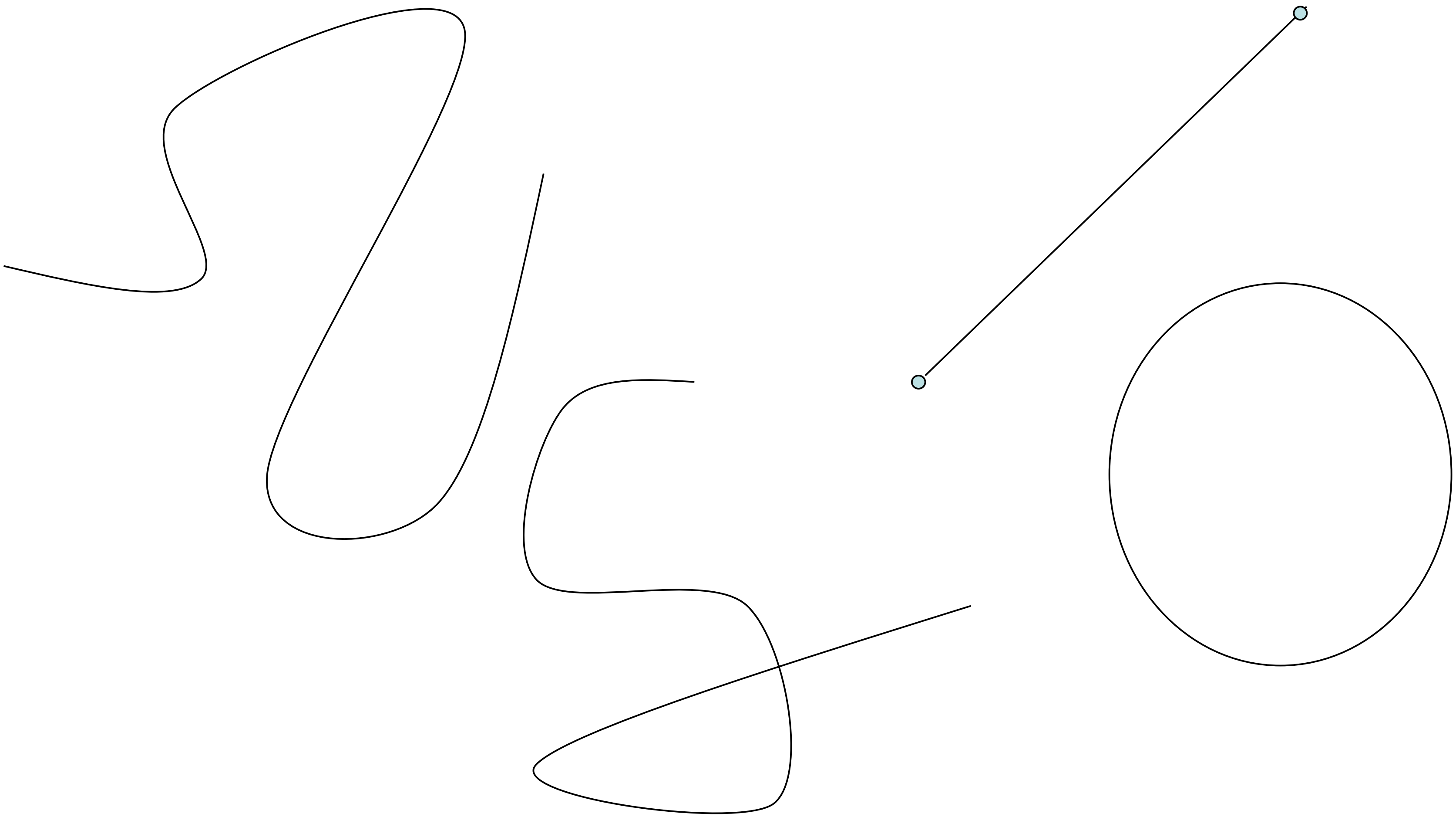
Geometric Modeling: curves

- Bézier Curves
- Spline Curves
- NURBS
- Subdivision curve

How to represent a ‘free form’?

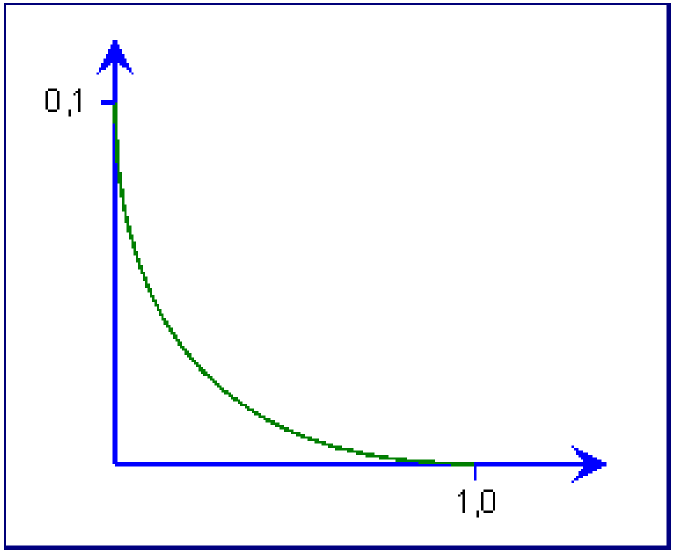


C(t): Curve which interpolates endpoints (0,1) and (1,0) tangent at x-y axes in (0,1) and (1,0).

Parametric Form: $C(t) = at^2 + bt + c$

a,b,c coefficient vectors

$$C(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} t^2 + \begin{pmatrix} -2 \\ 0 \end{pmatrix} t + \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

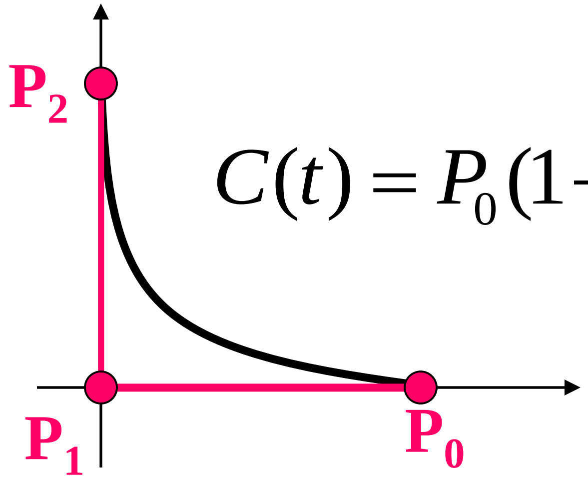


Rewrite as:

$$C(t) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} (1-t)^2 + \begin{pmatrix} 0 \\ 0 \end{pmatrix} (1-t)2t + \begin{pmatrix} 0 \\ 1 \end{pmatrix} t^2$$

Advantages in the new representation form?

The coefficients have a geometric interpretation: (1,0),(0,0),(0,1) are defined control points. When these points are connected in the order of their numbering with straight lines they form the control polygon which approximates the curve shape.

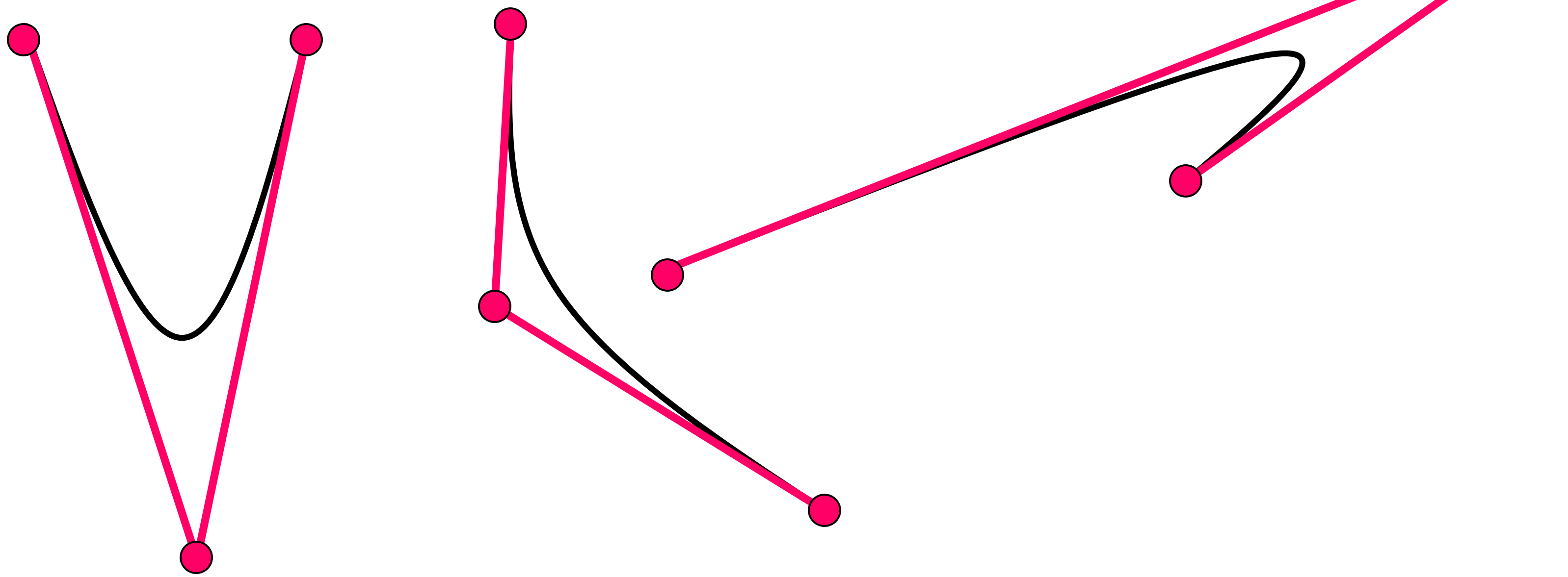


$$C(t) = P_0(1-t)^2 + P_1 2t(1-t) + P_2 t^2$$

This curve is defined Bézier curve (P. Bézier).

Bézier curves of degree 2

$$C(t) = P_0(1-t)^2 + P_1 2t(1-t) + P_2 t^2$$



Moving the control points gives the user an intuitive sense of how change/control the shape of the curve

Bézier Curves

Pierre Étienne Bézier
an engineer at Renault
(1910-1999)

A Bézier curve of degree n (order m=n+1) in parametric form is defined by an ordered sequence of points P_i i=0,..,n, in d-dimensional space $\mathbf{R}^d$, for d=2,3,4:

$$C(t) = \sum_{i=0}^{n} P_i B_i^n(t)$$

where

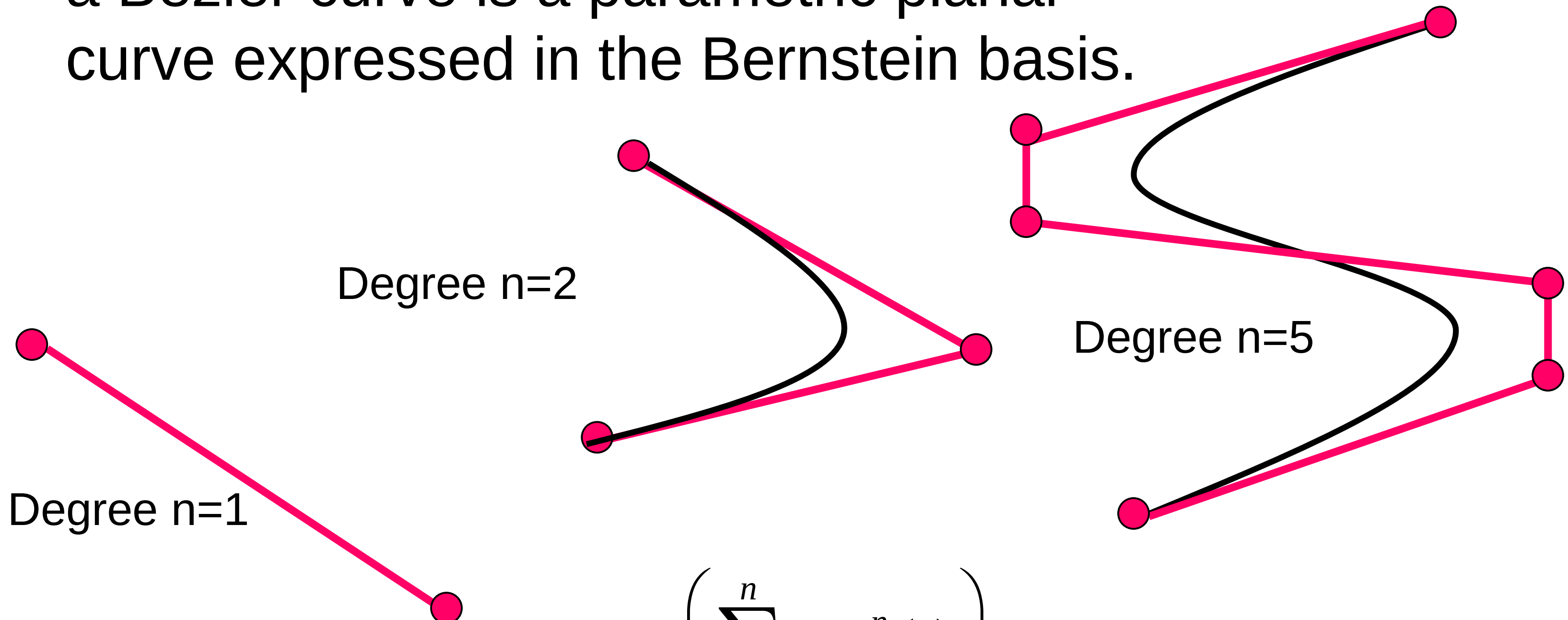
$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0,..,n$$

are Bernstein Basis Functions.

