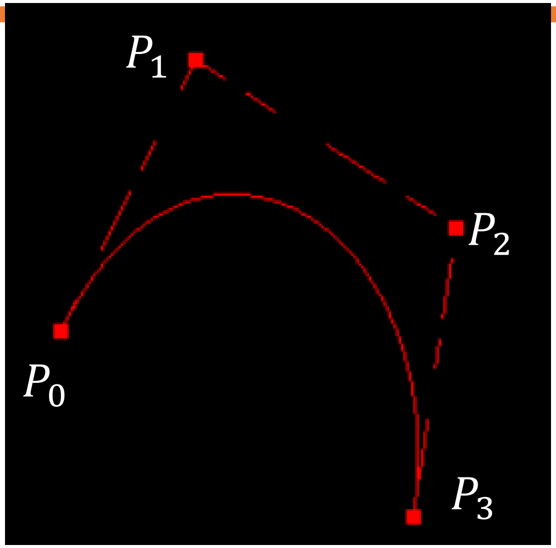
Curve di Bezier

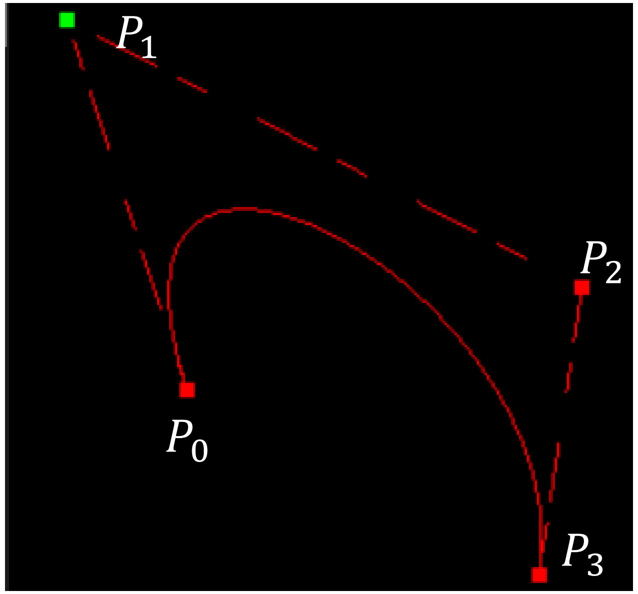
Curve di Bezier

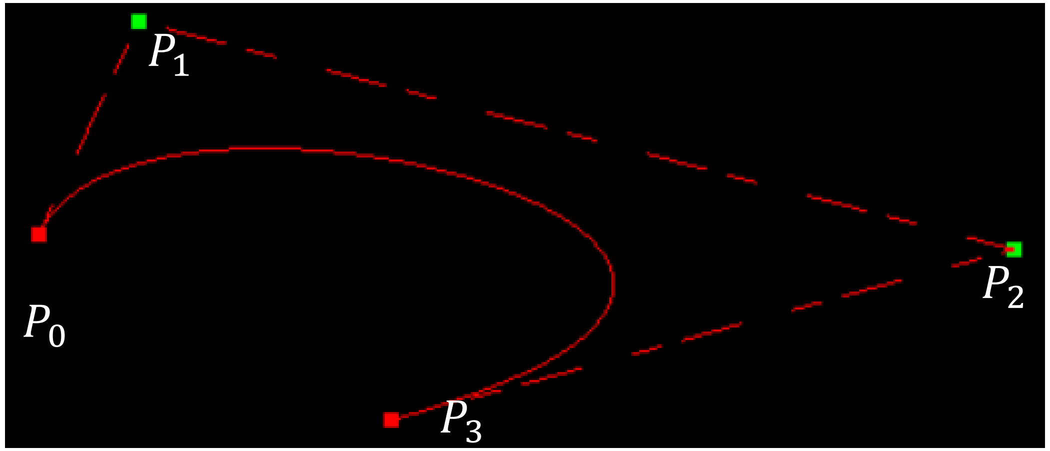
Abbiamo visto che nelle curve interpolanti di Hermite, abbiamo a che fare con segmenti di curva che sono dati dalla combinazione lineare di quattro valori P_1, P_2, D_1, D_2 mediante quattro funzioni base.
P_1, P_2 due assegnano il valore della curva, D_1, D_2 ne modellano la forma.

l'idea che è alla base delle curve di Bezier è di sostituire i valori D_1 e D_2 (che nella curva di Hermite vengono interpolati) con altri due valori puntuali, che non vengono interpolati ma solo approssimati e la cui posizione influenza la forma della curva.

I dati di un segmento di curva cubica di Bezier sono quindi quattro valori puntuali che chiameremo P_0, P_1, P_2, P_3, di cui P_0 e P_3 vengono interpolati, mentre gli altri due vengono solo approssimati. La loro posizione determina il vettore tangente all'inizio ed alla fine del segmento di curva.


P_1
P_2
P_0
P_3


P_1
P_2
P_0
P_3


P_1
P_2
P_0
P_3

Quali sono le funzioni base che permettono di ottenere il segmento di curva cubica di Bezier come combinazione lineare dei quattro valori P_0, P_1, P_2, P_3?

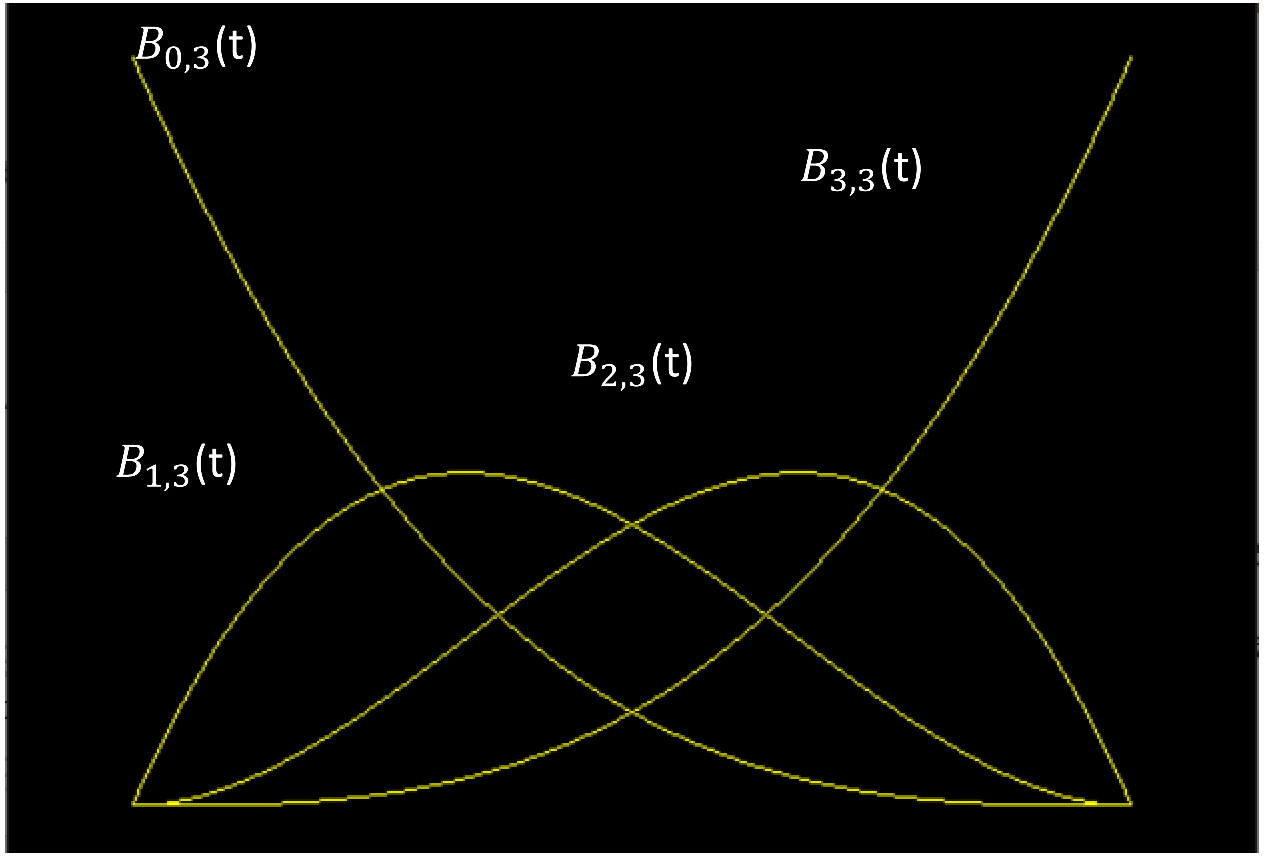
Bezier ha dimostrato che sono i polinomi di Bernstein (di grado 3) cioè:

$$B_{0,3}(t) = (1-t)^3$$

$$B_{1,3}(t) = 3t(1-t)^2$$

$$B_{2,3}(t) = 3t^2(1-t)$$

$$B_{3,3}(t) = t^3$$

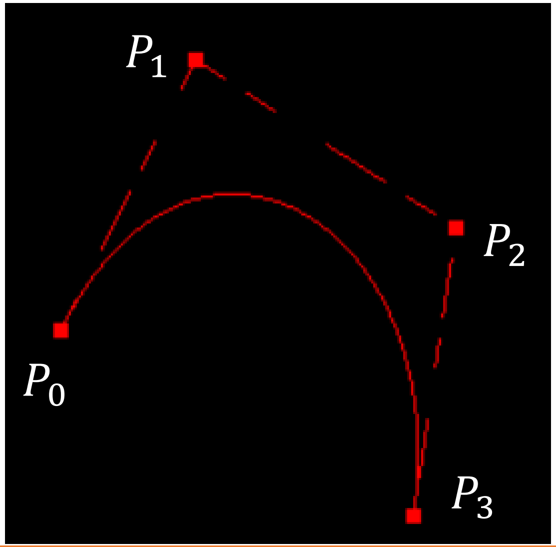


Il segmento di curva cubica di Bèzier che interpola P_0 e P_3 e che in P_0 e P_3 è tangente ai segmenti P_1-P_0 e P_2-P_3 rispettivamente è dato da:

$$C(t) = \sum_{i=0}^{3} P_i B_{i,3}(t) \qquad t \in [0,1]$$

$\in$

I punti P_i sono detti punti di controllo della curva, ed inviduano il poligono di controllo.



Nel caso più generale di curve di Bezier di grado n abbiamo a che fare con $n+1$ punti, di cui il primo e l'ultimo sono interpolati, il secondo e il penultimo danno la direzione della derivata negli estremi, mentre gli altri punti vengono "approssimati" e servono a modellare la forma della curva.

I polinomi di base di Bernstein di grado n definiti nell'intervallo [0,1] sono dati da:

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

dove

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

è il coefficiente binomiale

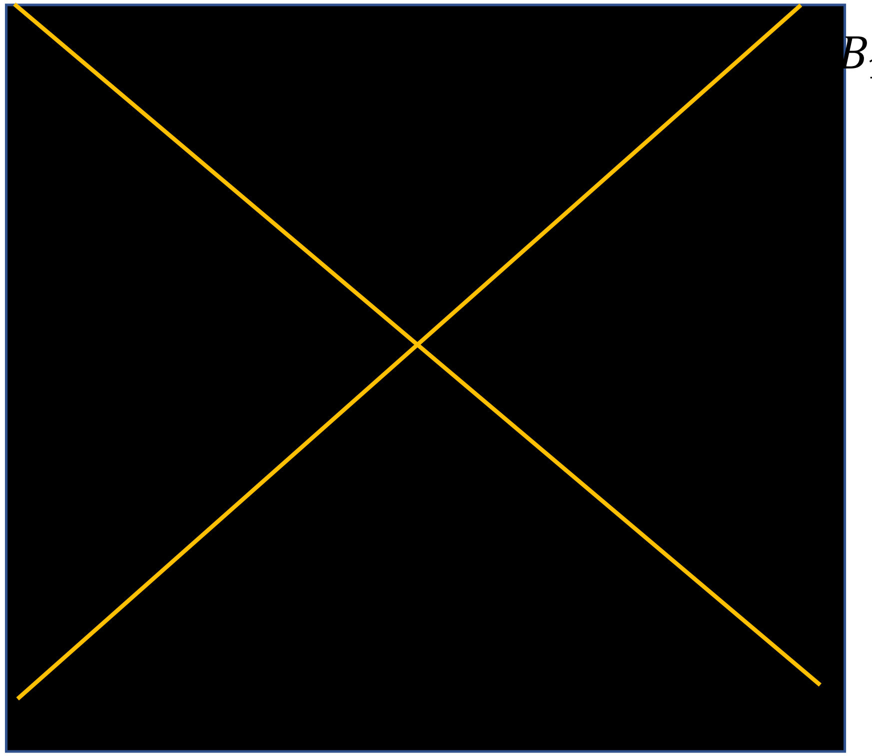
Vertici di controllo P_i , i=0,..,1

Curva di Bezier di grado 1

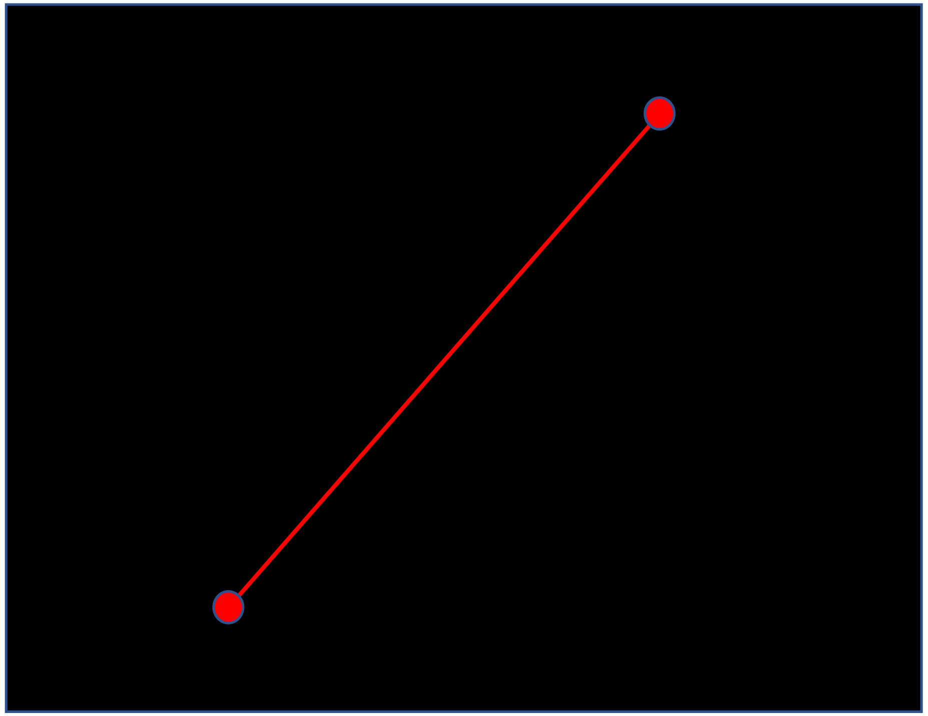
Funzioni base $B_{i,n}$(t), i=0,…n,   n=1

