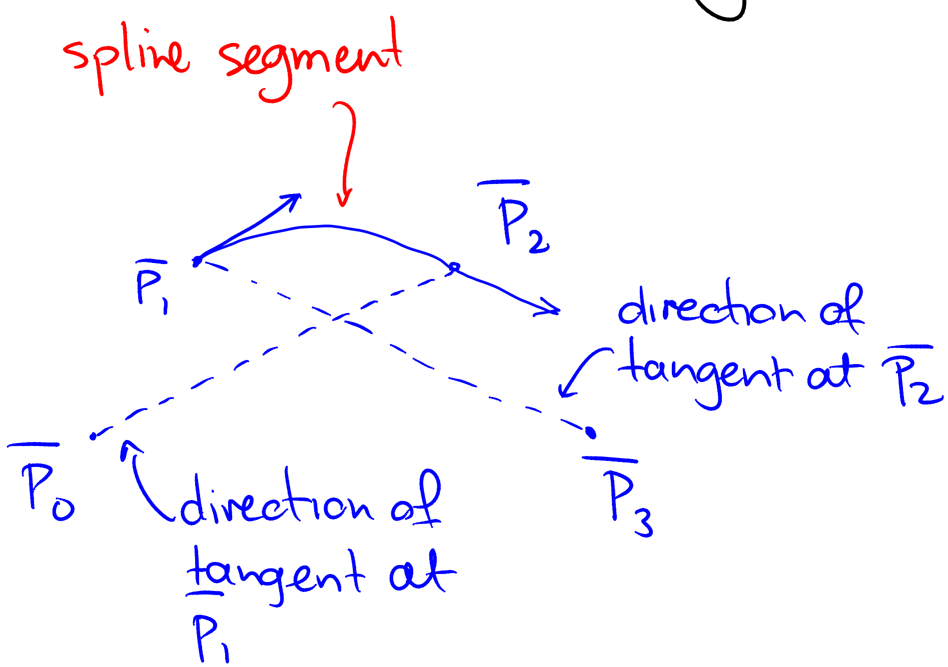
Cardinal Splines: The Strain Parameter

Approach: ① A user only specifies points $\bar{P}_0, \bar{P}_1, \ldots$

② tangent at $\bar{P}_i$ set to be parallel to vector connecting $\bar{P}_{i-1}$ and $\bar{P}_{i+1}$

$$\text{tangent at } \bar{P}_i = k(\bar{P}_{i+1} - \bar{P}_{i-1})$$

called a <u>strain</u> parameter



Catmull-Rom Splines

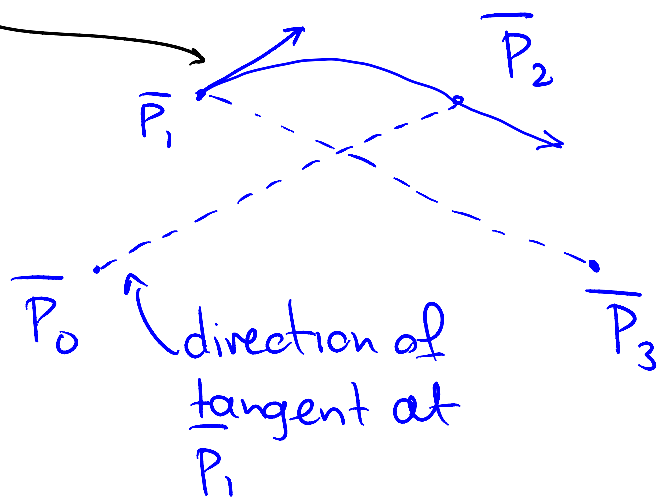
Approach: ① A user only specifies points $\bar{P}_0, \bar{P}_1, \ldots$

② tangent at $\bar{P}_i$ set to be parallel to vector connecting $\bar{P}_{i-1}$ and $\bar{P}_{i+1}$

$$\text{tangent at } \bar{P}_i = k(\bar{P}_{i+1} - \bar{P}_{i-1})$$

called a strain parameter

length of tangent = $\frac{1}{2}$ distance between $\bar{P}_0$ and $\bar{P}_2$



Note: if $k = \frac{1}{2}$ the spline is called a Catmull-Rom spline