P_i are called control points and the polygon joining these points is called Control Polygon

Bézier curves of degree n

In $\mathbf{R}^2$, given the control points $P_i=(x_i,y_i)$ a Bézier curve is a parametric planar curve expressed in the Bernstein basis.



$$C(t)=\begin{pmatrix} x(t) \\ y(t) \end{pmatrix}=\begin{pmatrix} \sum_{i=0}^{n} x_i B_i^n(t) \\ \sum_{i=0}^{n} y_i B_i^n(t) \end{pmatrix} \qquad x(t),y(t)\in P_n$$

Evaluation of a Bézier Curve

- Find the exact values of C(t) for a given value of t.
- Several ways to represent mathematically a Bézier curve:
  - By Bernstein Polynomial basis
  - Matrix Form
  - de Casteljau Algorithm (linear interpolation)

Bernstein Polynomials

Bernstein Polynomials are scalar-valued functions of degree n in the interval [0,1], for $0 \le t \le 1$

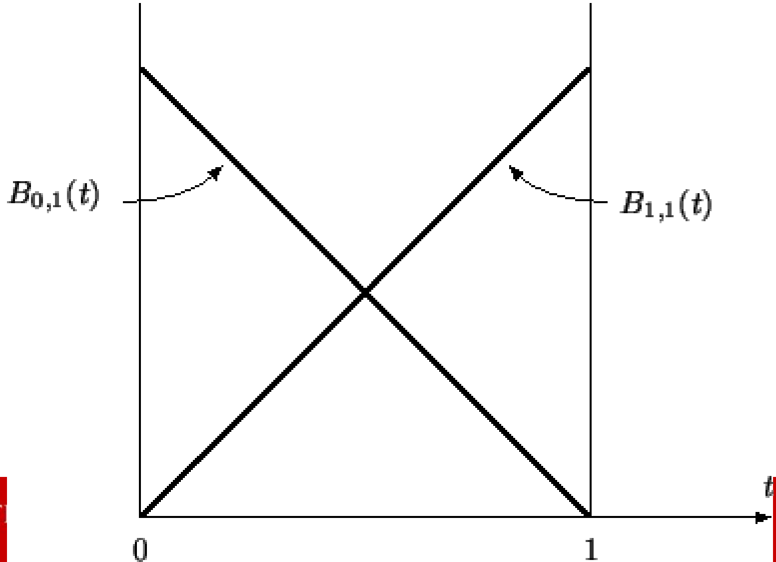
$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0,..,n \qquad \binom{n}{i} = \frac{n!}{i!(n-i)!}$$

Basis for the linear space of polynomial P_n of degree at most n

$$\left\{B_i^n(t)\right\}_{i=0}^{n} \qquad \left\{x^i\right\}_{i=0}^{n}$$

Bernstein Poly of degree 1:

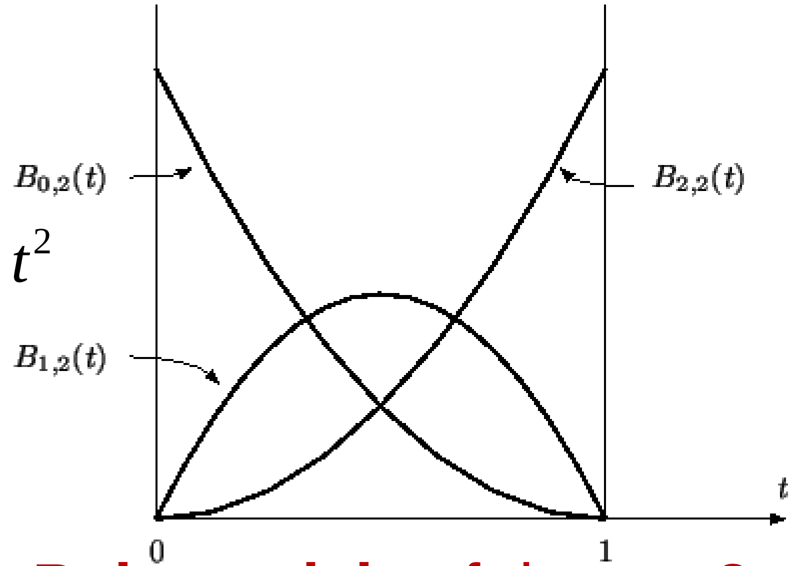
$$B_0^1(t) = 1 - t, \quad B_1^1(t) = t$$



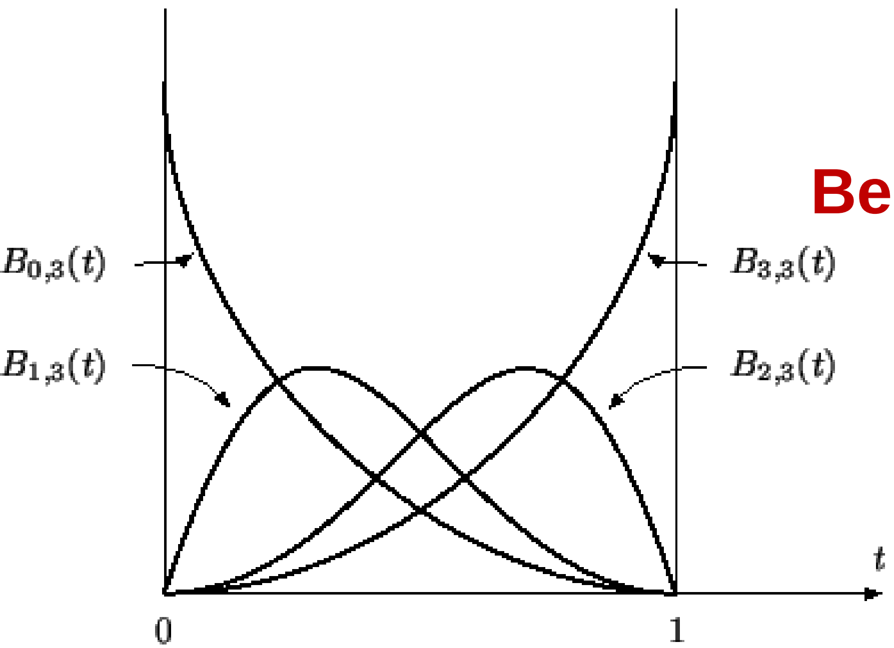
Bernstein Polynomials

Bernstein Polynomials of degree 2:

$$B_0^2(t) = (1-t)^2, \quad B_1^2(t) = 2t(1-t), \quad B_2^2 = t^2$$



Bernstein Polynomials of degree 3:



$$B_0^3(t) = (1-t)^3, \quad B_1^3(t) = 3t(1-t)^2,$$

$$B_2^3 = 3t^2(1-t), \quad B_3^3(t) = t^3$$

Bernstein Polynomials
of degree n over Arbitrary Parameter Intervals [a,b]

$$B_i^n(x) = \binom{n}{i} \frac{(b-x)^{n-i}(x-a)^i}{(b-a)^n}, \quad i = 0,..,n$$

Invariant under an affine reparametrization, or parameter transformation

$$x \in [a,b] \quad \rightarrow t \in [0,1]$$

$$x = a + t(b-a)$$

Poly are translation and scaling invariant

$$B_i^n(x) = B_i^n(a + t(b-a)) =$$

$$= \binom{n}{i} \frac{(b-a-t(b-a))^{n-i}(a+t(b-a)-a)^i}{(b-a)^n}$$

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

Polynomial representation using Bernstein basis

$$f(x)=\sum_{i=0}^{n} b_i B_i^n(x) \qquad x\in[a,b]$$

$$f(t)=\sum_{i=0}^{n} b_i B_i^n(t) \qquad t\in[0,1]$$

The change of variable does not alterate the coefficients.

To evaluate f(x) at x=$\underline{x}$, first determine $\underline{t}=(\underline{x}-a)/(b-a)$
then evaluate $f(\underline{t})$, finally $f(\underline{x})=f(\underline{t})$

Bernstein Polynomials: PROPERTIES

P1) They take on positive values in the interval [0,1]

$$B_i^n(t) \geq 0,$$

P2) They are a partition of unity

$$\sum_{i=0}^{n} B_i^n(t) = 1, \quad t \in [0,1]$$

If P2) holds, then for each set of points $P_0, P_1, .., P_n$, and for each t, the formula

$$\boldsymbol{f(t) = P_0 B_0^n(t) + P_1 B_1^n(t) + \ldots + P_n B_n^n(t)}$$

is an ***affine combination*** of the set of points; moreover, if t belongs to [0,1], then $1 \geq B_i^n(t) \geq 0,$

P1) holds and *f(t)* is a ***convex combination*** of the points.

Conversion to the power basis

$$\boxed{f(t) = \sum_{i=0}^{n} c_i B_i^n(t) = \sum_{i=0}^{n} a_i t^i}$$

$$\{1, t, t^2, \ldots, t^n\}, \quad \{B_i^n\}_{i=0}^{n}$$

$$B^n{}_i(t) = \sum_{k=i}^{n} (-1)^{k-i} \binom{n}{k} \binom{k}{i} t^k,$$

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

Bernstein polynomial in matrix form

A polynomial is a linear combination of Bernstein basis functions:

$$f(t) = c_0 B_0^n(t) + c_1 B_1^n(t) + \ldots + c_n B_n^n$$

$$f(t) = \begin{bmatrix} B_0^n(t) & B_1^n(t) & \ldots & B_n^n(t) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \ldots \\ c_n \end{bmatrix}$$

Using the representation in power basis functions:

$$f(t) = \begin{bmatrix} 1 & t & t^2 \ldots & t^n \end{bmatrix} \begin{bmatrix} b_{00} & 0 & 0 & 0 \\ b_{10} & b_{11} & \ldots & 0 \\ \ldots & \ldots & \ldots & 0 \\ b_{n0} & b_{n1} & \ldots & b_{nn} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \ldots \\ c_n \end{bmatrix}$$

Example

Bernstein polynomial of degree 2:

$$f(t) = c_0 B_0^2(t) + c_1 B_1^2(t) + c_2 B_2^2$$

$$f(t) = \begin{bmatrix} 1 & t & t^2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix}$$

In fact:

$$f(t) = [(1-t)^2 \quad 2t(1-t) \quad t^2] \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix}$$

Bézier Curve in R^3 in matrix form

- $C(t)=[x(t) \;\; y(t) \;\; z(t)]$
- $P = (Px,Py,Pz)$ vector of control points,
- M matrix of basis conversion
- $T=[1 \quad t \quad t^2 \quad t^3 \ldots \quad t^n]$

$$x(t)= T \cdot M \cdot Px$$

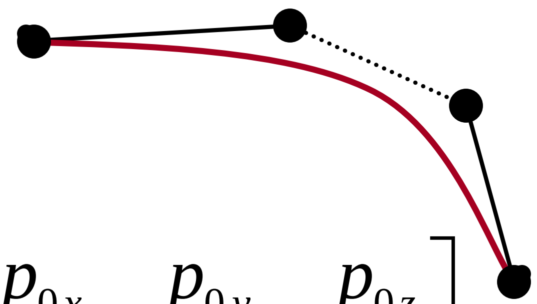
$$y(t)= T \cdot M \cdot Py$$

$$z(t)= T \cdot M \cdot Pz$$

Bézier Curve in R^3 in matrix form

Example: cubic curve, n=3

4 control points $P=[P_0\ P_1\ P_2\ P_3]$



$C(t)=T\bullet M\bullet P$

$$C(t)=\begin{bmatrix}1 & t & t^2 & t^3\end{bmatrix}\cdot\begin{bmatrix}1 & 0 & 0 & 0\\ -3 & 3 & 0 & 0\\ 3 & -6 & 3 & 0\\ -1 & 3 & -3 & 1\end{bmatrix}\cdot\begin{bmatrix}p_{0x} & p_{0y} & p_{0z}\\ p_{1x} & p_{1y} & p_{1z}\\ p_{2x} & p_{2y} & p_{2z}\\ p_{3x} & p_{3y} & p_{3z}\end{bmatrix}$$

M matrix of basis conversion

Derivatives of a Bézier curve

Cubic curve (n=3).

$$\boxed{\begin{aligned} C(t) &= B_0^3(t)P_0 + B_1^3(t)P_1 + B_2^3(t)P_2 + B_3^3(t)P_3 \\ &= (1-t)^3 P_0 + 3(1-t)^2 t\ P_1 + 3(1-t)t^2 P_2 + t^3 P_3 \end{aligned}}$$