$$C(t) = \sum_{i=0}^{1} P_i B_{i,1}(t)$$

$B_{0,1}(t)$

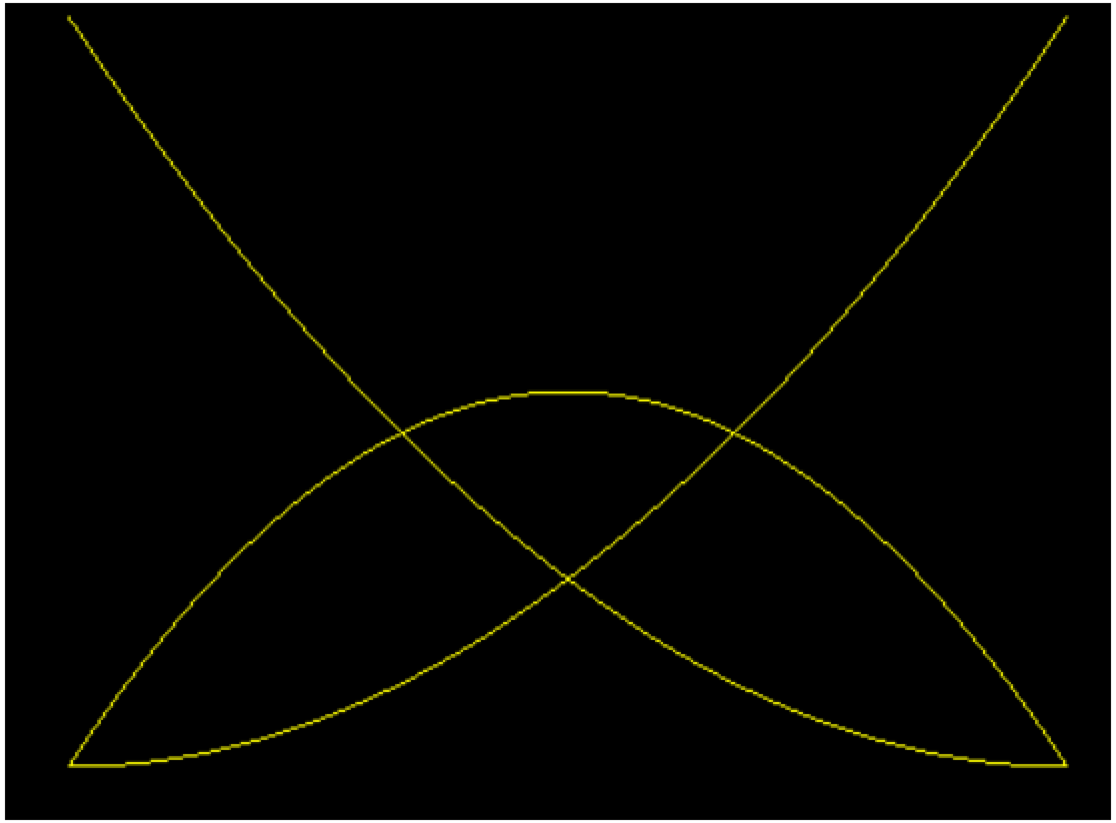


$B_{1,1}(t)$



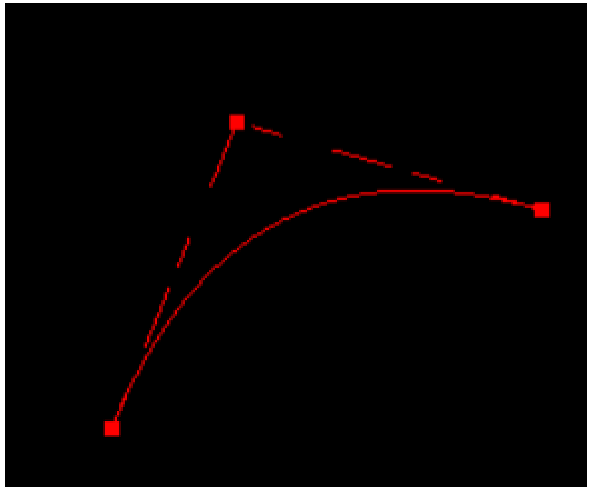
Vertici di controllo P_i , i=0,..,2

Funzioni base $B_{i,n}(t)$, i=0,…n, n=2



Curva di Bezier di grado 2

$$C(t) = \sum_{i=0}^{2} P_i B_{i,2}(t)$$

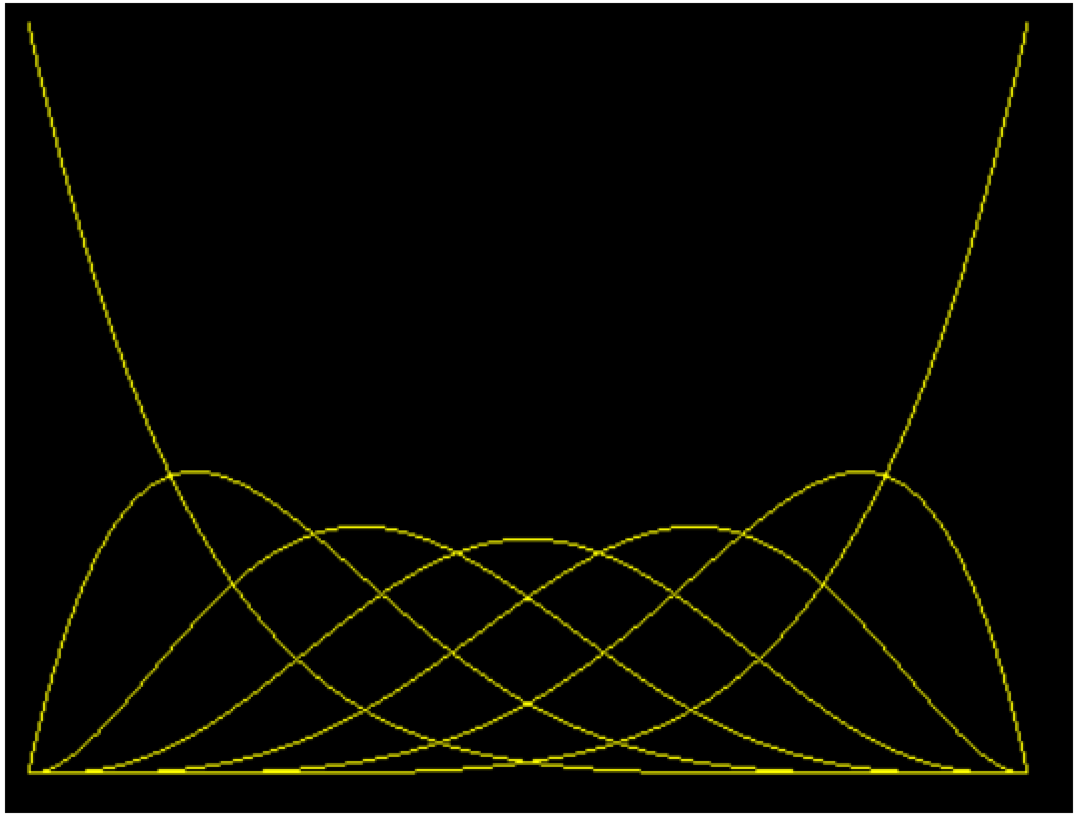


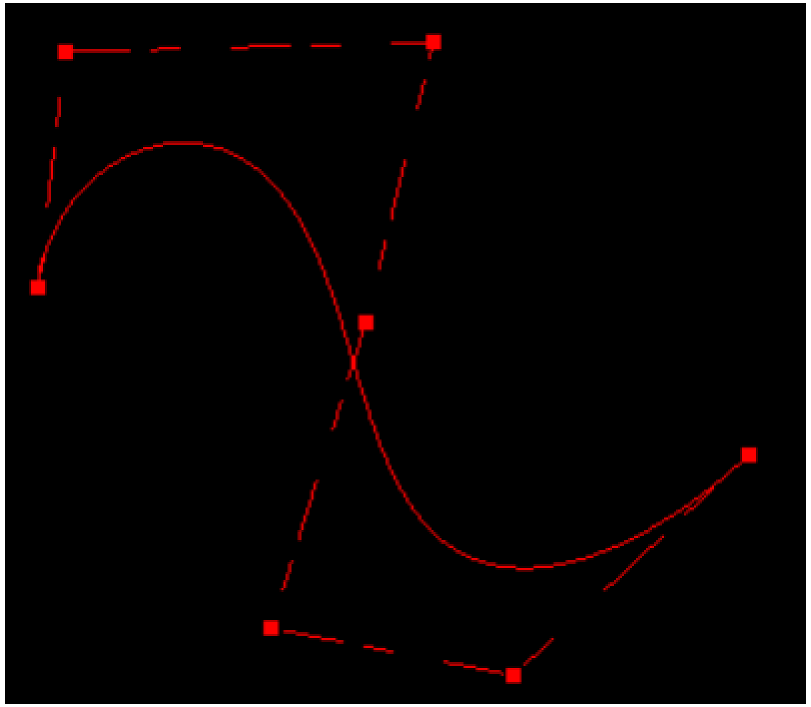
Vertici di controllo P_i , i=0,..,6

Curva di Bezier di grado 6

Funzioni base $B_{i,n}$(t), i=0,…n,   n=6

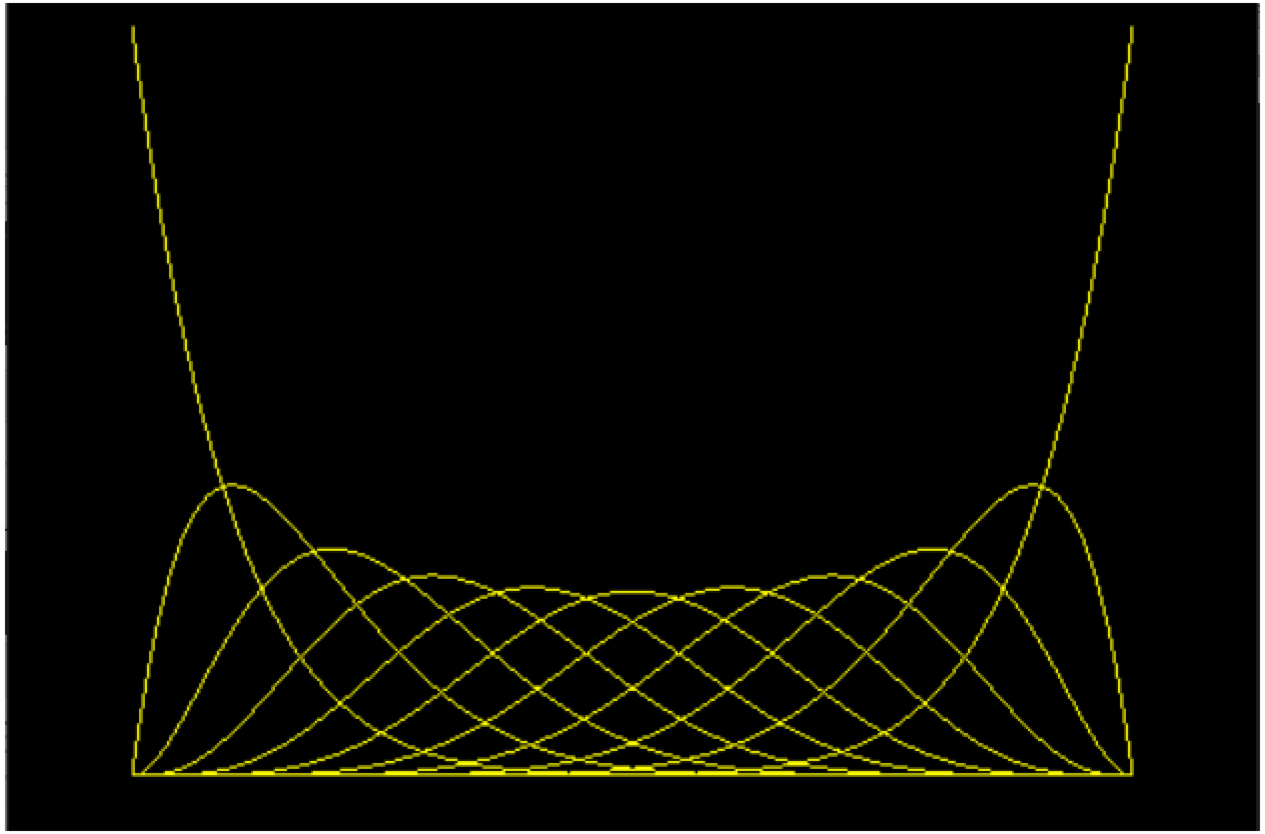
$$C(t) = \sum_{i=0}^{6} P_i B_{i,6}(t)$$





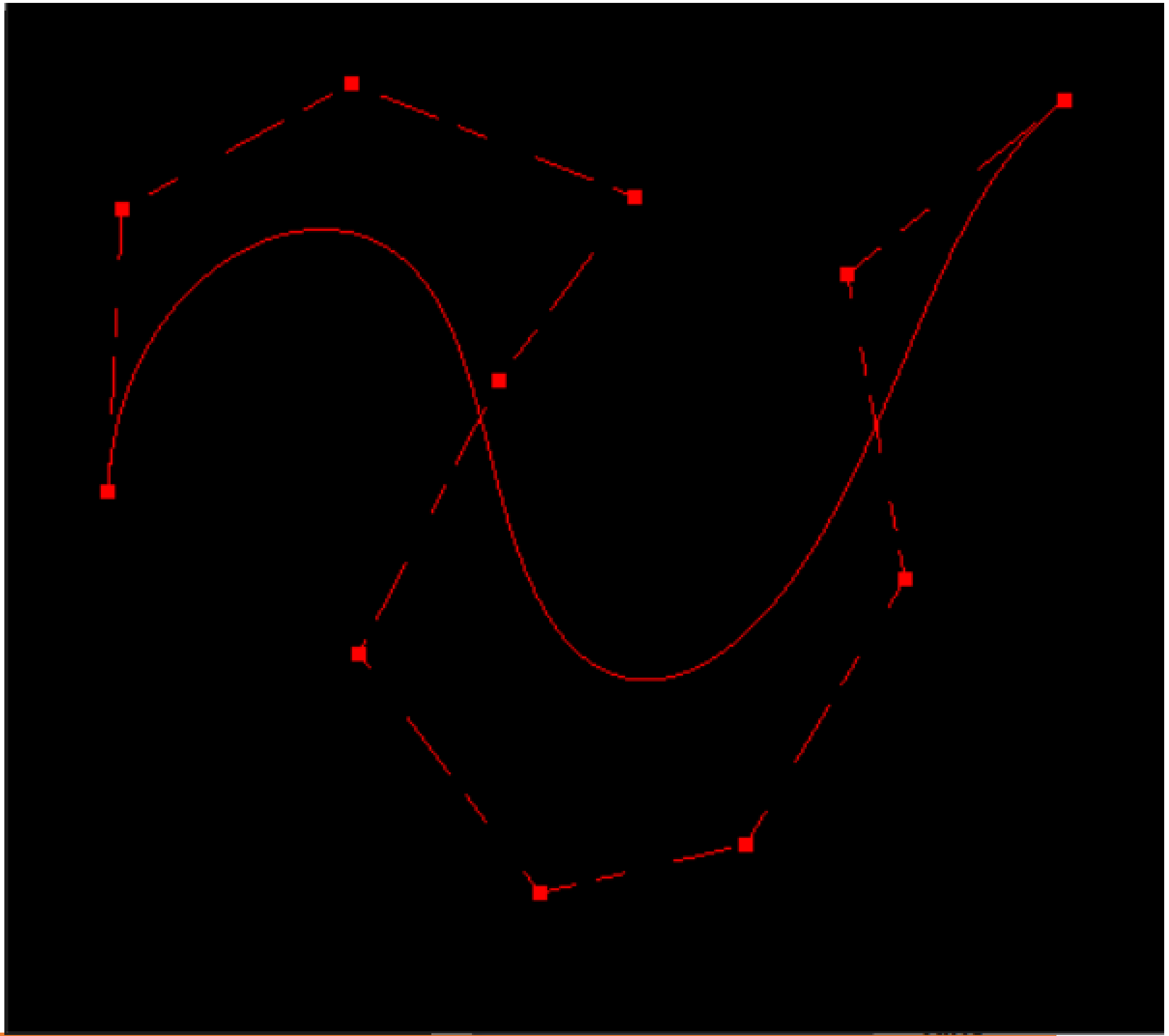
Vertici di controllo P_i , i=0,..,10

Funzioni base $B_{i,n}$(t), i=0,…n,   n=10



Curva di Bezier di grado 10

$$C(t) = \sum_{i=0}^{10} P_i B_{i,10}(t)$$



Il vantaggio di utilizzare la forma di Bezier di una curva, rispetto alla forma di Hermite è che, lavorando interattivamente si ottiene una modellazione della curva in modo molto più intuitivo. In un certo senso si può dire che la forma della curva approssima quella del poligono che si ottiene congiungendo con dei segmenti i punti di controllo. Questo poligono è detto poligono di controllo.