It's a vector function of 2 or 3 cubic poly components, compute the derivatives of each components:

$$\frac{dC(t)}{dt} = -3(1-t)^2 P_0 - 6(1-t)\ t\ P_1 + 3(1-t)^2 P_1 - 3t^2 P_2 + 6(1-t)tP_2 + 3t^2 P_3$$

$$\begin{aligned} \frac{dC(t)}{dt} &= 3(1-t)^2 (P_1 - P_0) + 6(1-t)t(P_2 - P_1) + 3t^2 (P_3 - P_2) \\ &= 3B_0^2(t)(P_1 - P_0) + 3B_1^2(P_2 - P_1) + 3B_2^2(P_3 - P_2) \end{aligned}$$

It's a **Bézier curve** of degree 2

Derivatives of a Bézier curve of degree n

The first derivative

The first derivative of a Bézier curve is itself a Bézier curve of degree decreased by one,

$$C'(t) = \frac{dC(t)}{dt} = n\sum_{i=0}^{n-1} \Delta P_i \, B_i^{n-1}(t)$$

and CP the vectors $\Delta P_i := (P_{i+1} - P_i)$

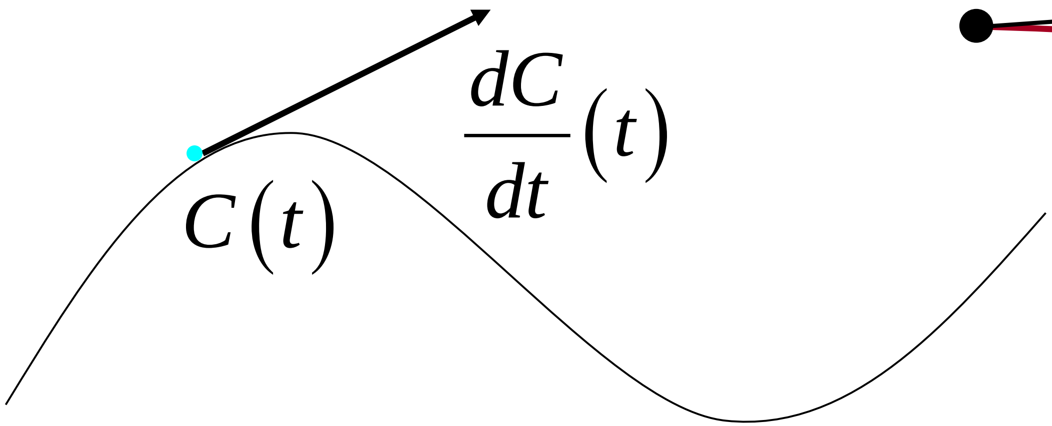
The r-th derivative

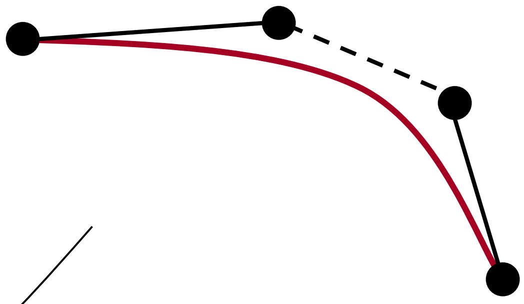
$$C^r(t) = \frac{n!}{(n-r)!}\sum_{i=0}^{n-r} \Delta^r P_i \, B_i^{n-r}(t)$$

where $\Delta^r P_i := \left(\Delta^{r-1} P_{i+1} - \Delta^{r-1} P_i\right) \quad \Delta^0 P_i := P_i$

Tangents

- The derivative of a curve represents the tangent vector to the curve at some point
- The control polygon of C(t) is tangent to the curve at the beginning and end of the curve





$$C'(t) = \frac{dC(t)}{dt} = 3(1-t)^2(P_1 - P_0) + 6(1-t)t(P_2 - P_1) + 3t^2(P_3 - P_2)$$

$$\Rightarrow C'(0) = 3(P_1 - P_0) \qquad C'(1) = 3(P_3 - P_2)$$

Excercise

A degree n=3 Bézier curve in R^2 satisfies

$C(0)=(0,1)$, $C(1)=(3,0)$

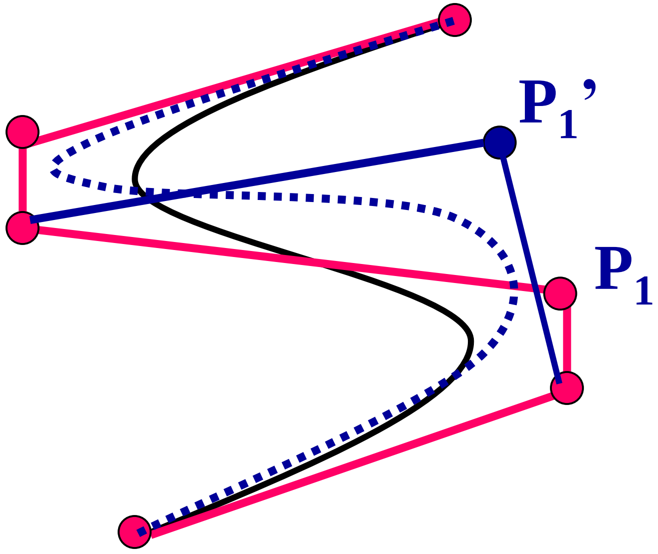
$C'(0)=(3,3)$, $C'(1)=(-3,0)$

What are the control points for this curve?

Give a rough freehand sketch of the curve, being sure to show the slopes at the beginning and end of the curve clearly.

Properties of a Bézier curve

- **Shape Control**



Moving a control point modifies the shape of the curve.

Properties of a Bézier curve

- The curve interpolates only its first and last control points

$$\boxed{C(0) = P_0 \quad C(1) = P_n}$$

$$C(t) = (1-t)^3 P_0 + 3(1-t)^2 t\ P_1 + 3(1-t)t^2 P_2 + t^3 P_3$$
$$t = 0 \quad C(0) = P_0 \qquad\qquad t = 1 \quad C(1) = P_3$$

- It's variation diminishing (without undesired oscillations) it has no more intersections with a line than its control polygon

Properties of a Bézier curve

- It's invariant under an affine transformation (translation, rotation, scaling, or shear): apply an affine transformation to the control points and then evaluate the curve represented by these transformed control points at t_i, is the same as apply an affine transformation to the point $C(t_i)$

- The curve is smooth with smooth derivatives

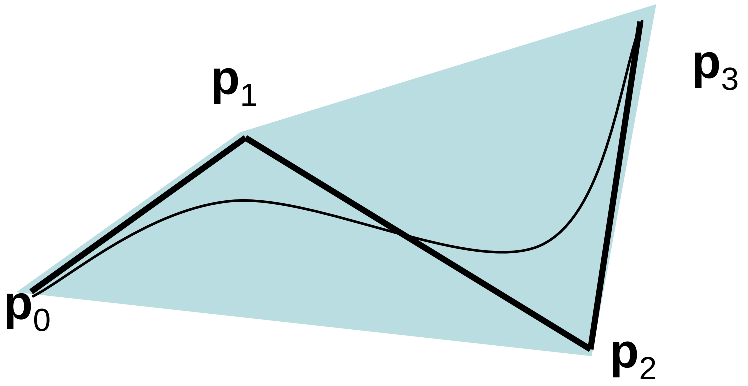
- The curve is tangent at the first and last control points, to the first and last line segments of the control polygon.

Properties of a Bézier curve

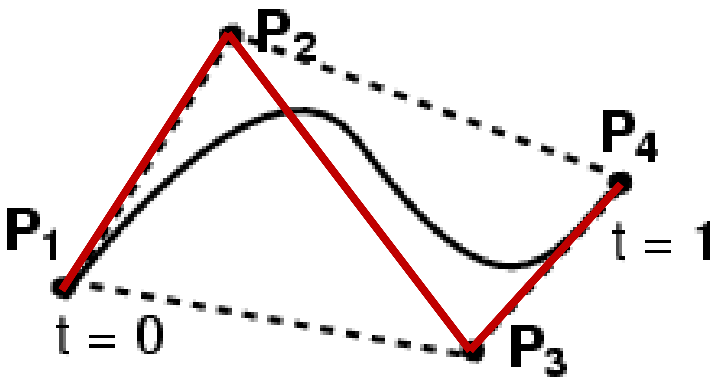
- The curve is contained into the *convex hull* of the control points, that is inside the smallest polygon formed by its control points

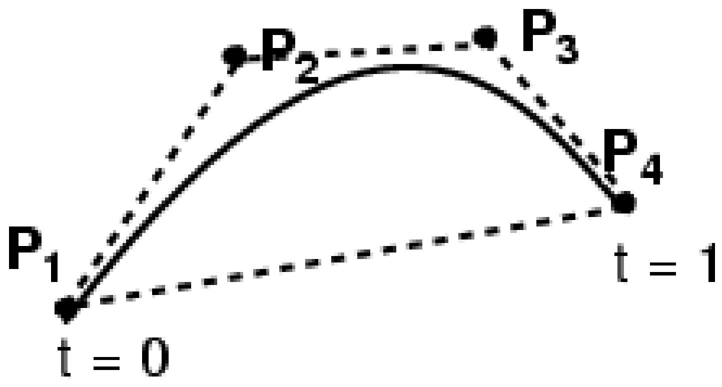
  A convex hull is the smallest convex set that contains a given set

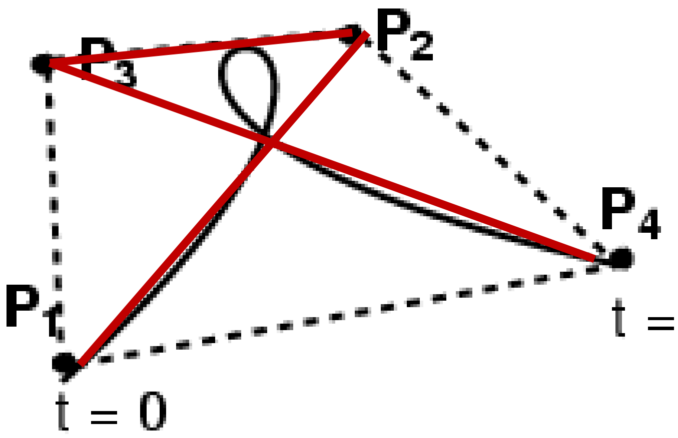
  All points on a Bézier curve lie within the convex hull of the control polygon.



Convex hull



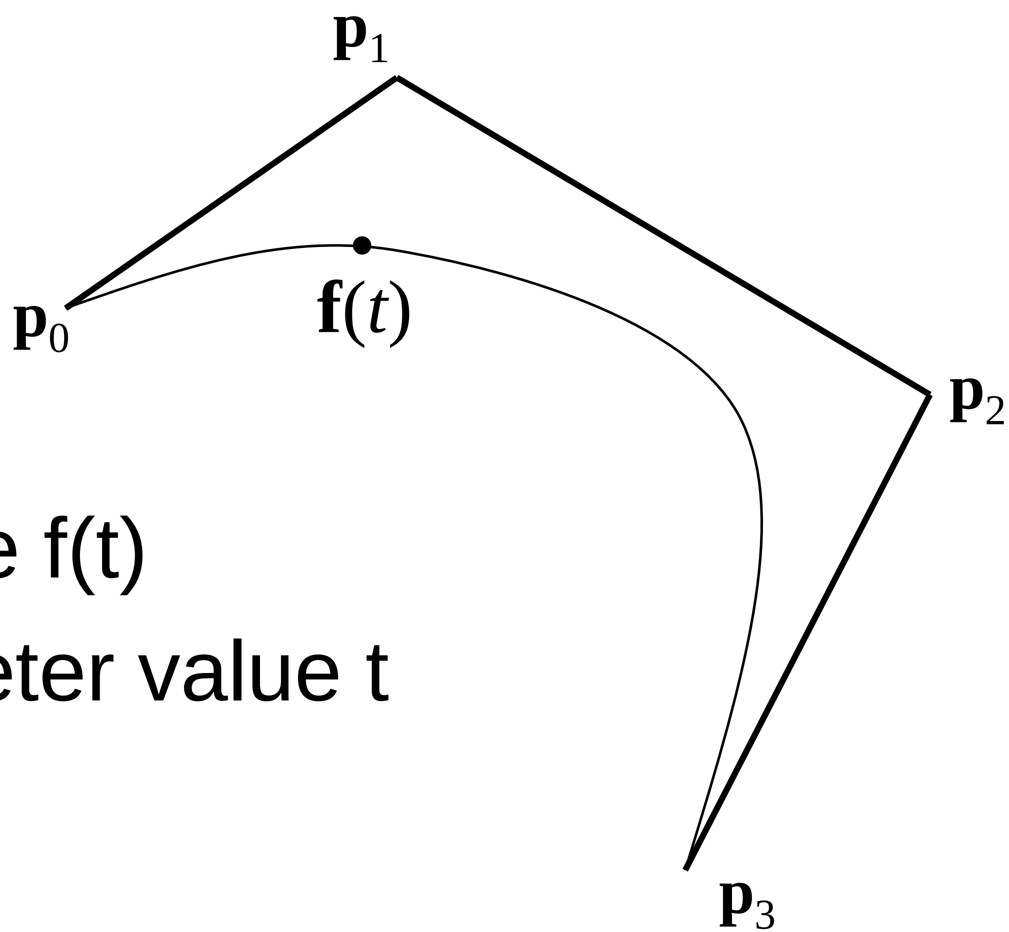




Properties of a Bézier curve

- Linear Precision: when all the control points lie on a line, then the Bézier curve is the segment line interpolating the points. (from convex hull property)