Mentre le curve di Hermite sono interpolanti, le curve di Bezier sono approssimanti di forma, approssimano la forma del poligono di controllo.

Le curve di Bèzier presentano molte proprietà interessanti ed utili per la computer graphics. Queste proprietà sono legate alle proprietà delle funzioni base utilizzate, cioè dei polinomi di Bernstein.

<u>Questi polinomi sono particolarmente adatti a rappresentare un polinomio definito in un intervallo [a,b], in quanto sono proprio definiti relativamente a quell'intervallo.</u>

Poiché nel caso di curve rappresentate in forma parametrica, se le componenti sono dei polinomi, il loro valore interessa solo relativamente all'intervallo di variabilità del parametro, la base dei polinomi di Bernestein è di particolare interesse.

In generale un polinomio di grado *n* sull'intervallo [a,b] può essere rappresentato come

$$P_n(x) = \sum_{i=0}^{n} c_i B_{i,n}(x) \qquad x \in [a,b]$$

dove i $B_{i,n}(x)$, i=0,…,*n* sono i polinomi di Bernstein di grado *n* definiti su [a,b] dalla relazione:

$$B_{i,n}(x) = \binom{n}{i} \frac{(b-x)^{n-i}(x-a)^i}{(b-a)^n} \qquad i = 0, \dots n$$

e $c_0, c_1, \dots, c_n$ sono i coefficienti che identificano il polinomio espresso nella base di Bernstein.

La rappresentazione data si contrappone alla rappresentazione mediante la base monomiale $1, x, x^2, \dots, x^n$

$$P_n(x) = \sum_{i=0}^{n} a_i x^i$$

Nelle applicazioni si suole usare l'intervallo [0,1] , ma questo non porta ad alcuna differenza, in quanto da un intervallo generico, ci si può ricondurre all'intervallo [0,1] senza alterare i coefficienti utilizzando la trasformazione

$$t = \frac{x-a}{b-a}$$

.

In [0,1] si ha

$$B_{i,n}(t) = \binom{n}{i}(1-t)^{n-i}t^{i} \qquad i = 0, \ldots n$$

$$P_n(t) = \sum_{i=0}^{n} c_i B_{i,n}(t) \qquad t \in [0,1]$$

E' interessante notare che questi polinomi sono tutti dello stesso grado, in contrapposizione alla base monomiale che usa monomi di grado diverso.

Nota. Il concetto di Curve di Bezier, nonché lo studio delle proprietà di queste curve, è stato sviluppato indipendentemente da P. De Casteljau nel 1959 e da P. Bezier nel 1962 perché potessero essere usate nei sistemi CAD della Renault e della Citroen, rispettivamente.

Proprietà dei polinomi di Bernstein di grado n:

1) I polinomi di Bernstein di grado n definiti in [a,b] sono legati dai polinomi di Bernstein di grado (n-1) definiti su [a,b] dalla seguente formula ricorrente:

$$B_{i,n}(x) = \left(\frac{x-a}{b-a} B_{i-1,n-1}(x) + \frac{b-x}{b-a} B_{i,n-1}(x) \right)$$

con condizione iniziale che $B_{0,0}(x)$, = 1 e $B_{i,n}(x) = 0 \;\; \forall \;\; i \notin [0,n]$

.

Se [a,b]=[0,1]

$$B_{i,n}(t) = t \cdot B_{i-1,n-1}(t) + (1-t) \cdot B_{i,n-1}(t) \qquad t \in [0,1]$$

Entrambe le formule, presentano solo somme di quantità positive, quindi sono numericamente stabili.

Ciascun polinomio di grado n è la combinazione convessa di polinomi di grado n-1: combinazione di quantità positive mediante coefficienti positivi, la cui somma vale 1.

2) Non negatività $B_{i,n}(x) \geq 0 \quad \forall\, x \in [0,1]$

3) I polinomi di Bernstein formano una **partizione dell'unità:**

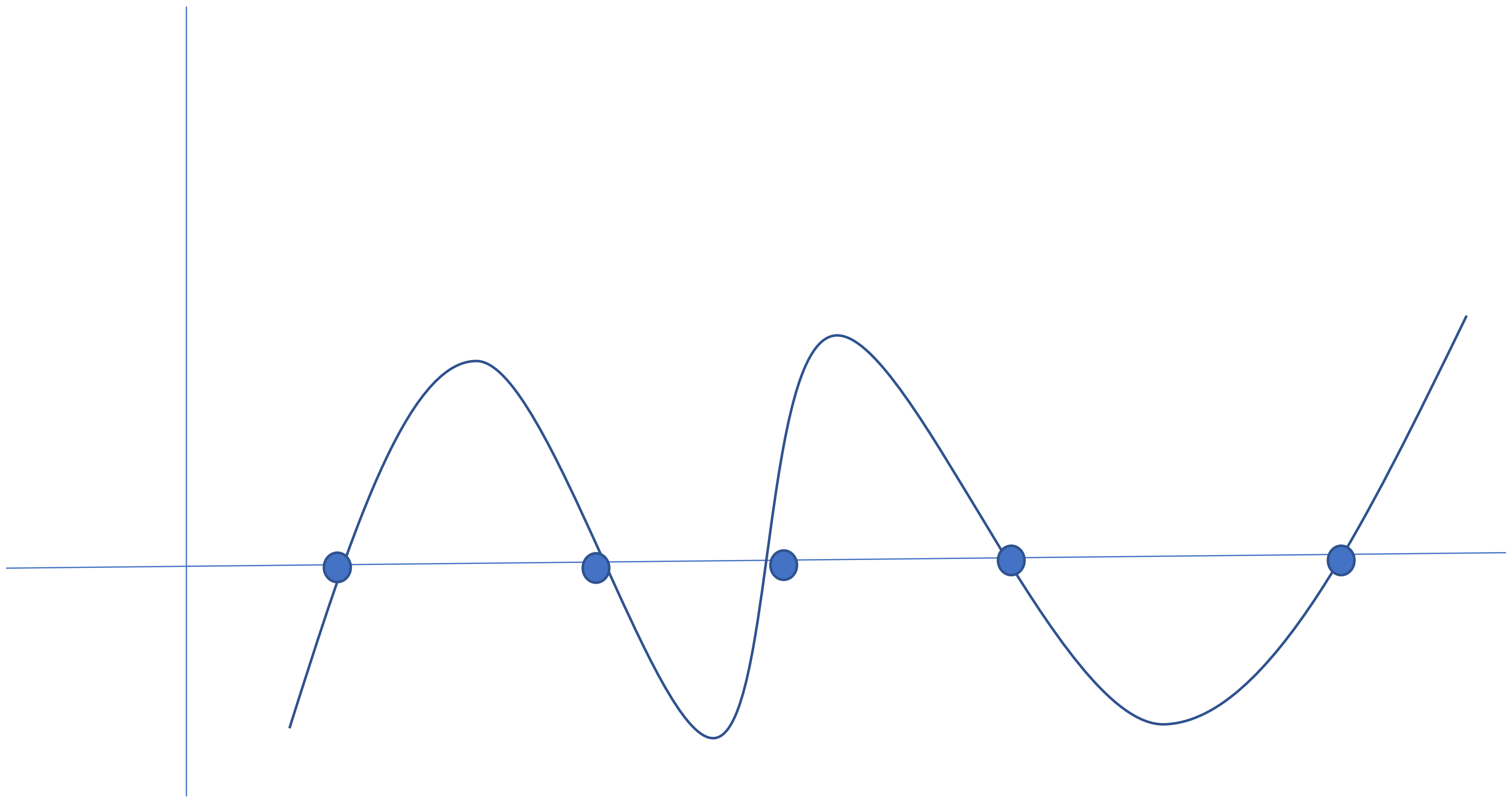
$$\sum_{i=0}^{n} B_{i,n}(t) = 1$$

Le proprietà 1) - 3) hanno una importante conseguenza.

4) Il polinomio $p(x) = \sum_{i=0}^{n} c_i B_{i,n}(x)$ è una **combinazione lineare convessa dei valori dei suoi coefficienti**, e quindi

$$\min_{i=0,\dots,n} \{c_i\} \leq p(x) \leq \max_{i=0,\dots,n} \{c_i\}$$

5) Il numero di **variazioni di segno di un polinomio** espresso nella base di Bernstein **è minore o uguale al numero di variazioni di segno dei suoi coefficienti**. Questo ci consente di avere per il polinomio un limite superiore per il numero dei suoi zeri semplici in [a,b].

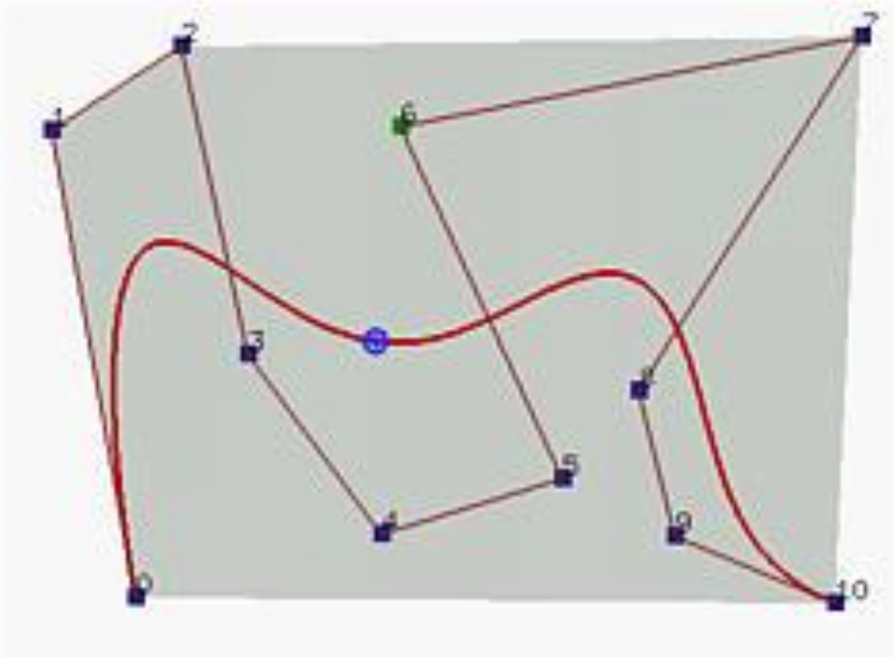


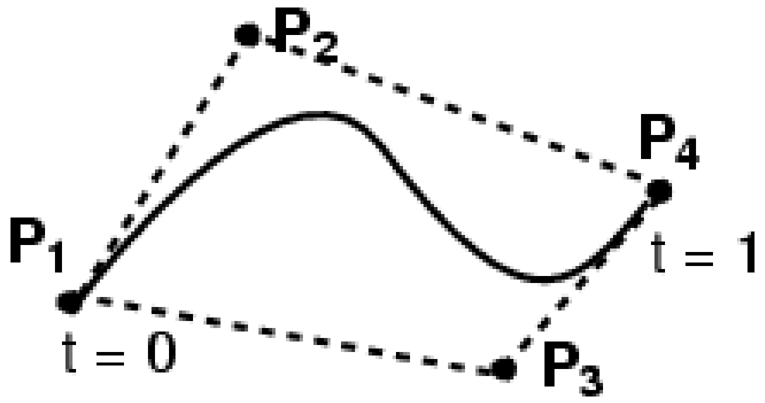
Proprietà dell’inviluppo convesso (convex Hull) :

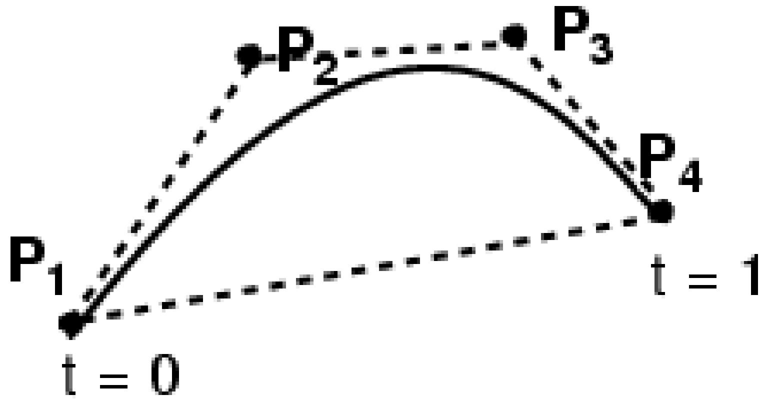
Definizione:

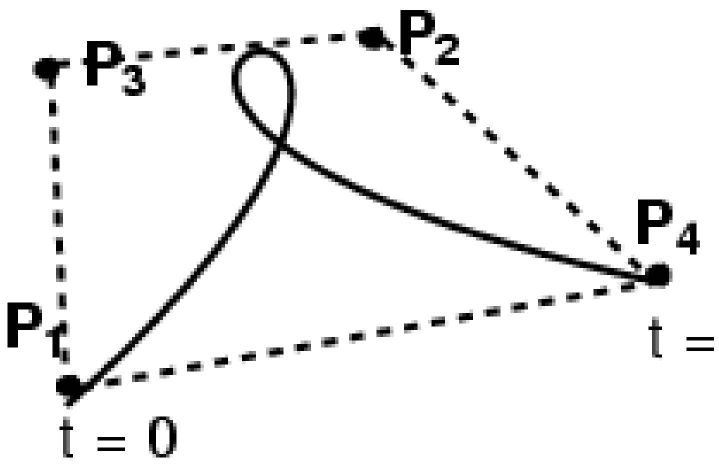
Assegnati i punti c_0, c_1,…, c_n, si definisce inviluppo convesso (convex hull) dell’insieme dei punti $\{c_i\}$ la più piccola regione convessa che contiene i punti dati.

Il grafico del polinomio espresso nella base di Bernstein è contenuto nell’inviluppo convesso dell’insieme dei vertici del suo poligono di controllo.




P1
P2
P3
P4
t = 0
t = 1


P1
P2
P3
P4
t = 0
t = 1

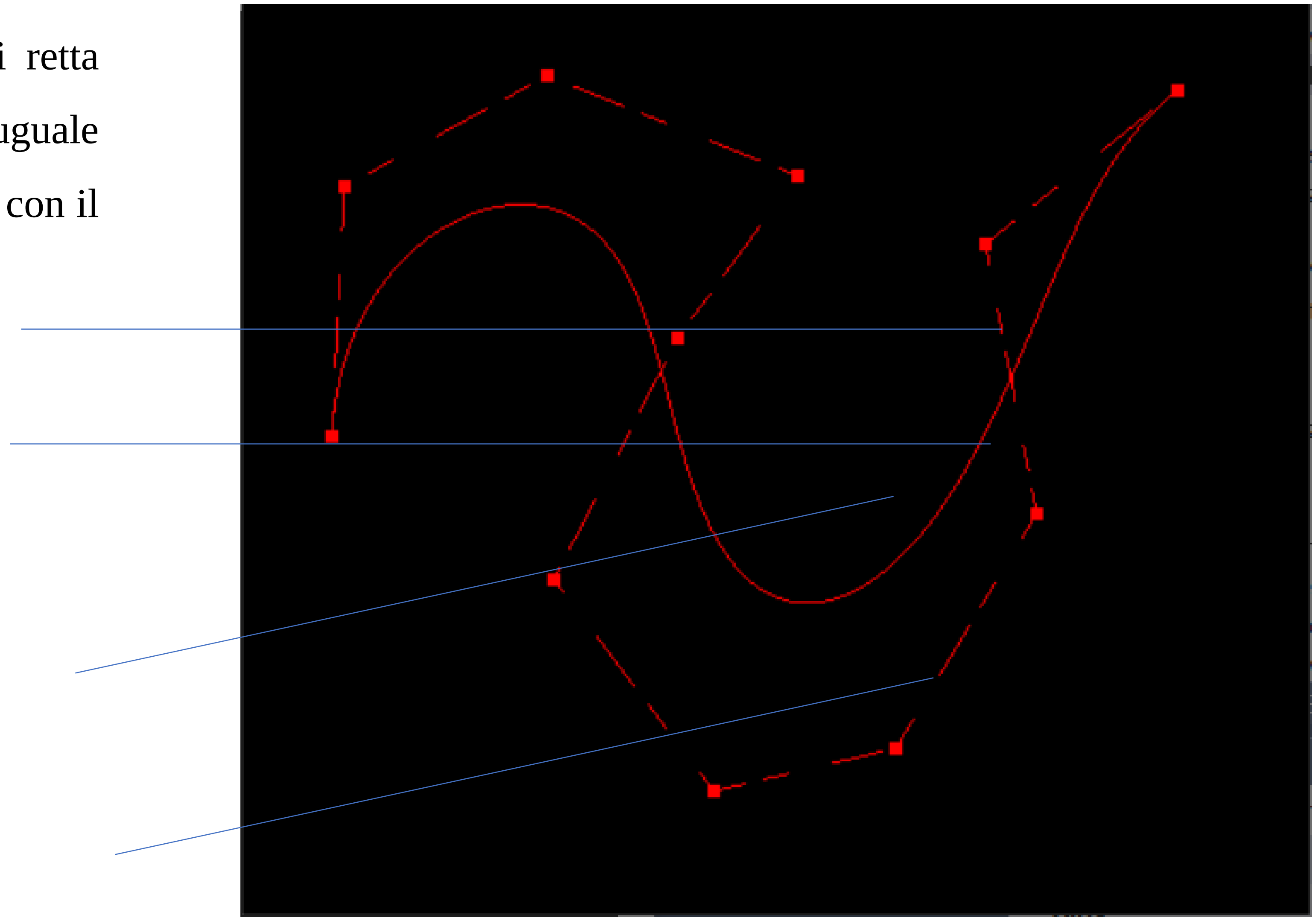

P1
P2
P3
P4
t = 0
t = 1

Proprietà di Precisione lineare:

- quando tutti i punti di controllo giacciono su una linea, la curva di Bézier è il segmento che interpola i punti. (segue dalla proprietà di Convex Hull)

Proprietà dell’approssimazione di forma.

Il numero di intersezioni di una qualsiasi retta con il poligono di controllo è maggiore o uguale del numero di intersezioni della stessa retta con il polinomio.



Dati i vertici di controllo $P_0 \equiv \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$, $P_1 \equiv \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$,…, $P_n \equiv \begin{bmatrix} x_n \\ y_n \end{bmatrix}$

la curva di Bezier di grado n si esprime come

$$C(t) = \begin{bmatrix} x(t) = \sum_{i=0}^{n} x_i B_{i,n}(t) \\ y(t) = \sum_{i=0}^{n} y_i B_{i,n}(t) \end{bmatrix} = \sum_{i=0}^{n} P_i B_{i,n}(t)$$

Valutazione di un polinomio nella base di Bernstein e algoritmo di de Casteljau:

E’ ben noto che la valutazione in un punto di un polinomio di grado n espresso nella base monomiale si può effettuare in modo efficiente facendo uso dello schema di Horner, cha ha una complessità computazione dell’ordine di O(n) ed è numericamente stabile.

L’algoritmo di De Casteljau rappresenta l’analogo per un polinomio espresso nella base di Bernstein. Esso consiste nell’applicare ripetutamente le formule ricorrenti viste per i polinomi della base di Bernstein.

La valutazione in un punto *t* di un polinomio espresso nella base di Bernstein consiste nel valutare:

$$P_n(t) = \sum_{i=0}^{n} c_i B_{i,n}(t) \qquad \text{in un punto } t \text{ assegnato.}$$

Dalle proprietà dei polinomi di base di Bernstein, ogni polinomio $B_{i,n}(t)$ è esprimibile come combinazione lineare convessa di due polinomi di base di Bernstein di grado n-1, sostituendo:

$$P(t) = \sum_{i=0}^{n} c_i \, [t \, B_{i-1,n-1}(t) + (1-t) \, B_{i,n-1}(t)] = \sum_{i=0}^{n} t \, c_i \, B_{i-1,n-1}(t) + \sum_{i=0}^{n} c_i \, (1-t) \, B_{i,n-1}(t) =$$

poichè $B_{-1,n-1}(t) = B_{n,n-1}(t) = 0$

$$= \sum_{i=1}^{n} t \, c_i \, (t) \, B_{i-1,n-1}(t) + \sum_{i=0}^{n-1} c_i \, (1-t) \, B_{i,n-1}(t) =$$

Vogliamo far variare entrambe le precedenti sommatorie da 0 ad n-1

$$= \sum_{i=0}^{n-1} t \, c_{i+1} \, (t) \, B_{i,n-1}(t) + \sum_{i=0}^{n-1} c_i \, (1-t) \, B_{i,n-1}(t) = \sum_{i=0}^{n-1} [t \, c_{i+1} + (1-t) \, c_i \,] B_{i,n-1}(t).$$

$$B_{i,n}(t) = t \cdot B_{i-1,n-1}(t) + (1-t) \cdot B_{i,n-1}(t) \quad t \in [0,1]$$

$$B_{i,n}(x) = 0 \quad \forall\ i \notin [0,n]$$

Ponendo $c_i^{[1]} = t\ c_{i+1}^{[0]} + (1\text{-}t)\ c_i^{[0]}$ = i=0,…,n-1 si ha

$$P(t) = \sum_{i=0}^{n-1} c_i^{[1]} \mathrm{B}_{\mathrm{i,n\text{-}1}}(\mathrm{t})$$

con coefficienti $c_i^{[1]}$ i=0,…,n-1.

Applicando *la formula ricorrente a* $B_{i,n-1(t)}$ nella formula precedente si ottiene

$$P(t) = \sum_{i=0}^{n-1} c_i^{[1]} B_{i,n-1}(t) = \sum_{i=0}^{n-1} c_i^{[1]} (t\, B_{i-1,n-2}(t) + (1-t) B_{i,n-2}(t)) =$$

$$= t \sum_{i=0}^{n-1} c_i^{[1]}\, B_{i-1,n-2}(t) + (1-t) \sum_{i=0}^{n-1} c_i^{[1]}\, B_{i,n-2}(t) =$$

i=0 -> $B_{-1,n-2}(t) = 0$ i=n-1 -> $B_{n-1,n-2}(t) = 0$

$$= t \sum_{i=1}^{n-1} c_i^{[1]}\, B_{i-1,n-2}(t) + (1-t) \sum_{i=0}^{n-2} c_i^{[1]}\, B_{i,n-2}(t) = \ t \sum_{i=0}^{n-2} c_{i+1}^{[1]}\, B_{i,n-2}(t) + (1-t) \sum_{i=0}^{n-2} c_i^{[1]}\, B_{i,n-2}(t) =$$

$$P_n(t) = \sum_{i=0}^{n-2} (t\, c_{i+1}^{[1]} + (1-t) c_i^{[1]}) \ \ B_{i,n-2}(t)$$

$$c_i^{[2]} = \text{t} \ c_{i+1}^{[1]} + (\text{1-t}) \ c_i^{[1]} \quad = \qquad \text{i}=0,\ldots,n\text{-}2$$

$$P(t) = \sum_{i=0}^{n-2} c_i^{[2]} B_{i,n-2(t)}$$

Continuando ad iterare nello stesso modo, al passo j-esimo avremo:

$$P(t) = \sum_{i=0}^{n-j} \ c_i^{[j]} \ \text{B}_{\text{i,n-j}}(\text{t})$$

con

$$\boxed{c_i^{[j]} = \text{t} \ c_{i+1}^{[j-1]} + (\text{1-t}) \ c_i^{[j-1]}} \qquad i=0,\ldots,n\text{-}j \qquad \text{j}=1,\ldots,n$$

Quando $j = n$ si ottiene :

$$P(t) = \sum_{i=0}^{0} c_i^{[n]} \, B_{i,0}(t) = c_0^{[n]} \cdot B_{0,0}(t) = c_0^{[n]}$$

con $c_0^{[n]} = t \, c_1^{[n-1]} + (1-t) \, c_0^{[n-1]}$

- L'algoritmo di De Casteljau non fa uso delle funzioni base, ma ad ogni passo fa una combinazione convessa dei coefficienti al passo precedente.

- E' computazionalmente valido sia perché elimina il problema di passare attraverso le funzioni base, sia perché permette di calcolare ciascuna nuova grandezza come combinazione convessa di due grandezze precedenti, mantenendo stabile l'algoritmo.

L'algoritmo può essere schematizzato nel seguente modo:

$$
\begin{array}{cccccc}
c_0^{[0]} & c_1^{[0]} & c_2^{[0]} & \cdots & c_{n-1}^{[0]} & c_n^{[0]} \\
(1\text{-}t) \quad t & (1\text{-}t) \quad t & & & (1\text{-}t) \quad t & \\
c_0^{[1]} & c_1^{[1]} & c_2^{[1]} & \cdots & c_{n-1}^{[1]} & \\
\cdot & & & & & \\
\cdot & & & & & \\
\cdot & & & & & \\
c_0^{[n-2]} & c_1^{[n-2]} & c_{n-1}^{[2]} & & & \\
c_0^{[n-1]} & c_1^{[n-1]} & & & & \\
c_0^{[n]} & & & & &
\end{array}
$$

j=1,…,n

i=0,…,n-j

$$\boxed{c_i^{[j]} = t\ c_{i+1}^{[j-1]} + (1\text{-}t)\ c_i^{[j-1]}}$$

Complessità computazionale

$$2\sum_{j=1}^{n} n-j+1 = 2\sum_{j=1}^{n} j = 2\cdot O\left(\frac{n^2}{2}\right)$$

Mediante tale algoritmo la valutazione di un polinomio nella base di Bernstein ha un costo computazionale di $O(n^2)$ operazioni moltiplicative

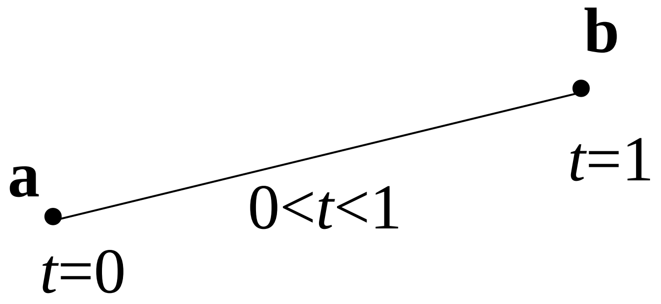
Valutazione curva di Bezier

Per ogni valore del parametro t, applicare l'algoritmo di De Casteljau ad ognuna delle sue componenti parametriche

$$C(t) = \sum_{i=0}^{n} P_i \, B_{i,n}(t) = \begin{cases} \sum_{i=0}^{n} x_i \, B_{i,n}(t) \\ \sum_{i=0}^{n} y_i \, B_{i,n}(t) \end{cases}$$

Interpolazione Lineare(Lerp)

- Interpolazione Lineare (Lerp) calcola un valore compreso tra due valori
- L'interpolante lineare tra due punti **a** and **b** con parametro t è data da:

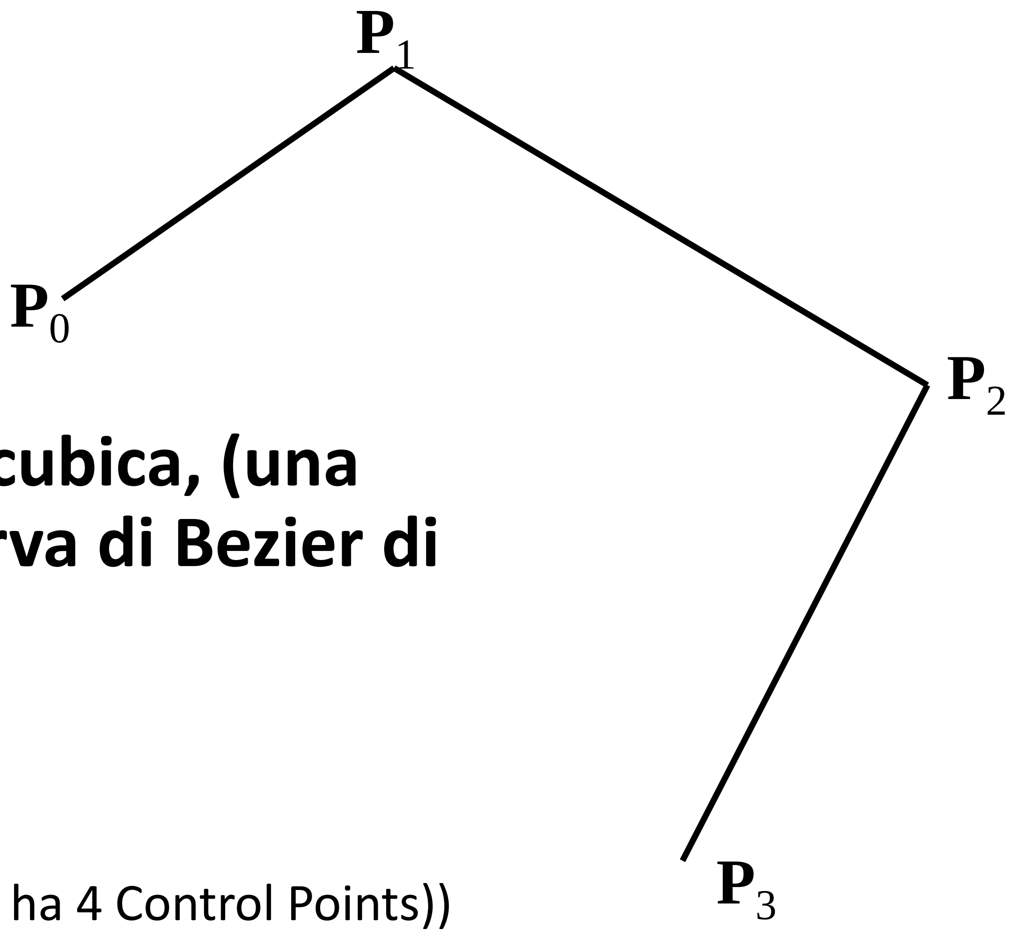


$$Lerp(t, \mathbf{a}, \mathbf{b}) = (1-t)\mathbf{a} + t\mathbf{b}$$

$$P_i^j = (1-\mathrm{t})P_i^{j-1} + \mathrm{t}\, P_{i+1}^{j-1} \longrightarrow P_i^j = Lerp(t, P_i^{j-1}, P_{i+1}^{j-1})$$

Interpretazione geometrica dell’algoritmo di de Casteljau

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



Dato un valore di t, valutare C(t)

Consideriamo una curva di Bezier cubica, (una costruzione simile vale per una curva di Bezier di qualsiasi grado)

- Partiamo con
  - Control points (Una curva di Bezier cubica ha 4 Control Points))
  - Un valore del parametro t (per esempio t=0.4)