Bézier Curve Evaluation



Compute the value f(t)

for a given parameter value t

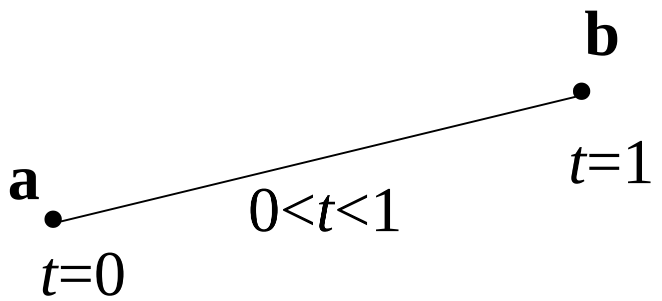
de Casteljau's Algorithm

- Given a parameter value t, evaluate the poly value f(t) by geometric construction
- Apply the algorithm to each curve component (x(t),y(t),z(t))

- Plot the curve by means of a sequence of recursive linear interpolations

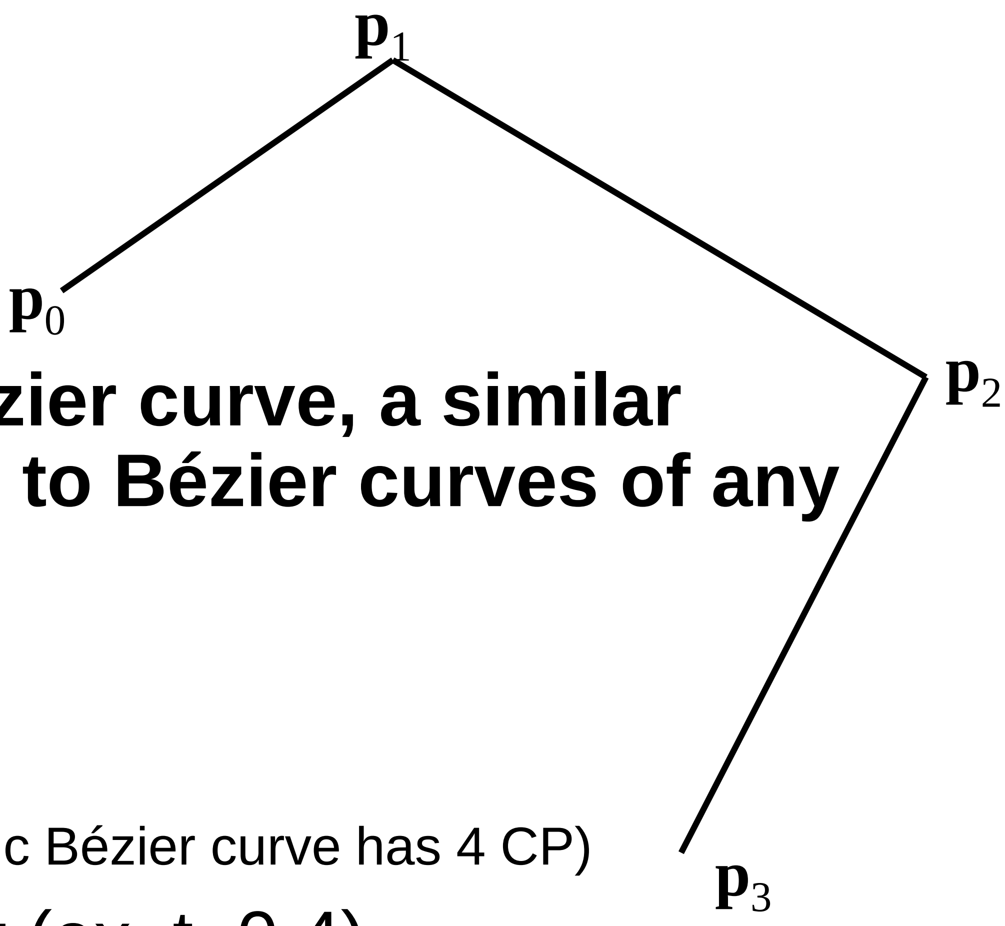
Linear Interpolation (Lerp)

- Linear interpolation (Lerp) compute a value inbetween two values
- A value could be a scalar number, a vector, a color,..
- The linear interpolant between points **a** and **b** with parameter t is given by



$$Lerp(t, \mathbf{a}, \mathbf{b}) = (1-t)\mathbf{a} + t\mathbf{b}$$

de Casteljau's Algorithm



Given t evaluate C(t)

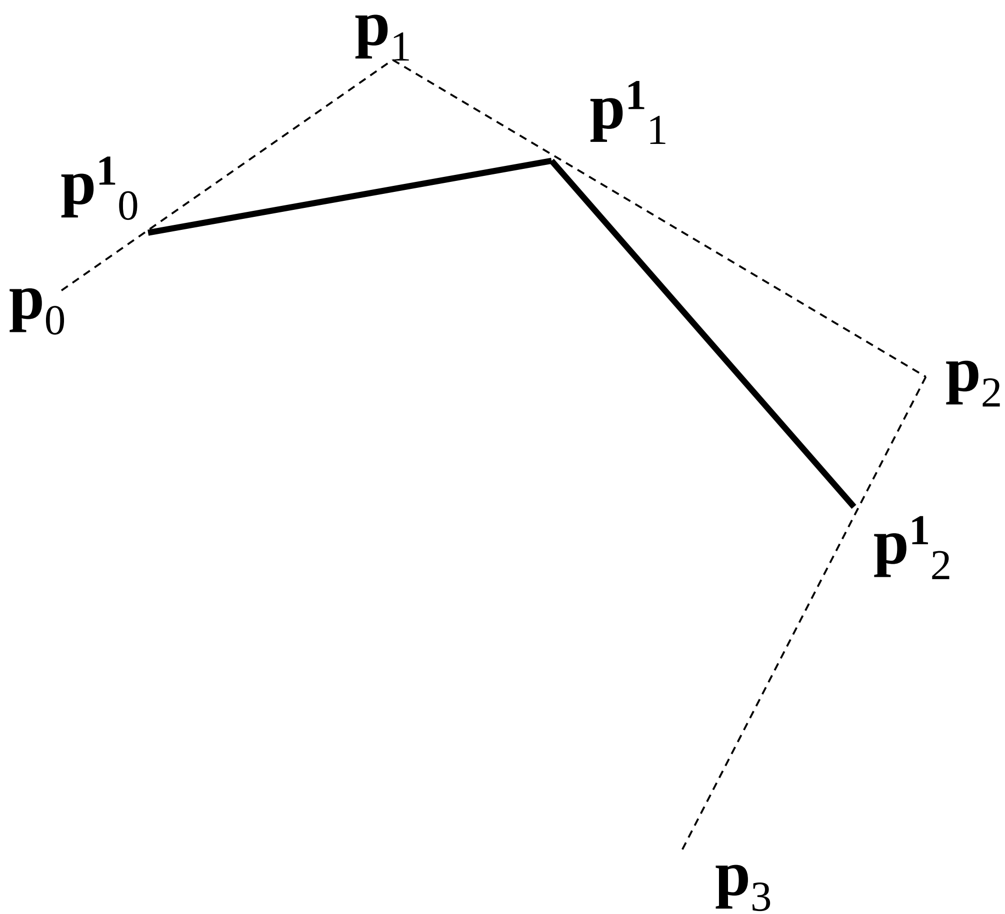
Consider a cubic Bézier curve, a similar construction applies to Bézier curves of any degree)

- Let's start with
  - Control points (A cubic Bézier curve has 4 CP)
  - A parameter value t (ex. t=0.4)

de Casteljau's Algorithm

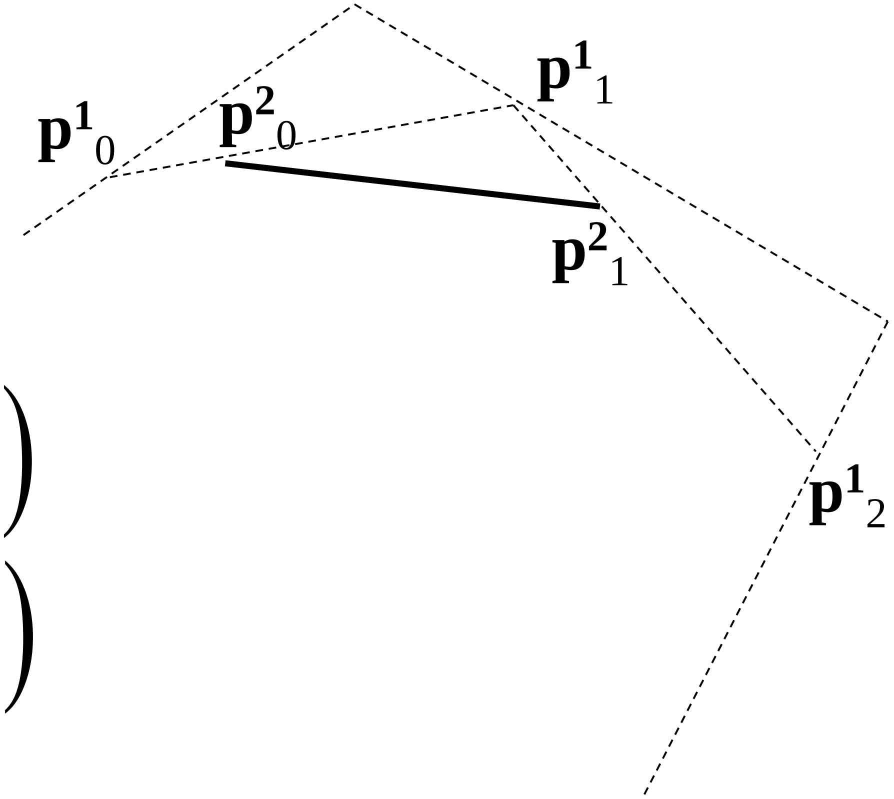
Step 1

$$p^1{}_0 = Lerp\left(t, \mathbf{p}_0, \mathbf{p}_1\right)$$
$$p^1{}_1 = Lerp\left(t, \mathbf{p}_1, \mathbf{p}_2\right)$$
$$p^1{}_2 = Lerp\left(t, \mathbf{p}_2, \mathbf{p}_3\right)$$



de Casteljau's Algorithm

Step 2



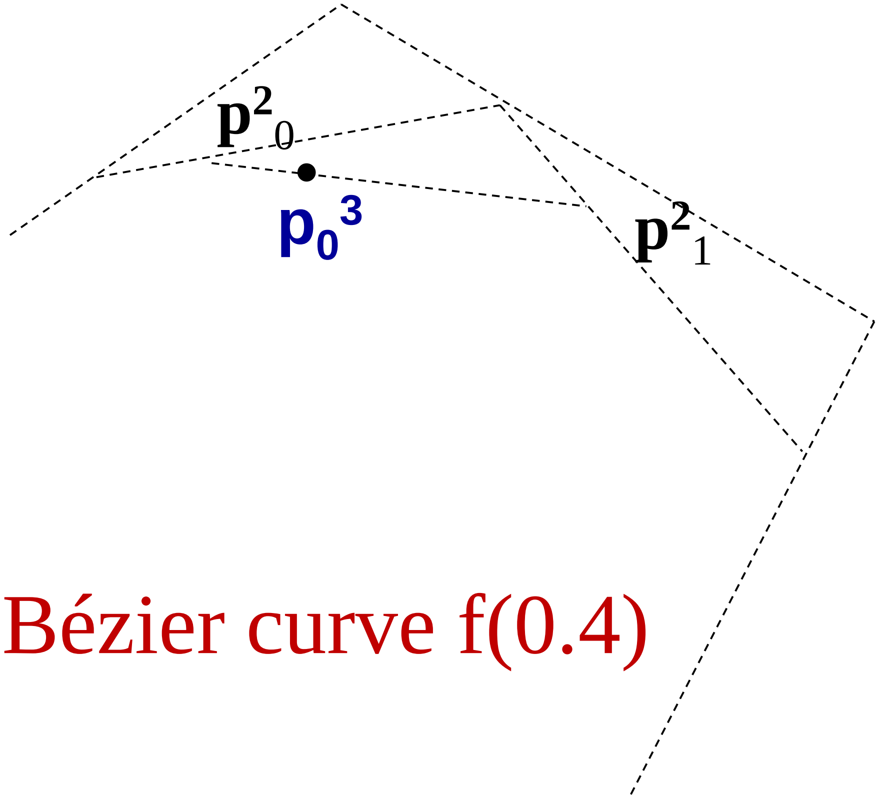
$$p_0^2 = Lerp\left(t, p_0^1, p_1^1\right)$$