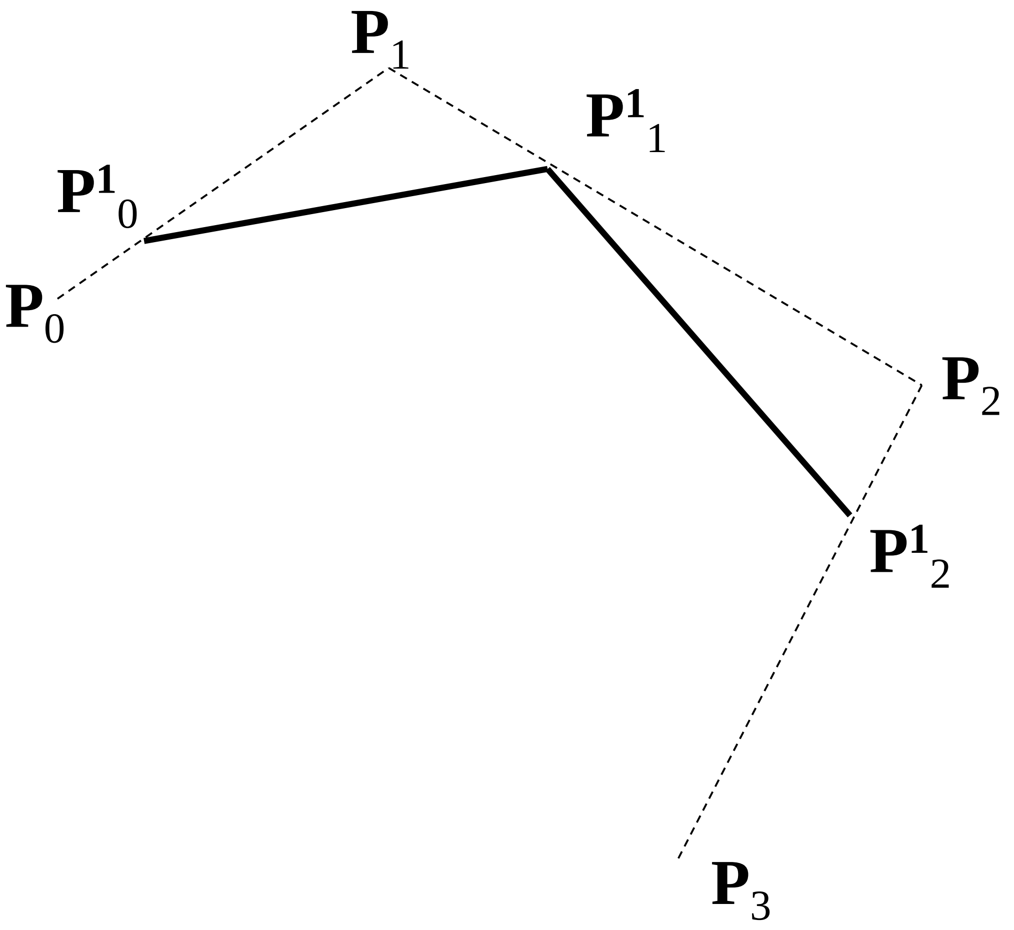
Algoritmo di de Casteljau

Step 1

$$P_0^1 = Lerp(t, P_0, P_1)$$

$$P_1^1 = Lerp(t, P_1, P_2)$$

$$P_2^1 = Lerp(t, P_2, P_3)$$

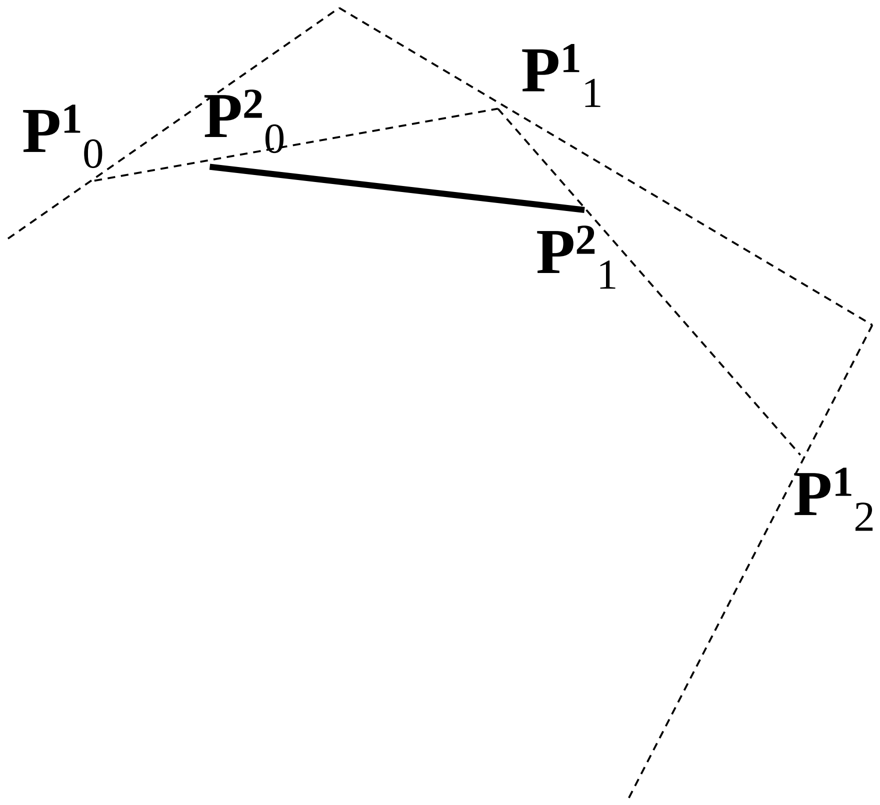


Algoritmo di de Casteljau

Step 2

$$P_0^2 = Lerp(t, P_0^1, P_1^1)$$

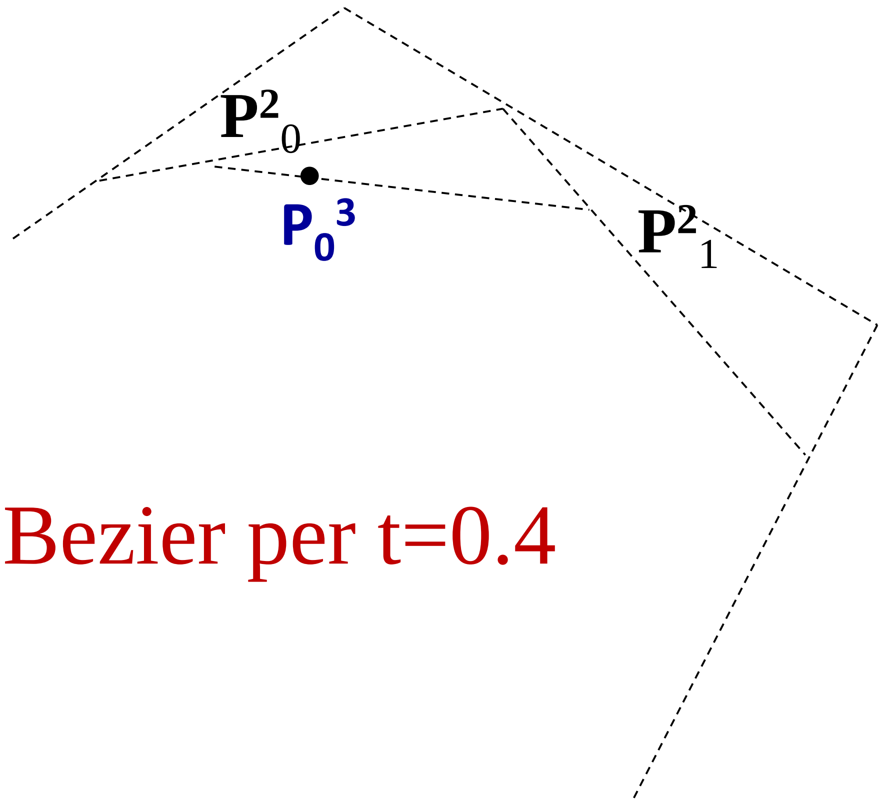
$$P_1^2 = Lerp(t, P_1^1, P_2^1)$$



Algoritmo di de Casteljau

Step 3

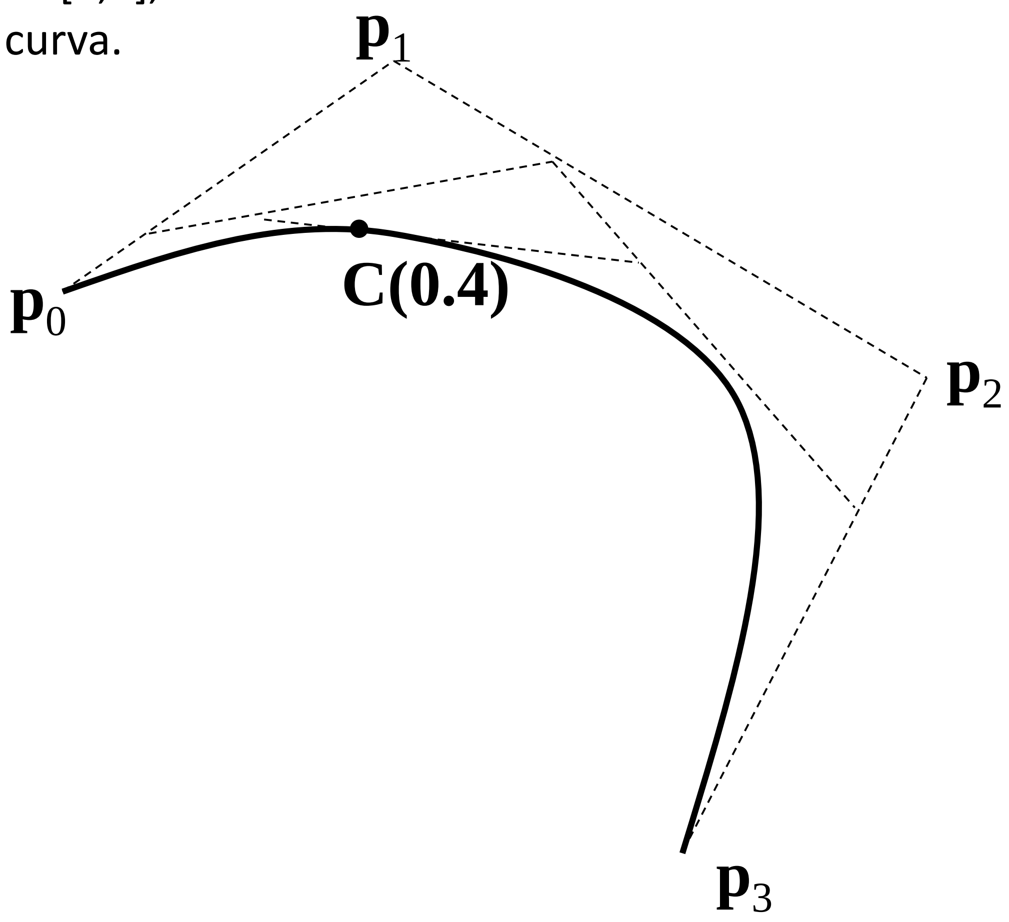
$P_0^3 = Lerp(t, P_0^2, P_1^2)$



$P_0{}^3$ è il valore della curva di Bezier per t=0.4

Curva di Bezier

Ripetendo questo procedimento per ogni valore di t in [0,1], sarà generata una sequenza di punti che definisce la curva.



Algoritmo di de Casteljau

Input: P_i , t

$P_i^0 = P_i$

```
for j=1:n
    for i=0:n-j
```

$P_i^j = (1-\mathrm{t})P_i^{j-1} + \mathrm{t}\, P_{i+1}^{j-1}$

```
    end
end
```

n grado della curva

P_i control points i=0,..,n

t parametro di valutazione

$P_i \equiv (x_i , y_i)$

Input: x_i , t

$x_i^0 = x_i$

```
for j=1:n
    for i=0:n-j
```

$x_i^j = (1-\mathrm{t})x_i^{j-1} + \mathrm{t}\, x_{i+1}^{j-1}$

```
    end
end
```

Input: P_i , t

$y_i^0 = y_i$

```
for j=1:n
    for i=0:n-j
```

$y_i^j = (1-\mathrm{t})y_i^{j-1} + \mathrm{t}\, y_{i+1}^{j-1}$

```
    end
end
```

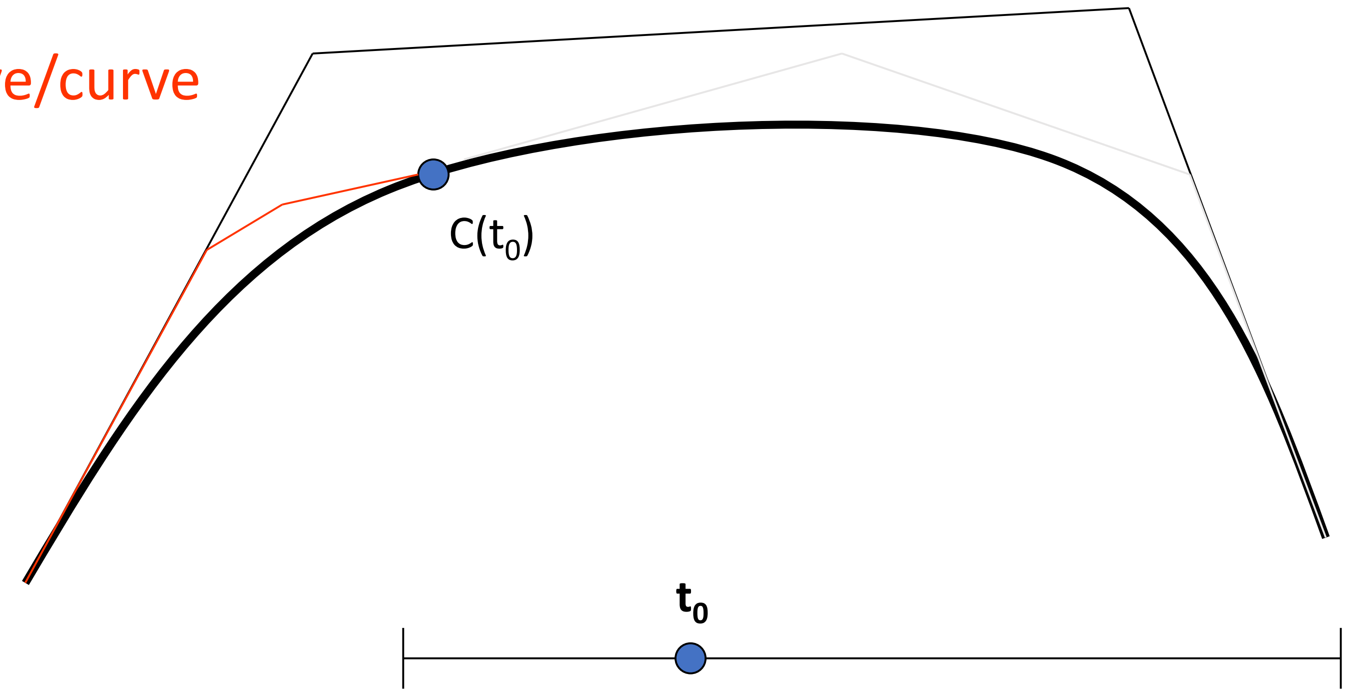
Suddivisione di una curva

E’ il processo di splittare una curva di Bezier di grado n, in due sottocurve di Bezier di grado n. Per realizzare questo splitting si utilizza l’algoritmo di de Casteljau

Strumento per:

-Raffinamento

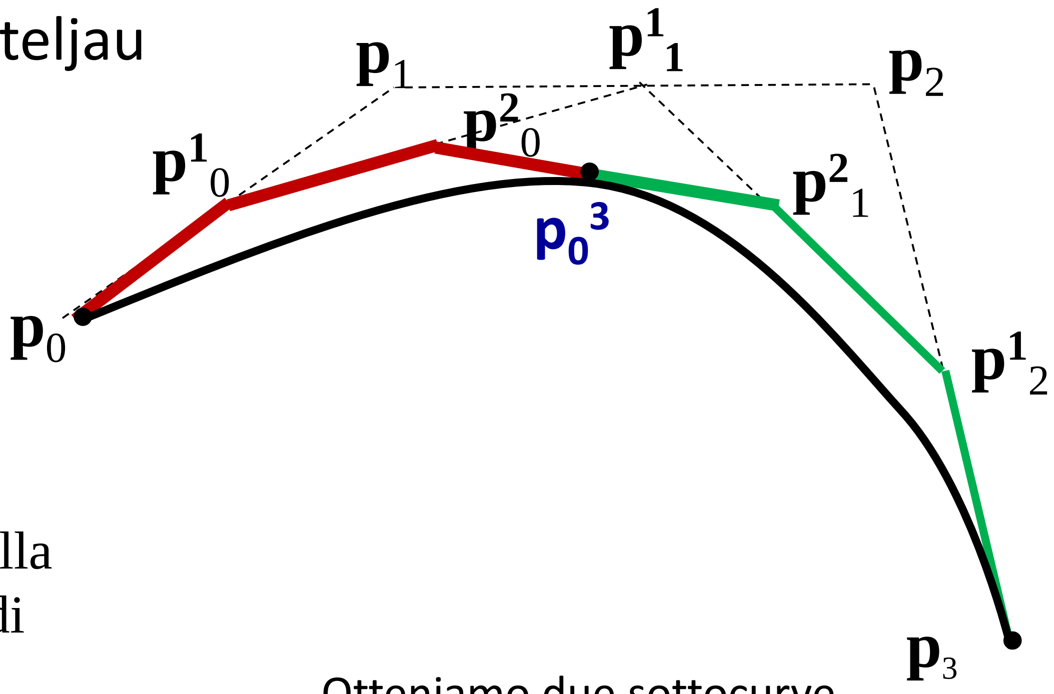
-Intersezione Curve/curve



Suddivisione di una curva di Bezier,utilizzando l’algoritmo di De Casteljau

Suddividiamo la curva C(t) definita in [0,1] in $t=t_0$. Applichiamo l’algoritmo di De Casteljau

Applichiamo l’algoritmo di de Casteljau per $t=t_0$



P_0^0	P_1^0	P_2^0	P_3^0
P_0^1	P_1^1	P_2^1	
P_0^2	P_1^2		
P_0^3			

Punti di controllo della seconda sottocurva di Bezier

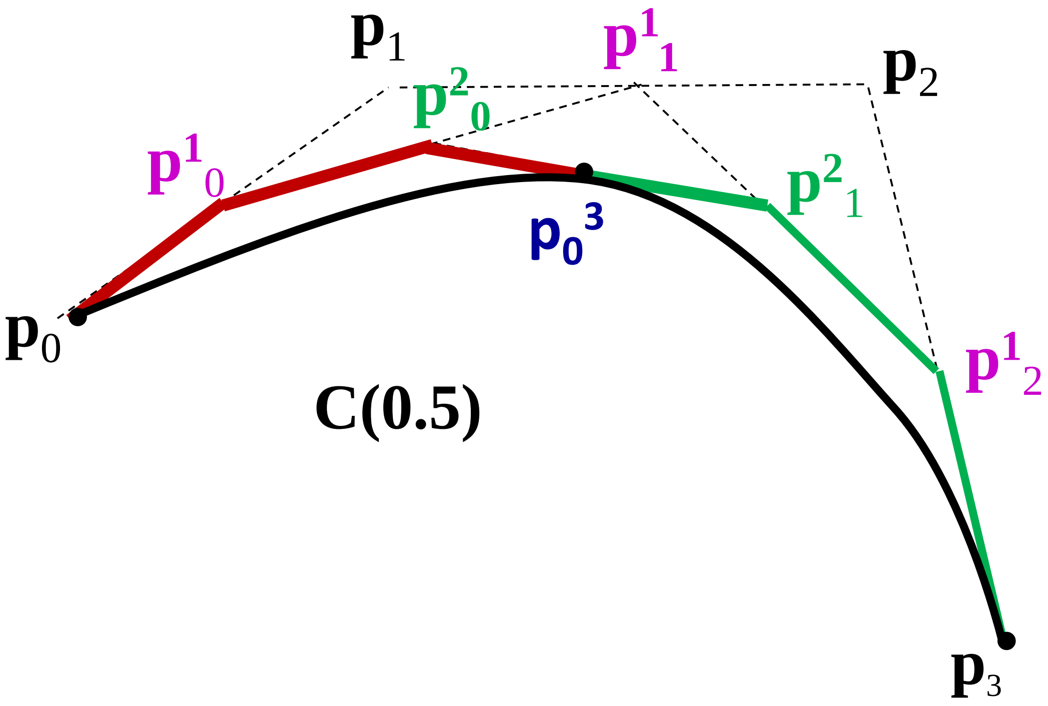
Punti di controllo della prima sottocurva di curva di Bezier

Otteniamo due sottocurve
C_1 in $[0,t_0]$
C_2 in $[t_0,1]$

$$C_1(t) = \sum_{i=0}^{n} P_0^i B_{i,n}(t) \quad t\in [0,t_0]$$

$$C_2(t) = \sum_{i=0}^{n} P_{n-i}^i B_{i,n}(t) \quad t\in [t_0,1]$$

Ex. Suddivisione di una curva in t=0.5 facendo uso dell’algoritmo di De Casteljau



- Se il procedimento di suddivisione è iterato, cioè ogni segmento di curva viene suddiviso a sua volta, si ottiene che la successione dei poligoni di controllo dei vari segmenti di curva ottenuti converge velocemente alla curva originale di Bezier, all’addensarsi dei punti di suddivisione all’interno dell’intervallo parametrico.
- Il metodo delle suddivisioni ripetute è spesso utilizzato per “rendere” la curva graficamente, sostituendo la curva teorica con l’insieme dei punti di controllo dei vari segmenti.

Proprietà dell'Algoritmo di de Casteljau

1) L'algoritmo di De CastelJau è numericamente stabile, infatti si eseguono solo somme e moltiplicazioni per coefficienti positivi.
2) È invariante per trasformazioni affini. Quindi, invece di valutare una curva e poi applicare una trasformazione, si può applicare la trasformazione affine ai punti di controllo P_i, $i=0,..,n$ e poi applicare l'algoritmo di De CastelJau.
3) Fornisce un metodo semplice per suddividere curve di Bezier. Infatti il punto $t=t_0$ in cui si valuta la curva rappresenta un punto di suddivisione della curva in due curve di Bezier. Una per $t\in[0, t_0]$ e l'altra per $t\in[t_0,1]$.

Elevamento di grado di un polinomio nella base di Bernstein (Degree Elevation):

Ci sono alcune situazioni in cui è necessario esprimere un polinomio di grado n come un polinomio di grado n+1, per esempio nel caso in cui bisogna sommare due polinomi di grado diverso. Ciò risulta semplice nel caso in cui il polinomio sia espresso nella base monomiale .

$$p_2(x) = a_0 + a_1 x + a_2 x^2$$

$$p_3(x) = b_0 + b_1 x + b_2 x^2 + b_3 x^3$$

$$p_2(x) + p_3(x) = (a_0 + b_0) + (a_1 + b_1)x + (a_2 + b_2)x^2 + b_3 x^3$$

Se si utilizza la base dei polinomi di Bernstein, il passaggio non è così immediato.

Se

$$p_1(t) = a_0 B_{0,3}(t) + a_1 B_{1,3}(t) + a_2 B_{2,3}(t) + a_3 B_{3,3}(t)$$

$$p_2(t) = c_0 B_{0,2}(t) + c_1 B_{1,2}(t) + c_2 B_{2,2}(t)$$

- Non è possibile sommare direttamente questi due polinomi.
- Non è possibile raccogliere le funzioni base e sommare i coefficienti, perché hanno grado diverso.
- Bisognerebbe passare alla forma nella base monomiale e poi fare la somma, oppure è possibile elevare di grado il polinomio p_2(t) cioè esprimerlo nella base $B_{i,3}$(t), i=0,1,2,3.

Bisogna fare quindi un degree-elevation, cioè esprimere un polinomio, dato nella base di Bernstein di grado n come se fosse un polinomio (nella base di Bernstein) di grado n+1. Ricordiamo che il polinomio è sempre lo stesso, lo esprimiamo solo in una nuova base.

Dato un polinomio espresso nella base di Bernstein di grado n, cioè nella base $B_{i,n}(t)$, $i=0,\ldots,n$, vogliamo esprimerlo nella base $B_{i,n+1}(t)$, $i=0,\ldots,n+1$.

Dato quindi il polinomio $$P_n(t) = \sum_{i=0}^{n} c_i \, B_{i,n}(t)$$

vogliamo poterlo esprimere come

$$P_{n+1}(t) = \sum_{i=0}^{n+1} d_i \, B_{i,n+1}(t)$$

Quali devono essere i coefficienti d_i, i=0,..n+1, affinchè $P_n(t) = P_{n+1}(t)$?

Noti i coefficienti c_i , vogliamo calcolare i nuovi coefficienti nella nuova base affinchè risulti $P_n(t) = P_{n+1}(t)$

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

$$P_n(t) = \sum_{i=0}^{n} c_i \binom{n}{i}(1-t)^{n-i}t^i$$

$$P_{n+1}(t) = \sum_{i=0}^{n+1} d_i \binom{n+1}{i}(1-t)^{n+1-i}t^i$$

Moltiplichiamo $P_n(t)$ per [(1-t)+t]

$$P_n(t)\,[(1-t)+t] = \sum_{i=0}^{n} c_i \binom{n}{i}\left[(1-t)^{n+1-i}t^i + (1-t)^{n-i}t^{i+1}\right] =$$

$$= \sum_{i=0}^{n} c_i \binom{n}{i}\left[(1-t)^{n+1-i}t^i\right] + \sum_{i=0}^{n} c_i \binom{n}{i}(1-t)^{n-i}t^{i+1}$$

Estendiamo la prima sommatoria per i=0,…,n+1, ponendo $c_{n+1} = 0$, ed esprimiamo la seconda sommatoria per i=1, …, n+1

$$= \sum_{i=0}^{n} c_i \binom{n}{i} \left[\quad (1-t)^{n+1-i} t^i \quad \right] + \sum_{i=0}^{n} \quad c_i \binom{n}{i} (1-t)^{n-i} t^{i+1} =$$

$$= \sum_{i=0}^{n+1} c_i \binom{n}{i} \left[\quad (1-t)^{n+1-i} t^i \quad \right] + \sum_{i=1}^{n+1} c_{i-1} \binom{n}{i-1} (1-t)^{n-i+1} t^i =$$

Adesso facciamo variare la seconda sommatoria per i=0,…, n+1, ponendo $c_{-1} = 0$. Ottenendo così:

$$= \sum_{i=0}^{n+1} c_i \binom{n}{i} \left[\quad (1-t)^{n+1-i} t^i \quad \right] + \sum_{i=0}^{n+1} c_{i-1} \binom{n}{i-1} (1-t)^{n-i+1} t^i$$

$$P_n(t) = \sum_{i=0}^{n+1} \left(c_i \binom{n}{i} + c_{i-1} \binom{n}{i-1} \right) (1-t)^{n-i+1} t^i$$

Confrontiamo i coefficienti di questo polinomio con i coefficienti del polinomio espresso nella base dei polinomi di Bernstein di grado n+1

$$P_{n+1}(t) = \sum_{i=0}^{n+1} d_i \binom{n+1}{i} (1-t)^{n+1-i} t^i$$

Affinchè rappresentino lo stesso polinomio,devono avere gli stessi coefficienti, cioè deve accadere che

$$d_i \binom{n+1}{i} = c_i \binom{n}{i} + c_{i-1} \binom{n}{i-1}$$

Da cui si ricava che

$$d_i = \frac{i}{n+1} c_{i-1} + \left(1 - \frac{i}{n+1}\right) c_i$$

con c_{-1} e c_{n+1} uguali a 0 per i=0,…,n+1.

Dimostrazione:

$$d_i \frac{(n+1)!}{i!(n+1-i)!} = c_i \frac{n!}{i!(n-i)!} + c_{i-1} \frac{n!}{(i-1)!(n-i+1)!}$$

$$d_i \frac{(n+1)n!}{i(i-1)!(n+1-i)(n-i)!} = \frac{c_i (n+1-i) n! + c_{i-1} n! i}{i(i-1)!(n+1-i)(n-i)!}$$

Semplificando otteniamo:

$$d_i = \frac{c_i (n+1-i) + c_{i-1} i}{n+1} \qquad d_i = \left(1 - \frac{i}{n+1}\right) c_i + \frac{i}{n+1} c_{i-1}$$

Cioè i nuovi coefficienti (che si trovano sul segmento che unisce $c_{i\text{-}1}$ e c_i) sono ottenuti

interpolando linearmente i vecchi coefficienti per il valore del parametro $\frac{i}{n+1}$.

La formula che fornisce i nuovi coefficienti nella base di grado n+1 non è altro che una combinazione convessa dei vecchi coefficienti. Si ha quindi che il nuovo poligono di controllo è interno all'inviluppo convesso dei nuovi coefficienti.

$$d_i = \left(1 - \frac{i}{n+1}\right) c_i + \frac{i}{n+1} c_{i-1}$$

$$d_i = Lerp\left(\frac{i}{n+1}, c_i, c_{i-1}\right)$$

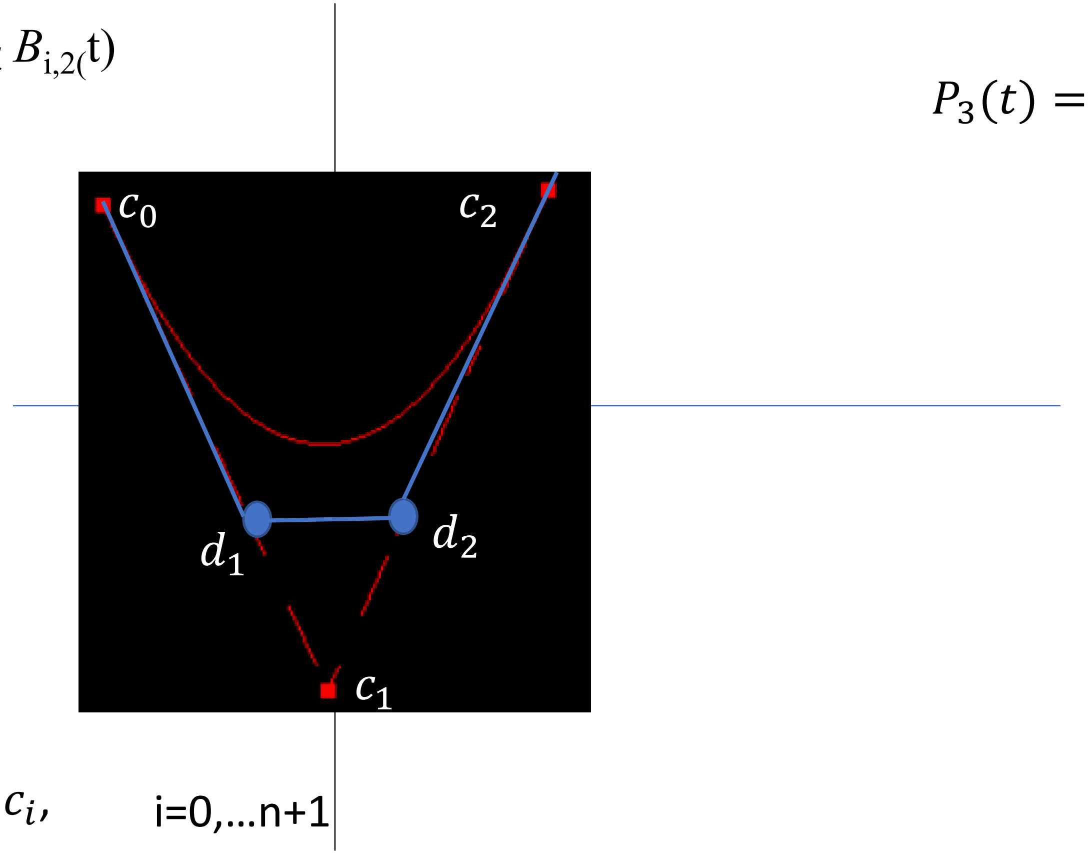
$$P_2(t) = \sum_{i=0}^{2} c_i \, B_{i,2}(t)$$

$$P_3(t) = \sum_{i=0}^{3} d_i \, B_{i,3}(t)$$

c_0=2

c_1=-2

c_2=2



$$d_i = \frac{i}{n+1} c_{i-1} + \left(1 - \frac{i}{n+1}\right) c_i,$$ i=0,…n+1

$$d_0 = \frac{0}{3} c_{-1} + \left(1 - \frac{0}{3}\right) c_0 \qquad d_0 = c_0 \qquad d_1 = \frac{1}{3} c_0 + \left(1 - \frac{1}{3}\right) c_1 = \frac{1}{3} c_0 + \frac{2}{3} c_1$$

$$d_3 = \frac{3}{3} c_2 + \left(1 - \frac{3}{3}\right) c_3 \qquad d_3 = c_2 \qquad d_2 = \frac{2}{3} c_1 + \left(1 - \frac{2}{3}\right) c_2 = \frac{2}{3} c_1 + \frac{1}{3} c_2$$

Il poligono di controllo del polinomio aumentato di grado è più vicino al grafico del polinomio.