$$p_1^2 = Lerp\left(t, p_1^1, p_2^1\right)$$

de Casteljau's Algorithm

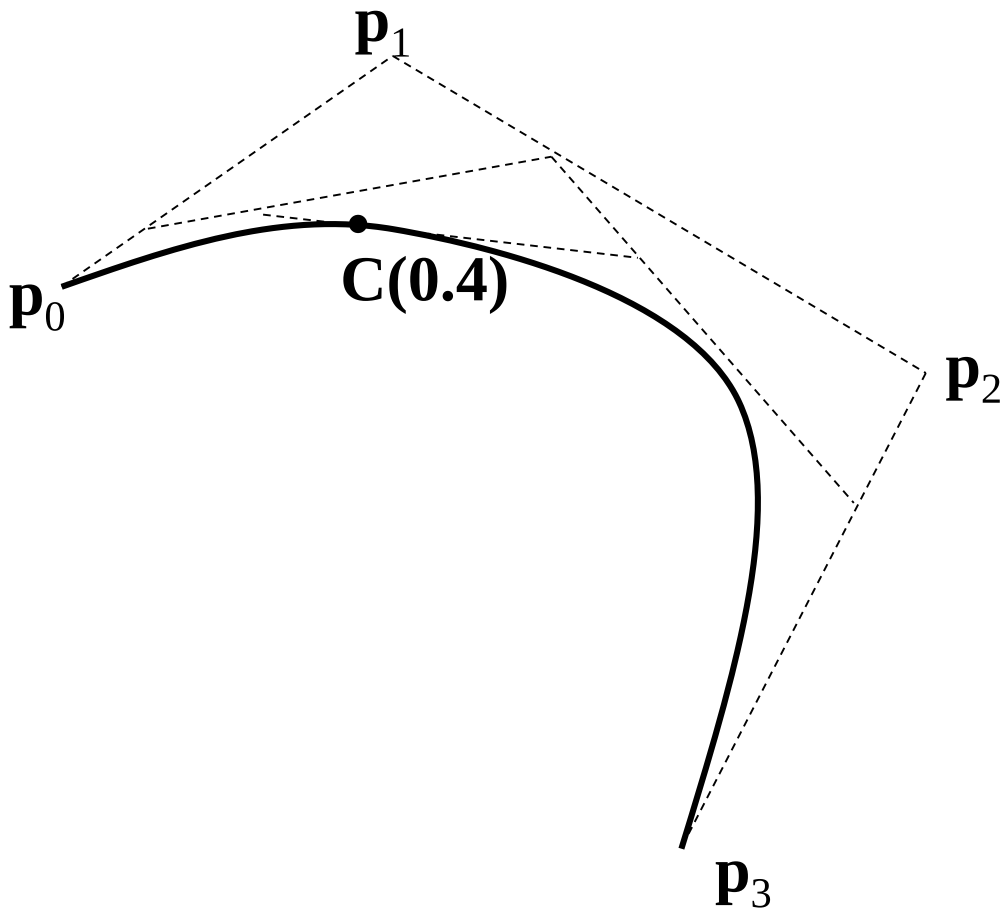
Step 3

$$p_0^3 = Lerp\left(t, p_0^2, p_1^2\right)$$



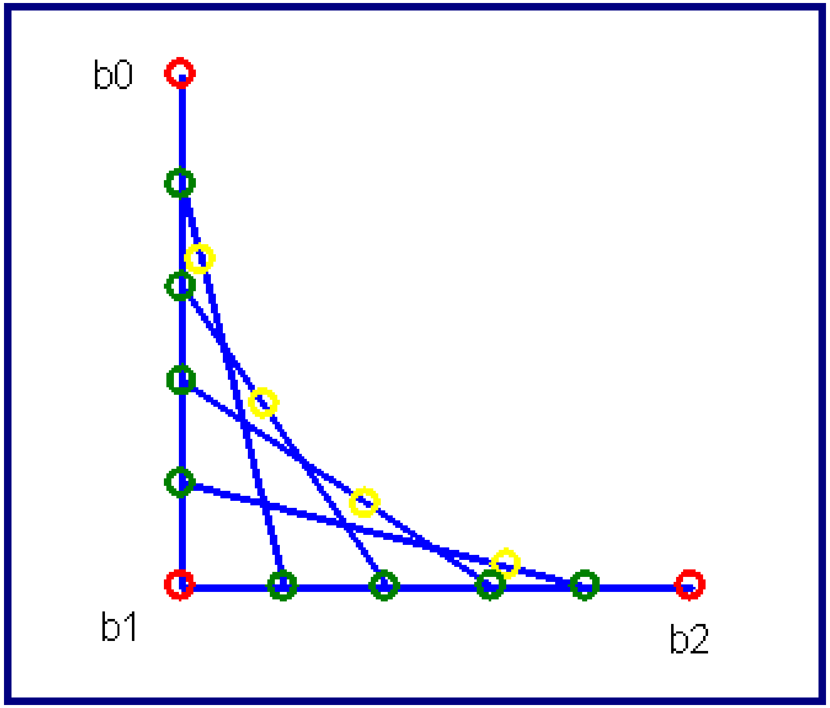
$p_0{}^3$ is the value of the Bézier curve f(0.4)

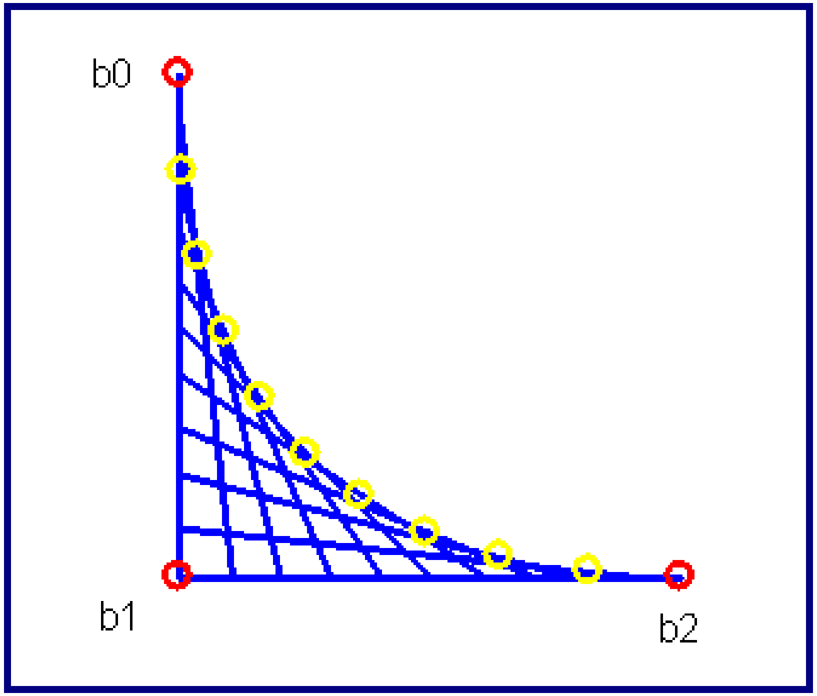
Bézier Curve



de Casteljau's Algorithm

Repeating this process for other values of t in [0,1] generates a sequence of points that define the curve.





A similar construction applies to Bézier curve of degree n, by simply applying n-1 recursive Lerp steps.

de Casteljau's Algorithm

$$input: \quad p_i \;, t$$

n degree of the curve
p_i control points i=0,..,n
t evaluation parameter

$$p_i^0(t) = p_i$$
$$for \quad i = 1,..,n$$
$$for \quad j = 0,..,n-i$$
$$p_j^i(t) = (1-t)\,p_j^{i-1}(t) + t p_{j+1}^{i-1}(t)$$
$$end$$
$$end$$

$$f(t) = p_0^n(t)$$ **point on the curve**

Recursive linear interpolation

$$p_0^3 = Lerp\left(t, p^2{}_0, p^2{}_1\right) \quad \begin{aligned} p_0^2 &= Lerp\left(t, p_0^1, p_1^1\right) \\ p_1^2 &= Lerp\left(t, p_1^1, p_2^1\right) \end{aligned} \quad \begin{aligned} p_0^1 &= Lerp\left(t, \mathbf{p}_0, \mathbf{p}_1\right) \\ p_1^1 &= Lerp\left(t, \mathbf{p}_1, \mathbf{p}_2\right) \\ p_2^1 &= Lerp\left(t, \mathbf{p}_2, \mathbf{p}_3\right) \end{aligned} \quad \begin{matrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{matrix}$$

To prove it, let's expand the Lerp

$$p_0^1 = Lerp\left(t, \mathbf{p}_0, \mathbf{p}_1\right) = \left(1-t\right)\mathbf{p}_0 + t\mathbf{p}_1$$
$$p_1^1 = Lerp\left(t, \mathbf{p}_1, \mathbf{p}_2\right) = \left(1-t\right)\mathbf{p}_1 + t\mathbf{p}_2$$
$$p_2^1 = Lerp\left(t, \mathbf{p}_2, \mathbf{p}_3\right) = \left(1-t\right)\mathbf{p}_2 + t\mathbf{p}_3$$

$$p_0^2 = Lerp\left(t, p_0^1, p_1^1\right) = \left(1-t\right)\left(\left(1-t\right)\mathbf{p}_0 + t\mathbf{p}_1\right) + t\left(\left(1-t\right)\mathbf{p}_1 + t\mathbf{p}_2\right)$$
$$p_1^2 = Lerp\left(t, p_1^1, p_2^1\right) = \left(1-t\right)\left(\left(1-t\right)\mathbf{p}_1 + t\mathbf{p}_2\right) + t\left(\left(1-t\right)\mathbf{p}_2 + t\mathbf{p}_3\right)$$

$$\begin{aligned} p_0^3 = Lerp\left(t, p_0^2, p_1^2\right) &= \left(1-t\right)\left(\left(1-t\right)\left(\left(1-t\right)\mathbf{p}_0 + t\mathbf{p}_1\right) + t\left(\left(1-t\right)\mathbf{p}_1 + t\mathbf{p}_2\right)\right) \\ &\quad + t\left(\left(1-t\right)\left(\left(1-t\right)\mathbf{p}_1 + t\mathbf{p}_2\right) + t\left(\left(1-t\right)\mathbf{p}_2 + t\mathbf{p}_3\right)\right) \end{aligned}$$

This reduces to a Bézier curve in Bernstein basis form

$$f(t)=(1-t)\Big((1-t)\big((1-t)\mathbf{p}_0+t\mathbf{p}_1\big)+t\big((1-t)\mathbf{p}_1+t\mathbf{p}_2\big)\Big)$$
$$+t\Big((1-t)\big((1-t)\mathbf{p}_1+t\mathbf{p}_2\big)+t\big((1-t)\mathbf{p}_2+t\mathbf{p}_3\big)\Big)$$

$$f(t)=(1-t)^3\,\mathbf{p}_0+3(1-t)^2\,t\mathbf{p}_1+3(1-t)t^2\mathbf{p}_2+t^3\mathbf{p}_3$$

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

Bézier curve in Bernstein basis form

$$f(t) = \left(-t^3 + 3t^2 - 3t + 1\right)\mathbf{p}_0 + \left(3t^3 - 6t^2 + 3t\right)\mathbf{p}_1 + \left(-3t^3 + 3t^2\right)\mathbf{p}_2 + \left(t^3\right)\mathbf{p}_3$$

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

$$B_0^3(t) = -t^3 + 3t^2 - 3t + 1$$
$$B_1^3(t) = 3t^3 - 6t^2 + 3t$$
$$B_2^3(t) = -3t^3 + 3t^2$$
$$B_3^3(t) = t^3$$

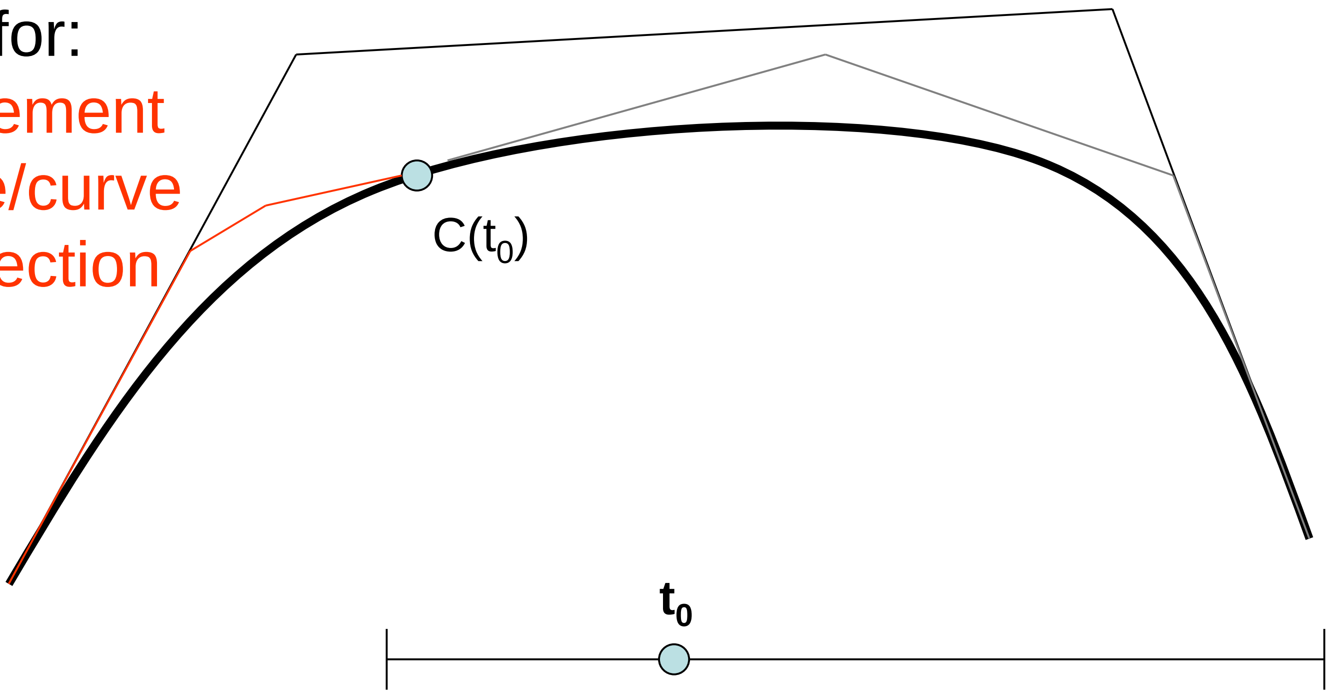
Curve Subdivision

It's the process of splitting a single Bézier curve of degree n into two subcurves of the degree n. de Casteljau's algorithm is used to perform the splitting.