Il processo di elevamento può essere ripetuto più volte. Si ottiene così una successione di poligoni di controllo che si dimostra che converge al polinomio medesimo.

La convergenza però è troppo lenta per essere usate per scopi pratici.

Derivata prima Curva di Bezier

Se $C(t) = \sum_{i=0}^{n} P_i B_{i,n}(t), \quad t \in [0,1]$

Allora

$$C'(t) = \sum_{i=0}^{n} P_i \, B'_{i,n}(t) \qquad\qquad t \in [0,1] \; .$$

di seguito dimostreremo che vale

$$B'_{i,n}(t) = n(B_{i-1,n-1}(t) - B_{i,n-1}(t))$$

Per ricavare questa formula, calcoliamo la derivata di un elemento della base di Bernstein:

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

$$B'_{i,n}(t) = \binom{n}{i}\frac{d}{dt}((1-t)^{n-i}t^i) = \binom{n}{i}\left[i(1-t)^{n-i}t^{i-1} + (n-i)(1-t)^{n-i-1}(-1)t^i\right] =$$

$$= \binom{n}{i}\left[i(1-t)^{n-i}t^{i-1} - (n-i)(1-t)^{n-i-1}t^i\right]$$

Poiché $\binom{n}{i} i = n\binom{n-1}{i-1}$ e $\binom{n}{i}(n-i) = n\binom{n-1}{i}$

si ha

$$B'_{i,n}(t) = n\left[\binom{n-1}{i-1}(1-t)^{n-i}t^{i-1} - \binom{n-1}{i}(1-t)^{n-i-1}t^i\right]$$

Per definizione:

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

si ha

$$B'_{i,n}(t) = n(B_{i-1,n-1}(t) - B_{i,n-1}(t)).$$

Questa espressione mette in evidenza il fatto che la curva di Bezier è tangente negli estremi al segmento che unisce i primi due punti e gli ultimo due punti di controllo rispettivamente.

$$C'(t) = \sum_{i=0}^{n} P_i \, B'_{i,n}(t)$$

$$C'(t) = \sum_{i=0}^{n} P_i B'_{i,n}(t) = n \sum_{i=0}^{n} P_i B_{i-1,n-1}(t) - n \sum_{i=0}^{n} P_i B_{i,n-1}(t)$$

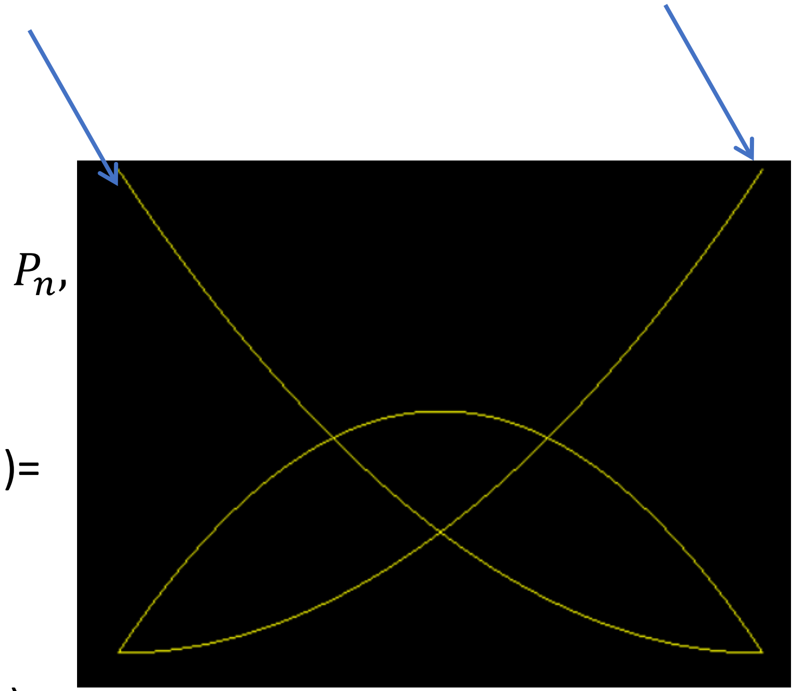
Se i=0 nella prima sommatoria $B_{-1,n-1}(t) = 0$

Se i=n nella seconda sommatoria $B_{n,n-1}(t) = 0$.

$$C'\ (t) = n \sum_{i=0}^{n-1} P_{i+1}\ B_{i,n-1}(t) - n \sum_{i=0}^{n-1} P_i\ B_{i,n-1}(t) = \sum_{i=0}^{n-1} n(P_{i+1} - P_i)B_{i,n-1}(t)$$

Ponendo $d_i = n(P_{i+1} - P_i)$

$$C'(t) = n \sum_{i=0}^{n-1} D_i B_{i,n-1}(\text{t})$$



Il vertice di controllo P_0, corrisponde al valore del parametro t=0, Il vertice di controllo P_n, corrisponde al valore del parametro t=1,

$$C'(0) = n \sum_{i=0}^{n-1} D_i B_{i,n-1}(0) = \text{n}(\ D_0 B_{0,n-1}(0) + D_1 B_{1,n-1}(0) + \text{....} + D_{n-1} B_{\text{n}-1,n-1}(0)) =$$

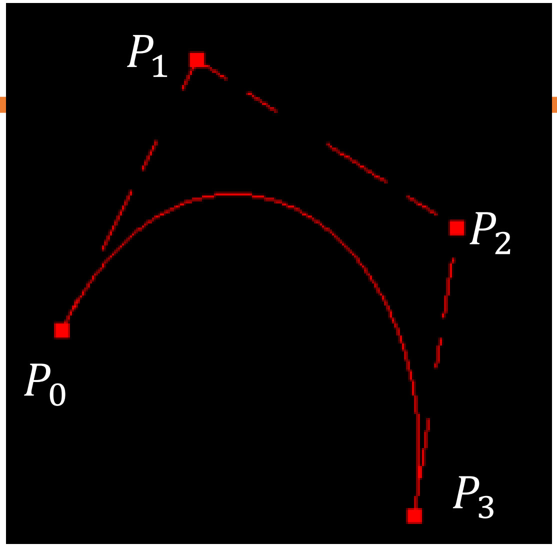
$$\text{C}'(0) = \text{n}D_0 = \text{n}\ (P_1 - P_0)$$

$$C'(0) = n \sum_{i=0}^{n-1} D_i B_{i,n-1}(0) = \text{n}(\ D_0 B_{0,n-1}(1) + D_1 B_{1,n-1}(1) + \text{....} + D_{n-1} B_{\text{n}-1,n-1}(1)) =$$

$$\text{C}'(1) = \text{n}D_{n-1} = \text{n}\ (P_n - P_{n-1})$$

$$C'(0) = \mathrm{n}D_0 = 3\,(P_1 - P_0)$$

$$C'(1) = \mathrm{n}D_2 = 3\,(P_3 - P_2)$$



La curva di Bezier interpola solo il primo e l'ultimo vertice di controllo

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

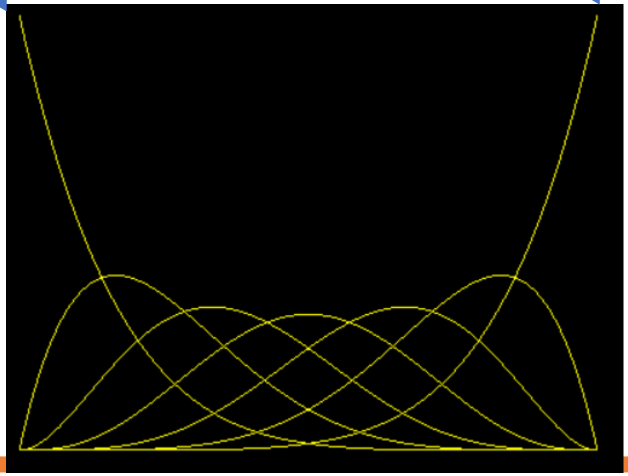
Il vertice di controllo P_0, corrisponde al valore del parametro t=0, Il vertice di controllo P_n, corrisponde al valore del parametro t=1,

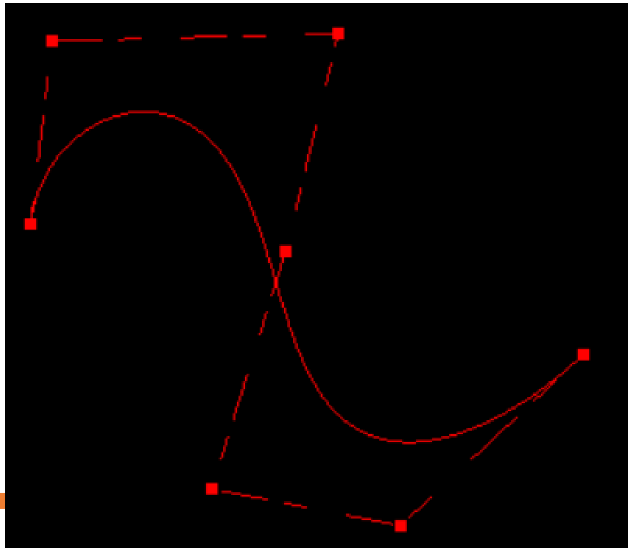
$$C(0) = P_0 B_{0,n}(0) + P_1 B_{1,n}(0) + \ldots. + P_n B_{\mathrm{n},n}(0)$$

$$C(0) = P_0 B_{0,n}(0) = P_0$$

$$C(1) = P_0 B_{0,n}(1) + P_1 B_{1,n}(1) + \ldots. + P_n B_{\mathrm{n},n}(1)$$

$$C(1) = P_n B_{\mathrm{n},n}(1) = P_n$$





Cambiamento di Base – Da base monomiale a base di Bernstein e viceversa

A volte è necessario passare dalla base di Bernstein alla base monomiale e viceversa.

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

La matrice di cambiamento di base che fa passare dalla base dei polinomi di Bernstein di grado n alla base monomiale di grado n è la matrice Λ di ordine n+1 i cui elementi sono dati da

$$\Lambda = \quad \lambda_{jk} = \begin{cases} \binom{k}{j}\binom{n}{j}^{-1} & k \geq j \\ 0 & altrimenti \end{cases}$$

Ogni monomio t^j si può esprimere nella forma:

$$t^j = \sum_{k=0}^{n} \lambda_{jk} \, B_{k,n}(t)$$

La matrice di cambiamento di base che fa passare dalla base monomiale di grado n alla base dei polinomi di Bernstein di grado n è la matrice Λ^{-1} di ordine n+1 i cui elementi sono dati da

$$\Lambda^{-1} = \lambda_{jk}^{-1} = \begin{cases} (-1)^{k-j}\binom{n}{k}\binom{k}{j} & k \geq j \\ 0 & \text{altrimenti} \end{cases}$$

Ogni funzione base$B_{j,n}(t)$ si può esprimere nella forma:

$$B_{j,n}(t) = \sum_{k=0}^{n} \lambda_{jk}^{-1} t^k$$

Esempio n=2

Matrice che fa passare dalla base di Bernstein alla base monomiale

$$\Lambda_{3\times 3} = \begin{bmatrix} \binom{0}{0}/\binom{2}{0} & \binom{1}{0}/\binom{2}{0} & \binom{2}{0}/\binom{2}{0} \\ 0 & \binom{1}{1}/\binom{2}{1} & \binom{2}{1}/\binom{2}{1} \\ 0 & 0 & \binom{2}{2}/\binom{2}{2} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1/2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrice che fa passare dalla base monomiale alla base di Bernstein.

$$\Lambda_{3\times 3}^{-1} = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

Per quanto riguarda i coefficienti avremo:

$$p(t) = \sum_{i=0}^{n} c_i \, B_{i,n}(t) = \sum_{i=0}^{n} a_i \, t^i$$

$$p(t) = \sum_{i=0}^{n} a_i t^i = \begin{bmatrix} a_0 & a_1 & \dots & a_n \end{bmatrix} \begin{bmatrix} 1 \\ t \\ t^2 \\ \dots \\ t^n \end{bmatrix} = \begin{bmatrix} c_0 & c_1 & \dots & c_n \end{bmatrix} \begin{bmatrix} B_{0,n}(t) \\ B_{1,n}(t) \\ B_{2,n}(t) \\ \dots \\ B_{\mathrm{n},n}(t) \end{bmatrix}$$

Ma,

$$p(t) = \begin{bmatrix} a_0 & a_1 & \dots & a_n \end{bmatrix} \Lambda \begin{bmatrix} B_{0,n}(t) \\ B_{1,n}(t) \\ B_{2,n}(t) \\ \dots \\ B_{\mathrm{n},n}(t) \end{bmatrix}$$

Segue quindi:

$$\begin{bmatrix} c_0 & c_1 & \dots & c_n \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & \dots & a_n \end{bmatrix} \Lambda$$

$$\begin{bmatrix} a_0 & a_1 & \dots & a_n \end{bmatrix} = \begin{bmatrix} c_0 & c_1 & \dots & c_n \end{bmatrix} \Lambda^{-1}$$

Sia

$$\boldsymbol{p}(\boldsymbol{x}) = \mathbf{2} + \mathbf{3}\boldsymbol{x} - \boldsymbol{x}^{\mathbf{2}} + \mathbf{4}\boldsymbol{x}^{\mathbf{3}}$$

La matrice di cambiamento di base Λ che fa passare dalla base dei polinomi di Bernstein alla base monomiale è:

$$\Lambda = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1/3 & 2/3 & 1 \\ 0 & 0 & 1/3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Quindi il polinomio $p(x)$ espresso nella base di Bernstein avrà per coefficienti:

$$[c_0 \quad c_1 \quad \dots \quad c_n] = [2 \quad 3 \quad -1 \quad 4] \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1/3 & 2/3 & 1 \\ 0 & 0 & 1/3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [2 \quad 3 \quad 11/3 \;\; 8\,]$$

Si passa dai coefficienti di una rappresentazione ai coefficienti dell’altra rappresentazione semplicemente moltiplicando i coefficienti della prima rappresentazione (monomiale) per la matrice di cambiamento di base.

Osservazione: Il primo e l’ultimo coefficiente nella base di Bernstein sono esattamente il valore che $p(x)$ nella base monomiale assume in 0 ed 1.