Good for:

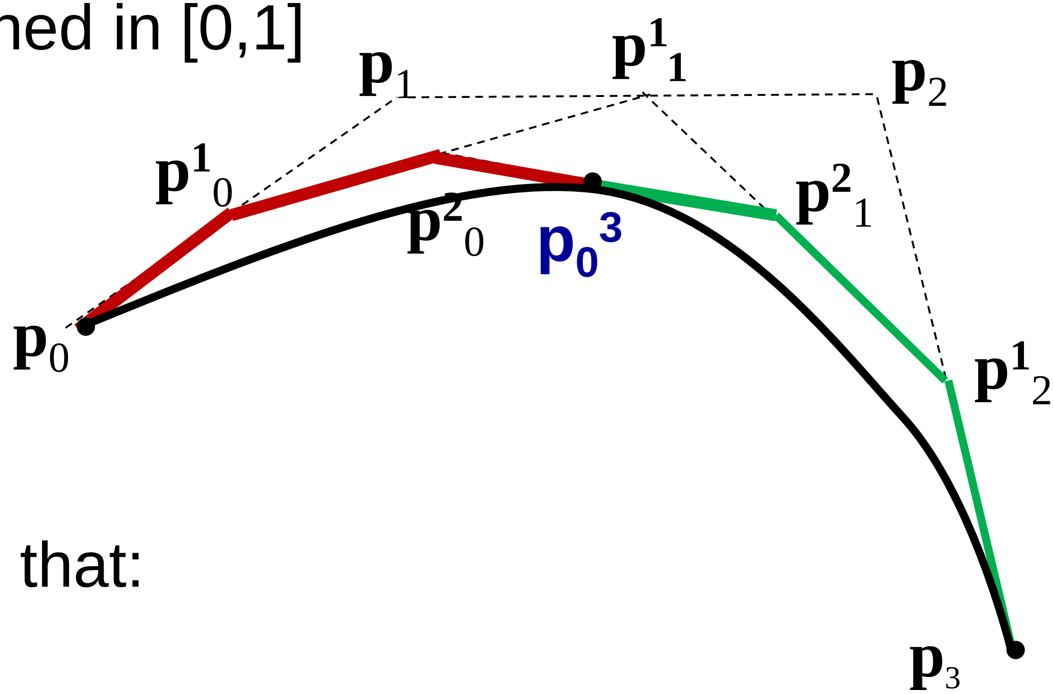
-Refinement

-Curve/curve intersection



Subdivide a Bézier curve at t by applying de Casteljau's method

Subdivide the curve C(t) defined in [0,1] at $t=t_0$, by applying the de Casteljau's algorithm.



We get the two subcurves:

C_1 in $[0,t_0]$ and
C_2 in $[t_0,1]$ such that:

$C_1(t)=C(t*t_0)$
$C_2(t)=C(t_0+(1-t_0)t)$

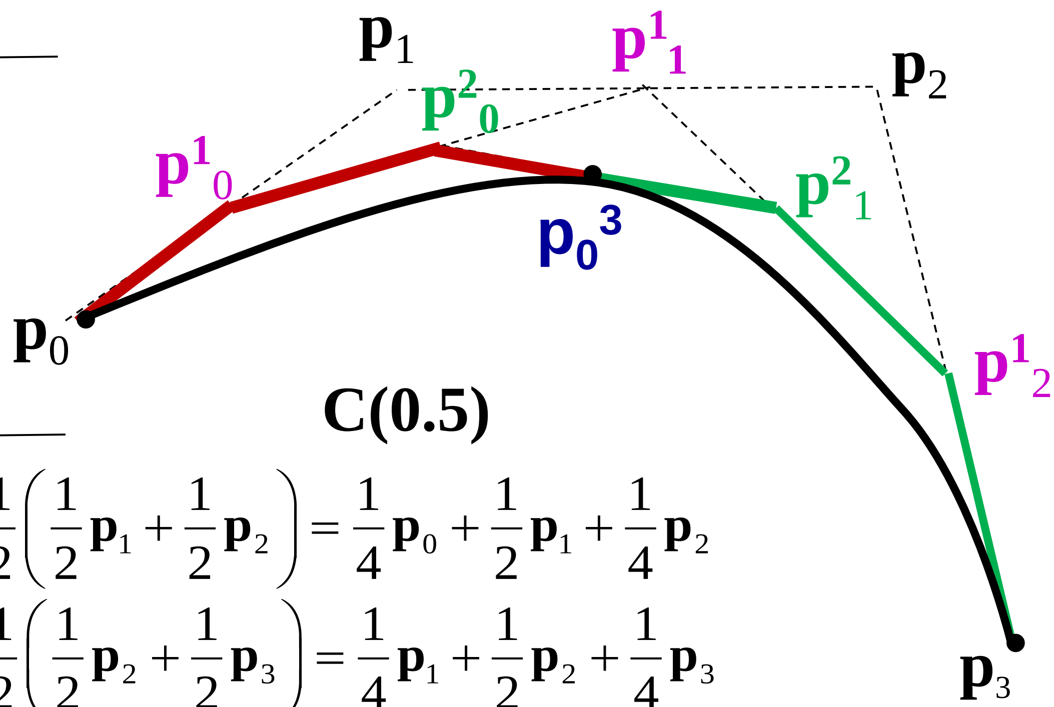
Ex. Subdivide a cubic curve at t=0.5 by applying de Casteljau's method

$$CP = \{\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$$

$$p_0^1 = Lerp(t, \mathbf{p}_0, \mathbf{p}_1) = \frac{1}{2}\mathbf{p}_0 + \frac{1}{2}\mathbf{p}_1$$

$$p_1^1 = Lerp(t, \mathbf{p}_1, \mathbf{p}_2) = \frac{1}{2}\mathbf{p}_1 + \frac{1}{2}\mathbf{p}_2$$

$$p_2^1 = Lerp(t, \mathbf{p}_2, \mathbf{p}_3) = \frac{1}{2}\mathbf{p}_2 + \frac{1}{2}\mathbf{p}_3$$



$$p_0^2 = Lerp(t, p_0^1, p_1^1) = \frac{1}{2}\left(\frac{1}{2}\mathbf{p}_0 + \frac{1}{2}\mathbf{p}_1\right) + \frac{1}{2}\left(\frac{1}{2}\mathbf{p}_1 + \frac{1}{2}\mathbf{p}_2\right) = \frac{1}{4}\mathbf{p}_0 + \frac{1}{2}\mathbf{p}_1 + \frac{1}{4}\mathbf{p}_2$$

$$p_1^2 = Lerp(t, p_1^1, p_2^1) = \frac{1}{2}\left(\frac{1}{2}\mathbf{p}_1 + \frac{1}{2}\mathbf{p}_2\right) + \frac{1}{2}\left(\frac{1}{2}\mathbf{p}_2 + \frac{1}{2}\mathbf{p}_3\right) = \frac{1}{4}\mathbf{p}_1 + \frac{1}{2}\mathbf{p}_2 + \frac{1}{4}\mathbf{p}_3$$

$$p_0^3 = Lerp(t, p_0^2, p_1^2) = \frac{1}{2}\left(\frac{1}{2}\left(\frac{1}{2}\mathbf{p}_0 + \frac{1}{2}\mathbf{p}_1\right) + \frac{1}{2}\left(\frac{1}{2}\mathbf{p}_1 + \frac{1}{2}\mathbf{p}_2\right)\right)$$
$$+ \frac{1}{2}\left(\frac{1}{2}\left(\frac{1}{2}\mathbf{p}_1 + \frac{1}{2}\mathbf{p}_2\right) + \frac{1}{2}\left(\frac{1}{2}\mathbf{p}_2 + \frac{1}{2}\mathbf{p}_3\right)\right) = \frac{1}{8}\mathbf{p}_0 + \frac{3}{8}\mathbf{p}_1 + \frac{3}{8}\mathbf{p}_2 \frac{1}{8}\mathbf{p}_3$$

Subdivision: matrix form

$$C_1(t) = \begin{bmatrix} 1 & \frac{t}{2} & \left(\frac{t}{2}\right)^2 & \left(\frac{t}{2}\right)^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

Curve in [0,1/2]

C(t)=C(t/2)

$$= \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 1/8 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} SM \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

Subdivision: matrix form

The subcurve is represented in Bernstein basis form [0,1/2]:

$$C_1(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} M \boxed{M^{-1} S M} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

$$S_{[0,1/2]} = M^{-1} S \; M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1/4 & 1/2 & 1/4 & 0 \\ 1/8 & 3/8 & 3/8 & 1/8 \end{bmatrix}$$

T · M =B

$$C_1(t) = TMS_{[0,1/2]}P = BS_{[0,1/2]}P$$

Subdivision: matrix form

$$C_1(t) = B\, S_{[0,1/2]} P = B\, P_{new}$$

$S_{[0,1/2]}$ is a subdivision matrix applied to the original CP to produce the CP of the new subcurve

$$P_{new} = S_{[0,1/2]} P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1/4 & 1/2 & 1/4 & 0 \\ 1/8 & 3/8 & 3/8 & 1/8 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} P_0 \\ \frac{1}{2}P_0 + \frac{1}{2}P_1 \\ \frac{1}{4}P_0 + \frac{1}{2}P_1 + \frac{1}{4}P_2 \\ \frac{1}{8}P_0 + \frac{3}{8}P_1 + \frac{3}{8}P_2 + \frac{1}{8}P_3 \end{bmatrix}$$

In a similar way for the second subcurve in [1/2,1] we get CP of the new subcurve $C_2(t)=C(1/2+(1-1/2)/2)$

Rendering a Bézier curve

Draw a Bézier curve as a series of straight line segments

- Uniform method

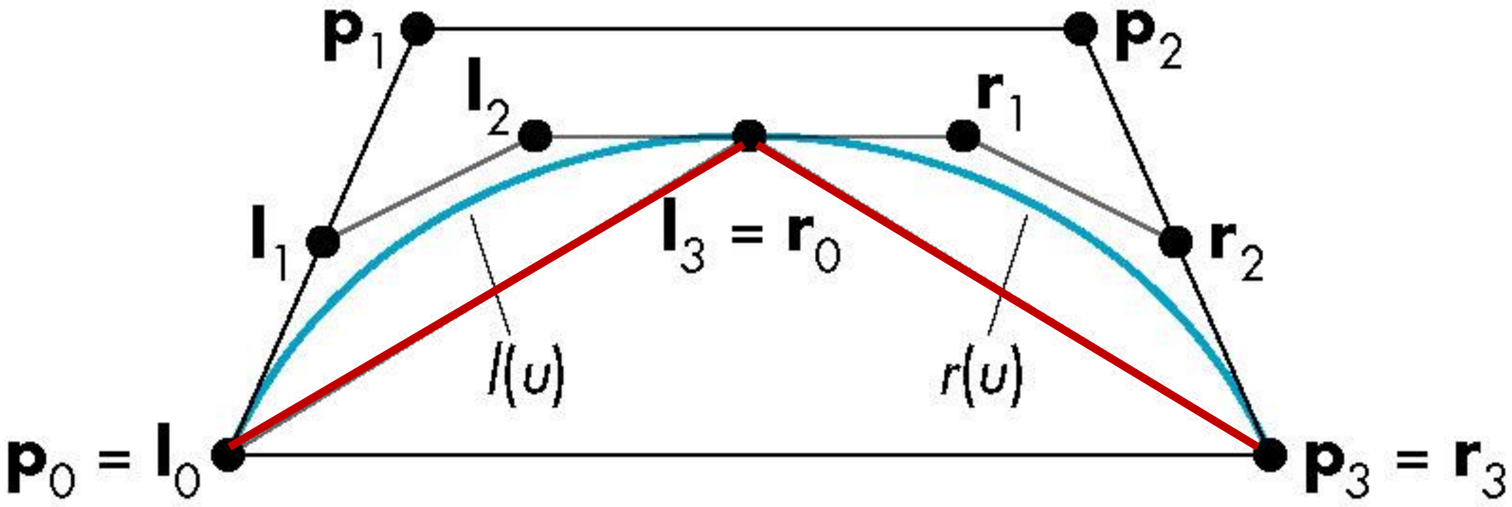
  Discretize the parametric interval into N equidistant points, then plot the polygonal joining corresponding evaluated points on the curve

- Adaptive Subdivision method

Adaptive subdivision: main idea

$l(u)$ with CP=$\{l_0, l_1, l_2, l_3\}$ and $r(u)$ with CP=$\{r_0, r_1, r_2, r_3\}$ have a convex hull that is closer to C(u) than the convex hull of C(u) defined by the CP $\{p_0, p_1, p_2, p_3\}$

The red polyline from l_0 to l_3 ($=r_0$) to r_3 is an approximation to C(u). Repeating recursively we get better approximations.

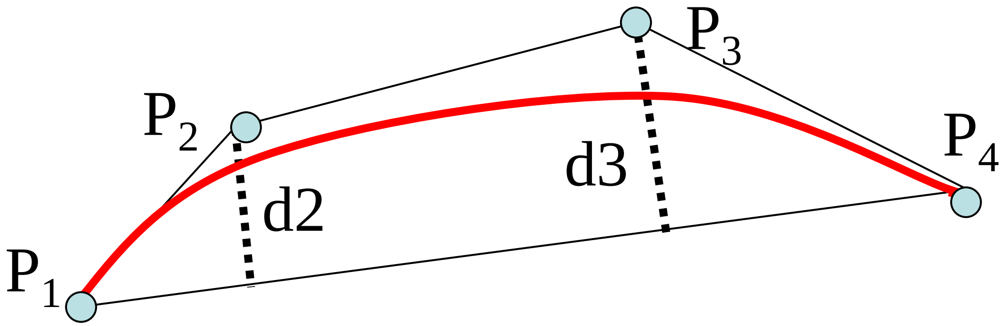


Adaptive Subdivision Method

It's an optimization rendering method based on adaptive subdivisions of the curve: break a curve into smaller an smaller subcurves until each subcurve is sufficiently close to being a straight line, so that rendering the subcurves as straight lines gives adequate results.

Flat test for a subcurve:

For every internal CP compute distance d to the chord



If both d2 and d3 are less of a given tolerance Tol>0, the subcurve is considered flat, and it is approximated by the straight line P_1P_4

Iterate:

Apply FlatTest to C(t)

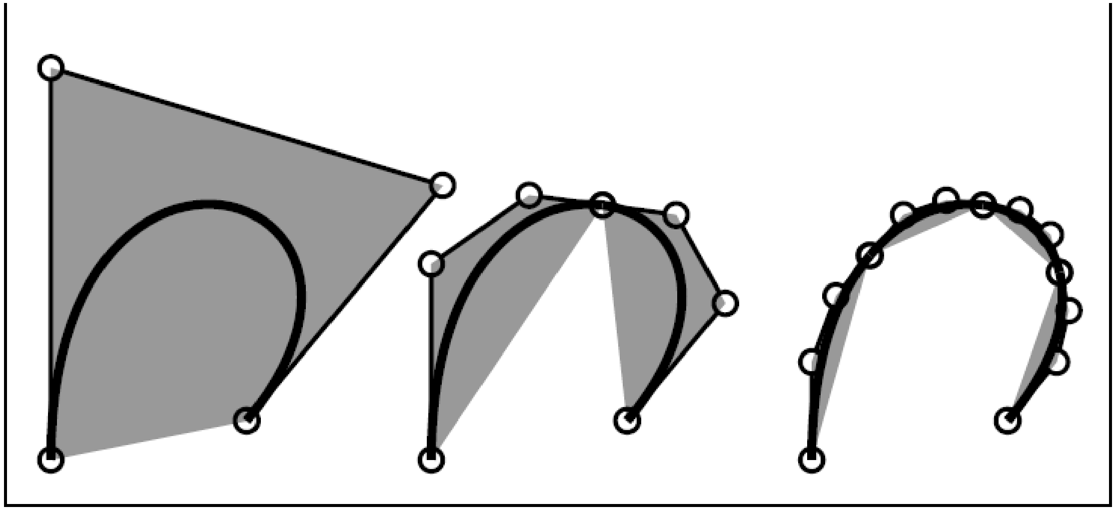
OK: Draw the segment between the curve CPend-points

KO: Subdivide the curve C(t) into 2 halves (t=0.5)

Apply FlatTest to C1(t) and C2(t)

(the two generated subcurves).

Stopping criterium: all the subcurve segments are flat

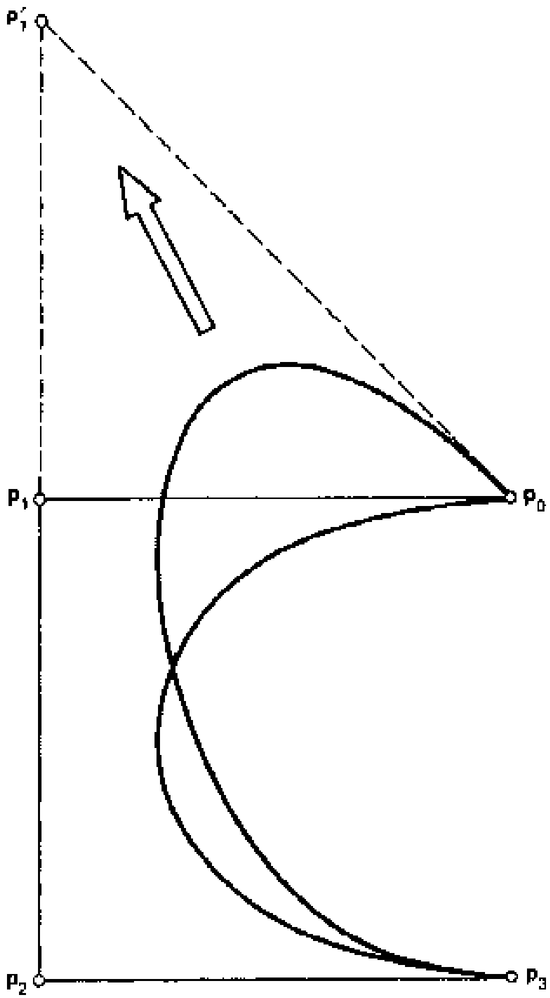


Bézier curves: limits

- Local Control vs. Global Control

- One would the degree of the curve to be independent on the number of control points

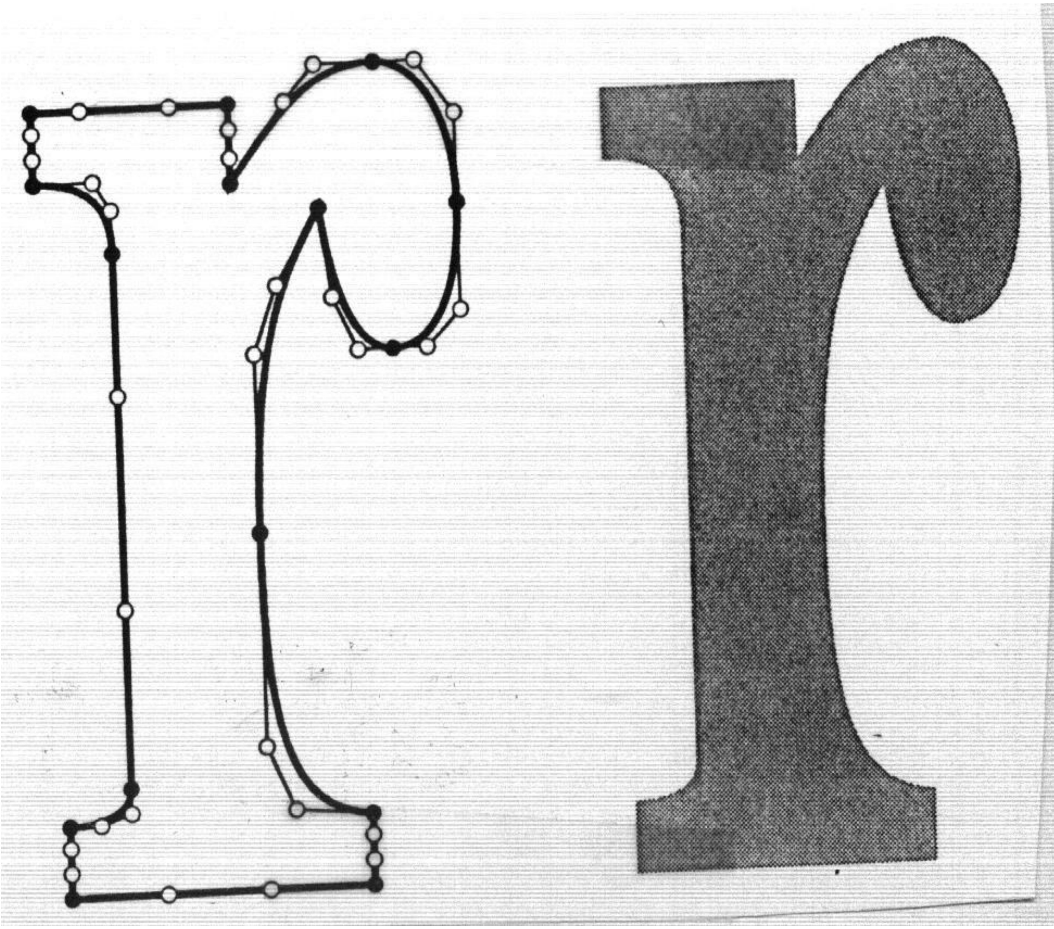
Bézier curves may be joined end-to-end to form a composite curve



Connecting Bézier curves

- How to impose continuity/ discontinuity at the joints?

1) **Piecewise Bézier curves**
   Piecewise polynomial, geometric relationship of the CP adjacent to the joints determines the continuity conditions



2) **Polynomial Spline**
   Piecewise polynomial with given regularity conditions at the joints

Piecewise Bézier curves

Definition:

Let $t_0<t_1<..<t_k$ be a partition of $[t_0,t_k]$. A composite Bézier curve of degree n is a piecewise Bézier curve, where each curve segment $C_i(t)$ on the interval $[t_i,t_{i+1}]$ (i=0,..,k-1) corresponds to a Bézier curve of degree n :

$$\underset{\uparrow}{C(t)} = \underset{\uparrow}{C_i(t)} = \sum_{j=0}^{n} P_{n*i+j} B_j^n\left(\frac{t-t_i}{t_{i+1}-t_i}\right) \qquad t \in [t_i, t_{i+1})$$

Piecewise Bézier

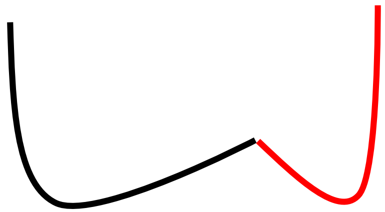
single Bézier

!! Only C^0 continuity is naturally satisfied

Parametric and Geometry Continuity

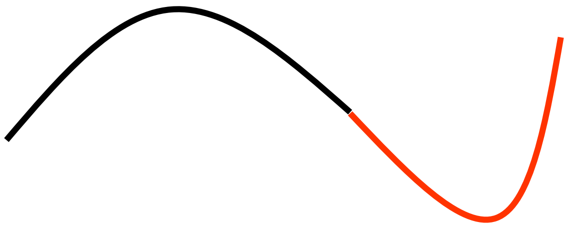
- C^0 = continuous

  The joint can be a sharp kink



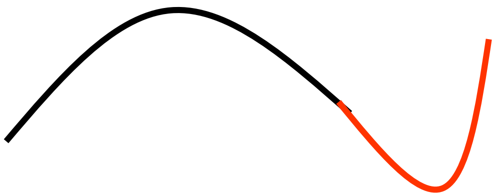
- C^1 = parametric continuity

  Tangents are the same at the joint



- G^1 = geometric continuity

  Tangents have the same directions



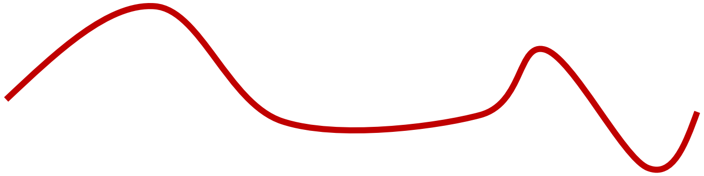
- C^2 = curvature continuity

  Tangents and their derivatives are the same

Parametric Continuity C^1

"A curve is C^0 if it can be drawn without lifting the pen off the paper sheet"

If the derived curve is continuous then the curve is also C^1



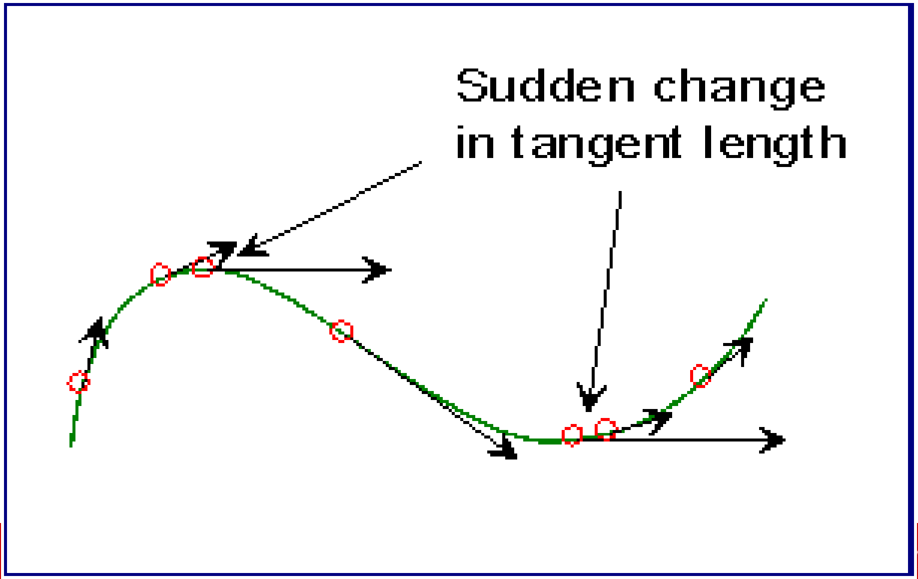
The parametric continuity ensures that only the motion of the particle moving along the curve is continuous, i.e. there are no sudden jumps in speed, it does not say that the path (the curve) is smooth.