Proprietà delle curve di Beziér

- Il grado dei polinomi di Bernstein che parametrizzano la curva è sempre dato dal numero dei punti di controllo meno 1. Se abbiamo $n+1$ punti di controllo, il grado è n.
- Questo è uno strumento di flessibilità perché è possibile aumentare i gradi di libertà della curva aumentando il numero dei punti di controllo, però può essere anche negativo, perché all'aumentare del numero dei vertici di controllo corrisponde un aumento del grado del polinomio e la complessità computazionale cresce, in quanto ogni valutazione costa $O(n^2)$.

La curva "segue" la forma del poligono di controllo e si trova sempre nell'inviluppo convesso dei punti di controllo

Il primo e l'ultimo punto di controllo sono l'inizio e la fine della curva e vengono interpolati

I vettori tangenti alla curva nel primo e nell'ultimo punto coincidono con il primo e l'ultimo lato del poligono di controllo.

Le intersezioni della curva con una retta sono sempre in numero minore o al massimo uguale a quelle che la retta ha con il poligono di controllo. (*variation diminishing*).

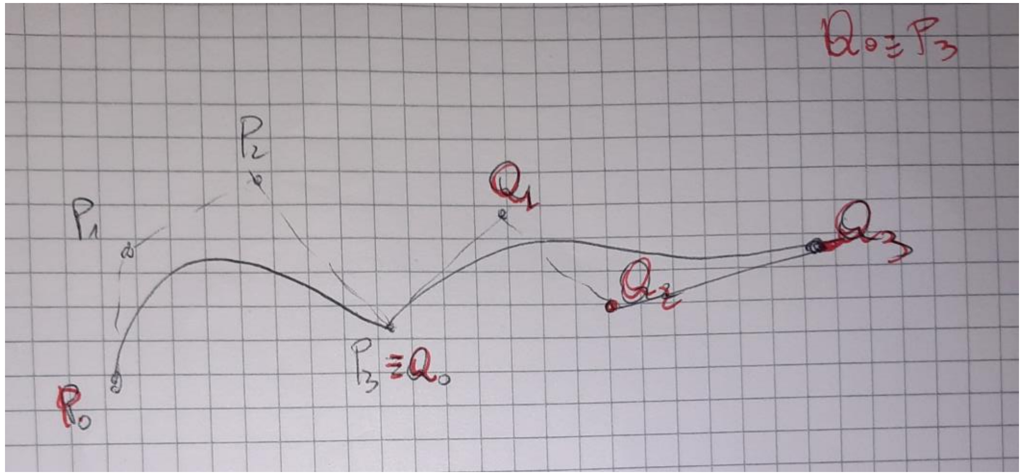
Le curve di Bézier sono invarianti per trasformazioni affini. La curva è trasformata applicando una qualunque trasformazione affine ai suoi punti di controllo.

Svantaggi delle curve di Bezier

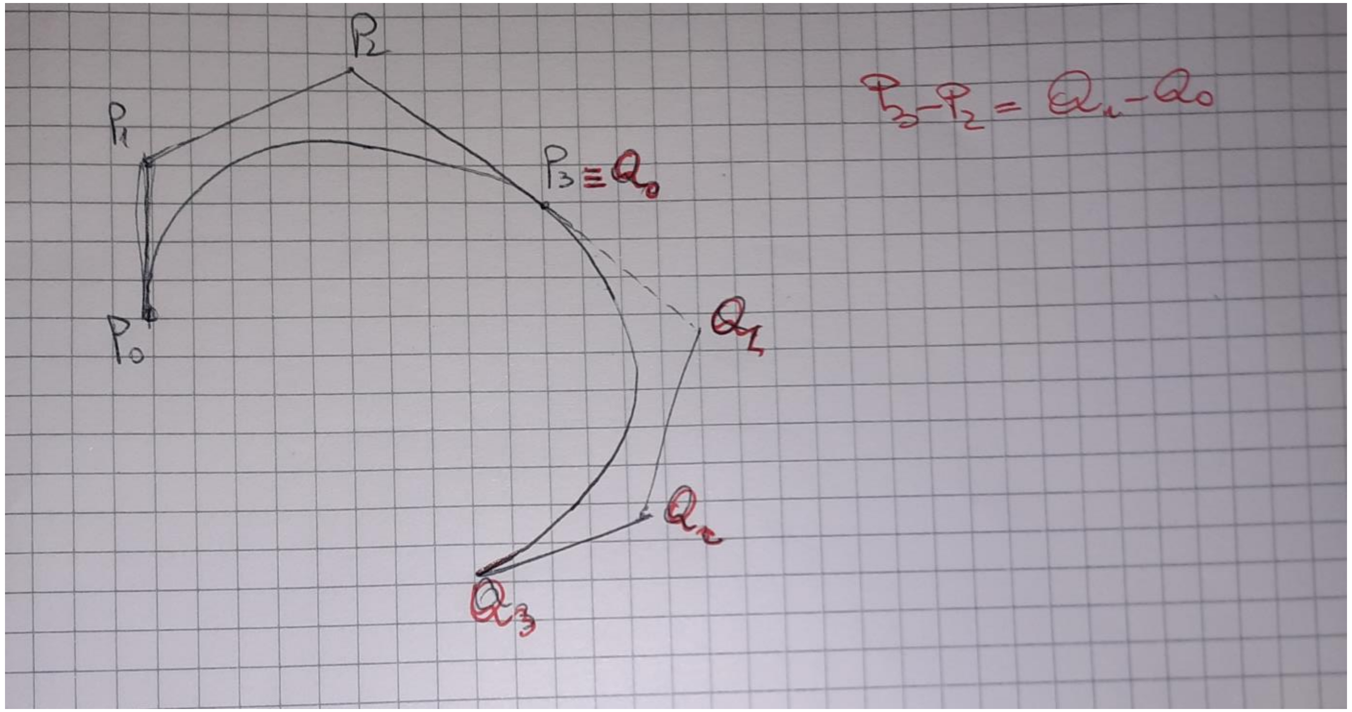
1) I vertici di controllo non permettono di effettuare un controllo locale della curva. Muovendo, infatti, uno qualunque dei suoi vertici di controllo si ottiene un effetto su tutta la curva, anche se esso è meno evidente nelle zone lontane dal punto che si è spostato. Questo perché i polinomi di base di Bernstein sono diversi da zero su tutto l'intervallo.
2) Relazione obbligata tra il numero dei vertici di controllo ed il grado della curva.

   Per ovviare a questo problema, è possibile utilizzare più segmenti di curve di Bezier di grado basso (es. n=3) nel caso si abbia un numero elevato di vertici di controllo. Bisogna però tenere presente il problema della continuità globale. La curva globale potrà essere G^0 se solo il punto di raccordo è in comune, se il punto che precede e quello che segue sono allineati allora potrà essere C^1. Per ottenere questo si perdono gradi di libertà.

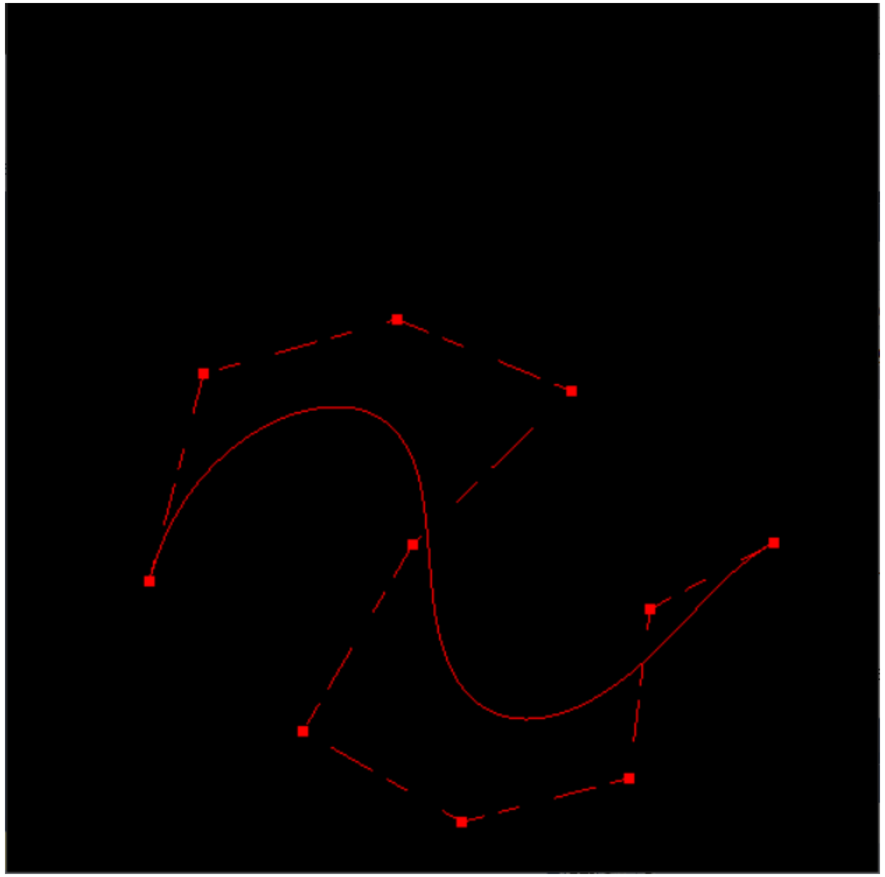
Esempio: Due archi di curva di Bezier di grado 3 che si congiungono in uno stesso punto

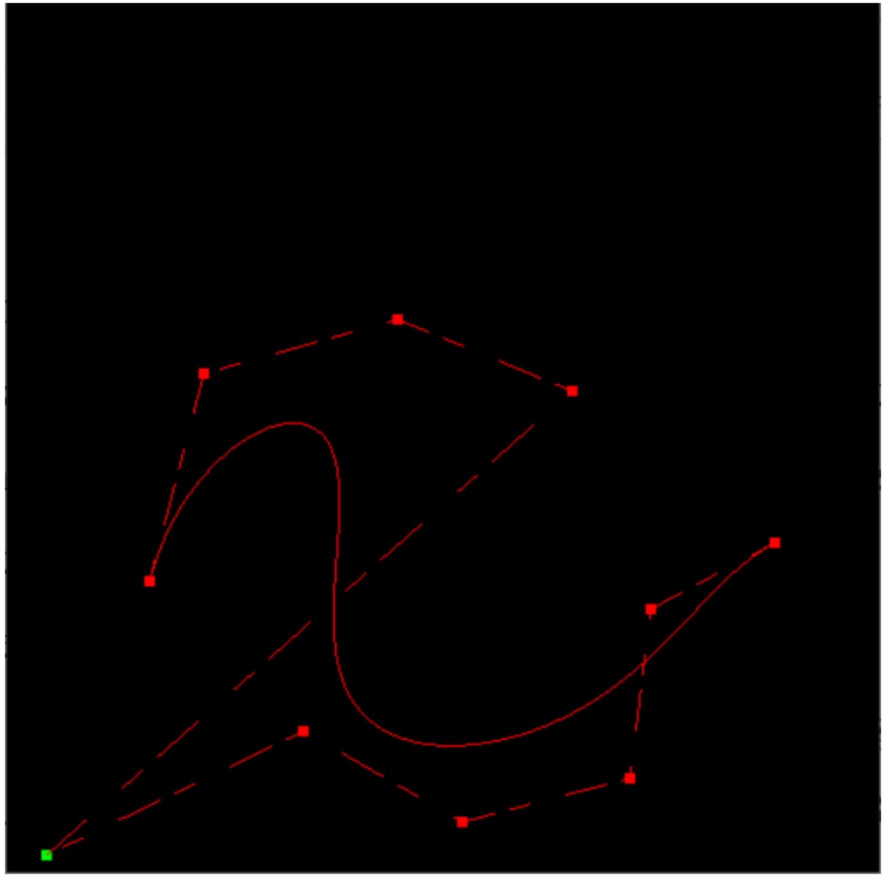


Continuità di classe C^0

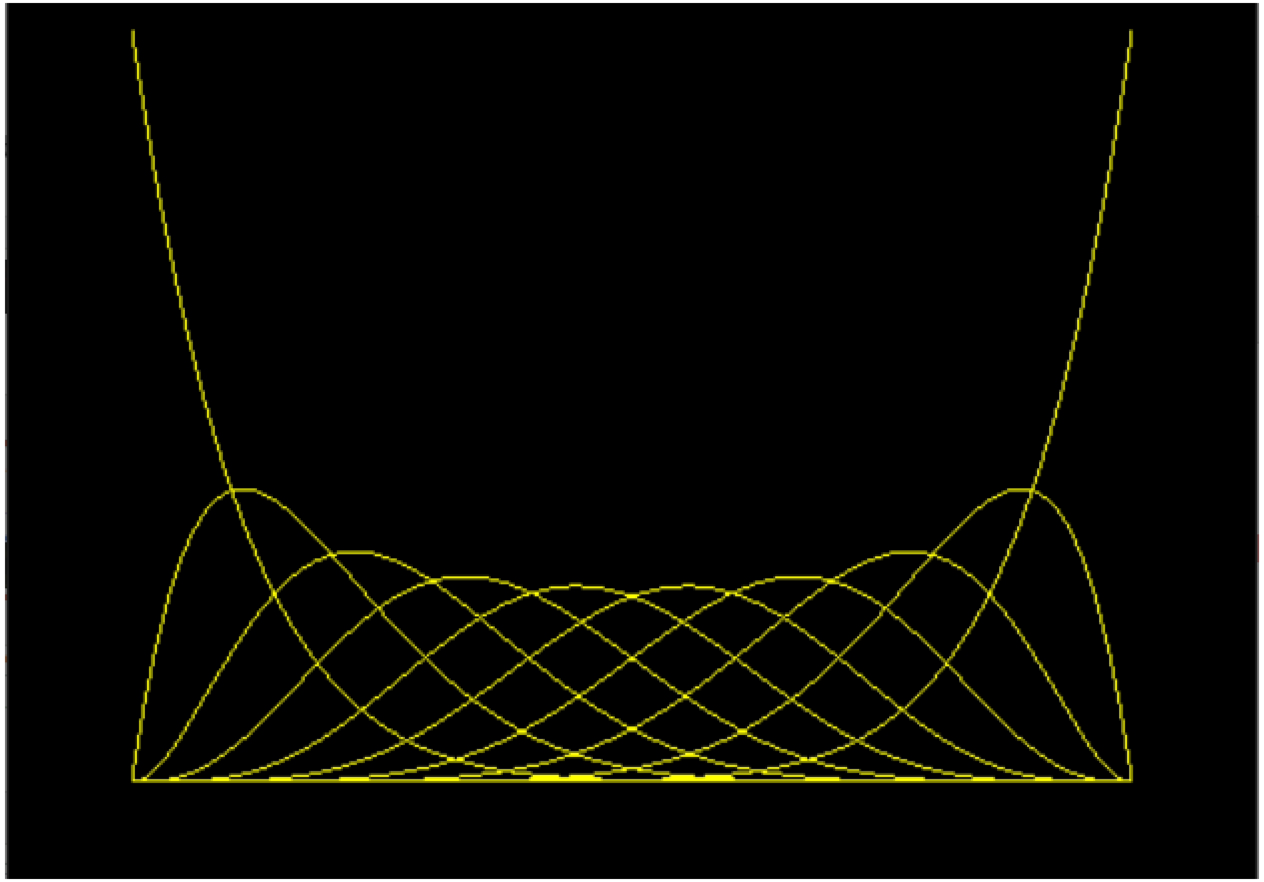


hanno la stessa tangente nel punto di contatto: Continuità di classe C^1





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



Se sposto un punto, poiché tutte le funzioni base sono diverse da zero per ogni valore del parametro t, modifico tutta la curva

$$C(t) = P_0 B_{0,n}(t) + P_1 B_{1,n}(t) + \ldots + P_n B_{\mathrm{n},n}(t)$$