Example: straight line with velocity jump. Is not C^1, but the curve is certainly smooth

Geometry Continuity G^1

If the direction tangent to a parametric curve varies in a continuous manner then it is continuous G^1

Its magnitude can also have discontinuous jumps, but the curve is still G^1. Then the particle moving along the curve varies its speed to rush but still along a continuous curve if its direction continuously changes.



Geometry continuity between two curves

- **Parametric Continuity C^1**

The <u>directions</u> v1, v2 and <u>modules</u> of the tangent vectors of the two curve segments in the contact point are equal

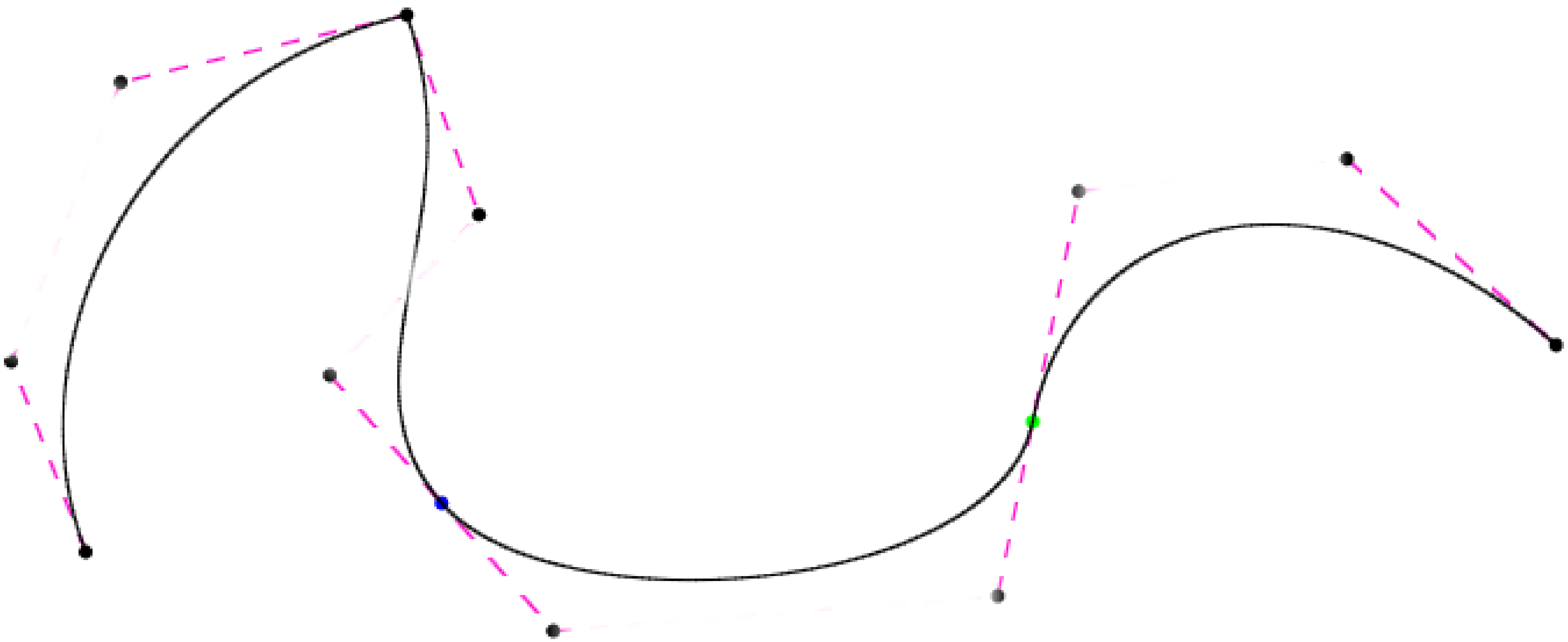
$$v_1 = k \cdot v_2, \quad k > 0$$

- **Geometry Continuity G^1:**

The <u>directions</u> v1, v2 of the tangent vectors of the two curve segments in the contact point are equal

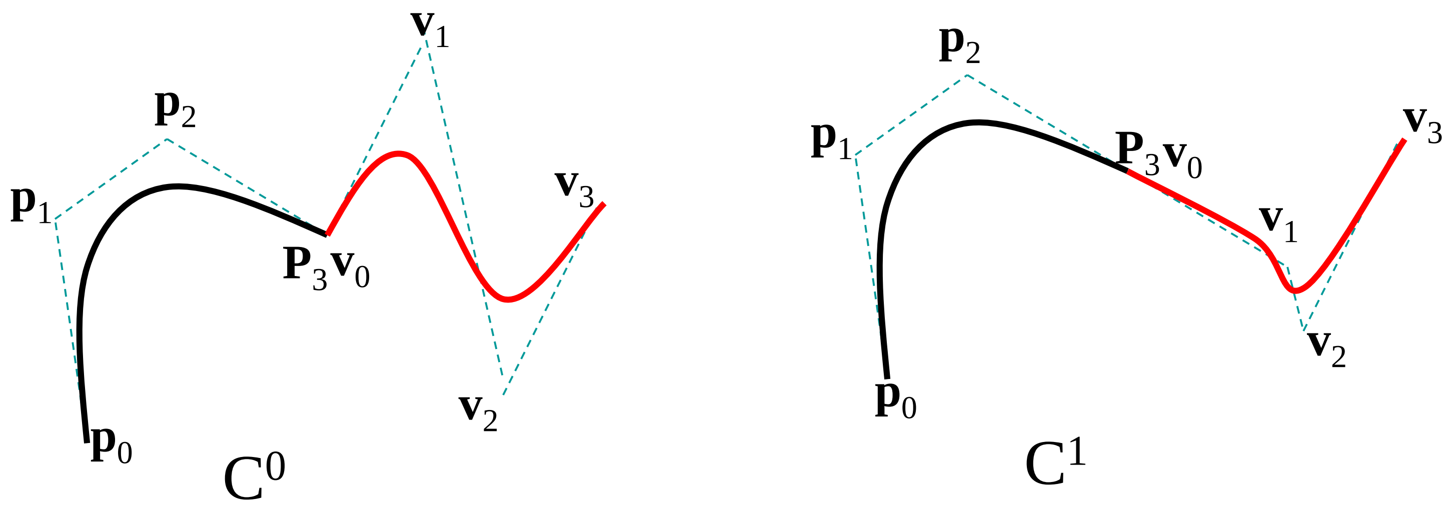
In general C^1 implies G^1, the reverse is not true in general

Example: 4 pieces of curves



Connecting two Bézier curves

- Consider two Bézier curves defined by $\mathbf{p}_0 \ldots \mathbf{p}_3$ and $\mathbf{v}_0 \ldots \mathbf{v}_3$
- If $\mathbf{p}_3 = \mathbf{v}_0$, then they will have C^0 continuity
- If $(\mathbf{p}_3 - \mathbf{p}_2) = (\mathbf{v}_1 - \mathbf{v}_0)$, then they will have C^1 continuity
- C^2 continuity is more difficult…



$C^0=G^0$

Bézier curves with G^1 –CONTINUITY

Two curves join at one end-point and have tangent vectors at that point with same direction.

CPs on the common tangent vectors are collinear.

Less severe than the C^1-Continuity condition.

C^r-Continuity of composed Bézier curves

Theorem

Let $$C_i(t)=\sum_{j=0}^{n} P_{n*i+j} B_j^n\left(\frac{t-t_i}{t_{i+1}-t_i}\right); \quad t\in[t_i,t_{i+1}]$$

$$C_{i+1}(t)=\sum_{j=0}^{n} P_{n*(i+1)+j} B_j^n\left(\frac{t-t_{i+1}}{t_{i+2}-t_{i+1}}\right); \quad t\in[t_{i+1},t_{i+2}]$$

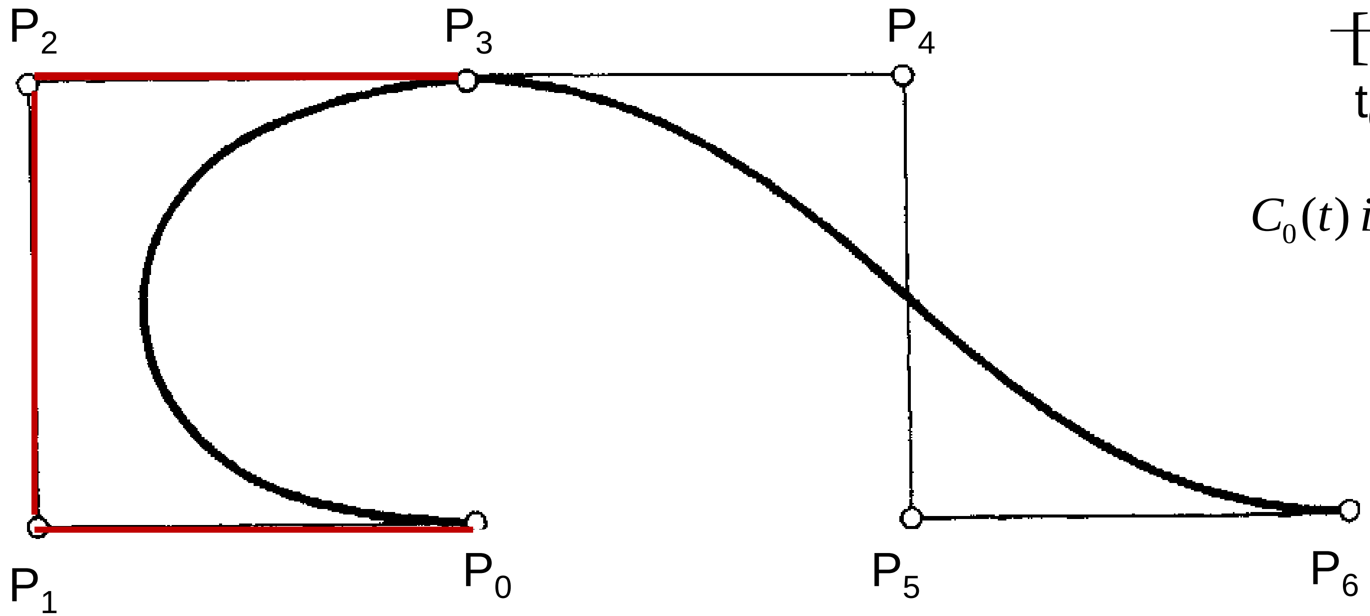
be two adjacent Bézier curve segments.Then the two segments join C^r-continuity at t_{i+1}, if and only if

$$\frac{1}{\delta_i^l}\Delta^l P_{n(i+1)-l}=\frac{1}{\delta_{i+1}^l}\Delta^l P_{n(i+1)}; \qquad l=0,..,r$$

$$\Delta^0 P_i = P_i \quad \Delta P_i = P_{i+1}-P_i, \quad \Delta^l P_i=\Delta^{l-1}P_{i+1}-\Delta^{l-1}P_i$$

$$\delta_i := t_{i+1}-t_i$$

Example r = 2: C^2 –Continuity at t_1



$C_0(t)\ in\ [t_0, t_1]$ $C_1(t)\ in\ [t_1, t_2]$

$$l = 0 \qquad C_0(t_1) = C_1(t_1) = P_3$$

$$l = 1 \qquad \frac{(P_3 - P_2)}{(t_1 - t_0)} = \frac{(P_4 - P_3)}{(t_2 - t_1)}$$

$$l = 2 \qquad \frac{(P_3 - 2P_2 + P_1)}{(t_1 - t_0)^2} = \frac{(P_5 - 2P_4 + P_3)}{(t_2 - t_1)^2}$$

Serena Morigi
Dipartimento di Matematica
serena.morigi@unibo.it
http://www.dm.unibo.it/~morigi